Report No. 8

31 JULY 1964

RESEARCH IN INFORMATION RETRIEVAL

Final Report

Contract No. DA 36-039-SC-90787
File No. 1160-PM-62-93-93(6509)

Technical Report 5400-TR-0096

U. S. Army Electronics Laboratories
Fort Monmouth, New Jersey

ITT Data and Information Systems Division
Route 17 and Garden State Parkway • Paramus, New Jersey

Report No. 8 31 July 1964

RESEARCH IN INFORMATION RETRIEVAL

Final Report

An investigation of the techniques and concepts of information retrieval

Contract No. DA 36-039-SC-90787

File No. 1160-PM-62-93-93(6509)

Signal Corps Technical Requirement

SCL-4218 12 January 1960

Technical Report 5400-TR-0096

Jacques Harlow, Principal Investigator

Paul W. Abrahams

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. PURPOSE

1.1 SCOPE

This report discusses the work performed for the U. S. Army Electronics Laboratory (USAEL) under Contract No. DA-36-039-SC-90787 during the period from 1 July 1962 to 30 June 1964.

1.2 OBJECTIVES

The objective of this project has been to investigate the techniques and concepts of information retrieval and to formulate and develop a general theory of information retrieval. The formalization of this theory is oriented to the automation of large-capacity information storage and retrieval systems. This theoretical framework is intended to serve as a basis for the use of general purpose stored-program digital computer systems to perform the storage and retrieval functions.

1.3 PROJECT TASKS

The primary task of this project has been the development of a research framework based on a general system model in which two processes take place simultaneously and independently: the insertion of documents into the system, and the response to queries. A description is attached to each document as part of the insertion process; most commonly, the description takes the form of a list of descriptors. The descriptions are stored in a file, together with indices that permit back-referencing to the documents themselves. The file is referenced during the processing of a query. Given this model, the analysis can be broken down into four questions:

(a) How is the descriptive structure of the retrieval system generated?

(b) How are descriptions assigned to documents?

(c) How is the file to be structured?

(d) How is a query processed in order to determine a response?

Each of these questions has generated a project task. The over-all framework is presented in Section 4.1, and the four questions are discussed in Sections 4.2, 4.3, 4.4, and 4.5, respectively.

2. ABSTRACT

The purpose of this report is to present the results of a research project on information retrieval. A general system model is presented, and this model is used to express the problem of system specification in terms of four questions:

(a) How is the descriptive structure of the retrieval system generated?

(b) How are descriptions assigned to documents?

(c) How is the file organized?

(d) How is a query processed in order to determine a response?

The treatment of these questions constitute the major subdivisions of the report. Under question (a), the economical assignment of descriptors is discussed and some measures of accessibility are presented; the nature of relatedness of descriptors is also examined. Under question (b), the principal topic is the development of a method for clue word selection in automatic classification methods based on word occurrence; the question of automatic abstracting is also treated under this topic. Under question (c), the relative efficiency of different types of file organizations is examined quantitatively, and the Multi-List system is described and analyzed. Under question (d), the topics treated include the development of a method of probabilistic retrieval and a more searching consideration of the problems involved in retrieval systems with a high degree of man-machine interaction.

3. PUBLICATIONS, REPORTS, AND CONFERENCES

3.1 PUBLICATIONS

A paper by Alfred Trachtenberg entitled "Automatic Document Classification Using Information Theoretical Methods" was presented at the 26th Annual Meeting of the American Documentation Institute and published in the proceedings of that meeting.

3.2 REPORTS

The following reports were issued during the period of this contract:

3.2.1 Monthly Letter Reports

(a) MONTHLY LETTER REPORT NO. 1, 1 July 1962 - 31 July 1962, File No. P-AA-TR-(0006), 3 August 1962; Research in Information Retrieval, Alfred Trachtenberg.

(b) MONTHLY LETTER REPORT NO. 2, 1 August 1962 - 31 August 1962, File No. P-AA-TR-(0009), 31 August 1962; Research in Information Retrieval, Alfred Trachtenberg.

(c) MONTHLY LETTER REPORT NO. 3, 1 October 1962 - 31 October 1962, File No. P-AA-TR-(0012), 31 October 1962; Research in Information Retrieval, Alfred Trachtenberg.

(d) MONTHLY LETTER REPORT NO. 4, 1 November 1962 - 30 November 1962, File No. P-AA-TR-(0025), 30 November 1962; Research in Information Retrieval, Alfred Trachtenberg.

(e) MONTHLY LETTER REPORT NO. 5, 1 January 1963 - 31 January 1963, File No. P-AA-TR-(0032), 31 January 1963; Research in Information Retrieval, Alfred Trachtenberg.

(f) MONTHLY LETTER REPORT NO. 6, 1 February 1963 - 28 February 1963, File No. P-AA-TR-(0033), 28 February 1963; Research in Information Retrieval, Alfred Trachtenberg.

(g) MONTHLY LETTER REPORT NO. 7, 1 April 1963 - 30 April 1963, File No. P-AA-TR-(0046), 30 April 1963; Research in Information Retrieval, George Greenberg.

(h) MONTHLY LETTER REPORT NO. 8, 1 May 1963 - 31 May 1963, File No. P-AA-TR-(0048), 31 May 1963; Research in Information Retrieval, George Greenberg.

(i) MONTHLY LETTER REPORT NO. 9, 1 July 1963 - 31 July 1963, File No. 5201-TR-0059, 31 July 1963; Research in Information Retrieval, George Greenberg.

(j) MONTHLY LETTER REPORT NO. 10, 1 August 1963 - 31 August 1963, File No. 5201-TR-0063, 31 August 1963; Research in Information Retrieval, George Greenberg.

(k) MONTHLY LETTER REPORT NO. 11, 1 October 1963 - 31 October 1963, File No. 5201-TR-0070, 31 October 1963; Research in Information Retrieval, George Greenberg.

(l) MONTHLY LETTER REPORT NO. 12, 1 November 1963 - 30 November 1963, File No. 5201-TR-0075, 30 November 1963; Research in Information Retrieval, Paul W. Abrahams.

(m) MONTHLY LETTER REPORT NO. 13, 1 January 1964 - 31 January 1964, File No. 5201-TR-0079, 31 January 1964; Research in Information Retrieval, Paul W. Abrahams.

(n) MONTHLY LETTER REPORT NO. 14, 1 February 1964 - 29 February 1964, File No. 5201-TR-0081, 2 March 1964; Research in Information Retrieval, Paul W. Abrahams.

3.2.2 Quarterly Progress Reports

(a) RESEARCH IN INFORMATION RETRIEVAL: First Quarterly Report, 1 July 1962 - 30 September 1962, Technical Report P-AA-TR-(0010), 30 October 1962.

(b) RESEARCH IN INFORMATION RETRIEVAL: Second Quarterly Report, 1 October 1962 - 31 December 1962, Technical Report P-AA-TR-(0031), 31 January 1963.

(c) RESEARCH IN INFORMATION RETRIEVAL: Third Quarterly Report, 1 January 1963 - 31 March 1963, Technical Report P-AA-TR-(0044), 30 April 1963.

(d) RESEARCH IN INFORMATION RETRIEVAL: Fourth Quarterly Report, 1 April 1963 - 30 June 1963, Technical Report 5201-TR-0058, 31 July 1963.

(e) RESEARCH IN INFORMATION RETRIEVAL: Fifth Quarterly Report, 1 July 1963 - 30 September 1963, Technical Report 5201-TR-0069, 31 October 1963.

(f) RESEARCH IN INFORMATION RETRIEVAL: Sixth Quarterly Report, 1 October 1963 - 31 December 1963, Technical Report 5201-TR-0078, 31 January 1964.

(g) RESEARCH IN INFORMATION RETRIEVAL: Seventh Quarterly Report, 1 January 1964 - 31 March 1964, Technical Report 5201-TR-0088, 30 April 1964.

3.3 CONFERENCES

3.3.1 Conferences with USAEL Personnel

The following conferences were held between DISD personnel and USAEL personnel:

(a) 5 July 1962--Meeting at DISD. Discussions of objectives and plans for the research activity were initiated. The formulation of a method of approach was requested for presentation at the next meeting.

(b) 17 July 1962--Meeting at DISD. A technical note prepared by the project staff was used as the basis of discussions pertaining to the scope, development phases, alternative plans, and recommended direction for the project.

(c) 18 July 1962--Meeting at DISD. Informal discussion of Signal Corps objectives and goals for research activity.

(d) 9 August 1962--Meeting at Fort Monmouth, New Jersey. Discussions were held concerning the functional characteristics of information retrieval systems. No particular area of activity was selected for further study.

(e) 10 September 1962--Meeting at DISD. Several methods of relating descriptor systems in a generalized sense were discussed in relation to the requirements for a file structure. The analysis and development of a general theory was recommended as the objective of the project.

(f) 29 November 1962--Meeting at DISD. DISD personnel were introduced to Mr. Anthony V. Campi, who had recently been assigned as Project Engineer. Several aspects of the First Quarterly Report were discussed, and the concepts pertaining to measure of relevance were clarified. DISD accepted the suggestion that the discussion in the report should be elaborated in more detail.

(g) 28 February 1963--Meeting at DISD. DISD personnel met with Mr. Anthony V. Campi, who had recently been assigned as Project Engineer. Several aspects of the Second Quarterly Report were discussed. A few minor corrections and elaborations were requested, and a general emphasis on the importance of user requirements was indicated.

(h) 25 April 1963--Meeting at DISD. Mr. David Haretz and Mr. Larry Sarlo Conferred with project personnel on the general impact and significance of the report on scientific information prepared by the President's Science Advisory Committee. This report is entitled Science, Government, and Information.

(i) 6 June 1963--Meeting at DISD. Lt. Fred Hill and Mr. Larry Sarlo conferred with project personnel about the manuscript version of the Third Quarterly Report. Difficult concepts were explained,

and questions were discussed. Several suggested changes were accepted for inclusion in the published form. The plans for the current quarter and for future activity were also discussed.

(j) 2 August 1963--Meeting at DISD. Mr. Larry Sarlo and Lt. Fred Hill were briefed on progress made during the fourth quarter of the information retrieval project. Researchers presented aspects of their work during the quarter which were included in the Fourth Quarterly Report. Plans for the fifth quarter and future activity were also discussed.

(k) 21 October 1963--Meeting at DISD. Mr. David Haretz and Mr. Larry Sarlo conferred with project personnel to review the first draft of the Fourth Quarterly Report. The report was reviewed in detail, and some concepts relating to the Multi-List system were clarified.

(l) 20 November 1963--Meeting at USAEL. Project personnel conferred with Mr. David Haretz, Mr. Larry Sarlo, and Mr. Serafino Amoroso. Several problems were discussed and settled, and some of the difficulties of system integration were examined.

(m) 4 March 1964--Meeting at DISD. Mr. Larry Sarlo of USAEL reviewed the first draft of the Sixth Quarterly Report. Several minor corrections were made, and some technical difficulties were clarified.

(n) 25 June 1964--Meeting at DISD. A discussion was held between project personnel and Mr. David Haretz, Mr. Anthony V. Campi, and Mr. David Hadden, Jr., of USAEL. The current status and accomplishments of the project were discussed, and the content of the final report was considered.

3.3.2 Other Conferences

During the term of this project, various project personnel attended conferences relating to information retrieval. Attendance at these

conferences was sponsored by DISD; the knowledge gained was of considerable help in pursuing specific research areas within this project.

(a) 3 December 1962 - 7 December 1962--Mathematics of Information Storage and Retrieval. Quentin A. Darmstadt attended this conference, which was conducted by Dr. Robert M. Hayes under the auspices of the Georgia Institute of Technology.

During this period several ancillary conferences were also attended:

(b) 2 May 1963--NASA Scientific and Technical Information Conference. This conference was held in Atlanta, Georgia, and was attended by George Greenberg. The conference presented NASA's methods and techniques for acquiring, processing, storing, disseminating, and retrieving information.

(c) 17 June 1963--Simulation of Cognitive Processes. This seminar was conducted for six weeks at the RAND Corporation. Its purpose was to discuss the problems of information systems. George Greenberg was an invited participant. During the time spent at the seminar Dr. Greenberg had the opportunity to discuss the problems of information retrieval with several other research organizations.

(d) 6 October 1963 - 11 October 1963--26th Annual Meeting of the American Documentation Institute. This meeting was attended by Jacques Harlow and Alfred Trachtenberg; Mr. Trachtenberg presented some of the results of the project in an invited paper at the conference.

(e) 19 February 1964--Meeting with Dr. Harold Borko. Paul Abrahams met with Dr. Borko at the System Development Corporation in Santa Monica, California. The research carried on under this contract was discussed, and Dr. Borko offered a number of helpful suggestions.

4. FACTUAL DATA

4.1 STATEMENT OF THE PROBLEM

4.1.1 Original Formulation - The technical requirement of the Signal Corps, as specified in SCL-4355, is for "...a research investigation of techniques and concepts necessary for the efficient mechanization of large-capacity information storage and retrieval systems." Among the applied objectives suggested as guides for such research are "...problems of military significance; i.e., personnel files, intelligence data, etc."

4.1.2 System Model and Definitions - The purpose of an information storage and retrieval system is to record a body of information and to provide to a group of users a means of answering questions pertaining to this information. The information is ordinarily provided in the form of a discrete set of documents, such as books, parts listings, personnel records, or newspaper articles. Information retrieval systems may be either document retrieval systems or content retrieval systems; a document retrieval system responds to a query with a set of documents that are relevant to the user's question, while a content retrieval system provides the actual answer to the question. Document retrieval systems may further be subdivided into those systems that provide the actual documents and those that merely tell where the documents are located.

Most of the research described in this report has been concerned with document retrieval systems that provide the locations of documents rather than the documents themselves. In order to clarify the terminology, it will be helpful to present a generalized model of how such systems operate. A diagram of this model is shown in Figure 1. There are two major

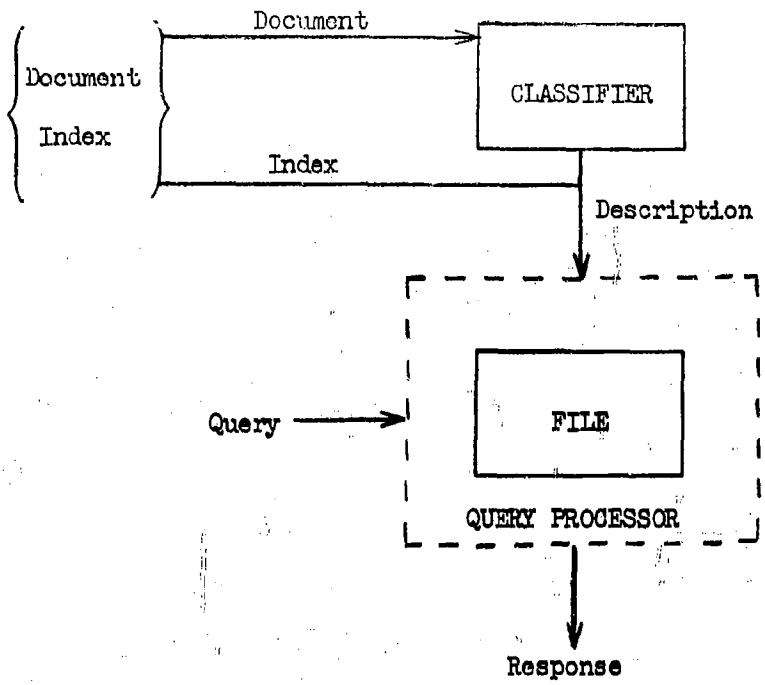


FIGURE 1. Basic System Model

processes taking place in the system: the incorporation of documents into a file, and the response to queries. These processes take place asynchronously. A query as we use it here is not quite the same thing as a question; a question is the user's own description of the information he needs, while a query is in a form that the system can operate upon and respond to. Questions may be vague and formless; queries must be specific and formal.

Associated with each document stored in the system are an index and a description. The index specifies either directly or indirectly where the document is physically stored. (For instance, the personnel file of

an employee can be located physically if the employee's name is known, or even if only the serial number of a card giving his name is known.) The description relates to the content of the document, and consists of that information about the document that is available for matching against queries. The file contains the index and description of each available document. The query processor operates on queries, making use of the file, to produce the indices of those documents that are responsive to the query. The file should really be thought of as an integral part of the query processor.

When a document is entered into the system it is presented to a classifier that generates the description of the document. The output of the classifier is then paired with the index of the document and stored in the file. A variation on this configuration is to have the index derived from the description; the ordinary library follows this procedure, since the physical location of a book depends on its description.

These concepts can be clarified by means of a simple example. Consider a library of technical journals. Since each journal may contain several unrelated articles, each article is treated as a separate document. The index of each document is the journal name, volume number, and page number. The librarian records, for each document, a list of subject headings that describe the document; this list is the document description. A separate card is made up for each appropriate subject heading, listing and subject heading and the document index. The file consists of the subject cards for all the available documents. If the cards are stored

alphabetically by subject, then a query consists of the name of a single subject, and the processing of a query consists of locating the set of cards for that subject. The response is the set of index numbers listed on the cards. Of course, the system of this example will not be particularly effective, but it does serve to illustrate the concepts.

The most common form of document description consists of a list of descriptors such as the subject headings of the previous example. The descriptors may have additional information associated with them, or they may be related to one another in rather complex ways. It should be emphasized that the descriptor list is not the only possible form of a document description.

Various modifications of the model in Figure 1 are possible. One such modification is to have the index as an output of the classification process rather than having it be independent of that process. A different variation would be a system that produced documents rather than indices in response to queries. In such a system the description of a document would be the document itself. The document--or significant and definable parts of a document--would then absorb the function of the index when the query capabilities were activated.

It is also possible to conceive of query capabilities with the ability to retrieve only the relevant portions of documents. At the least sophisticated level this variation simply involves refining the organization of the total data so that a larger number of functional documents is available for output. This procedure could be achieved by applying the same

processes to small subunits of conventional documents. Finally, the system may be able to produce responses to queries that are neither documents nor portions of documents but responses derived from the information contained in various documents. Such a system must have the capacity to perform inferential processing on the content of the documents. This type of system would correspond to a content retrieval system.

4.1.3 Problem Formulation and Task Breakdown - A model may describe the operation of an information retrieval system; but in order to develop an operating system, questions relating to the requirements of an information retrieval system must first be answered.

An analysis of the system model given in Figure 1 leads to the breakdown of the problem into four tasks. The form of the descriptions that are transmitted from the classifier to the file must be defined; in addition, the three major system components--the classifier, the file, and the query processor--must be specified. In order to account for the static as well as the dynamic aspects of information retrieval, these requirements may be expressed in terms of four questions:

(a) How is the descriptive structure of the retrieval system generated?

(b) How are descriptions assigned to documents?

(c) How is the file to be structured?

(d) How is a query processed in order to determine a response?

Although the answers to each of these questions are interdependent, it is still possible to consider each of them separately. Question (a) must be answered first since the very definitions of the other questions depend

upon it. For instance, it is impossible to talk about the assignment of descriptions to documents until the class of possible descriptions has been settled upon. Since a good file organization will be based upon the descriptive structure in use, question (a) must be answered before question (c) can be considered. Question (d), in turn, depends upon question (c) since the file is an integral part of the query processor.

In considering question (a) it must be recognized that the descriptive structure of a retrieval system will depend upon the particular corpus of information that it is to operate upon. It is the method of generating descriptions rather than the descriptions themselves that are invariant from one corpus to another. Furthermore, the class of possible descriptions may itself vary with time as new types of documents are introduced into the system and rarely used ones dropped out.

Questions (a) and (c) may be regarded as concerned with the static aspects of a retrieval system, while questions (b) and (d) deal with the dynamic aspects. The descriptive structure and file structure are usually fixed before the system becomes operational and are modified, at worst, at a slow rate thereafter. The assignment of descriptions to documents and the answering of queries, on the other hand, are on-going processes.

In order to clarify these questions, each of them will be discussed in greater detail in the following section.

4.1.4 Explication of the System Requirements

4.1.4.1 Descriptive Structures - Most descriptive structures

are based on the use of descriptors. Descriptors are introduced into information retrieval systems in order to reduce the language recognition and transformation requirements and to reduce the complexity of the data structures or content relationships. In short, descriptors represent an artificially restricted standard language used to increase the convenience of handling requests, constructing and organizing files, and searching for answers.

One of the major problems in constructing a descriptor system is the proper selection of the descriptors that are class names for synonyms so as to maximize retrieval of relevant information and to minimize noise, the retrieval of irrelevant data. The descriptors must be words in common use, as unambiguous as possible, and sufficiently numerous to delineate relatively fine distinctions. Obviously, the more documents filed under a given descriptor, the larger the noise is likely to be.

To increase the number of relevant documents retrieved in response to a given request, descriptors for the request can be weighted. These weights can be assigned according to the relevance and the importance of the particular descriptor under consideration. The system can then produce responses ordered according to weights assigned descriptors or responses greater than a fixed weight of relevance and importance. Another scheme for reducing irrelevance in responses is to assign descriptors to each section of documents added to the file. This method, of course, increases the degree of content retrieval.

Increasing the flexibility of descriptors by introducing role indicators or specifying terms as actions, relations, results, means, purpose, or locations is a further step toward content retrieval in the sense that it is the beginning of syntactical and semantic specification of request terms.

4.1.4.2 Assignment of Descriptions to Documents - If the selected form of description for documents is the descriptor list, then the simplest method of classification would be simply to assign to a document those descriptors that occurred within its title. This rule is the basis of the quite popular KWIC indexing system. Its defect is that a descriptor must have associated with it a large number of synonyms, since the occurrence of the intended descriptor in a title is usually rather unlikely.

More elaborate classification schemes can be based upon the occurrence of words other than the descriptors themselves within either the title, the abstract, or the text of a document. These methods are also capable of generalization to account for word frequency as well as word occurrence information and to assign different weights to words according to their relevance to the category. Such approaches are particularly amenable to automatic classification; their defect is that they cannot be quite so readily adapted to descriptions more complicated than the simple descriptor list.

For more complicated kinds of descriptions, such as descriptors interrelated through the use of connectives, more sophisticated textual

analyses are necessary. Word occurrences can still be used as aids in locating key sentences within the document, but for this type of classification the use of syntactic analyzers is probably unavoidable.

4.1.4.3 Organization and Structure of Files - If information retrieval is viewed generally, it can be defined as locating and presenting a specific informative and accurate answer or piece of information in response to a specific question. Accomplishing this function requires a classification scheme that groups larger units of related information; e.g., documents or sections of documents. Descriptors are assigned to units of information. The file consists of the system of descriptors and of information units ordered in some fashion to indicate the relations between descriptors and information. Generally, a descriptor is associated with many units of information and a unit of information may be described by several descriptors. In addition, the file structure must provide for relations among information units and among descriptors.

One of the best known systems that can be used to relate descriptors is the hierarchical classification or tree structure originally developed for biological classification. This type of structure forms a Boolean algebra under the relation of class inclusion. This model is only appropriate for a limited field of information in which a class is immediately subordinate to only one other class. This restriction requires a breakdown into small units of information, which means that the descriptor file would be composed of a large number of hierarchies of class inclusion. (The Multi-List system is a device for circumventing

the limitations of ordinary list processing or hierarchies by allowing for relations among branches.)

For information fields of some diversity, the relations among descriptors usually form complicated networks to which the tree theory is not directly applicable. A general model of a complicated descriptor network is represented by means of a complemented modular lattice. This model is of sufficient generality to cover a wide variety of situations. Most elements are multiply connected rather than singly connected as in a tree. The lattice model is referred to as a weak hierarchy--an element may have more than one predecessor. The tree is a strong hierarchy--an element has only one predecessor. The principal problem with the lattice model is that the number of nodes in the network quickly reaches into the millions if all relations between descriptors are represented. Consequently, the problem becomes one of effectively limiting the number of relations represented among descriptors.

The descriptor file associates descriptors with information units or items of data. These associations can be represented by a matrix of ones and zeros, where descriptors may be ordered as rows and information units as columns. A one indicates a relation; a zero, none. For a rich information store, this matrix will be large and most of its elements will be zeros. It is, therefore, an uneconomical representation. The matrix can be compressed by listing rows or columns (descriptors or data) and related items only for each entry. Of course, access to the file is much simpler for descriptor entry. Search time for these types

of files can be reduced by using multiple entry of terms or by an ordered arrangement of both descriptors and data. Generic relations among terms can be shown by direct cross references, carried with each descriptor, or by a code of hierarchical class numbers showing the generic structure of the terms.

4.1.4.4 Query Response - In a retrieval system based upon descriptors there are two requirements for effective response to queries. The first is the transformation of the query into the standard search terms. The second is the particular strategy or methodology for searching the descriptor file effectively and fruitfully.

Transforming a query into standard descriptor terms is basically a form of translation from a rich language into a summary language or the matching of two sets of terms, one large, the other smaller. In order to accomplish this transformation, the meaning and relations between terms of the two sets or languages must be understood. Aid may be provided in the form of a dictionary or glossary of subject matter. The knowledge required to transform requests into descriptors is most simply provided to a computer by furnishing it with a thesaurus. Any more sophisticated means would involve a considerable capability for linguistic transformation on the part of the computer.

The formulation of a query and its transformation into a limited set of descriptors often does not provide sufficient information and direction to obtain exhaustive information concerning a subject that may exist in the data file. Effective search procedures are closely related

to the way in which the descriptor file is structured and what sort of relations are indicated there. The most common method of searching is the conjunctive search, which retrieves only that information related to or encompassed by all the request descriptors in conjunction. It is also possible to construct search procedures in terms of logical sums, differences, complements, and more complicated combinations of these functions as well as weighted logical functions in terms of set densities.

4.1.5 Relation to Specific Problems - In the following three subsections the four questions posed in Section 4.1.3 will be examined in relation to three specific information retrieval problems: personnel files, literature, and intelligence information. These three problems will be examined in increasing order of difficulty.

4.1.5.1 Personnel Files - In extracting information from personnel files, the critical questions are (c) and (d), namely, file structure and response to queries. Each personnel record will normally be composed of a set of fields, each giving some characteristic of the individual person. Some of these fields may be variable in length; e.g., there may be a field listing the age and sex of all dependent children. The descriptive structure of such a file is trivial, since the description of any document (i.e., individual record) is simply that subset of the fields that may be used for retrieval purposes. The process of assigning descriptions to documents is nothing more than deletion followed by straightforward encoding.

The file structure problem in this case concerns the specific

device used to store the information and the arrangement of the items within this device. For instance, it may be possible physically to string together those items that possess a common characteristic; this technique is effectively what is done by the Multi-List system (cf., Section 4.4.2). On the other hand, items may be placed in a special order, with appropriate indexing systems. Query processing consists of nothing more than matching, but the mechanization of this matching may be quite complex and will certainly be closely related to the file organization. For personnel files the problem of deciding whether a particular document is responsive to a particular query is quite trivial.

4.1.5.2 Literature Retrieval - In a literature retrieval system, unlike the personnel file, the problem of selecting a descriptive structure and then of classifying documents is no longer trivial. Furthermore, the question of whether or not a particular document is responsive to a particular query cannot be answered with certainty but only with probability.

The most common form of description for literature retrieval systems is the descriptor list. In this case the choice of descriptors becomes critical, since the descriptors are used both for classification and for querying. The particular descriptors used will depend on the subject matter of the literature being classified, although the nature of the interrelation may be subject-independent if the descriptors within a description are interrelated.

Given a set of descriptors, the problem of classifying documents

is still quite difficult. An approach to this problem that utilized the occurrence of clue words is discussed in this report. A complicating factor is the difference between the use of a descriptor in a document and the meaning of that descriptor as understood by a user of the system. If these meanings are divergent, then poor system performance may result.

The problem of file structure and query response in a literature retrieval system is similar to that of a personnel file. Once descriptors have been assigned to documents, the process of answering a query is again purely a matching process. The guesswork occurs not in the response to queries but in the classification.

4.1.5.3 Intelligence Information - In retrieving intelligence information all the difficulties that exist in literature retrieval are retained, but in addition the problem of query processing is no longer merely a matter of matching. In its more elaborate forms, in fact, intelligence information processing really requires the use of implicit information retrieval techniques. On a lesser level, it may still be necessary to consider the interrelationships of different items of data in order to decide which ones are to be provided in the response to a query. Items that are useless by themselves may become useful as part of a chain of related events.

Processing of intelligence information will almost certainly require the use of syntactic and semantic analysis. For information of this type it is virtually impossible for a system to respond to queries unless it is capable of extracting the meaning of a sentence or a document.

Terms of interest will ordinarily occur far to frequently within the corpus of information for mere word occurrence or frequency data to be particularly helpful in isolating salient data. In addition, much of the required output will be useful only when presented in appropriate combination.

A further salient aspect of intelligence information processing is that ordinarily one would expect to ask many queries in order to answer a question. Thus there exists a feedback relationship between the system and the user, in which each query is largely determined by the response to the last one. The structure of the query processor, and consequently of the query language, must be constructed to account for this feedback relationship.

4.2 DESCRIPTIVE STRUCTURE OF RETRIEVAL SYSTEMS

In an information retrieval system a description is attached to each document, and this description represents all the information about the document that is available to the system for retrieval purposes. The descriptive structure of the system is concerned with the class of possible descriptions, but not with how descriptions are actually assigned to documents. The descriptive systems examined within the scope of this project, with the exception of the material on automatic abstracting, have assumed that there exists a set of descriptors from which descriptions are constructed. Three key questions then remain:

(a) How are the descriptors to be selected?

(b) What information is to be attached to a descriptor?

(c) How are several descriptors in a description to be related?

In dealing with the first of these questions in particular, one can examine methods of descriptor selection that operate through improvement of an initial set on the basis of experience with the retrieval system.

In the approaches considered here it has been assumed that descriptions consist of Boolean combinations of descriptors, and the possibility of attaching probabilities to the descriptors has been explicitly admitted. The major task is then the selection of the particular descriptors to be used. This section discusses the role of efficiency in descriptor selection and some corrective methods for improving a descriptor set under actual operating conditions. In Appendix C, Section 9.3, some of the more popular existing descriptive schemes are described and discussed.

4.2.1 Efficiency Considerations in Descriptor Selection

4.2.1.1 General Criteria - In a collection of n items there is only a finite number of subcollections of items that are theoretically possible responses in item retrieval systems. The number is 2^n if zero items are considered a subcollection. In practice, not all 2^n answers are equally likely to be searched for by a user. Intuition suggests that this disparity is an essential criterion for the effective design of a query or descriptor language.

There are several possible approaches to specifying which of these 2^n subcollections is being referenced. In one sense the simplest means of specification is to assign a name or descriptor to each of the n items in the collection. In the case when all 2^n subcollections are requested equally often and when the questioner knows the name of each

item he is interested in, this method produces an adequate system. If, however, some subcollections are considerably more popular than others, then an obvious improvement in coding efficiency would result from giving popular collections special category names.

There are, however, considerations other than information theoretic measures of coding efficiency that are relevant to the selection of a descriptor language. Asking for all the items in a subcollection by name is possible only when the names of all the documents in the subcollection that are of interest are known. Under these circumstances the general problem of information retrieval becomes a special case, and only considerations of coding efficiency and, perhaps, user compatibility are relevant criteria for descriptor language design.

In an ordinary library search the questioner does not know the names of the items he needs. He wants the system to supply a subcollection of items that will provide information relevant to his query after he reads them. The system must go from his query or a transformation of his query to an appropriate subcollection of items, even though the user does not yet know in advance what is in this subcollection.

How can the system do this? One approach is to ask, perhaps implicitly, questions in advance and to search, again implicitly, the entire collection to find the items that contain information relevant to each question. The system would then have the stored answer available whenever the same question arose. In a sizable collection it is not feasible to ask all questions in advance. There are two reasons: first,

there are a large number of ways of asking essentially the same question; another way of putting this point is that the same answer subcollection would satisfy many possible question variations. Second, there are too many possible answers--specifically, 2^n--in any sizable system.

Each of these difficulties requires a different approach. The approach to the former involves standardization; that is, the possible ways of asking essentially the same question must be restricted. This solution is primarily a language problem. The approach to the latter difficulties involves exclusion of less probable questions and their resultant answers from advance treatment. This solution is primarily a system design and organization problem.

How is explicit or implicit advance treatment of questions possible? One theoretically possible method would be to have all documents in the library unordered, except perhaps by author and title, for those searches in which the querier already knows which documents he wants. Anyone wishing to use the library could then be asked to submit both a copy of his question and a list of the documents he found relevant after making his search of the library. This information could then be stored for occasions when the same or similar questions are asked.

Of course, this scheme is impractical, but listing some of its inherent difficulties may lead to an understanding of the requirements of an ideal descriptor-query language.

(a) There is no assurance that any initial questioner will do a good or thorough job in searching all the documents in the library.

(b) Even if the initial questioner has done a perfect job at the time he searched the library, there would be a lack of information about the relevance of new accessions to the question. Of course, new accessions could be re-searched by subsequent questioners in order to keep the answer list up to date.

(c) Many questions will recur imprecisely; even if the statement of the question is identical, different users are likely to have different meanings of intentions that would influence which documents they considered appropriate for the answer list. Thus, even if there is a perfect and up-to-date search performed by the initial questioner, it is not likely to be perfect for a subsequent questioner.

(d) Such a system would impose an unacceptable search burden, not only upon initial questioners but also upon subsequent questioners, if there are a substantial number of new acquisitions. Furthermore, the askers of somewhat unusual questions would always tend to be in the role of initial questioners, regardless of how long the system has been in operation. Their extensive search efforts would rarely be applied by subsequent users.

The technique currently used by most libraries, in order to deal with these objections, is implicitly to select a range of questions to be pre-answered and then to assess the relevance of each accession--i.e., index it--to all these questions as it is entered into the library file.

To the extent that a document's relevance to many questions can be assessed nearly simultaneously, this technique has obvious advantages over repeatedly scanning each document for each question in some sequence of questions.

The approach of classifying each accession for all questions will deal completely only with difficulties (a) and (b). Difficulties (c) and (d) will be resolved only to the extent that the question list, against which each document is implicitly being checked, is sufficiently extensive and to the extent that the meaning of these implicit questions is sufficiently clear to the system users.

It is likely that none of the difficulties will ever be resolved completely. Even a user searching on the basis of his own question is likely to introduce inadvertent errors of both inclusion and exclusion on the answer list if he is scanning a large file collection. Similar errors will occur when a librarian classifies a book. But additional errors will result from the fact that the meaning of the implicit questions reflected by the classification varies from person to person.

These errors, while often significant, are not as basic a problem as the limitation on possible questions that can be answered. These limitations are a necessary concomitant of indexing a large collection. As has already been suggested, there are two kinds of limitations:

(a) Basic limitations on the retrieval of all 2^n answers. In general, no indexing scheme for a sizable collection is sufficiently articulated to allow retrieval of all possible answers without knowing the names of individual documents.

(b) Secondary limitations on the acceptability, or communicability, of a specific question formulation that does in fact correspond to one of the accessible answers.

The latter limitation does not necessarily imply any change in the logical organization of the indexing or query-descriptor language. The problem is one of using appropriate names or labels for the index terms or combinations of index terms that correspond to those of the 2^n answers that the system is capable of generating. Of course, the problem is not one that can be solved merely by the judicious selection of terms. It is necessary that the questioner and the library system use these terms in essentially the same sense. Furthermore, it is necessary that alternate descriptions of the same answer or question be interconvertible, either by the library system or by the user. To date, the only methods of dealing with this problem have been to provide the user with a dictionary-type description of the index terms, an over-view of the relationship among the terms used by the system, and/or a thesaurus type of referral ("see" and "see also") to related terms.

The problem of converting synonymous descriptions probably cannot be approached by considering the relative frequency of subcollection questions. Of course, the more popular a subcollection, the more valuable it might be to be able to deal with alternate ways of describing it. The problem of unaskable questions, however, can only be approached fruitfully from this point of view. If the system is to be insufficiently articulated for the retrieval of all 2^n possible answer collections, it seems that the criteria (other than random exclusion based upon cost considerations) for

deciding which subcollections are to be retrievable should ultimately be based upon the frequency of user demand. Only those questions that will rarely or never be asked should in principle be unanswerable--without searching the entire collection--because of limitations in the query language and the accompanying file structures and search procedures.

This conclusion suggests that a second consideration, besides the relative frequency of user demand for various possible answers, may be important. This consideration is the absolute level of demand for a possible answer subcollection. The absolute level of demand is readily calculated from estimates of relative demand and the total number of questions asked. An estimate for the number of questions may be the length of time for which the collection of items will be used multiplied by levels of use such as questions per day during this interval. As absolute use of the system as a whole increases, more articulate indexing becomes necessary to include the relatively less frequently asked questions, which now are asked a significant number of times in the system's lifetime.

Answer subcollections should not merely be regarded as accessible or inaccessible with a given query capability. Even if a subcollection is not immediately accessible, there are degrees of desirability that can be discriminated with respect to its inaccessibility. Thus a desired answer subcollection may not be directly accessible <u>per se</u>, yet it may be wholly embedded in another subcollection that is accessible and that contains few additional items. Clearly, there is no great

deficiency in query capability under such circumstances so long as the user can identify and ask for the appropriate inexact subcollection. If, however, the items in a desired inaccessible subcollection are widely scattered--that is, the items cannot be obtained without searching a number of accessible subcollections--the situation is quite different. This difficulty is likely to be further complicated by the inherent unavailability of information about which accessible subcollections contain the items the user needs. Under such circumstances the user may be reduced to searching the entire collection, or unacceptably large parts of it, in order to obtain the needed information. It might be fruitful to develop rigorous measures of degree of inaccessibility based upon minimal and/or maximal false drops and/or misses.

Such a measure of accessibility could be used to evaluate the goodness of any descriptor scheme for any item collection. More precisely, it could be used to measure the average (in)accessibility for the power set of items, the set of 2^n possible answers, for a given descriptor scheme. When combined with information about relative frequencies of the members of the possible answer set, such a measure can provide information about the average accessibility of items per request. One purpose of a general theory of information retrieval is to provide an analytical framework in which this quantity, the average accessibility per request, can be optimized, given a context of relevant system parameters.

4.2.1.2 Factors That Govern the Criteria of Relative Importance of Descriptors - If a large collection of documents is classified in some

fashion, each document in this collection is labeled by one or more descriptors. However, not all descriptors are equally important; the deletion of some would hardly affect the retrieval processes, but the deletion of others would be detrimental.

The unchecked proliferation of descriptors may diminish the usefulness of a collection of documents either by lengthening the physical processes involved in retrieval, by confusing the taxonomical logic of the collection, or by simply straying too far from the natural usage of terms. In any case, it is usually prudent to restrict the number of new descriptors that may be introduced in order to keep the retrieval processes near peak efficiency.

Under such conditions the choice and the allocation of descriptors may be governed by criteria of descriptor importance. In addition, the criteria used in automatic indexing procedures may necessarily lean more towards the use of statistical information about the collection than is the case when indexing is done manually. To put the same ideas differently and more strikingly, when indexing is performed automatically the governing criteria may pertain more to statistical distributions of descriptors among the documents than to explicit relations between the subject matter of a given document and a descriptor.

Given these premises, the factors that govern the relative importance of descriptors are:

(a) Let us suppose that a certain descriptor is never mentioned in any of the retrieval requests. Obviously such a descriptor

could be deleted from the collection without loss. Conversely, descriptors used with high frequencies have a high probability of being important. At present, we can only speak of the higher probability of importance since the relation of various factors to each other has not been formalized. So far as frequency relations are concerned, a certain assymetrical situation exists. Below a certain frequency threshold the frequency considerations are overwhelming. If a descriptor is not used with a certain minimum frequency, it cannot be ranked high. However, the high frequency descriptors are not necessarily important. For example, a high frequency descriptor may be synonymous with another descriptor.

(b) Descriptors are usually employed jointly. The importance of a descriptor is influenced by "the company it keeps." A descriptor may have little "actual discriminatory power" vis-a-vis descriptors that co-occur in a representative retrieval request. For example, let us assume that a certain descriptor say D is used jointly with descriptors:

$A_1A_2A_3A_4$

$B_1B_2B_3B_4$

and

$C_1C_2C_3C_4$

Let us assume that the increment of the retrieval collection due to the deletion of D is in each of the cases from 498 documents to 500 documents. The average "actual discriminatory

power" of the D-descriptor is low.

(c) The average number of descriptors used in retrieval calls containing a given descriptor is an important indicator of the order of importance. Other things being equal, one may expect that a descriptor that co-occurs with large numbers of other descriptors in retrieval requests is of lesser importance than one that co-occurs with few, since the absolute number of documents excluded by the descriptor in the latter case is greater.

These considerations dealt with descriptors as used in retrieval and have to do with the actual usage of descriptors. To distinguish these considerations from those pertaining to the potential usage, the next set of factors deal with factors not directly related to actual usage. These factors are dependent only upon the distribution of descriptors among documents and not with their occurrence in retrieval calls:

(d) The larger the size of a document set spanned by a descriptor, the greater will be its ranking on the importance scale.

(e) Corresponding to the "actual discriminatory power" of a descriptor there is the "potential discriminatory power." The potential discriminatory power of a descriptor measures the uniqueness of its coverage. It is computed in the same way as the actual discriminatory power, except that the descriptor combinations to be considered are not derived from usage statistics. A descriptor will have potential discriminatory power of zero, if any retrieval request involving that descriptor

can be replaced by a different request not involving that descriptor with no change in the set of retrieved documents. On the other hand, if many sets of documents can be retrieved only as small subsets of other retrievable sets when the descriptor in question is not used, then the descriptor has high potential discriminatory power.

(f) A set spanned by a descriptor may intersect sets spanned by closely related descriptors or by sets spanned by descriptors remote from one another. Such characteristics may be called a measure of dispersion of a descriptor. Other things being equal, the more dispersed a descriptor is, the less highly will it rank. This fact is so because with high dispersion in any particular retrieval call the higher proportion of retrieved documents may be expected to be only marginally relevant to the request.

4.2.1.3 Statistical Data Required for the Determination of the Order of Descriptor Importance - Unfortunately not all the factors mentioned in the preceding section can be conveniently measured. For some factors the amount of bookkeeping required is close to astronomical. Therefore, one must take recourse to convenient substitutes that encapsulate the essential information without too much leakage and, at the same time, reduce the requisite amount of data handling and bookkeeping.

The important consideration that has to be kept in mind is that detailed accounts of intradescriptor relationships cannot be kept. For

example with 10,000 descriptors there are $2^{10,000}$ possible combinations or descriptors and if even .01 percent of these are active (i.e., there are some documents that are indexed by them) the number of entries that would have to be retained is astronomical. It is only necessary, therefore, to keep track of selective data on the basis of which the important intradescriptor relationships could be approximately reconstructed.

The most difficult problem will consist of trying to reconstruct the "dispersion" and the "discriminatory power" of the descriptor set. Tentatively, the following set of parameters is suggested as a basis for further study:

(a) Total document span of individual descriptors.

(b) Frequency of recall of individual descriptors.

(c) The number of documents spanned by a given descriptor in company with either k descriptors where k is 1,2,...,n - 1.

(d) The document span of an average descriptor contained in a set of k of them present with a given descriptor.

(e) The frequency of recall of an average descriptor contained in a set of k of them present with any given descriptor.

(f) The number giving an overlap measure of an average descriptor contained in a set of k descriptors present with a given descriptor.

4.2.1.4 Summary - In a collection of n documents, there are 2^n possible subcollections if the empty collection is included. In practice, not all 2^n subcollections are equally likely to be searched for by a user. Any descriptive scheme should be based on this fact and designed in such a way that useful subcollections have simple descriptions. A query to the system selects a particular subcollection. With each

subcollection a measure of accessibility can be associated; this measure indicates the complexity of the query required to retrieve the subcollection. Certain statistical measures have been presented that could be used to measure the value of a descriptor in constructing descriptions; specifically, the concepts of "dispersion" and "discriminatory power" are defined. Some of the data that would be useful in computing these measures have also been described.

4.2.2 Corrective Procedures for Indexing Systems

4.2.2.1 General - This section investigates the methods and feasibility of applying corrective procedures to indexing systems. A fundamental aspect of these concepts is their ultimate adaptability to automated procedures. The first part of this discussion presents the basic ideas of this concept; the second part develops the concept formally.

4.2.2.2 The Taxonomy of Indexing Systems - Information retrieval systems consist of a collection of documents and set of indexing rules and procedures for linking descriptors to documents. The documents in this context refer to the smallest ensemble of information subject to retrieval; these documents are considered as being indivisible. The indexing rules and procedures theoretically select descriptors that bear some relation to the descriptors used by people who will interrogate the system.

The system may accept new documents; the documents are then classified according to the rules and procedures of the indexing scheme

of the system. The system is not necessarily committed to the use of old descriptors. The indexing rules allow for the supply of new descriptors with the acceptance of the new documents by the library.

The user specifies his requests for information by writing a sequence of acceptable descriptors in the form of a Boolean function; that is, the descriptors are joined by OR and AND. The user's disposition of the descriptors implies the existence of an ideal taxonomic system. The taxonomy imposed by the indexing rules and procedures constitutes an external taxonomy or a priori taxonomy.

A corrective procedure will cause the external taxonomy to evolve into the ideal taxonomy on the basis of information concerning the adequacy of the sets of documents retrieved. This information is supplied by the user.

The central problem is: On what factors does the functioning of a corrective procedures depend? The answer to this problem depends upon the elucidation of the relation between the ideal and the external taxonomy. More specifically, the hypothesis depends upon the concept of invariance. Invariance pertains to the a priori postulated constancy between descriptors in the two taxonomies.

This discussion, then, will advance the hypothesis that:

(a) The concept of relatedness of descriptors can be made numerically precise.

(b) The concept of relatedness can serve as a building block for more complex relationships between descriptors.

(c) Some such relationships are postulated as being constant; i.e., these relationships remain invariant in both the external and the ideal taxonomies.

(d) The existence of such constancies forms the basis for selecting rules of reassigning descriptors among documents.

The remainder of this section will attempt to validate this hypothesis and describe the resultant consequences.

4.2.2.3 Formalization of the Hypothesis - Let $d_1, d_2, \ldots, d_n$ and $D_1, D_2, \ldots, D_n$ be descriptors and documents, respectively. For every descriptor there corresponds a class of documents spanned by this descriptor. In set-theoretic notation this concept becomes:

$$[D: \; d(D) = d_i(D)] \tag{1}$$

which may be read as "the set of all documents such that descriptor d_i applies to the set." To avoid cumbersome notation, the abbreviation $[D(d)]$ will be used to represent the set. The number of documents contained in such a set will be denoted by M. Then $M[D(d_i)]$ stands for the number of documents contained in the set spanned by the descriptor d_i.

In general, to every Boolean function of descriptors there corresponds a set of documents spanned by these descriptors. Therefore, "the set of all documents that are indexed by B(d)," becomes:

$$[D(B(d))] \tag{2}$$

For example,

$$[D(d_1 \wedge (d_2 \vee d_3))] \tag{3}$$

is a set of all documents that have as their indices the descriptors d_1 and d_2 or d_3 or both, among others. It is clear that the following relation holds:

$$[D(B(d))] = B[(D(d))] \qquad (4)$$

This expression signifies that the set of all documents spanned by a Boolean function of descriptors is equivalent to the Boolean function of sets spanned by these descriptors. By analogy, the expression $[d(B(D))]$ represents a set of predicates contained in the set of documents described by the Boolean function $B(D)$.

The relatedness of descriptors or their Boolean functions is defined as the number of documents contained in the intersection of classes spanned by these descriptors or their Boolean functions divided by the number of documents spanned by the union. Formally, this definition becomes:

$$R_d[B_i(d), B_j(d)] = \frac{M[B_i(D(d)) \wedge B_j(D(d))]}{M[B_i(D(d)) \vee B_j(D(d))]} \qquad \text{(Definition 1)} \quad (5)$$

A similar concept of the relatedness of documents or their Boolean functions is defined analogously:

$$R_D[B_i(D), B_j(D)] = \frac{M[B_i(d(D)) \wedge B_j(d(D))]}{M[B_i(d(D)) \vee B_j(d(D))]} \qquad \text{(Definition 2)} \quad (6)$$

It is important to note that throughout this discussion the concepts for descriptors can be analogously applied to documents. The subsequent development, however, will be limited to the relatedness of descriptors.

Since the external taxonomy by hypothesis does not precisely correspond to the ideal taxonomy, the distinct symbol, δ, is introduced to represent the descriptors of the user. These descriptors are only different insofar as they index classes of documents that are not identical

with the classes of documents indexed by the descriptors of the external taxonomy. Thus for any descriptor or index i, $[d_i(D)]$ and $[\delta_i(D)]$ are not necessarily identical, even though the descriptors themselves may be the same. The objective of corrective procedures is to adjust the application of descriptors to documents so that the two sets become identical. The corrective procedures may have fulfilled their task if the objective is approximated to the extent that any divergence has a negligible impact upon the user.

4.2.2.4 **The Basis of Corrective Procedures** - Assume that all retrieval requests consist of single descriptors. The user formulates his request in terms of a descriptor δ_i related to the ideal taxonomy. The system retrieves all documents spanned by this descriptor, except that this descriptor is d_i in the external taxonomy. The user then decides whether the retrieved collection of documents is satisfactory. The collection may not satisfactorily fulfill the user's requirements for three reasons:

(a) Too many documents were collected.

(b) Too few documents were collected.

(c) Some documents are superfluous and some are missing.

The corrective procedures should select documents more in consonance with user's needs and then effect permanent changes in the application of descriptors to documents.

If the system retrieves too many documents, the system may select a set of descriptors that are most related to the user's descriptor

and then remove from the retrieved set those documents spanned by the related descriptors. This method conceals a difficulty. Although a measure for relatedness of two descriptors has been defined, no technique has yet been specified to select clusters of most related descriptors.

If the system retrieves too few documents, a set of descriptors most closely related to the given descriptor is assembled; the set may be limited to a single descriptor. A Boolean function of these descriptors is then constructed, and documents spanned by the Boolean function are retrieved. The factors that determine the nature of the particular Boolean function of descriptors must still be defined.

If some documents are superfluous and some are missing, the problem may be handled as a combination of the specific problems of too many or too few documents. More realistically, however, some problems of this type are sui generis, and specific solutions must be developed.

After the originally inadequate set of documents is deleted to the satisfaction of the user, the corrective procedures must effect permanent changes in the extension of some descriptors so that the denotation of the external and ideal descriptors approach equivalence. The problem is to render the sets $[\delta_i(D)]$ and $[d_i(D)]$ extensionally as similar as possible. Several corrective procedures may be used:

(a) To affix the user's descriptor to all the documents and only those documents in the acceptable retrieved set.

(b) To delete or add some descriptors selectively from the set of documents spanned; after the process of deletion or augmentation.

(c) To delete or add some descriptors selectively to the documents that were deleted or complemented from the originally inadequate retrieved set.

(d) To effect other descriptor changes on the document not affected by the processes of complementation or deletion.

The first procedure by itself will not produce the desired transformation until all descriptors have been used in retrieval processes at least once. This prospect is uninviting for any document collection with a large number of descriptors. If such procedure were feasible, there would be no reason not to index the entire collection in the ideal taxonomy, in the first place. In addition, the procedure of complementing the original set of documents need not necessarily lead to the formation of a taxonomy whose extension is identical to the ideal. Rather, the process may only be an approximation; that is, a set obtained after a series of complementations may only approximate the ideal taxonomy.

A closer look at the remaining three procedures and their inherent problems is necessary. Consider a class of documents $[D(d_i)]$ spanned by descriptor d_i. Suppose that the user requests all documents under the descriptor δ_i, a descriptor corresponding to d_i. The class $[D(d_i)]$ is retrieved; it does not fulfill the user's requirements. The complementation procedure results in formation of a new class $[D'(d_i)]$. The corrective procedure should then implement changes pertaining to the distribution of the remaining descriptors among documents. How should these changes be made? Or, to rephrase this question, on what should the inferential processes be based in order to ensure that the ideal taxomony is approximated?

Assume that there is no relation between the external and the ideal taxonomies. In this case the first stage of the corrective procedure--that is, the complementation of the selected set--must proceed at random. If the taxonomy imposed upon the collection of documents is not correlated with the taxonomy implied by the user, then the relatedness of descriptors to one another will be of no help either in reassigning descriptors or in complementing the original sets.

The possibility of developing corrective procedures depends, therefore, upon some a priori relation between the two taxonomic systems. If such relationships exist, then it must be expressible in terms of the concept of relatedness. The relatedness of descriptors, in one system, must resemble the relatedness in the other. The concept of a relatedness between two taxonomic systems isolates the particular invariance that characterizes the sets of documents designated by certain descriptors. Formally, an invariance exists if $d_i R d_j$ is true whenever $\delta_i R \delta_j$ is true, where R is a relationship between descriptors. There need not be some universal type of invariance present whenever there is a resemblance between two taxonomic systems. On the contrary, depending upon the nature of the data to be retrieved, the invariance between the ideal and the external taxonomy may differ.

Some formal examples may clarify the concept of invariance. First, if a set of documents spanned by a descriptor in one system contains another set of documents spanned by another descriptor and if this condition implies the same condition for the corresponding descriptors

in the other system, then the invariance might be called nested invariance. Formally:

$$[D(d_j)] \supset [D(d_k)] \rightarrow [D(\delta_j)] \supset [D(\delta_k)] \qquad (7)$$

where $\rightarrow$ indicates "implies," and $\supset$ indicates set inclusion.

In a second example the most closely related descriptors in one system are also most closely related in another. To represent this type of invariance formally, let $(d_i, d_j)^*$ be an ordered pair of descriptors that are related to each other as follows:

$$R_d[(d_i), (d_j)] = \text{Max } R_d[(d_i), (d_k)] \text{ (for all k)} \qquad (8)$$

If then $(d_i, d_j)^* \rightarrow (\delta_i, \delta_j)^*$, the relationship of being most closely related is preserved.

The third example replaces MAX by MIN to obtain an invariance of being the least closely related descriptor. In spite of the formal similarity between the most and least closely related conditions, there is a formidable practical difference. The most closely related condition preserves an invariance between a descriptor and a descriptor; the least closely related condition preserves an invariance between a descriptor and a class of descriptors.

As a fourth example the concept of most closely related descriptors may be applied to chains of descriptors. In such a relationship one descriptor leads to another to form an associative chain. There are many non-equivalent ways of formulating the conditions for the existence of such a chain. One is to let $< d_1, d_2, \ldots, d_n >$ be an associative chain of

n^{th} order. Then this chain is defined as:

(a) The set $[d_1, d_2, \ldots, d_n]$ of descriptors comprised in the chain contains each element except the first and the last only once.

(b) The first element appears twice; it is also the last element. The first and last elements are linked to complete the chain.

(c) Each element except the first determines its successor by selecting the second most related descriptor. The first descriptor determines its successor by selecting its most related neighbor.

Then, if every associative chain of n^{th} order in one taxonomic system corresponds to a chain in another, a chain invariance of n^{th} order exists. The elements in one chain correspond to the elements in the other, but not necessarily in the same order.

There are a number of additional possible relationships that remain invariant. The problem is to select those that realistically relate to the properties of data structures and their associated indexing systems.

If these invariances exist, formal rules for reassigning the descriptors may be deduced. The concept of invariance places a strong constraint upon the type of admissible rules that can be formulated. There is also a relation between the invariances and the nature of the convergence and efficiency criteria imposed upon the corrective procedures. The important question is: Given a specific form of invariance and the appropriate rules for complementing sets and for reassigning descriptors, how many queries must elapse before the external taxonomy approximates the ideal? (Approximation in this sense may mean either the probability of obtaining a set that is too small or too large by a specified margin.)

A comparison between one type of invariance and another now becomes possible. These invariances that result in a quick convergence of the corrective procedures are desirable. Conversely, it is possible to investigate the suitability of rules for complementing and reassigning descriptors by keeping a set of invariant relationships constant. All these problems can be investigated mathematically.

4.2.2.5 Summary - There is an inherent problem in accommodating the descriptors selected for a set of documents by indexing rules to the descriptors used by the user of a system. This problem is related to the extensional difference in the denotation of descriptors or words in an external and an ideal taxonomy. This discussion described methods for developing corrective procedures, which could be applied automatically, to relate the external to the ideal taxonomy. The basis for developing the inferential rules for these procedures is the concept of invariance.

4.3 ASSIGNMENT OF DESCRIPTORS TO DOCUMENTS

The major work performed on the assignment of descriptions to documents has been on the development of automatic indexing methods based on clue words. The approach assumes that an initial set of categories has been set up by a group of human experts and that there is available a test body of documents that can be used to extract the basic parameters used in automatic indexing. The basic thesis of this approach is that the occurrence of certain words in a document indicates the correct categorization of that document; i.e., the descriptor most appropriate to it. A variation on this approach is the use of game-theoretic methods to find

those clue words that maximize the probability of correct classification; using this approach, the choice of clue words determines the choice of a classification algorithm.

The description assigned to a document need not be composed of descriptors in the usual sense. Automatic abstracting provides a technique for generating more logically complex document descriptions; the descriptive language in this case has all the richness of human language. Some investigations in this area are described below.

4.3.1 Information Theoretical Methods of Document Categorization - This section presents some applications of information theory to the problem of document classification or categorization. Criteria for a good categorizer are presented, and various information theoretical measures that measure the goodness of categorizers are examined.

The problem of document categorization is the problem of selecting from a set of possible categories those categories to which a document may belong. This selection would have to be based upon certain clues or indications found in the document itself. Thus, as Maron [47] has stated, the problem of categorization can be divided into two parts: the selection of certain relevant aspects of a document as clues toward classification; and the use of these clues to predict the proper category to which the document belongs. Once the method of classification has been defined, then the procedures could be automated.

Many authors [3, 4, 8, 16, 44, 54, 67] have felt that the occurrence

of certain words in a document provided excellent indications of the category to which that document belonged. Based upon word occurrence statistics, document categories would be predicted automatically, This approach is also developed here, but certain information theoretical techniques are applied that do not appear to have been applied elsewhere.

This approach assumes that a group of human experts will initially classify a number of documents into a given set of categories. A basic assumption is that all categories that receive one or more documents will be retained as permanent categories, which will be the only categories used in the future. Another assumption is that the number of documents initially classified by experts is large enough so that the statistics of this group may be assumed to reflect the statistics of the body of documents that may later be automatically categorized. In other words, relative frequencies of categorization obtained from the initial group will be used as the probabilities of categorization of the larger group.

4.3.1.1 Basic Approach to Automatic Classification Using Word Occurrence Information

4.3.1.1.1 Criteria for Selecting Predictors - It is expected that the occurrence of certain words in a document indicates the categorization of that document. It follows that one of the criteria for selecting a particular word to predict categories is that its occurrence in documents be strongly correlated with the appearance of those documents in a particular category--for those documents that were initially classified. In other words, a word that appears in every document of a

particular category and appears in no document of any other category seems to be an ideal predictor of that category. In practice there may be few of these ideal predictors; that it is necessary to look for words for which occurrence in a document means a particular category for that document is much more likely than any other category.

This criterion would be sufficient for choosing indicator words if the distribution of documents in the categories were uniform. In practice, this condition would generally not be the case; some categories would have many more documents than others. Then a word that would seem to be an excellent indicator might be found to supply no more information than the total distribution of documents supplied. Thus the occurrence of the good indicator word in documents must not only be strongly correlated with the classification of these documents in one particular category, but the distribution of documents containing this word must also markedly differ from the distribution of all the documents.

4.3.1.1.2 Mathematical Statement of the Problem - The problem can now be expressed mathematically: Given N documents* classified into C_j categories, where $j = 1,\ldots,k$. The vocabulary of the N documents contains m words, W_i, $i = 1,\ldots,m$. Word W_i occurs in N_i documents, and n_{ij} of these documents fall into category C_j.

Let:

$p(C_j)$ = the probability that a document falls into category C_j

*The classification of a document into two or more categories is counted as the classification into one category each of two or more documents.

$p(C_j|W_i)$ = the probability that a document with the word W_i falls into category C_j.

Then: $$p(C_j) = p_j = n_j/N \tag{9}$$

and: $$p(C_j|W_i) = p_{ij} = n_{ij}/N_i \tag{10}$$

The following relationships hold by definition:

$$\left.\begin{aligned} &\sum_j n_{ij} = N_i \\ &\sum_j n_j = N \\ &\sum_j p_j = \sum_j p_{ij} = 1 \end{aligned}\right\} \tag{11}$$

It has been assumed that there exists at least one document in each category; i.e., the smallest possible $p_j = 1/N$. If there were no documents in a category C_e, then p_e would be zero; consequently all the p_{ie} would be zero. Such a category would be of no use and would be discarded. Having at least one document in each category also implies that $k \leq N$, and that the largest possible $p_j = 1 - \frac{k-1}{N}$, for there are $k - 1$ categories that would have to have the minimum p_j. Therefore:

$$\left.\begin{aligned} &\frac{1}{N} \leq p_j \leq 1 - \frac{k-1}{N} \\ \text{and:}\quad &0 \leq p_{ij} \leq 1 \end{aligned}\right\} \tag{12}$$

4.3.1.1.3 Definitions of Measures of Goodness - The non-correlation of word occurrence and category or the uncertainty of category, given the occurrence of a word W_i, can be expressed by Shannon's formula for entropy:

$$H_i = H(C_j|W_i) = -\sum_j p_{ij} \log p_{ij} \tag{13}$$

Thus a good indicator word would have a low H_i. But is this word supplying more information than the total document distribution? Maron suggests a measure:

$$M_1 = H - H_i \tag{14}$$

where: $$H = H(C_j) = - \sum_j p_j \log p_j \tag{15}$$

H is simply the uncertainty of categorization when no word occurrences are known; that is, H is the entropy of the a priori distribution of all of the documents.

This measure, however, does not seem adequate. Difficulty arises when the a priori p_j are unequal and have the same numerical value as the p_{ij} of different categories. In this case, $H = H_i$ and $M_1 = 0$, which indicates a poor predictor; but W_i may actually be a good predictor in terms of the given criteria. The example in Figure 2 illustrates this difficulty. Clearly $H = H_r$ and $M_1 = 0$ in Figure 2, but W_r is a good predictor and supplies a great deal of information.

More effective measures of the adequacy of an indicator word can be based on a relative entropy function of the type found in Watanabe [84]. This function is similar to the previous entropy functions, but it accounts for the a priori probabilities directly. The relative entropy, S_i, is defined by:

$$S_i = S(C_j|W_i) = - \sum_j p_{ij} \log \frac{p_{ij}}{A p_j} \tag{16}$$

where A is a positive constant chosen to keep S_i non-negative. A should be chosen such that $A = 1/p_e$, where $p_e \leq p_j$ for all j, so that $S_{imin} = 0$.

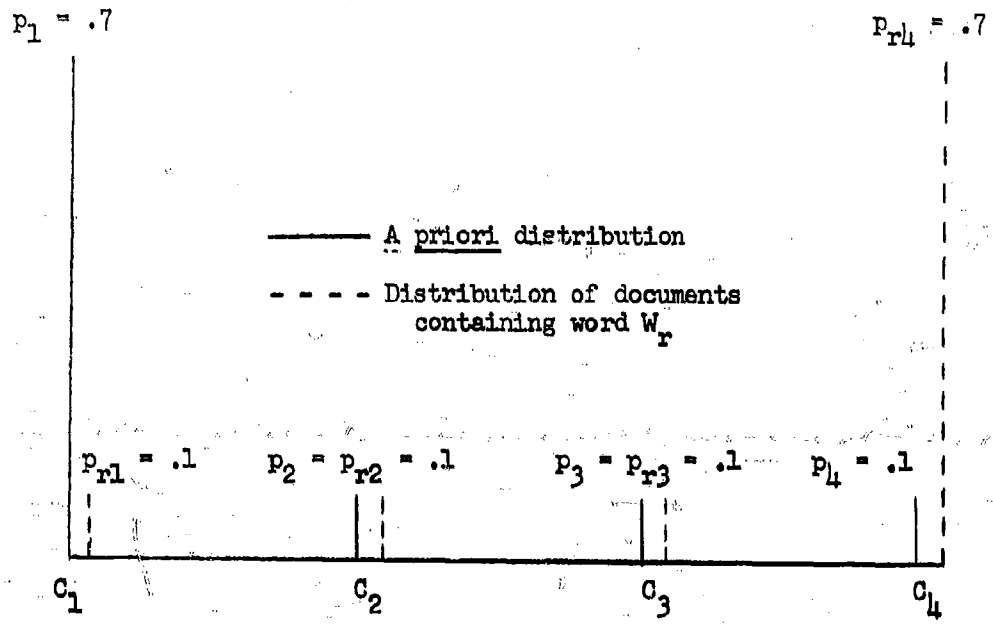


FIGURE 2. Probability Distributions for a Class of Documents

This condition means that $k \leq A \leq N$, since $1/N \leq p_e \leq 1/k$.

Before these measures are defined and examined, one more entropy function must be defined:

$$H_A = -\sum_j p_j \log p_j/A = H + \log A \tag{17}$$

Three possible measures will now be defined, in addition to the measure M_1 that Maron has suggested.

$$\left.\begin{aligned} M_1 &= H - H_i && \text{(Maron's measure)} \\ M_2 &= H - S_i \\ M_3 &= H_A - S_i \\ M_4 &= \log A - S_i \end{aligned}\right\} \tag{18}$$

Now:

$$\left.\begin{aligned} M_2 &= H - H_i - \sum_j p_{ij} \log p_j - \log A \\ M_3 &= H - H_i - \sum_j p_{ij} \log p_j = M_2 + \log A \\ M_4 &= -\sum_j p_{ij} \log p_j - H_i = \sum_j p_{ij} \log \frac{p_{ij}}{p_j} \end{aligned}\right\} \quad (19)$$

The new M_2 and M_3 are similar to M_1, except for a cross-term that relates the p_j and the p_{ij}. M_4 also has this cross-term. M_3 is simply M_2 with the constant term missing. The behavior of these measures of goodness and the various entropy functions are developed in Appendix A, Section 9.1.

4.3.1.1.4 Evaluation of the Measures - Measure M_1 was shown to be inadequate, since it may erroneously indicate that a good predictor is a bad predictor. In addition, M_1 can assume negative values. M_2 can also assume negative values, which may make it inconvenient to use. M_2 is also inconvenient to calculate, since it requires the calculation of two sums, $\sum_j p_j \log p_j$ and $\sum_j p_{ij} \log \frac{p_{ij}}{p_j}$, and since the last summation also includes a division operation. M_3 requires the calculation of these same sums, although it is slightly more convenient to use since M_3 is always positive. M_1, M_2, and M_3 have fairly complex expressions for maxima and minima; M_1 and M_2 become negative and M_3 never reaches zero. M_4, on the other hand, is always positive, has a simple expression for the maximum, has a zero minimum, and is easier to calculate than the others.

An additional argument in favor of M_4 is that it can be justified on the basis of more fundamental definitions of information. This

justification can proceed in either of two ways; on the basis of probabilities or on the basis of entropies. In either case, M_4 can be shown to be the amount of information provided by the occurrence of word i in a document. The proofs are given in Appendix B, Section 9.2.

It seems clear, then, that M_4 is the best measure of the group, both in terms of ease of calculation and in terms of theoretical justification. For these reasons, it will be adopted as one of the two basic measures for category prediction; since there is a different M_4 for each word, the notation M_i will be used instead of M_4 to indicate the dependence of the measure on the particular word being considered.

4.3.1.1.5 Mathematical Expression of Predictor Criteria - The correlation of the occurrence of an indicator word in a document and the classification of that document in a particular category would be measured by H_i.

$$H_i = -\sum_j p_{ij} \log p_{ij} \qquad (0 \leq H_i \leq \log k) \qquad (20)$$

A low H_i indicates a good predictor; a high H_i, a bad predictor.

A measure that also accounts for the a priori distribution of documents and indicates how much more information the predictor supplies than this distribution is M_i.

$$M_i = \sum_j p_{ij} \log \frac{p_{ij}}{p_j} \qquad (0 \leq M_i \leq -\log p_e) \qquad (21)$$

$$(1/N \leq p_e \leq 1/k)$$

A high M_i indicates a good predictor; a low M_i, a bad one. Both of these measures must be taken into account when choosing indicator words.

4.3.1.1.6 Predictors - On the basis of these mathematical criteria, it is now possible to select clues or predictors. A word that has a high value for M_i and a low value for H_i will be selected. The cutoff point for these functions for good predictors must be determined experimentally. It is difficult to say how high a value for M_i or how low a value for H_i is actually needed for a good predictor without empirical verification.

Not only can single words be used as predictors, but word pairs, word triplets, and higher word combinations can also be used with an expected improvement in prediction. The mathematics for these cases is essentially the same; the only difference is that the occurrence of word pair $[W_a\ W_b]$ or word triplet $[W_a\ W_b\ W_c]$ is considered instead of the single word W_i. These word pairs and word triplets can be ranked together with single words on the same scale, and their effectiveness as predictors can then be compared.

4.3.1.1.7 Application of Clues to Predicting Categories - Once the significant predictors have been determined, it is possible to obtain the probability that a document appears in a category on the basis of those predictors. This probability is:

$$p(C_j|W_a\ W_b.....) \qquad (22)$$

Maron gives an approximation to this probability. In general, this approximation would require a great deal of calculation. One way of approximating the probability would be to take the weighted average

of the category probabilities using each of the most significant indicator words. Other functions of these words might also approximate the probability. Thus, in general, the predicted category would be some function of the category probabilities for each of the words. Methods for determining suitable functions of this kind should be investigated.

4.3.1.1.8 Modification of Categories - Implied in this discussion are criteria for modifying and combining categories to get better classification. What is needed is a set of categories that would be strongly correlated with word occurrence and that would yield approximately equal a priori category probabilities. In this way, there would be words with high M_i and low H_i. In fact, these two measures would then be almost the same; for if $p_j \doteq 1/k$ for all j, then:

$$M_i \doteq \sum_j p_{ij} \log p_{ij} + \log k = \log k - H_i \qquad (23)$$

Thus in equalizing the categories, if for some W_i, M_i is high and there exists at least one such W_i for each category, then the classification would be a good one.

4.3.1.2 Extension of Concepts to Include Word Frequency Information - There are several ways in which word frequency information can be taken into account to determine good predictors of document categories. The first two methods use absolute values of word occurrence in a document, while the third method uses relative word frequency in a document to obtain more information.

4.3.1.2.1 Additional Definition - Let:

N = the total number of documents in the initial group.

N_i = the number of documents in which word W_i occurs.

$N_i(x)$ = the number of documents in which word W_i occurs x times.

n_j = the number of documents in category C_j

n_{ij} = the number of documents in category C_j which have word W_i.

$n_{ij}(x)$ = the number of documents in category C_j which have word W_i x times.

Now:

$$\left.\begin{aligned} N_i &= \sum_x N_i(x) \\ n_{ij} &= \sum_x n_{ij}(x) \end{aligned}\right\} \quad (24)$$

In addition to the probabilities p_{ij} and p_j, the following probabilities can be defined. Let:

p_i = the probability that a document contains word W_i.

$p_i(x)$ = the probability that a document contains word W_i x times.

$p_{ij}(x)$ = the probability that a document containing word W_i x times falls into category C_j.

$p(C_j,W_i)$ = the joint probability that a document is in category C_j and contains word W_i.

$p[C_j,W_i(x)]$ = the joint probability that a document is in category C_j and contains word W_i x times.

Then the probabilities can be approximated as follows:

$$\left.\begin{aligned} p_i &= \frac{N_i}{N} \\ p_j &= \frac{n_j}{N} \end{aligned}\right\} \quad (25)$$

$$p_{ij} = \frac{n_{ij}}{N_i}$$

$$p_i(x) = \frac{N_i(x)}{N}$$

$$p_{ij}(x) = \frac{n_{ij}(x)}{N_i(x)}$$

$$p(C_j,W_i) = \frac{n_{ij}}{N}$$

$$p[C_j,W_i(x)] = \frac{n_{ij}(x)}{N}$$

Of course:

$$p_i = \sum_x p_i(x)$$

$$p(C_j,W_i) = \sum_x p[C_j,W_i(x)] \qquad (26)$$

and $p_{ij}(x)$ is related to p_{ij} by the expression:

$$p_{ij} = \frac{\sum_x p_{ij}(x)\, N_i(x)}{\sum_x N_i(x)} \qquad (27)$$

4.3.1.2.2 Derivation of Measures

(a) Method 1 - The measures H_i and M_i can easily be generalized to include frequency information by considering word W_i occurring x and only x times in a document as a clue. Then, instead of using p_{ij} in H_i and M_i, a new probability $p_{ij}(x)$ can be used. Two new measures, $H_i(x)$ and $M_i(x)$, can now be defined:

$$\left.\begin{aligned} H_i(x) &= -\sum_j p_{ij}(x) \log p_{ij}(x) \\ M_i(x) &= \sum_j p_{ij}(x) \log \frac{p_{ij}(x)}{p_j} \end{aligned}\right\} \qquad (28)$$

With these measures, the effectiveness of word W_i as a predictor, when it occurs x times in a document, can be evaluated. As before, $H_i(x)$ must be low and $M_i(x)$ must be high for a good predictor.

The average effectiveness of a word W_i as a predictor can be measured by:

$$\left.\begin{aligned} \overline{H_i(x)} &= \langle H_i(x) \rangle_x \\ \overline{M_i(x)} &= \langle M_i(x) \rangle_x \end{aligned}\right\} \qquad (29)$$

where $\langle f(z) \rangle_z$ denotes the probabilistically weighted average value of the function f over its domain. Then, on the basis of Equations (25) and (26), it follows that:

$$\overline{H_i(x)} = \frac{\sum_x p_i(x)\, H_i(x)}{\sum_x p_i(x)} \qquad (30)$$

and;

$$\overline{H_i(x)} = -\frac{1}{p_i} \sum_x \sum_j p[C_j, W_i(x)] \log p_{ij}(x) \qquad (31)$$

Similarly:

$$\overline{M_i(x)} = \frac{\sum_x p_i(x)\, M_i(x)}{\sum_x p_i(x)} \qquad (32)$$

But:

$$M_i(x) + H_i(x) = - \sum_j p_{ij}(x) \log p_j \tag{33}$$

therefore;

$$\langle M_i(x) + H_i(x) \rangle_x = \overline{M_i(x)} + \overline{H_i(x)}$$

$$= - \frac{1}{p_i} \sum_x \sum_j p[C_j, W_i(x)] \log p_j$$

$$= - \frac{1}{p_i} \sum_j p(C_j, W_i) \log p_j \tag{34}$$

and, by substituting Equation (25);

$$\overline{M_i(x)} + \overline{H_i(x)} = - \sum_j p_{ij} \log p_j \tag{35}$$

But:

$$M_i + H_i = - \sum_j p_{ij} \log p_j \tag{36}$$

therefore;

$$\overline{M_i(x)} + \overline{H_i(x)} = M_i + H_i \tag{37}$$

(b) Method 2 - This method is similar to Method 1. Instead of considering that a word occurs exactly x times in a document, this method considers that a word occurs between x_a and x_b times in a document. In other words, word frequency information is grouped in intervals of frequency of occurrence, B_r. For example, the frequency intervals might be 1-5 times, 6-10 times, etc.

New probabilities must be introduced. Let:

$p_i(B_r)$ = the probability that a document contains word W_i x times, where x is in interval B_r.

$p_{ij}(B_r)$ = the probability that a document containing word W_i x times falls into category C_j, where x is in interval B_r.

$p[C_j, W_i(B_r)]$ = the joint probability that a document is in category C_j and contains word W_i x times, where x is in interval B_r.

Now the probabilities can be expressed as:

$$\left.\begin{aligned} p_i(B_r) &= \sum_{x \in B_r} p_i(x) \\ p[C_j, W_i(B_r)] &= \sum_{x \in B_r} p[C_j, W_i(x)] \\ p_{ij}(B_r) &= \frac{\sum_{x \in B_r} p_{ij}(x)\, N_i(x)}{\sum_{x \in B_r} N_i(x)} \\ &= \frac{\sum_{x \in B_r} p[C_j, W_i(x)]}{\sum_{x \in B_r} p_i(x)} \end{aligned}\right\} \quad (38)$$

Then, following Method 1 and Equation (28), expressions may be written for $H_i(B_r)$ and $M_i(B_r)$.

$$\left.\begin{aligned} H_i(B_r) &= -\sum_j p_{ij}(B_r) \log p_{ij}(B_r) \\ M_i(B_r) &= \sum_j p_{ij}(B_r) \log \frac{p_{ij}(B_r)}{p_j} \end{aligned}\right\} \quad (39)$$

$H_i(B_r)$ should be low and $M_i(B_r)$ should be high for a good predictor.

Another set of functions that measure the effectiveness of word W_i as a predictor, when W_i occurs x times and x is in interval

B_r, can be obtained by taking the average values of $H_i(x)$ and $M_i(x)$ over the interval B_r. The average effectiveness is measured by:

$$\left.\begin{aligned} \overline{H_i(x,r)} &= \langle H_i(x) \rangle_{x \in B_r} \\ \overline{M_i(x,r)} &= \langle M_i(x) \rangle_{x \in B_r} \end{aligned}\right\} \qquad (40)$$

Then, by using Equation (33) as in Method 1:

$$\langle H_i(x) + M_i(x) \rangle_{x \in B_r} = - \frac{\sum_{x \in B_r} p_i(x)\, p_{ij}(x) \log p_j}{\sum_{x \in B_r} p_i(x)}$$

$$= - \frac{1}{p_i(B_r)} \sum_j p[C_j, W_i(B_r)] \log p_j$$

$$= - \sum_j p_{ij}(B_r) \log p_j \qquad (41)$$

But:

$$H_i(B_r) + M_i(B_r) = - \sum_j p_{ij}(B_r) \log p_j \qquad (42)$$

therefore;

$$H_i(B_r) + M_i(B_r) = \langle H_i(x) + M_i(x) \rangle_{x \in B_r}$$

$$= \overline{H_i(x,r)} + \overline{M_i(x,r)} \qquad (43)$$

If this quantity $[H_i(B_r) + M_i(B_r)]$ is averaged over all r, then by the proof outlined for Method 1:

$$H_i + M_i = \overline{H_i(x)} + \overline{M_i(x)}$$

$$= \overline{H_i(B_r)} + \overline{M_i(B_r)}$$

$$= \langle \overline{H_i(x,r)} \rangle_r + \langle \overline{M_i(x,r)} \rangle_r \qquad (44)$$

Thus the sum of the averages of the two measures remains constant and is independent of the size of the intervals or frequency of occurrence.

(c) Method 3 - This method considers the number of times a word appears in a document in relation to the total number of words in a document as a clue. Using this relative frequency information as clues should provide even better category prediction than word occurrence or simple word frequency information.

Let f be the relative frequency of a word in a document; the relative frequency is the ratio of the number of occurrences of the word in the document to the total number of words in the document. Let f_s be an interval of relative frequencies, where the interval is defined by the limits f_a and f_b. Then, $p_i(f_s)$ is simply the probability of word W_i occurring in a document with a relative frequency in the interval f_s, and $p_{ij}(f_s)$ is the probability that a document falls in category C_j, given that the document contains word W_i with a relative frequency within the interval f_s.

The probabilities $p_i(f_s)$ and $p_{ij}(f_s)$ are approximated by:

$$\left.\begin{aligned} p_i(f_s) &= \frac{N_i(f_s)}{N} \\ p_{ij}(f_s) &= \frac{n_{ij}(f_s)}{N_i(f_s)} \end{aligned}\right\} \qquad (45)$$

where $N_i(f_s)$ is the number of documents containing word W_i with a relative frequency within the interval f_s, and $n_{ij}(f_s)$ is the number of documents in category C_j containing word W_i with a relative frequency within the interval f_s.

Following the previous analyses, expressions for $H_i(f_s)$ and $M_i(f_s)$ can be written:

$$\left.\begin{aligned} H_i(f_s) &= -\sum_j p_{ij}(f_s) \log p_{ij}(f_s) \\ M_i(f_s) &= \sum_j p_{ij}(f_s) \log \frac{p_{ij}(f_s)}{p_j} \end{aligned}\right\} \qquad (46)$$

By analogy to the proofs developed for Methods 1 and 2, $\overline{M_i(f_s)} + \overline{H_i(f_s)}$ can be calculated where:

$$\left.\begin{aligned} \overline{H_i(f_s)} &= \langle H_i(f_s) \rangle_s \\ \overline{M_i(f_s)} &= \langle M_i(f_s) \rangle_s \end{aligned}\right\} \qquad (47)$$

Since, as compared to Equation (33):

$$M_i(f_s) + H_i(f_s) = -\sum_j p_{ij}(f_s) \log p_j \qquad (48)$$

then;

$$\langle M_i(f_s) + H_i(f_s) \rangle_s = \overline{M_i(f_s)} + \overline{H_i(f_s)}$$

$$= - \sum_j p_{ij} \log p_j$$

$$= M_i + H_i \qquad (49)$$

Therefore, as before:

$$\overline{M_i(f_s)} + \overline{H_i(f_s)} = M_i + H_i \qquad (50)$$

One of the major experimental problems is the proper selection of frequency intervals to evaluate. For some areas of the relative frequency spectrum a small change in interval size might lead to a large change in effectiveness; for other areas of the spectrum, however, changing the interval might have a negligible effect on effectiveness. These intervals will in general not be uniform over the spectrum and will be different for each word. Although this selection and evaluation appears difficult, it will lead to better category prediction.

4.3.1.2.3 Improvement in Effectiveness - We have previously defined and used measures to indicate the improvement in effectiveness of prediction using word or word frequency information rather than simple category statistics alone. Instead of evaluating word frequency information with respect to simple category statistics, this information may be evaluated with respect to word occurrence information. This new measure would indicate how much more information the word frequency information supplied than the word occurrence information. Call this measure $M_{ia}(x)$ where

$$M_{ia}(x) = \sum_j p_{ij}(x) \log \frac{p_{ij}(x)}{p_{ij}} \tag{51}$$

Now:

$$M_{ia}(x) = -\sum_j p_{ij}(x) \log p_{ij} - H_i(x) \tag{52}$$

but this relationship does not seem very meaningful. The important relationships would relate $M_{ia}(x)$ with $M_i(x)$ and M_i. Consider the quantity $M_i(x) - M_{ia}(x)$. Now

$$M_i(x) - M_{ia}(x) = \sum_j p_{ij}(x) \log \frac{p_{ij}(x)}{p_j} - \sum_j p_{ij}(x) \log \frac{p_{ij}(x)}{p_{ij}}$$

$$= \sum_j p_{ij}(x) \log \frac{p_{ij}}{p_j} \tag{53}$$

Let us now take the average of this quantity over x. Then:

$$\langle M_i(x) - M_{ia}(x) \rangle_x = \frac{\sum_x p_i(x) \sum_j p_{ij}(x) \log \frac{p_{ij}}{p_j}}{\sum_x p_i(x)} \tag{54}$$

where

$p_i(x)$ = the probability that a document contains word W_i x times.

Now:

$$\sum_x p_i(x) = p_i \text{ and } \sum_x p_i(x)\, p_{ij}(x) = p(C_j, W_i) \tag{55}$$

where

p_i = the probability that a document contains word W_i, and

$p(C_j, W_i)$ = the joint probability that a document is in category C_j and contains word W_i.

Thus:

$$\langle M_i(x) - M_{ia}(x) \rangle_x = \frac{1}{p_i} \sum_j p(C_j,W_i) \log \frac{p_{ij}}{p_j} \tag{56}$$

But;

$$\frac{p(C_j,W_i)}{p_i} = p_{ij} \tag{57}$$

Then:

$$\langle M_i(x) - M_{ia}(x) \rangle_x = \sum_j p_{ij} \log \frac{p_{ij}}{p_j} = M_i \tag{58}$$

Therefore;

$$\langle M_i(x) - M_{ia}(x) \rangle_x = \overline{M_i(x)} - \overline{M_{ia}(x)} = M_i \tag{59}$$

Now M_i is a measure of the information that W_i supplies about the categorization. In addition,

$$\overline{M_i(x)} = \langle \log \frac{p_{ij}(x)}{p_j} \rangle_{x,j} = \langle \log \frac{p(C_j,W_i(x))}{p_i(x)\ p_j} \rangle_{x,j} \tag{60}$$

where

$p(C_j,W_i(x))$ = the joint probability that a document is in category C_j and contains word W_i x times.

$\overline{M_i(x)}$ closely resembles an information function, and is a measure of the average information that W_i occurring x times supplies about the categorization. Now:

$$M_{ia}(x) = \langle \log \frac{p_{ij}(x)}{p_{ij}} \rangle_{x,j} = \langle \log \frac{p(C_j,W_i(x))}{p_i(x)\ p_{ij}} \rangle_{x,j} \tag{61}$$

$\overline{M_{ia}(x)}$ then represents the average information that W_i occurring x times supplies about the categorization, knowing that the document

contains W_i at least once, i.e., $\overline{M_{ia}(x)}$ represents the average information that word frequency information supplies above the word occurrence information. Thus the equation:

$$\overline{M_i(x)} = M_i + \overline{M_{ia}(x)} \tag{62}$$

can be expressed verbally as follows:

$$\left\{\begin{array}{l}\text{Average information about } C_j \text{ supplied by word frequency information.}\end{array}\right\} = \left\{\begin{array}{l}\text{Information about } C_j \text{ supplied by word occurrence information.}\end{array}\right\} + \left\{\begin{array}{l}\text{Average information about } C_j \text{ supplied by word frequency information when word occurrence is known.}\end{array}\right\}$$

This equation satisfies our intuitive notions about information and the additivity of information. The equation justifies to some extent the choice of the particular information measures M_i, $M_i(x)$, and $M_{ia}(x)$.

In a similar manner, the equation for the relative frequency case can be developed. Therefore;

$$\overline{M_i(f_s)} = \overline{M_{ia}(f_s)} + M_i \tag{63}$$

where f_s indicates an interval of relative frequencies.

4.3.2 Game-Theoretic Aspects of Clue Word Selection - The motivation for the work described in this section arose from a consideration of the role of clue word selection in an operational document classification system. In such a system, the problem is to optimize the probability of correct classification; among the parameters that can be varied are the number of clue words chosen, the particular algorithm used for employing

these words, and the particular words used. The principal constraints are the cost of processing and the amount of information available about the relationship of the clue words to categories. The information-theoretic approach described in the previous reports presents a method of ranking clue words relative to each other in terms of information-theoretic measures. The game-theoretic approach yields somewhat more specific advice on how to choose clue words, but the necessary data for the choice seems to be far more obscure. The best one would hope for would be a game-theoretic justification of the information-theoretic measures, in which a maximum payoff would be achieved by maximizing (or minimizing) a function whose arguments are information-theoretic measures. In an attempt to achieve this goal we have analyzed a number of specific cases, which are described below.

4.3.2.1 The Approach - Consider the classification problem to be a two-person game in which nature is one of the players. Further consider that the probability that nature is in a particular state in this game is known. There are a number of acts that the player may perform and associated with each act, for each particular state of nature, there is a certain utility. Let:

State C_j = the document is in category j

Act A_r = put the document into category r

Utility u_{rj} = the utility of act A_r when nature is in state C_j

Thus we have a k x k utility matrix of the following form:

	C_1	C_2	. . .	C_k
A_1	u_{11}	u_{12}	. . .	u_{1k}
A_2	u_{21}	u_{22}	. . .	u_{2k}
.	.	.	. . .	.
A_k	u_{k1}	u_{k2}	. . .	u_{kk}

Let $p(C_j)$ be the probability that nature is in state C_j (nature plays a mixed strategy, playing C_j with probability $p(C_j)$). Then the utility of act A_r, $U(A_r)$, can be written:

$$U(A_r) = \sum_j p(C_j)u_{rj} \tag{64}$$

Utilities would be calculated for each act, and the act with the highest utility performed.

Consider a further addition to our model. Before we chose an act, we are permitted to perform an experiment e which has outcomes θ_i. Consider also that we have determined statistically the probabilities of the states of nature when the outcome θ_i has occurred. These probabilities, written $p(C_j|\theta_i)$, might have to be determined using Bayes' rule and the probabilities $p(\theta_i|C_j)$, which are generally more easily available (or deducible). Then:

$$p(C_j\ \theta_i) = \frac{p(\theta_i|C_j)\ p(C_j)}{\sum_n p(\theta_i|C_n)\ p(C_n)} \tag{65}$$

The utility of act A_r, given that the experiment e has had an outcome θ_i is now:

$$U(A_r|\theta_i) = \sum_j p(C_j|\theta_i)u_{rj} \tag{66}$$

These remarks may be related to the classification problem if we consider the experiment e to be the scanning of a document hunting for a clue word. If the first clue word we find is W_i, then the outcome of the experiment is θ_i. In this case, the utility u_{rj} is unity if r = j, i.e., if the selected category r is the same as the correct category j, and 0 otherwise. In other words,

$$u_{rj} = \delta_{rj} \tag{67}$$

where δ is the Kronecker delta function. We then have

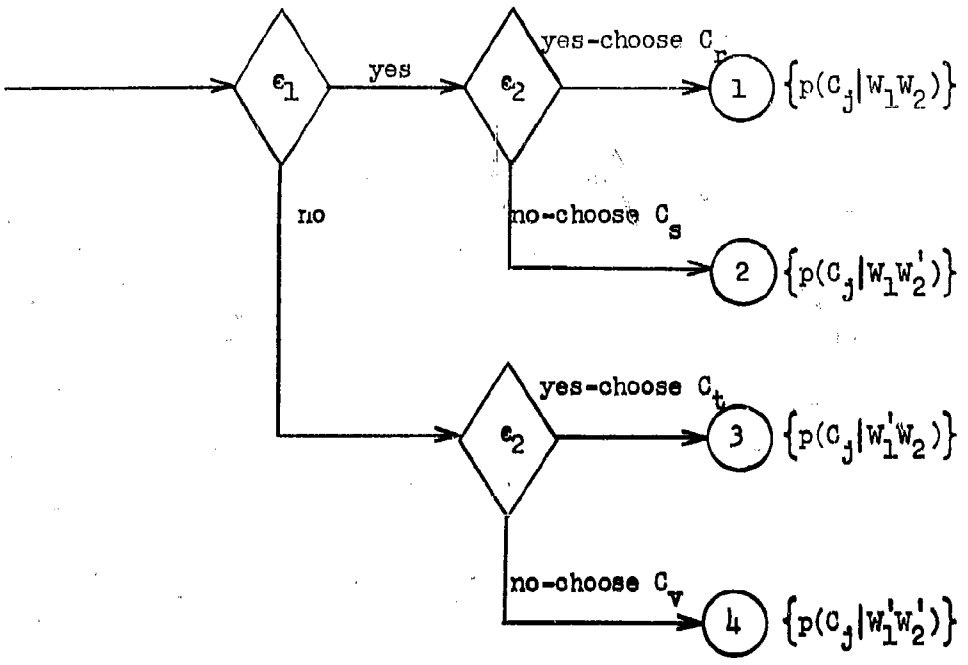
$$U(A_j|\theta_i) = p(C_j|\theta_i) \tag{68}$$

We want to pick the value of $U(A_j|\theta_i)$ that is maximum for a given θ_i. This is:

$$\underset{j}{\text{Max}}\ U(A_j|\theta_i) = \underset{j}{\text{Max}}\ p(C_j|\theta_i) \tag{69}$$

4.3.2.2 Maximization of Correct Classification - Consider a classification procedure based on two experiments e_1 and e_2. e_1 checks to see if word W_1 is present in a document and e_2 checks to see if word W_2 is present. This procedure is represented in Figure 3. For each of the four possible outcomes we get a set of probabilities, $\{p(C_j|W_1W_2)\}$, $\{p(C_j|W_1W_2')\}$, $\{p(C_j|W_1'W_2)\}$, and $\{p(C_j|W_1'W_2')\}$, where W_i' indicates the absence of word W_i. In accordance with this procedure, we would choose those categories which have the highest probability of each set.

Let us now group the documents we are trying to classify on the basis of these experiments. In group 1, where W_1 and W_2 are present, there are on the average the fraction $p(W_1W_2)$ of the total number of



e_1 = Is W_1 present?

e_2 = Is W_2 present?

Choose A_r such that

(1) $p(C_r|W_1W_2) = \max_j p(C_j|W_1W_2)$

(2) $p(C_s|W_1W_2') = \max_j p(C_j|W_1W_2')$

(3) $p(C_t|W_1'W_2) = \max_j p(C_j|W_1'W_2)$

(4) $p(C_v|W_1'W_2') = \max_j p(C_j|W_1'W_2')$

FIGURE 3. A Decision Procedure for Category Selection

documents. If, for every document in this group we perform A_r, we will correctly classify the fraction $p(C_r|W_1W_2)$ of this group. Thus the total

fraction of correctly classified documents is, on the average:

$$G = p(C_r|W_1W_2)\ p(W_1W_2) + p(C_s|W_1W_2')\ p(W_1W_2')$$
$$+ p(C_t|W_1'W_2)\ p(W_1'W_2) + p(C_v|W_1'W_2')\ p(W_1'W_2')$$
$$= p(C_rW_1W_2) + p(C_sW_1W_2') + p(C_tW_1'W_2) + p(C_vW_1'W_2') \qquad (70)$$

But A_r, A_s, etc., are optimal and so the conditional probabilities are the maxima. Then:

$$G_o = \max_j p(C_jW_1W_2) + \max_j p(C_jW_1W_2') + \max_j p(C_jW_1'W_2)$$
$$+ \max_j p(C_jW_1'W_2') \qquad (71)$$

In general if $\{\alpha_i\}$ represents the set of outcomes,

$$G_o = \sum_i \max_j p(C_j\alpha_i) \qquad (72)$$

Then we want to choose our experiments such that the outcomes lead to a maximum value of G_o.

Let the series of experiments $e_1, e_2, \ldots, e_n$ associated with clue words $W_1, W_2, \ldots, W_n$ be called the experiment α. Let the possible outcomes of this experiment be designated $\{\alpha_i\}$ as before. If we take different combinations of words, we will generate an associated set of experiments $\{\alpha\}$. Let γ be all possible experiments we can generate in this manner. Then:

$$G_{omax} = \max_{\alpha\in\gamma} \sum_i \max_j p(C_j\alpha_i) \qquad (73)$$

In general it would be very difficult to find the best set of words, for every set would have to be examined. This procedure is

clearly different from the information-theoretical methods and in general may lead to somewhat different results. What the differences are and why they occur should be investigated, but it might be possible to derive the information theoretic formulae from the game theoretic approach if appropriate utility values are used.

4.3.2.3 Departures from the Ideal Procedure - In the previous analysis we have assumed that all of the conditional statistics were obtainable. This condition may not be the case, however. Only a partial set of conditional probabilities may be obtainable, or it may not be practical to obtain them. It may also be impractical to perform the complete set of experiments on all documents.

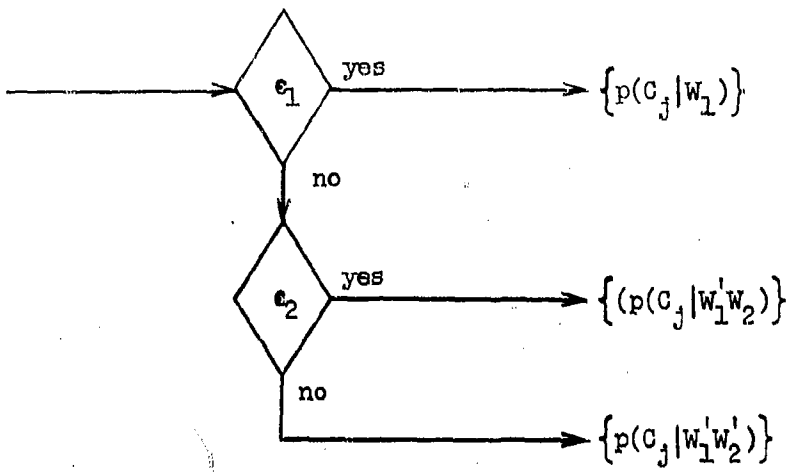
4.3.2.3.1 Simplifying the Choice Rule - Consider the following example, in which documents having W_1 are not tested for W_2 (see Figure 4). Then the decision procedure would be similar to the first case, and we would obtain the total fraction of correctly classified documents for the optimal procedure, G_{2o}:

$$G_{2o} = \max_j p(C_j W_1) + \max_j p(C_j W_1' W_2) + \max_j p(C_j W_1' W_2') \qquad (74)$$

The degradation in result caused by not testing all documents for W_2 is $G_o - G_{2o}$:

$$G_o - G_{2o} = \max_j p(C_j W_1 W_2) + \max_j p(C_j W_1 W_2') - \max_j p(C_j W_1) \qquad (75)$$

The maximum of G_{2o} would be found by trying all possible word combinations as in the first case. If the reason for not performing the W_2



c_1 = Is W_1 present?

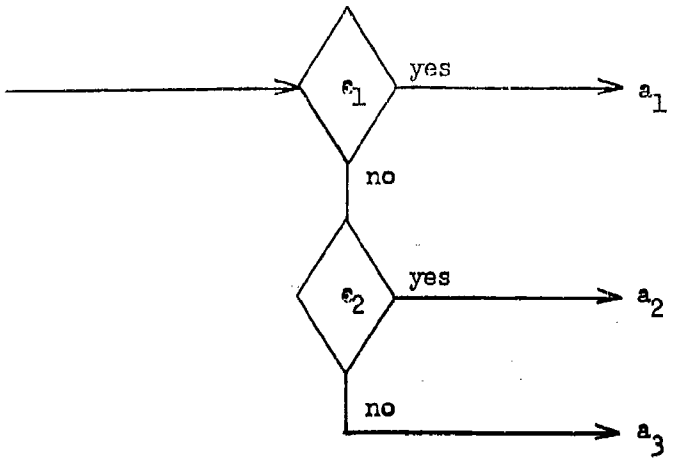
c_2 = Is W_2 present?

FIGURE 4. A Second Procedure for Classification

test is the cost of the test, then certainly W_1 should be chosen such that $p(W_1)$ is large and fewer documents would need two tests. This consideration can be introduced into the equation by including a testing cost factor in C_{2o}.

4.3.2.3.2 Lack of Information - Consider the situation in which the only sets of probabilities available are $\{p(C_j)\}$, $\{p(C_j|W_1)\}$, and $\{p(C_j|W_2)\}$. Also assume that W_1 will be tested for first and only those documents not having W_1 will be tested for W_2. This procedure is shown in Figure 5. The fraction of correctly classified documents if categories a_1, a_2, and a_3 are chosen for the respective groupings is:

$$C_3 = p(a_1W_1) + p(a_2W_1'W_2) + p(a_3W_1'W_2') \qquad (76)$$



e_1 = Is W_1 present?

e_2 = Is W_2 present?

FIGURE 5. A Third Procedure for Classification

It would seem reasonable to choose

$$\left.\begin{aligned} &a_1 \text{ on the basis of } p(a_1|W_1) = \max_j p(C_j|W_1); \\ &a_2 \text{ on the basis of } p(a_2|W_2) = \max_j p(C_j|W_2); \\ &\text{and} \\ &a_3 \text{ on the basis of } p(a_3) = \max_j p(C_j), \end{aligned}\right\} \qquad (77)$$

because this method seems to make the best use of the available information. Then to calculate the maximum value of G_{3o} over the entire set of clue word combinations would seem practically quite difficult, although conceptually it appears easy.

4.3.2.4 Summary - Some game theoretical considerations of the classification problem have been presented. It seems that any theoretical

analysis of the "non-ideal" case is extremely limited; however, this should be investigated further. It is not yet clear how the information-theoretical and game-theoretical approaches are related; but if there is a simple relationship in the ideal case, it may shed light on a combined approach for the non-ideal cases.

4.3.3 An Approach to a Criterion for Automatically Generated Extracts - Automatic extracting was originally described by Luhn [46] some time ago. While he refers to the end products of his process as abstracts, they are more accurately characterized as extracts of what are hopefully the more central, critical, or descriptive sentences in a document. Luhn's technique is purely statistical. Sentences are selected for extracting on the basis of two related facts about their word content:

(a) The relative frequency of the words in the sentence, except for common words.

(b) The distance between high frequency words in the sentence, based upon the number of intervening non-clue words.

While Luhn present a rather vague theoretical rationale for the validity of such an approach, there is no attempt to justify it in detail, except on the grounds that it can produce useful extracts. No attempt is made to show whether extracts generated by any other technique are more or less useful. Recently Guiliano et al [22] at Arthur D. Little have proposed a technique for incorporating syntactic information into the distance measure in order to make the technique more useful.

There seem to be two things lacking in this approach to automatic abstracting or extracting:

(a) A lack of any criterion or perhaps of multiple criteria, depending on the context in which the extract is to be used, for determining the adequacy of any given extract or extracting scheme.

(b) A lack of understanding of the fundamental processes involved in human abstracting, extracting, condensation, or perception of statement saliency in a longer argument of presentation.

It would seem that a combination of the approach of Newell and Simon [53] to the simulation of cognitive processes--theorem proving and problem solving more generally--and the approach of Maron [47] to the automatic classification of documents might be appropriate. While each of these studies is well known, it might be appropriate to indicate briefly which aspects of their methodology are relevant to alleviating the two shortcomings in present automatic extracting systems.

Newell et al, in order to simulate cognitive functioning, first used a method of observation and introspection to gain insight into the method by which humans proved logic theorems. In the context of information retrieval the major emphasis is on useful extraction rather than on the simulation of human extraction. It may nevertheless pay to observe human extracting behavior in order to develop more useful algorithms for obtaining automatic extracts.

The work of Maron and Kuhns has already been described in previous reports. It involved the use of human classification of a set of items as a criteria for automatic classification. The automatic classification, however, was not based on the unknown techniques of the human classifiers. The automatic algorithm was based rather upon purely statistical features of some of the classified documents. Human classification was also available, however, to provide the criteria for checking

the adequacy of the automatic algorithm once it was derived.

In the case of automatic extracting both of these techniques might prove useful. That is, the use of observation and introspection would help alleviate the difficulty caused by the lack of understanding of human functions and allow for the development of more rational extracting algorithms. Perhaps these techniques could be ultimately extended to abstracting per se. The records of humanly generated extracts could be used as a criterion for evaluating the adequacy of various automatic algorithms. The latter would alleviate the difficulty caused by the non-existence of suitable criteria.

The paradigm for such research and development would be as follows:

(a) A series of documents, either large texts or shorter articles for research convenience, would be selected for extracting.

(b) Ground rules for desired extracts would be developed; e.g.:

  (1) How long should each extract be? Should it be some fixed proportion of the total document?

  (2) What sentential units should be extracted? Whole sentences only? Parts of sentences? Parts that can be recombined to form larger sentences?

  (3) What is the focal purpose of the extract? To extract as much factual information as possible within the limits imposed by the length of an extract? To characterize the document as well as possible in order that the reader might know what information it contains? Both of these?

  (4) What information or techniques may be used in generating the extract? Anything that occurs to the user based upon his total knowledge? Anything based on the explicit and implicit content of the document? Only explicit content? Only Rigorously formulated rules?

(c) The documents would then be subjected to human extracting using instructions based upon the ground rules.

(d) A portion of the humanly extracted documents would be carefully subjected to introspective report and an analysis of the implicit rules followed in extracting.

(e) Based on this analysis, one or several automatic algorithms would be developed for achieving essentially the same extracts from readily treated information in the documents. For the sake of generality, an attempt would also be made to incorporate those rules manifest in introspective protocols that could be handled by computers.

(f) Measures of correspondence between humanly and automatically generated extracts would then be developed.

(f) Finally, the automated techniques would be applied to the remaining documents in the sample and the extracts generated would be validated against the criterion of the human extracts already available.

While this approach depends upon research and development strategies already developed by others, its application to the information retrieval problem is unique. Further research along these lines appears warranted.

4.4 FILE STRUCTURE

File structure is concerned with the organization of document descriptions in a storage medium. The assumption has generally been that the storage medium is attached to a computer, though much of the work can be applied more generally. With every file organization there is associated an algorithm for obtaining the addresses of those documents that satisfy a given description. The file structure depends, of course, on the descriptive structure to be used. File structure is concerned only peripherally with the method by which descriptions are assigned to documents.

In this section, two topics will be presented. The first and major topic is a mathematical analysis and discussion of the efficiency of

certain types of file organizations. The second topic is a description and evaluation of the Multi-List system.

4.4.1 Comparative Analysis of Some File Organizations

4.4.1.1 Introduction - This section contains a discussion of a number of file organizations that may be suitable for the retrieval of documents or other items of information. The exposition largely follows the order of mathematical development rather than some didactic organization for easily communicating the results. This method of exposition is used because it is impossible in work of this kind to know at the beginning where fruitful mathematical analysis will lead.

For each file structure considered, expressions are derived for the average or expected values of the number of items and the subject or category headings examined to retrieve a single item, known to be in the file, in response to a request. The file organizations are then compared and evaluated in terms of these expected values for a wide range of file sizes. To aid in the comparison, variances are derived and plotted.

Three different types of file organizations or structures will be compared. They are:

(a) Single-level subject headings.

(b) Hierarchical trees of items.

(c) Hierarchical trees of subject headings.

The first type consists of a single level of unrelated subject headings or category names under which items are grouped or filed. Both the order of subject headings within the file and the order of items within a subject are random.

The second type of file organization is a multi-level tree of items. The connectivity of the tree does not necessarily imply a corresponding logical relation among the items.

The tree of subject headings, on the other hand, is a multi-level categorization of subject headings where each heading is divided into two or more sub-headings down to the lowest level of detail. The tree of subject headings is intended to imply the logical relation among them. The items may be filed in a linear sequence or in a hierarchical tree under the last row of headings.

More than one may of searching the nodes of a tree will be used. Further subdivisions of the three types of file organizations will be discussed in the following detailed analysis. Trees of both items and subject headings will be considered in various cases in the section on hierarchical trees. First, however, single-level subject headings will be analyzed. This analysis will include the case of a sequentially ordered file that, when searched logarithmically, makes the transition between single-level subject headings and hierarchical trees one of generalizing a special case.

For each type of file structure a mathematical expression can be derived for the expected number of headings and items searched and examined in order to locate a single item in the file. Some simplifying assumptions will be made to keep the mathematics relatively uncomplicated. Similar expressions can be derived, however, under less restrictive assumptions.

4.4.1.2 Single-Level Subject Headings - Suppose there are s subject headings. It is assumed that the subject heading under which the item is to be found is supplied with the request. It is further assumed for the sake of simplicity that the items in the file are evenly distributed under the subject headings. That is, it is equally likely that any subject heading and any item under a subject heading will be requested and each subject heading will have the same number of items filed under it. The probability p_1 of searching one subject heading is:

$$p_1 = \frac{1}{s} \tag{78}$$

The probability of searching two subject headings to find the requested one is:

$$p_2 = \frac{s - 1}{s} \frac{1}{s - 1} = \frac{1}{s} \tag{79}$$

Similarly:

$$p_i = \frac{1}{s} \tag{80}$$

The expected number E(i) of subject headings searched is:

$$E(i) = \sum_{i=1}^{s} i \frac{1}{s}$$

$$= \frac{1}{s} \frac{s(s + 1)}{2} \tag{81}$$

or

$$E = \frac{s + 1}{2} \tag{82}$$

The number of items N_s under each subject heading is:

$$N_s = \frac{N}{s} \tag{83}$$

By an argument analogous to that for subject headings, the expected number E(i) of items searched is:

$$E(i) = \sum_{i=1}^{N_s} \frac{i}{N_s}$$

$$= \frac{N + s}{2s} \qquad (84)$$

The expected number of items and subject headings searched for in a linear file is then:

$$E = \frac{s + 1}{2} + \frac{N + s}{2s}$$

$$= \frac{1}{2} (s + N/s + 2) \qquad (85)$$

A file of items arranged sequentially by some ordering rule--e.g., a file of part or drawing numbers or any other numbered or ordered items--can be arranged and searched by the method of subject headings previously described. Another method of search is the following: Go to the middle of the file. Compare the item requested with the item there. A decision can then be made on the basis of the ordering of the items as to whether the item sought is in the first (lower) half of the file or in the second (higher) half. Whichever half it is in, go to the middle of that half and repeat the procedure. This process is continued until the item is located. The process of going to the middle of any portion of the file will be called a cut. Since a single file item is examined for each cut, the expected number of cuts is equal to the expected number of file items which will be examined. This method is called the Binary Logarithmic search.

Consider a file of N items. By the search procedure just described, the number of items N_1 that can possibly be retrieved on the first cut is 1; on the second cut, 2; and, in general, on the $j^{\underline{th}}$ cut:

$$N_j = 2^{j-1} \tag{86}$$

The maximum number of cuts n required to retrieve any item whatsoever in the file can be determined from Equation (86) as follows:

$$\begin{aligned} N &= \sum_{j=1}^{n} N_j \\ &= \sum_{j=1}^{n} 2^{j-1} \\ &= 2^n - 1 \end{aligned} \tag{87}$$

Solving for n gives:

$$n = \log_2(N + 1) \tag{88}$$

The origin of the name <u>logarithmic search</u> is obvious from Equation (88).

It is evident from Equation (86) that the probability p_j of retrieving the correct item in response to a given random request on the $j^{\underline{th}}$ cut is:

$$p_j = \frac{2^{j-1}}{N} \tag{89}$$

The expression for the expected number of cuts j (or, equivalently, the number of items examined) is:

$$E = \sum_{j=1}^{n} j \frac{2^{j-1}}{N} \tag{90}$$

where n is obtained from Equation (88). The series in Equation (90) is

the derivative of a geometric progression, and the expression for its sum can be obtained by differentiating the expression for the sum of a geometric progression with a finite number of terms. This procedure yields the following expression for E:

$$E = \left[\frac{N + 1}{N}\right] \log_2(N + 1) - 1 \qquad (91)$$

4.4.1.3 Hierarchical Trees - Only regular rooted trees will be considered for hierarchical trees. A tree is rooted if all its branches are connected ultimately to a single node (the root). A tree is regular if the number of branches k emanating from each node is a constant. Another way of thinking of this file structure is that every heading or grouping of the file organization is divided into the same number of subheadings.

Four cases of retrieving items from trees will be considered. These cases are designated I to IV, respectively.

4.4.1.3.1 Case I - In this case the tree is considered as a hierarchy composed entirely of file items, each of which is equally likely to be the answer to a given random request. Hence, retrieving a given node will be considered as providing a single-item response. The level of the node then represents the generality of the response, which is presumably related directly to the generality of the request. The node provided as a response can be considered as the name or term or descriptor for all the nodes at lower levels of the tree that are connected to the node provided as a response. If the node is a category

name, all the connected nodes--the items in the category--could be provided as part of the response. It is assumed that the tree is indexed; that is, each node of the tree contains indexes of the nodes on the next lower level connected to it. It is also assumed that these indexes are sufficient to ascertain which node to examine at the next level. Thus only one node is examined at each level searched.

If each node of the tree contains indexes that are identifiers of the nodes at the next level at the end of the branches emanating from it, then by examining a given node a decision can be made as to which node to examine at the next level. Searching a tree of this type is a generalization of the binary logarithmic search. For example, consider a regular binary tree; that is, $k = 2$. Examining the first node, the root, is analogous to going to the middle of the file. There are two nodes at the next level. Selecting one is analogous to going to the middle of the lower half of the file; selecting the other is equivalent to going to the middle of the upper half of the file. The generalization of this process for larger integral values of k is obvious. The mathematics is analogous to the binary logarithmic search.

The number of levels L to be examined in order to guarantee the retrieval of any item in a regular tree of order k is:

$$L = \log_k[(k - 1)N + 1] \tag{92}$$

The expected number of items examined becomes:

$$E = \frac{1}{N} \sum_{j=1}^{L} jk^{j-1}$$

$$= \left[\frac{(k - 1)N + 1}{(k - 1)N}\right] \log_k[(k - 1)N + 1] - \frac{1}{k - 1} \tag{93}$$

where L is determined from Equation (92). Thus Equations (88) and (91) are merely special cases of Equations (92) and (93), respectively, for regular binary trees.

4.4.1.3.2 Case II - In this case only the nodes at the bottom level of the tree represent file items. It is assumed that each such node represents a group of file items. Thus a search consists of tracing a path through the tree to one node at the bottom and searching the items filed under that node to provide a single file item as a response. Again, it is assumed that each node is equally likely to be the answer. If this case is restricted to regular trees with no method of indexing or determining which connected node at the next level is the correct one, then this case generalizes the simple subject heading file to a multi-level subject heading or classification file. Only non-indexed trees will be considered in this case. A non-indexed tree is one that has no mechanism for selecting the proper node at the next lower level without examining the nodes at that level connected to the node at which the searcher is presently located.

Assume there are s nodes or subject headings on a regular tree of order k. Then let there be N file items listed under the bottom nodes and assume that the file items are evenly distributed among these nodes. Assume also that there are L levels of nodes in the tree.

Since the only nodes searched at each level are those connected to the node selected at the next higher level, the probability p_j of finding the desired subject heading at a given node is:

$$p_j = \frac{1}{k} \tag{94}$$

Therefore, the expected number of nodes examined at any level j, except the first level or the root node* where the expected number is 1, is:

$$E_j(i) = \sum_{i=1}^{k} \frac{i}{k}$$

$$= \frac{k + 1}{2} \tag{95}$$

where $2 \leq j \leq L$. Hence, the expected number of nodes examined for the entire tree including the root node is:

$$E_s = [\frac{k + 1}{2}](L - 1) + 1 \tag{96}$$

The required number of levels L in the tree is determined by k and s, and is obtained from Equation (92), which gives:

$$L = \log_k[(k - 1)s + 1] \tag{97}$$

and, by substituting and simplifying:

$$E_s = [\frac{k + 1}{2}] \log_k[(k - 1)s + 1] + \frac{1 - k}{2} \tag{98}$$

At this stage, no file items have been examined. Equation (98) gives the expected number of subject headings examined to find the heading at the lowest level under which the file item sought is listed. Therefore, the file items under that heading must now be examined. The number of items N_s filed under a given subject heading is:

$$N_s = \frac{N}{s_L} \tag{99}$$

---

*It is assumed that this node is examined to identify the tree and locate the nodes at the second level.

where s_L is the number of subject headings, or nodes, at the lowest level of the tree. This sequence is a simple linear file like the first one examined. The expected number of file items searched E_n is then:

$$E_n(i) = \sum_{i=1}^{N_s} \frac{i}{N_s}$$

$$= \frac{N_s + 1}{2} \tag{100}$$

The number of nodes s_j at level j of a regular tree of order k is given by:

$$s_j = k^{j-1} \tag{101}$$

therefore;

$$s_L = k^{L-1} \tag{102}$$

Substituting Equation (97) into Equation (102) yields:

$$s_L = \frac{(k - 1)s + 1}{k} \tag{103}$$

and, from Equations (99) and (103);

$$N_s = \frac{kN}{(k - 1)s + 1} \tag{104}$$

Substituting Equation (104) in Equation (100) gives:

$$E_n = \frac{kN + (k - 1)s + 1}{2[(k - 1)s + 1]} \tag{105}$$

The expected value of the number of subject headings and file items examined to retrieve one file item in this type of file organization is Equation (98) plus Equation (105):

$$E = \frac{kN + (k - 1)s + 1}{2[(k - 1)s + 1]} + \left[\frac{k + 1}{2}\right] \log_k[(k - 1)s + 1] + \frac{1 - k}{2} \qquad (106)$$

It is now evident that when file items are related it may be possible to arrange each set of N_s items so that it can be searched logarithmically. In this case Equation (106) becomes:

$$E = \left[\frac{(k - 1)N + s_L}{(k - 1)N}\right] \log_k[(k - 1)\frac{N}{s_L} + 1] - \frac{1}{k - 1} + \left[\frac{k + 1}{2}\right] \log_k[(k - 1)s + 1] + \frac{1 - k}{2} \qquad (107)$$

This equation is obtained from Equations (93), (98), and (99). Equation (103) was used to obtain the value of s_L.

4.4.1.3.3 Case III - This case is the same as Case I except that the tree is not indexed. That is, any node may be a satisfactory response to a request; but after selecting a node at a given level, it is necessary to examine the nodes at the next lower level connected to the selected node in order to ascertain which one is the next appropriate subheading.

In this case the maximum number of nodes examined at each level except the first is simply k. The number of nodes examined at the first level is 1. Therefore, the maximum number of nodes examined in any search is:

$$n = k(L - 1) + 1 \qquad (108)$$

hence, from Equations (92) and (108):

$$n = k \log_k[(k - 1)N + 1] + (1 - k) \tag{109}$$

Therefore, the expected number of nodes examined is:

$$E = \sum_{i=1}^{n} \frac{i}{n}$$

$$= \frac{k}{2} \log_k[(k - 1)N + 1] + \frac{2 - k}{2} \tag{110}$$

where n is determined from Equation (109).

4.4.1.3.4 Case IV - This case considers an indexed tree of subject headings rather than file items with the file items located under the lowest row of nodes or subject headings. The equally likely assumption is involved, as usual. Two variations can be considered. First, the file items are sequential and searched in order. Second, the file items are searched logarithmically; in this variation the items are actually filed in a tree structure.

Since the subject headings in this case are not responses, the expected number of headings examined is fixed and equal to the number of levels L in the tree. Therefore, from Equation (97):

$$E_s = \log_k[(k - 1)s + 1] \tag{111}$$

For a sequentially searched file, the expected number of items searched is obtained from Equation (105). Therefore, the expected number of subject headings and items searched is:

$$E = \frac{kN + (k - 1)s + 1}{2[(k - 1)s + 1]} + \log_k[(k - 1)s + 1] \tag{112}$$

If the items are searched logarithmically, the expected number is obtained by taking N equal to N_s and then substituting Equation (104)

in Equation (93). The resulting equation is:

$$E_n = \left[\frac{(k-1)(kN+s)+1}{k(k-1)N}\right] \log_k\left[\frac{(k-1)(kN+s)+1}{(k-1)s+1}\right] - \frac{1}{k-1} \qquad (113)$$

Therefore, the expected number of subject headings and items examined is Equation (111) plus Equation (113):

$$E = \log_k[(k-1)s+1] + \left[\frac{(k-1)(kN+s)+1}{k(k-1)N}\right] \log_k\left[\frac{(k-1)(kN+s)+1}{(k-1)s+1}\right] - \frac{1}{k-1} \qquad (114)$$

4.4.1.4 Analysis and Comparison of the Expected Values - The major purpose of deriving expressions for the expected values of the number of headings and items examined in various file structures is that these values provide a convenient (if oversimplified) means of comparing the effectiveness of different file structures. These file organizations and their corresponding average values are summarized in Table 1.

For general purposes of comparison the equations identified in Table 1 can be rewritten in simpler form. The simplified versions are given below with their original numbers followed by "A". The subscript s stands for subject headings; N for file items. For a file with single-level subject headings, and no other structure,

$$E = \frac{1}{2}[s + N/s + 2] = \frac{s+1}{2} + \frac{N_s + 1}{2} \qquad (85A)$$

where N_s is obtained from Equation (83).

For an indexed tree of items (Case I),

$$E \doteq L_N - \frac{1}{k-1} \qquad \left(N \geq \frac{100}{k-1} \right) \qquad (93A)$$

where $L_N = n$ is obtained from Equation (92).

For a non-indexed tree of subject headings with items stored sequentially (Case II-A),

$$E = \left[\frac{(k+1)}{2} \right] (L_s - 1) + 1 + \frac{N_s + 1}{2} \qquad (106A)$$

where L_s is obtained from Equation (97), and N_s, from Equation (104).

For a non-indexed tree of subject headings with items stored in an indexed tree (Case II-B),

$$E \doteq \left[\frac{k+1}{2} \right] (L_s - 1) + 1 + L_{N_s} - \frac{1}{k-1} \left(N \geq \frac{100}{k-1} \right) \qquad (107A)$$

where L_s and L_{N_s} are obtained from Equation (97), and N_s, from Equation (104).

For a non-indexed tree of items (Case III),

$$E = \frac{k}{2} (L_N - 1) + 1 \qquad (110A)$$

where $L_N = n$ is obtained from Equation (92).

For an indexed tree of subject headings with items stored sequentially (Case IV-A),

$$E = L_s + \frac{N_s + 1}{2} \qquad (112A)$$

where L_s is obtained from Equation (97), and N_s, from Equation (104).

TABLE 1. SUMMARY OF FILE ORGANIZATIONS

SUBJECT HEADINGS	ITEMS	CASE	AVERAGE NUMBER OF HEADINGS AND ITEMS EXAMINED TO FIND ONE ITEM	EQUATION NUMBER
Linear	Random under each heading	Single Level	$E = \frac{s+1}{2} + \frac{N \div s}{2s} = \frac{1}{2}(s + N/s + 2)$	85
None	Indexed tree	Case I	$E = \frac{1}{N}\sum_{j=1}^{L} jk^{j-1}$ $= \left[\frac{(k-1)N+1}{(k-1)N}\right] \log_k[(k-1)N+1] - \frac{1}{k-1}$	93
Non-indexed tree	Sequential under last row of nodes	Case II-A	$E = \frac{kN + (k-1)s + \square}{2[(k-1)s+1]}$ $+ \left[\frac{k+1}{2}\right] \log_k[(k-1)s+1] + \frac{1-k}{2}$	106
Non-indexed tree	Indexed trees under last row of headings	Case II-B	$E = \left[\frac{(k-1)N + s_L}{(k-1)N}\right] \log_k[(k-1)\frac{N}{s_L} + 1]$ $+ \left[\frac{k+1}{2}\right] \log_k[(k-1)s+1] - \frac{(k^2-2k+3)}{2(k-1)}$	107

TABLE 1 (Continued). SUMMARY OF FILE ORGANIZATIONS

SUBJECT HEADINGS	ITEMS	CASE	AVERAGE NUMBER OF HEADINGS AND ITEMS EXAMINED TO FIND ONE ITEM	EQUATION NUMBER
None	Non-indexed tree	Case III	$E = \sum_{i=1}^{n} \frac{i}{n} = \frac{k}{2} \log_k[(k-1)N + 1] + \frac{2-k}{2}$	110
Indexed tree	Sequential under last row of nodes	Case IV-A	$E = \frac{kN + (k-1)s + 1}{2[(k-1)s + 1]} + \log_k[(k-1)s + 1]$	112
Indexed tree	Indexed trees under last row of headings	Case IV-B	$E = \log_k[(k-1)s + 1] + \left[\frac{(k-1)(kN + s) + 1}{k(k-1)s + 1}\right] \cdot \log_k\left[\frac{(k-1)(kN + s) + 1}{(k-1)s + 1}\right] - \frac{1}{k-1}$	114

For an indexed tree of subject headings with items stored in an indexed tree (Case IV-B),

$$E \doteq L_s + L_{N_s} - \frac{1}{k-1} \qquad \left(N \geq \frac{100}{k-1} \right) \qquad (114A)$$

where L_s and L_{N_s} are obtained from Equation (97), and N_s, from Equation (104).

These equations can be analyzed in two major ways with respect to E. The first is to ascertain within a given equation whether there is a relationship between s and N that will minimize E for that type of file organization. The second is to compare the equations with each other to determine whether some file structures are always superior to others.

To carry out the first analysis it is sufficient to assume that s can take any positive real value and to differentiate each of the equations with respect to s, considering N as a constant, and checking to see if the resulting extremum is indeed a minimum. If there is such a relationship between s and N, it provides the proper number of subject headings s to minimize E for a file of N items with that type of organization.*

---

*In the following discussion the values of s, which optimize the expected number of headings and items examined, are obtained for several of the file organizations. This derivation is accomplished by differentiating the expression for E with respect to s to obtain the appropriate s as a function of N that minimizes E. Strictly speaking, such a procedure is not permissible because all the distributions considered are discrete. E is defined only for positive integral values of s and N. Nevertheless, the equations for E in all cases are continuous functions for the domains

For example, taking the partial derivative of E with respect to s in Equation (85A) and setting the result equal to zero yields:

$$s = \sqrt{N} \qquad (115)$$

for a file with single-level subject headings only. A check reveals that the appropriate conditions for a minimum are satisfied. That is, the value of s given in Equation (115) will always result in a minimum E for that N. Substituting Equation (115) in Equation (85A) gives:

$$E_{min} = 1 + \sqrt{N} \qquad (116)$$

From Equations (83) and (115), the optimum number of items under the subject headings is:

$$N_s = \sqrt{N} \qquad (117)$$

Equation (93A) for Case I cannot be treated in this manner because it is a function of N only (and k). However, as k increases, E decreases for a constant N. This fact must be interpreted carefully because no two aribtrarily selected values of k will necessarily yield an integral value of L for a fixed N.

---

of k, s, and N that are of interest. Consequently, these differentiations can be carried out formally and the relative minima obtained. To obtain the integral values of s that minimize E, it is then necessary to substitute the two integers closest to the minimum s into the equation for E to ascertain which gives the smaller E. This integer is then used as the minimum, provided it is positive. Even this procedure would not be sufficient were it not for the fact that these functions, in the cases considered, have only one relative minimum, and, therefore, this relative minimum is also an absolute minimum. The ultimate justification for these unrigorous techniques is that they do provide the real minima and, therefore, have considerable utility.

Application of the same method to Equation (106A) for Case II-A yields:

$$s = \frac{1}{k-1}\left[\frac{kN}{(k+1)\log_k e} - 1\right] \tag{118}$$

This value of s for any N will yield the minimum E, and the value of E is:

$$E_{min} = \frac{3}{2} + \left[\frac{k+1}{2}\right]\log_k\left[\frac{eN}{(k+1)\log_k e}\right] \tag{119}$$

Equation (107A) for Case II-B has no relative minimum. However, the optimum value for s can be obtained by observation. By substituting Equations (97) and (104) in (107A) and simplifying, the result is:

$$E \doteq \left[\frac{k-1}{2}\right]\left[\log_k[(k-1)s+1] - 1\right]$$

$$+ \log_k[k(k-1)N + (k-1)s + 1] - \frac{1}{k-1} \tag{107B}$$

This equation is defined for $s \geq 1$. For this range of s, Equation (107B) has a minimum at $s = 1$. This minimum gives for E:

$$E \doteq 1 + L_N - \frac{1}{k-1}$$

The single subject heading is superfluous and can be eliminated. The minimum E thus becomes:

$$E_{min} \doteq L_N - \frac{1}{k-1} \tag{120}$$

Therefore, the optimum s for Equation (107A) is zero, and the equation has been reduced to Equation (93A). Consequently, it is disadvantageous to superimpose a non-indexed tree of subject headings on an indexed tree of file items.

Equation (110A) for Case III is also a function of N and k only. In this case a minimum E cannot be easily derived analytically. But solving Equation (110) numerically indicates that E is a minimum when k = 2 for $N \leq 100$ and when k = 2 for $N \geq 500$.

For Equation (112A) (Case IV-A) the s that gives minimum E is:

$$s = \frac{1}{k - 1}\left[\frac{kN}{2 \log_k e} - 1\right] \tag{121}$$

The minimum E becomes:

$$E_{min} = \frac{3}{2} + \log_k \left[\frac{eN}{2 \log_k e}\right] \tag{122}$$

Equation (114A) for Case IV-B has no relative minimum. However, the optimum value for s can be obtained as follows. By substituting Equations (94) and (97) in Equation (114A) and simplifying, it becomes:

$$E \doteq \log_k[k(k - 1)N + (k - 1)s + 1] - \frac{1}{k - 1} \tag{114B}$$

This equation is defined for $s \geq 1$. Obviously, it has an absolute minimum at s = 1, which gives:

$$E \doteq 1 + L_N - \frac{1}{k - 1}$$

The single subject heading again is superfluous, and E becomes:

$$E_{min} \doteq L_n - \frac{1}{k - 1} \tag{123}$$

Thus the optimum s for Equation (114A) is zero, and this equation is also reduced to Equation (93A). In other words, wherever it is possible to construct an indexed tree of items, it is pointless to superimpose an indexed tree of subject headings upon it. It is also pointless to

establish any other system of subject headings. One example, namely Equation (107A), has already been considered.

The second type of analysis compares one equation with another for an arbitrary but specified file size N and for a number of headings s; the objective is to determine whether E is always less in one type of file organization than in another. Equations (107A) and (114A) have been shown to be superfluous and will not be considered.

The files with no subject headings will be considered first. For a given N, an indexed tree of items, Equation (93A), will yield a lower average number of items searched than a non-indexed tree of items, Equation (110A), if:

$$L_N - \frac{1}{k-1} < \frac{k}{2}(L_N - 1) + 1$$

This inequality can be written:

$$(L_N - 1) - \frac{1}{k-1} < \frac{k}{2}(L_N - 1) \qquad (124)$$

The inequality is clearly valid for $k \geq 2$. Consequently, the average number of items examined in searching an indexed tree of N items is always less than the average number examined in a non-indexed tree.

Indexed and non-indexed trees with sequentially stored items can be compared in the case wher the number of headings in both trees is the same. Equation (106A) for non-indexed trees and Equation (112A) for indexed trees can be compared in terms of:

$$\left[\frac{k+1}{2}\right](L_s - 1) + 1 > L_s$$

or

$$\left[\frac{k+1}{2}\right](L_s - 1) > L_s - 1 \tag{125}$$

This inequality is clearly valid for $k \geq 2$ and $L_s \geq 1$. Therefore, Equation (112A) gives a smaller E than Equation (106A). It is clear, however, from Equations (118) and (121) that the optimum s's for the two trees of Equations (112A) and (106A) are not identical. Nevertheless, it can be shown directly from Equations (119) and (122) that Equation (112A) also yields a smaller E than Equation (106A) when s is optimized in each case. This optimization would require:

$$\log_k \left[\frac{eN}{2\log_k e}\right] < \log_k \left[\frac{eN}{(k+1)\log_k e}\right]^{(k+1)/2}$$

or (126)

$$\frac{eN}{2\log_k e} < \left[\frac{eN}{(k+1)\log_k e}\right]^{(k+1)/2}$$

This inequality is valid for:

$$N > \frac{1}{2^{2/(k-1)}} \frac{(k+1)^{(k+1)/(k-1)}}{e \log_e k} \tag{127}$$

This condition presents no restriction for a practical case. For example, Equation (127) requires $N \geq 4$ if $k = 2$; $N \geq 3$, if $k = 10$; $N \geq 6$, if $k = 100$.

For a given N and a fixed $s > 1$, an indexed tree of subject headings, Equation (112A), always gives a lower value of E than a single level of subject headings, Equation (85A). The conditions would require:

$$L_s < \frac{s+1}{2}$$

This inequality can be transformed by algebra to:

$$k^{-(s+1)/2}[(k - 1)s + 1] < 1 \tag{128}$$

By differentiating the left member of Equation (128) with respect to k and setting it equal to zero, a value for k can be obtained to make it an extremum. This value is:

$$k = \frac{s + 1}{s} \tag{129}$$

By examining the second derivative at this point, it is observed that Equation (129) maximizes the left member of Equation (128) when $s > 1$. This maximum value is:

$$2\left[\frac{s}{s + 1}\right]^{(s+1)/2} \tag{130}$$

For $s > 1$, the Value (130) is always less than 1. Since the maximum value satisfies Equation (128), any other value, in particular any $k \geq 2$, will also satisfy it.

When s is optimized in each case, these two file structures can be compared by Equations (116) and (122). Equation (112A) will give a lower E than Equation (85A) in the optimum case when:

$$\frac{3}{2} + \log_k \left[\frac{eN}{2 \log_k e}\right] < \sqrt{N} + 1$$

By algebraic transformations, this inequality can be written:

$$\frac{N \ln k}{k^{\sqrt{N}-\frac{1}{2}}} < \frac{2}{e} \tag{131}$$

When $k = 2$, this inequality is valid for $N \geq 27$; when $k = 4$, it is valid for $N \geq 4$; when $k \geq 6$, it holds for $N \geq 1$.

The optimum cases of Equations (106A) and (85A) can be compared by using Equations (116) and (119). Equation (106A) will yield a smaller E when:

$$\frac{3}{2} + \left[\frac{k+1}{2}\right] \log_k \left[\frac{eN}{(k+1)\log_k e}\right] < \sqrt{N} + 1$$

that is, when

$$\frac{N \ln k}{[(k+1)k]^{[(2\sqrt{N}-1)/(k+1)]}} < \frac{1}{e} \qquad (132)$$

Equation (132) is generally valid for larger files. For example, a simple calculation with k = 10 shows that this equation is valid for N roughly greater than 115 and invalid for smaller N. Hence, the single level subject heading file results in a smaller average number of items searched in files with less than 115 items. This conclusion is shown clearly in Figure 6.

Figure 6 depicts the average number of headings and items examined for a wide range of file sizes. Only optimum values for s are shown. The figure indicates the superiority of indexed trees over non-indexed trees and of non-indexed trees over single-level subject headings, except for small files as indicated by Equation (132). However, the degree of superiority of the indexed trees is somewhat misleading. Although it is true that the average number of headings and items examined or searched for such trees is much smaller than for the other file structures, this fact does not imply much faster response times. By omitting consideration of the indexing function itself, the burden of search has in a sense merely been shifted elsewhere. Unless the indexing function

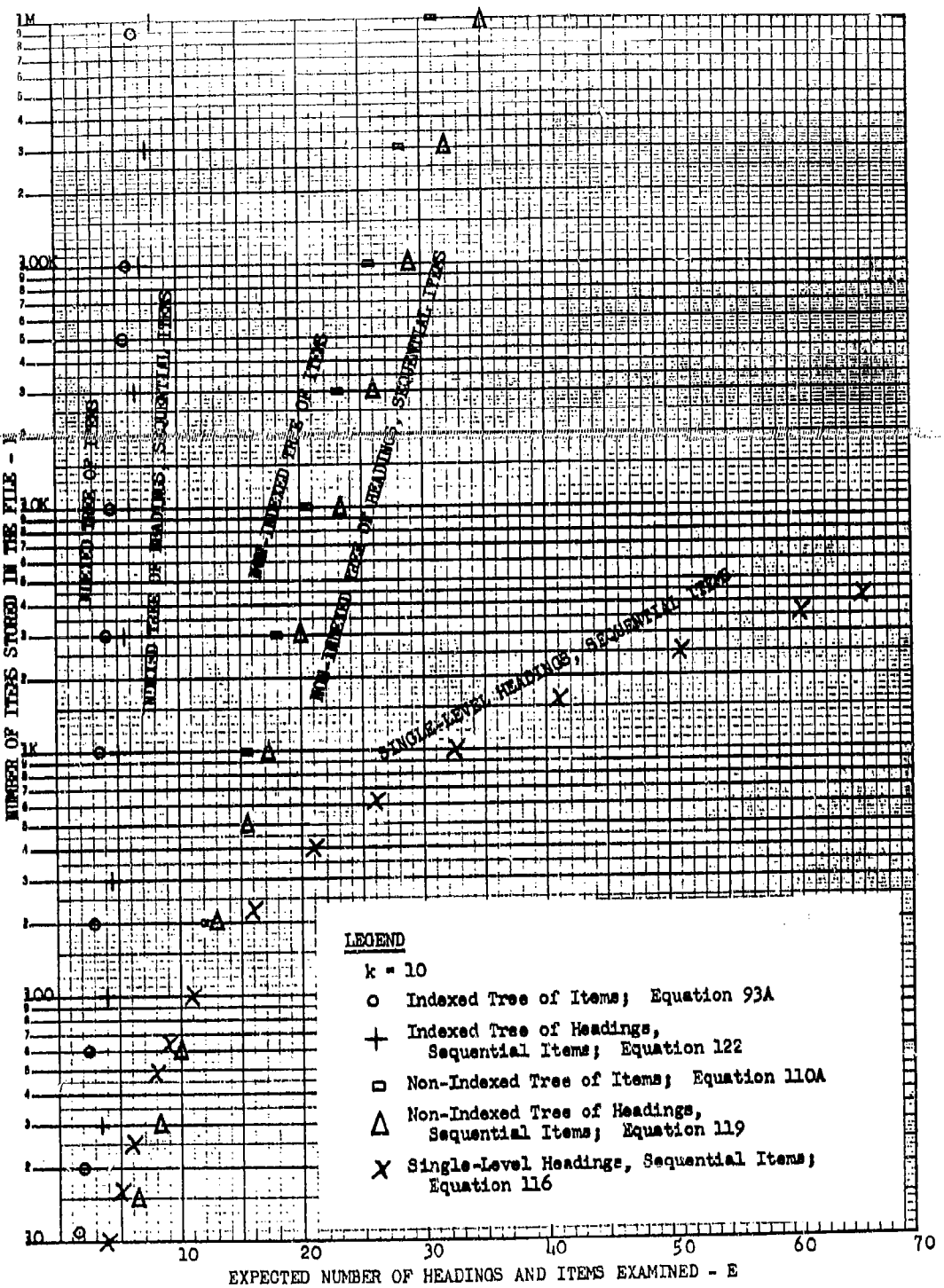


FIGURE 6. Average Number of Headings and Items Examined in a Search of Differently Organized Files

is powerful, the search procedure in an indexed tree, particularly where k is large, may spend almost as much time examining indexes to determine the appropriate paths as would be involved in examining the headings themselves.

A singular feature of Figure 6 is that the indexed tree of items, Equation (93A), and the indexed tree of headings, Equation (112A), give similar values of E. The same is true for the non-indexed trees represented by Equations (110A) and (106A). The explanation, however, is simple. Equations (118) and (121) require that the number of subject headings should be so large that essentially only a few items or even a single item are filed sequentially under each node of the last row. In other words, N_s is small. This fact can be seen from the values of N_s derived by inserting Equations (118) and (121), respectively, into Equation (104). These values are:

$$N_s = (k + 1)\log_k e \tag{133A}$$

$$\left.\begin{aligned} N_s &= 2 \log_k e \qquad (k \leq 7) \\ N_s &= 1 \qquad\qquad\quad (k > 7) \end{aligned}\right\} \tag{133B}$$

Consequently, almost all the searching is performed in the tree of headings where it is most economical. Hence, the close correspondence arises between trees of headings and between trees of items. Of course, in practice, it may frequently be impossible to achieve a meaningful breakdown of related headings to such a detailed level. Therefore, the optimum values of s, N_s, and E should be regarded as interesting idealizations. In practice, only integral values of s and N_s can be used.

In cases where the optimum curves plotted in Figure 6 are unrealistic because they restrict s too much, the equations developed in this and the previous section can be used to generate complete sets of design charts. From these charts the best file organization can be read in terms of whatever value s must have to reflect the logical relationships and the nature of the subject matter to be classified.

In the interest of completeness, Figure 7 is included for reference. It relates the number of levels of nodes in a regular tree of order k to accommodate N items, one item per node. Figure 7 is obtained from Equations (92) or (97).

4.4.1.5 Variance From the Expected Values - The utility of the average or expected number of items and headings examined in different file structures depends upon the likelihood that the number of items and headings searched will generally be near the average value. An estimate of this likelihood is provided by the statistical variance of the number of items and headings searched from the average number. Expressions for the variance relative to Equations (85A), (93)*, (106A), (110A), and (112A) will be developed and analyzed.

Directly from the definition, the variance σ^2 of the single-level subject heading file can be written:

---

*In this case Equation (93) will be used instead of Equation (93A). Equation (93A) is not sufficiently accurate to be used in computing the variances, because the variances are small. The computation is based upon differences between numbers that are approximately equal.

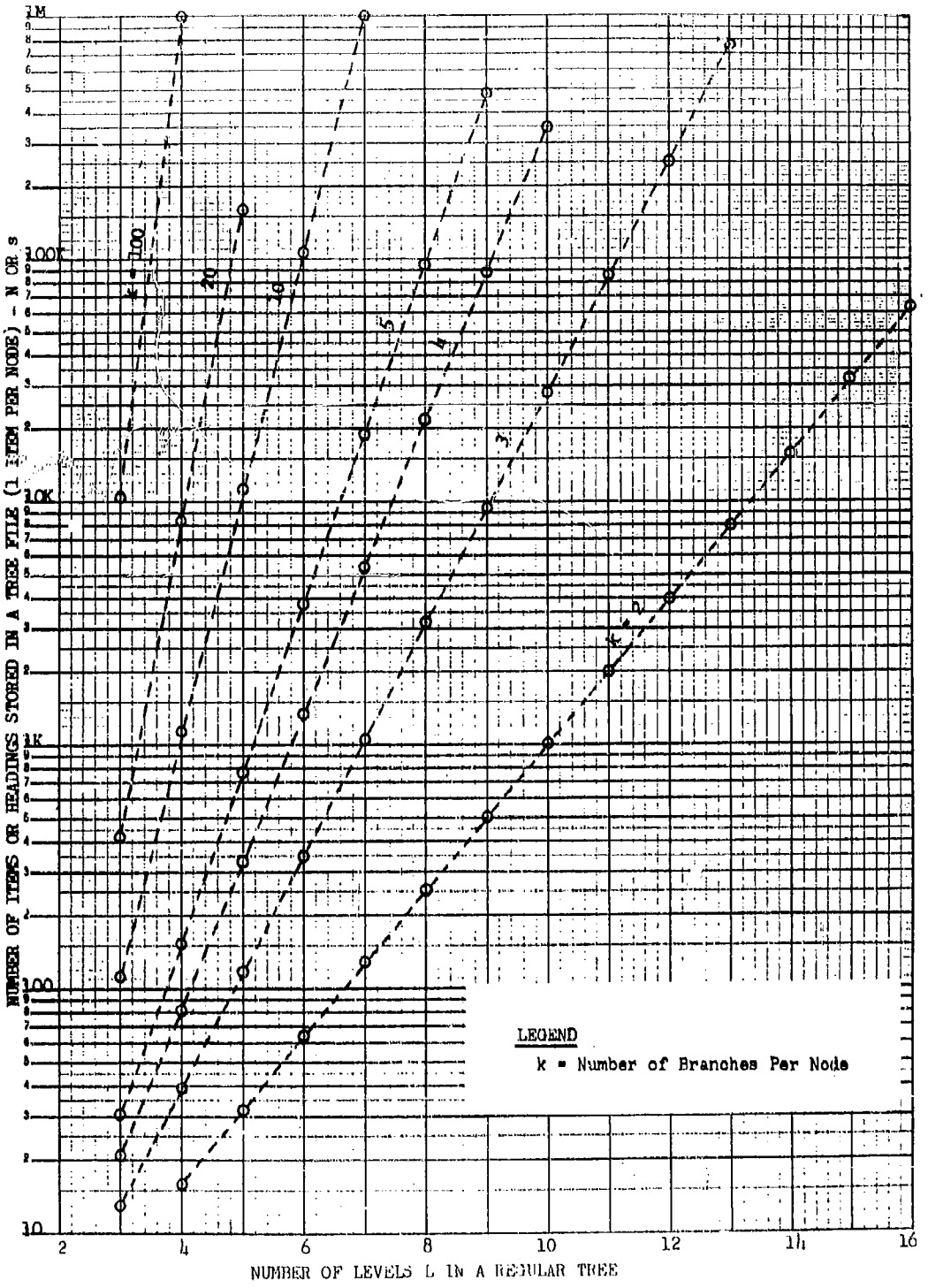


FIGURE 7. Number of Levels Required to Store N Items in a Regular Tree

$$\sigma^2 = \sum_{i=1}^{s} \frac{1}{s} \left[i - \frac{s+1}{2} \right]^2 + \sum_{i=1}^{N_s} \frac{1}{N_s} \left[i - \frac{N+s}{2s} \right]^2 \tag{134}$$

Carrying out the summations yields:

$$\sigma^2 = \frac{(s-1)(s+1)}{12} + \frac{(N/s)^2 - 1}{12} \tag{135}$$

$$\sigma^2_{(85A)} = \frac{1}{12} \left[s^2 + (N/s)^2 - 2 \right] \tag{136}$$

[Note: the subscript such as (85A) references the equation related to a given variance.]

By differentiating Equation (136) with respect to s, setting the result equal to zero, and checking the appropriate requirements, it can be shown that:

$$s = \sqrt{N} \tag{137}$$

gives the minimum variance. Thus the s that gives minimum E, Equations (115) and (116), also gives the minimum variance. This value is:

$$\sigma^2_{min} = \frac{N-1}{6} \tag{138}$$

For the indexed tree of items, the variance is:

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{n} k^{j-1} \left[j - E_{(93)} \right]^2 \tag{139}$$

where n is given by Equation (92). An elementary theorem of mathematical statistics states that Equation (139) is equal to:

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{n} j^2 k^{j-1} - E^2 \tag{140}$$

where E is the expected value obtained from Equation (93). The sum in

Equation (140) can be evaluated by using some relationships among the derivatives of arithmetic and geometric series. Generating functions can also be employed directly and effectively, in this case, to obtain the variance. Using either of these methods, the following expression for the variance can be derived:

$$\sigma^2_{(93)} = \frac{1}{k-1}\left[\frac{L_N^2}{N} - \frac{2L_N}{(k-1)N} - 2L_N + \frac{k+1}{k-1}\right] + L_N^2 - E^2 \qquad (141)$$

where $n = L_N$ is obtained from Equation (92) and E from Equation (93). Equation (141) can be used to compute the variance for relatively small size files (moderately large N).

As N becomes arbitrarily large, however, Equation (141) approaches the following limiting value:

$$\sigma^2_{(93)} \doteq \frac{k}{(k-1)^2} \qquad (142)$$

Equation (141) converges relatively rapidly to Equation (142). For example, when k = 10, the following errors in the variance are introduced:

N	Error in Equation (142)
10^3	1.11%
10^4	.70%
10^5	.05%

This point is primarily of academic interest, since the variances given by Equations (141) and (142) are insignificant. For $k \geq 3$, the variance

given by Equation (141) is less than 1. It can be shown that the variance is a monotonically increasing function of N, and that Equation (142) is an upper limit for the variance.

Applying similar methods, the variances for the other file structures were derived. They are:

$$\sigma^2_{(106A)} = \frac{(k+1)(k-1)}{12}(L_s - 1) + \frac{N_s^2 - 1}{12} \tag{143}$$

where L_s is obtained from Equation (97); N_s, from Equation (104).

$$\sigma^2_{(110A)} = \frac{n^2 - 1}{12} \tag{144}$$

where n is obtained from Equation (109).

$$\sigma^2_{(112A)} = \frac{N_s^2 - 1}{12} \tag{145}$$

where N_s is obtained from Equation (104).

The variances of Equations (106A) and (112A) can now be derived for optimum s. From Equations (97) and (118):

$$L_{s_{opt}} = \log_k \left[\frac{kN}{(k+1)\log_k e}\right]$$

$$= 1 + \log_k \left[\frac{N}{(k+1)\log_k e}\right] \tag{146}$$

Substituting Equations (133A) and (146) into Equation (143) yields:

$$\sigma^2_{(106A)_{opt}} = \frac{1}{12}\left\{(k^2 - 1)\log_k \left[\frac{N}{(k+1)\log_k e}\right] + (k+1)^2(\log_k e)^2 - 1\right\} \tag{147}$$

In the case of Equation (112A), substituting Equation (133B) into Equation (145) gives:

$$\sigma^2_{(112A)_{opt}} = \frac{4(\log_k e)^2 - 1}{12} \qquad (148)$$

Whenever the optimum N_s given by Equation (133B) is less than 1, N_s is taken as 1 and the variance given by Equation (148) is zero. The reason is, of course, that in this case there is a unique indexed procedure to locate any item in a fixed number of steps.

The standard deviations from the expected values are shown in Figure 8. In other words, Figure 8 is a graph of $\sigma_{(85A)_{opt}}$, $\sigma_{(93)}$, $\sigma_{(110A)}$, and $\sigma_{(106A)_{opt}}$ obtained by taking the positive square root of Equations (138), (141), (144), and (147), respectively. The graph was plotted for k = 10. For this value of k, the standard deviation of the indexed tree of headings with sequential items is zero for the reason given after Equation (148). Consequently, this standard deviation has not been included in the graph. As Figure 8 indicates, the standard deviation of the indexed tree of items, Equation (141), is also negligible. Hence, the expected value is a good indicator of the actual number of headings and items examined in a single search of an indexed tree. The standard deviation for the non-indexed tree of headings, Equation (147), is somewhat larger; for the non-indexed tree of items, Equation (144), it is still larger. For reasonably large files, the largest deviation is the single level subject heading file, Equation (138). Consequently, the expected number of headings and items examined is not a good indicator

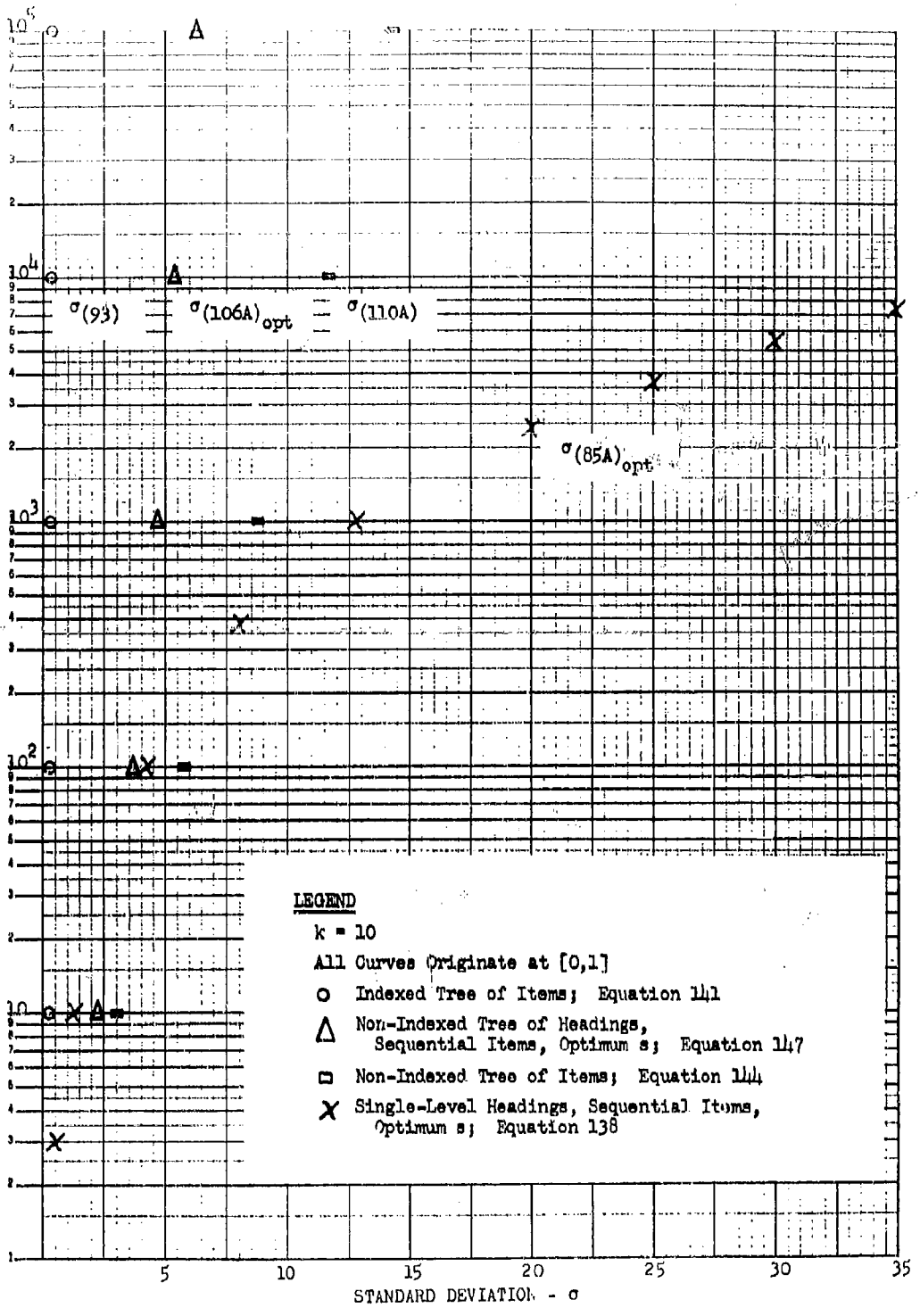


FIGURE 8. Standard Deviation From Average Number of Headings and Items Examined in a Search

of what will occur in any given search of a single-level file. This point is verified by anyone's experience with this kind of file.

Figure 9 compares the cumulative probability distributions for three types of files. It indicates rather clearly the wide variation in n among the file types (with a fixed file size) for any given probability that the number of headings and items searched will be not greater than n in any single search. For example, in a file of 111,111 items the probability is .5 that fewer than 7 items will be examined in an indexed tree; fewer than 25 in a non-indexed tree; but fewer than 335 in a sequential, single-level heading file.

4.4.1.6 Generalized Expressions for Expected Values - The purpose of this section is to present generalized expressions for the expected number of headings and items searched, when two previous assumptions are removed. These assumptions are:

(a) Each subject heading or item is equally likely to be the one sought.

(b) The same number of items is filed under each heading.

For example, if information is available on anticipated or past activity of the file items--and if this information indicates the likelihood of a given heading or item being requested--then the expected number of headings and items searched can be obtained in terms of the available data that approximate the probability distribution of file activity. Generally, the more specialized the contents of a file, the better known and more stable will be its activity. When the activity of the file is known and it is relatively stable, it is clearly advantageous to organize the file so that

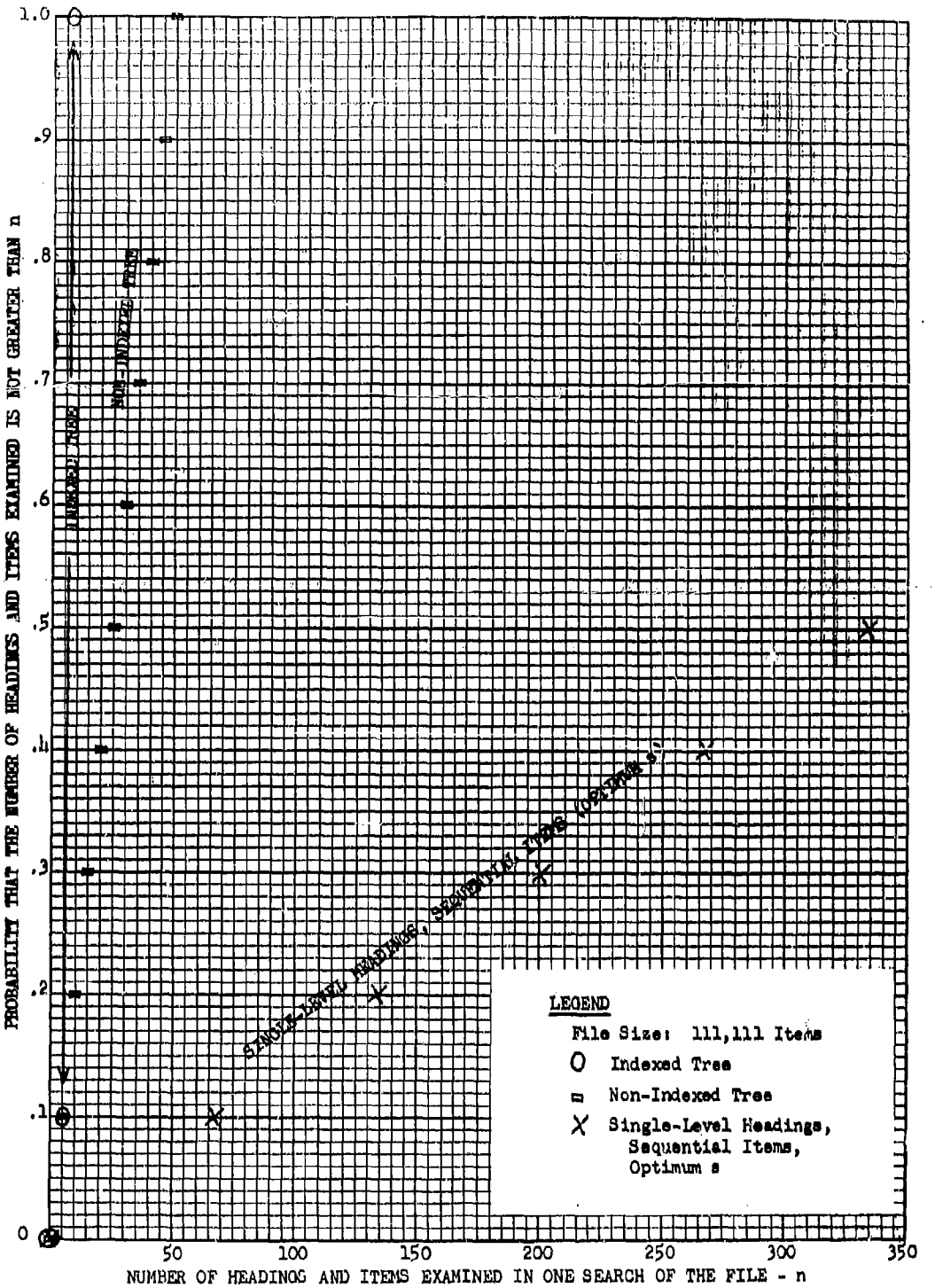


FIGURE 9. Cumulative Probability Distributions for a Search of Differently Organized Files

the items that have the greatest likelihood of being requested are the most accessible. For obvious reasons such a file is called activity organized. It is the intent of this section to provide a general background for the investigation of activity organized files in terms similar to those appearing in previous sections. For the sake of simplicity, expressions for expected values will be presented for only two of the file organizations. These expressions will provide a starting point for the analysis of activity organized files. In each case, p(i) indicates the probability that the $i^{\underline{th}}$ item or heading is the answer to a request.

The single-level subject headings with sequential items, Equation (85), generalizes to:

$$E = \sum_{i=1}^{s} i p_s(i) + \sum_{i=1}^{s} \left[\sum_{j=1}^{n_i} j p_i(j) \right] p_s(i) \qquad (149)$$

where s = the number of subject headings in the file.

n_i = the number of items under heading i.

$p_s(i)$ = the probability that the answer to a request is under heading i.

$p(j)$ = the probability that item j is the answer to a request.

$p_i(j)$ = the probability that item j will be requested, given that it is filed under heading i.

This last probability is obtained from:

$$p(j) = p_s(i) \cdot p_i(j) \qquad (150)$$

The expected value for the indexed tree of items, Equation (93), generalizes to:

$$E = \sum_{j=1}^{n} jp(j) \tag{151}$$

where p(j) is the probability of finding the answer on the j^{th} cut; it is given by:

$$p(j) = \sum_{i=1}^{k^{j-1}} p_j(i) \tag{152}$$

where $p_j(i)$ is the probability that the i^{th} node on level j is the requested item. Values for n are obtained from Equation (92).

4.4.1.7 Summary - Conclusions have been developed and presented throughout this section and will be summarized only briefly. These conclusions are valid only for files where every heading and item is equally likely to be required for a response.

(a) In terms of expected values, indexed trees give a lower average number of headings and items examined than non-indexed trees. Non-indexed trees give lower values than single-level subject headings, except for small files. The break-even points can be determined precisely from the equations in Section 4.4.1.4.

(b) Whenever a file of items can be indexed or ordered into a tree structure, it is disadvantageous, in terms of expected values, to superimpose any heading structure on the items.

(c) For trees and single-level subject heading files, relationships between the number of headings s and the number of items N in the file minimize the expected number of headings and items that will be examined in a file search.

(d) The standard deviation from the average number of headings and items examined for indexed trees is small. Consequently, these average numbers are excellent indicators of the number of headings and items likely to be examined in a single search. The deviations for non-indexed trees are somewhat larger, so expected values have less utility. Finally, the deviation from the expected values of the file with single-level headings and sequential items is so large that the average values are poor indicators of the number of headings and items examined in any single search.

4.4.2 The Multi-List System

4.4.2.1 General - This section surveys and summarizes some basic concepts of information storage and retrieval and their related mathematical models. It is intended primarily to provide a comprehensive review and evaluation of the Prywes and Gray Multi-List system, but within the constraints of their report [62].

The need for a new approach to the solution of information retrieval problems had led some investigators to abandon the addressable memory in favor of an associative type of memory, in which information can be retrieved on the basis of content rather than physical location or address. However, it is possible to use an addressable memory in such a way that information can be retrieved on the basis of its description by simulating an associative memory. For instance, Newell, Shaw, and Simon [52] simulated by programming a type of associative memory in which

lists of arbitrary length and organization could be generated by annexing registers from a common store.

One major advantage of an associative store is that the allocation of storage space for data is coordinated with the actual generation of the data, thus achieving a sort of local optimization, since each basic item of data occupies a minimal amount of space. A second advantage is that data having multiple occurrences usually need not be stored in more than one place, since there is an overlapping or intersection of lists. The Multi-List system extends the associative memory-list storage concept; each item of data appears only once in an addressable memory, and descriptors and control information place the data item on a number of separate lists. Although this technique requires a large amount of storage, it has fast access and retrieval. The advantage of using an addressable memory to simulate an associative memory is that this method permits a versatility of requests and responses that are not attainable by building the associative memory features into the hardware.

Much of the literature on file organizations and storage allocation techniques indicates that chain allocations and tree structures are among the best techniques available for efficient storage and retrieval systems. The chained allocation is simply a type of list processing technique in which each item is associated with the addresses of other related items of the file. The tree structures often encompass several types of allocation techniques; for example, combining random and ordered allocation. A system that provides an efficient combination of the tree

structure and list storage techniques would appear to be a promising solution to the information storage and retrieval problem; hence, the investigation of the Prywes and Gray Multi-List system, which combines these two techniques.

4.4.2.2 Description of the Multi-List System - The Multi-List system described in the Prywes and Gray report has the following system requirements:

(a) The use of an associative memory--for storing, deleting, and reading of information without requiring addressing.

(b) A hierarchy of memories varying in speed and storage capacity.

(c) Processor organization and timing that are intended to minimize the time for instruction retrieval and housekeeping routine.

(d) Processor instructions that can process items of data of varying length.

(e) Built-in automatic retrieval of programs by name to allow for much greater vocabulary and ease of communication with the computer.

4.4.2.2.1 The Descriptive Structure - Information is stored in the Multi-List system in the form of a set of items, each with an associated set of descriptors. Each descriptor specifies a single property of the item. A descriptor consists of an attribute and a value; the attribute specifies a class of descriptors (e.g., color, account number), and the value specifies the actual element of the class (e.g., chartreuse, 20178). Two descriptors are mutually exclusive if no single item can be described by both of them; attributes are defined so as to ensure that any two descriptors with the same associated attribute (e.g., color-chartreuse and color-green) are mutually exclusive. For the sake

of efficiency, attributes are organized into groups called superfields, and the values associated with the attributes in the superfield are combined to form a numerical key. Thus, keys bear the same relation to superfields that values bear to attributes.

4.4.2.2.2 The Memory Structure - An addressable memory is used to simulate the associative memory. This memory is divided into two parts:

(a) The tree structure - The tree structure is used in order to provide access to all items having a given set of descriptors. In describing the tree, the terms branch and node are used in the usual sense. Each branch emanating from the top node represents a superfield. The lowest-level nodes under a superfield give the individual keys associated with that superfield, and each intermediate node represents the set of nodes below it. Thus, as one traverses the tree from bottom to top, one starts with an individual key and encounters successively larger sets of keys, each of which contains the preceding set. Since all keys are numerical, an appropriate arrangement makes it possible to label each node with an indication of the set of keys it represents. Consequently, it is easy to trace down the tree from top to bottom and locate the node at the bottom level corresponding to a particular given key.

(b) The multi-association area - In this area the file is contained in the form of lists. A list consists of a sequence of items.

Each item contains the machine address of the next item on the list. A list emanates from each bottom node of the tree; the list contains precisely those items that are associated with the key corresponding to the node. An item can be contained in as many lists as there are superfields, though it may be contained in a smaller number of lists. Each item consists of a sequence of catenae; catenae are of several types. Two of these types are data catenae and associative catenae. Data catenae provide information not given by any of the descriptors represented in the tree. Associative catenae record a key and the next item on the list associated with that key. Thus, each item has as many list successors as it has keys (unless, of course, it is the last item on a key list).

A search down the tree structure is used to translate the combination of descriptors given in a retrieval or change request into the address of the first item on a list containing the items satisfying such a description. This list, which originates at a bottom tree node, is followed to retrieve or change the contents of the items. The list may, however, contain extraneous items. One advantage of this type of storage organization is the efficiency of retrieval, since a search is required through only a small part of the total storage, while duplication of items is still avoided. Other advantages include the ability to retrieve by partial description and the ease of adding items and descriptors. Deletion, however, is less economical.

Any available space can be used to store information in the Multi-List system. The addresses of the available spaces are kept in a List of Available Space (LAS); when an item is added or deleted, the LAS is changed to record the appropriate modification. The information structure of the Multi-List system is such that multiple paths to each item are provided in the storage space; namely, through the trees for each superfield associated with the item. The computer must be programmed to choose the appropriate superfield when more than one is involved in a retrieval request.

Several assumptions were made with respect to the organization of the memory. First, it is assumed that a tree with the same number of branches emanating from each node except at the lowest level (a balanced tree) can always be constructed. A process for generating these trees is described. Another assumption is that it is possible to divide the totality of descriptors in an arbitrary information retrieval file into attributes. In complex problems this separation of descriptors into exclusive attribute groups may not be an easy task. A process for machine analysis of the file to determine these groupings is also described.

4.4.2.2.3 Maximization of Efficiency - Different types of file organizations are usually compared on the basis of operating time and storage capacity to determine relative efficiency. These criteria are not always the best measures to use, since it is often possible to improve one at the expense of the other. One function that overcomes this type of difficulty is the product of search time and storage capacity, which

can be considered as the cost of operating the system, since storage capacity measures the amount of equipment required and search time measures the time the equipment is in use. In the work on file organizations by Hayes [31] maximum efficiency and minimum cost is achieved by minimizing search time; a method for computing average search time is also described. The Multi-List system employs a technique for the maximization of efficiency based upon a minimum of the product of storage capacity and retrieval.

The balanced tree is particularly well suited as a decoding network for retrieval requests since the search time is almost equal for all terminal tree nodes. The ability to have branches of the tree associated with monotonically increasing numerical values makes the tree an efficient tool for sorting an arbitrarily arranged ensemble of numbers. The mutual exclusion of descriptors of an item can be used as a criterion by which the computer can separate descriptors into distinct attribute groups. The tree mechanism appears to be an efficient tool in the process of establishing attribute groups whose members (descriptors) are mutually exclusive.

A balanced tree is built by progressively adding more keys as more data items are entered into the Multi-List memory. Keys of the first data item to be filed form the nodes of the initial tree. Keys of subsequent data items are incorporated into the tree structure according to various rules.

When a new item is to be added, the relationship of the new

item to all other items in the tree is determined. If any items having the same keys as the current item were filed previously, then the lists are located and the new item is incorporated in the corresponding lists according to the established order (push-down fashion, alphabetical order, etc.). When only part or none of the required lists exist, new lists are added. To maintain the monotonic order of the key values, the keys corresponding to the new lists are entered at specified locations in the lowest level of the tree. If the tree item corresponding to this location contains a vacant catena, then the key corresponding to the new list and the address referring to the new item are inserted so as to preserve the monotonic order of the keys. If no vacant catena is available, a procedure that creates one is invoked. The depth of the tree is increased whenever the required number of keys for an attribute increases beyond a power of the number of nodes per level.

4.4.2.2.4 Automatic Stratification of Information - The use of content addressed memories alone is not sufficient to solve the retrieval problem without additional stratification of the descriptor language. In the Multi-List system the input data are semi-automatically processed into attribute groups for input to the Multi-List trees; this process improves efficiency in terms of speed, storage capacity, and versatility of retrieval. The desired stratification of the descriptor language consists of separating the entire vocabulary into attributes, each consisting of a set of mutually exclusive descriptors.

In many problems, there exists a natural set of attributes. This is true, for instance, of the example discussed in Section 4.4.2.2.7.

More generally, it is necessary to discover such a set. The fewer the number of attributes needed, the more efficient the system will be. The lower bound on the number of attributes is the number of descriptors that can simultaneously apply to an item; the upper bound is the total number of possible descriptors, and this is usually quite large. The problem of selecting exclusive attributes is somewhat analogous to the problem of orthogonalizing a set of vectors via a linear transformation, where the vectors may be of different dimensions. Each item corresponds to a vector with as many components as the item has descriptors. The minimal number of attributes required is analogous to the minimum dimension of the space in which the vectors can be made orthogonal.

The Multi-List system includes an algorithm for assigning descriptors to attributes at the time that the descriptors show up attached to input items. Thus, the attribute assignment program receives inputs at successive moments of time; each input consists of a set of descriptors, some of which are new and some of which have already been assigned attributes. Each of these descriptors must then be associated with a different attribute. Performing this assignment may be quite complicated, and may involve the creation of new attributes. The system includes provision for assistance to the machine at this task from a human being. The algorithm as stated appears to be rather inefficient in terms of minimizing the number of attributes needed.

4.4.2.2.5 The Memory Synchronizer - The list organization of the memory permits the flow of data in and out of the memory on the basis

of content rather than location. This organization, which behaves like an associative memory, would employ a variety of storage devices, e.g., core storage for fast access and limited capacity, drum of disc for intermediate access and capacity, and tapes for slow access and large capacity. In using a hierarchy of memories a coordinating or synchronizer unit is required. The memory synchronizer is designed to be incorporated into the hardware of the list machine memory. It has four basic instructions: read item, store item, replace catena, and delete item. Its purpose is to handle the memory space assignment of incoming or deleted data and to synchronize the processor and the memory.

4.4.2.2.6 The Multi-List Processor - The design of a Multi-List processor for this system is approached in two different ways. The primary difference in the two approaches is that the second approach uses an instruction memory for storing micro-instruction routines; the first approach is based upon macro-instructions. In the first approach the processor is developed from the basic operations of transfer and compare. In the second approach, more complex processes were selected as the basic processes--finding, filing, and deleting an item of information; these processes require sets of micro-instructions to carry out each function.

Both design approaches call for a hierarchy of memories--for example, a parallel, read-only memory such as the UNIVAC search memory for high speed operations, and a slower access memory for storing the mass of data. These design approaches deal mainly with programming and

hardware to implement the Multi-List system and need not be described in detail.

4.4.2.2.7 Sample Problem - Consider a personnel file of approximately 10^6 items. The file contains the names of the personnel and their descriptions in terms of a fixed set of attributes or categories of information. The description will be made up of 15 exclusive attributes where each attribute can have a fixed number of values. Ten values per attribute are assumed. The 15 attributes used for this problem are as follows:

A_1	Height	Superfield I
A_2	Weight	
A_3	Age	
A_4	Hair Color	Superfield II
A_5	Eye Color	
A_6	Race	
A_7	Rank	Superfield III
A_8	Branch of Service	
A_9	Years in Service	
A_{10}	Nationality	Superfield IV
A_{11}	Education	
A_{12}	Religion	
A_{13}	Marital Status	Superfield V
A_{14}	No. of Children	
A_{15}	No. of Dependents	

The values of the attributes (or descriptors) will be represented by the digits 0,1,2,...,9. The attributes are grouped into superfields, in which each superfield represents 3 attributes; hence, there are five superfields. This is done in order to represent these attributes efficiently in a tree structure. Three descriptors per superfield will give the values of the attributes; the combined values form a key. For each key, the values range from 0 to 999.

An item of information such as a person's name and description is represented in this system by two types of catenae: data catenae and associative catenae. The data catenae contain the name; the associative catenae contain the descriptors and addresses associated with them. The attributes have positional significance in an item, as shown in Figure 10.

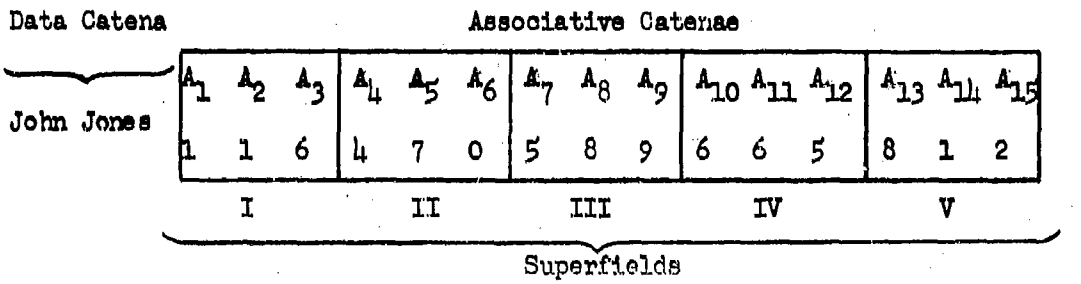


FIGURE 10. Relation of Data Catena and Associative Catenae

The tree structure for this example is shown in Figure 11. The upper part of the diagram represents the tree structure, and the lower part represents the multi-association area. Each point in the tree represents the multi-association area. Each point in the tree represents a set of keys, some of which are explicitly indicated in the diagram. The

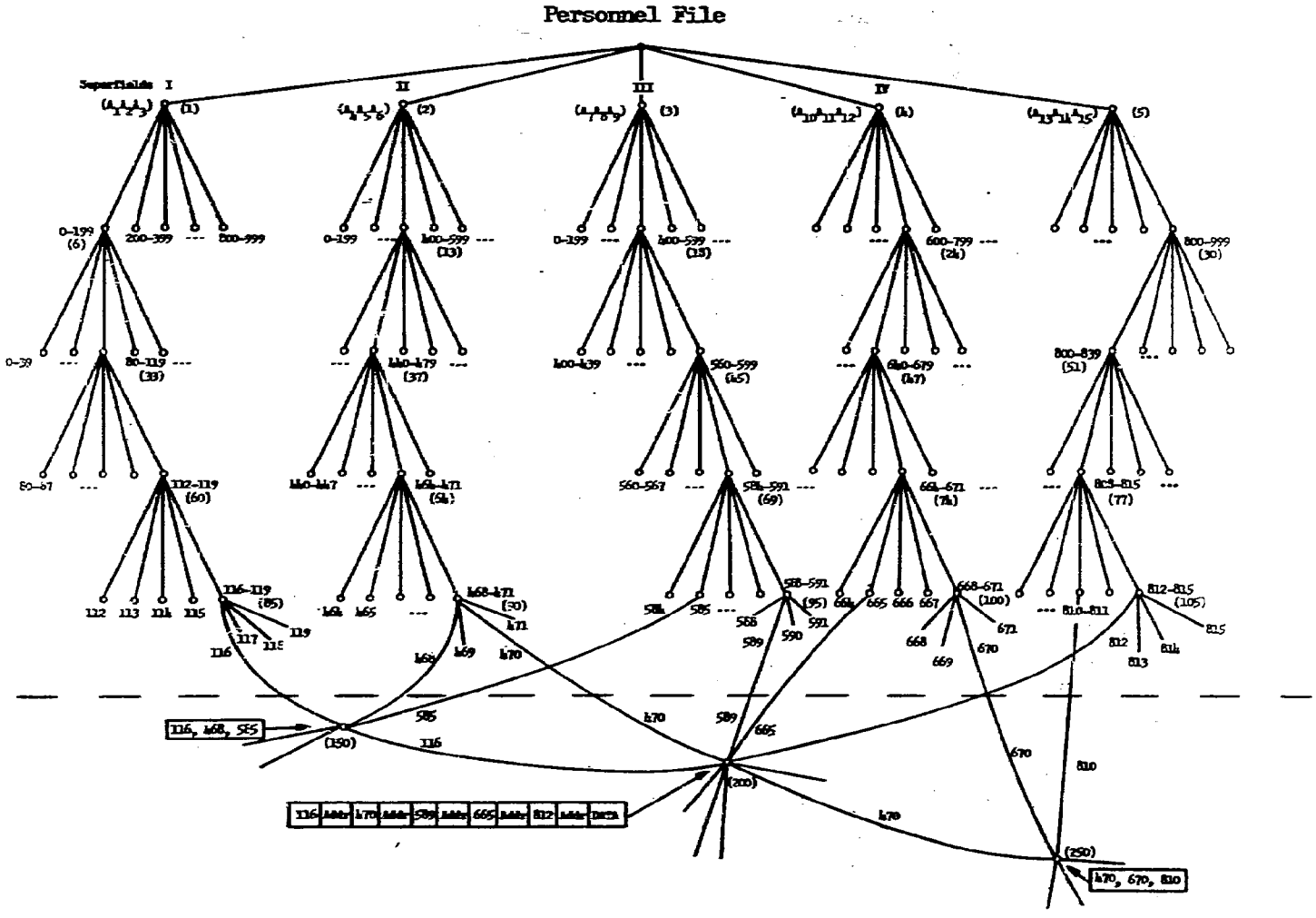


FIGURE 11. Multi-List Organization for a Personnel File

numbers in parentheses associated with each node (in either area) represent hypothetical memory locations; these are used for illustrative purposes in the sample problem. A trace can be made on any value of any one of the 5 keys; the trace will lead to a node on the lowest level of the tree. At this level the address of the head of a list containing all items having that same key will be retrieved. The intersection of the lists for each key contained in the item will yield the appropriate item. Figure 12 illustrates the partial contents of the Multi-List memory for the sample problem. The arrows indicate the path to be taken if a search is made on the key for Superfield I (116) in order to arrive at the appropriate item.

4.4.2.3 Summary and Evaluation - The Multi-List system for information retrieval utilizes a conventional memory to simulate an associative memory, thus gaining some of the advantages of each. It employs a novel memory organization that incorporates both a conventional tree structure and an unconventional list structure; the list structure differs from most others in that each a single element may actually appear as part of several lists. This is accomplished by permitting an element to have several distinct list successors. In the Multi-List system, searching is extremely rapid and searching on at least some types of partial description can be performed with no loss of time; if the partial descriptions to be used can be anticipated in advance, then the memory can be organized to handle them efficiently. There has been considerable investigation of machine organizations and logic that can handle the Multi-List system efficiently.

On paper, the system appears quite reasonable. However, it cannot be operated on most conventional computers without significant loss of efficiency. The problem of efficient deletions remains unsolved. Difficulties arise, also, in organizing data into the attribute-value descriptors used by the system. It is necessary to structure the data so that the number of attributes will not be unduly large, and no really general way of doing this has yet been found.

In a paper on automatic stratification of information presented at the 1963 SJCC [42] a hand-simulated example is given using natural language (represented by a 2-digit code). This simulation has also been programmed for the IBM 7090 using artificial input and a small amount of ASTIA (or, presently, DDC) live data. This technique looks promising for at least certain types of information retrieval problems once the technique is fully developed. As is generally the case, the examples used have a limited scope; a great deal of development is required before the concept can be practicably implemented. The question remains as to whether all types of information retrieval data will be adaptable to descriptor/attribute stratification.

4.5 QUERY PROCESSING

In an important sense the answers to all the preceding questions determine in large measure the query capabilities of a system. Conversely, the descriptor and processing structures must be designed to accommodate query requirements. The state-of-the-art in query capabilities of operating information retrieval systems at the inception of this project was

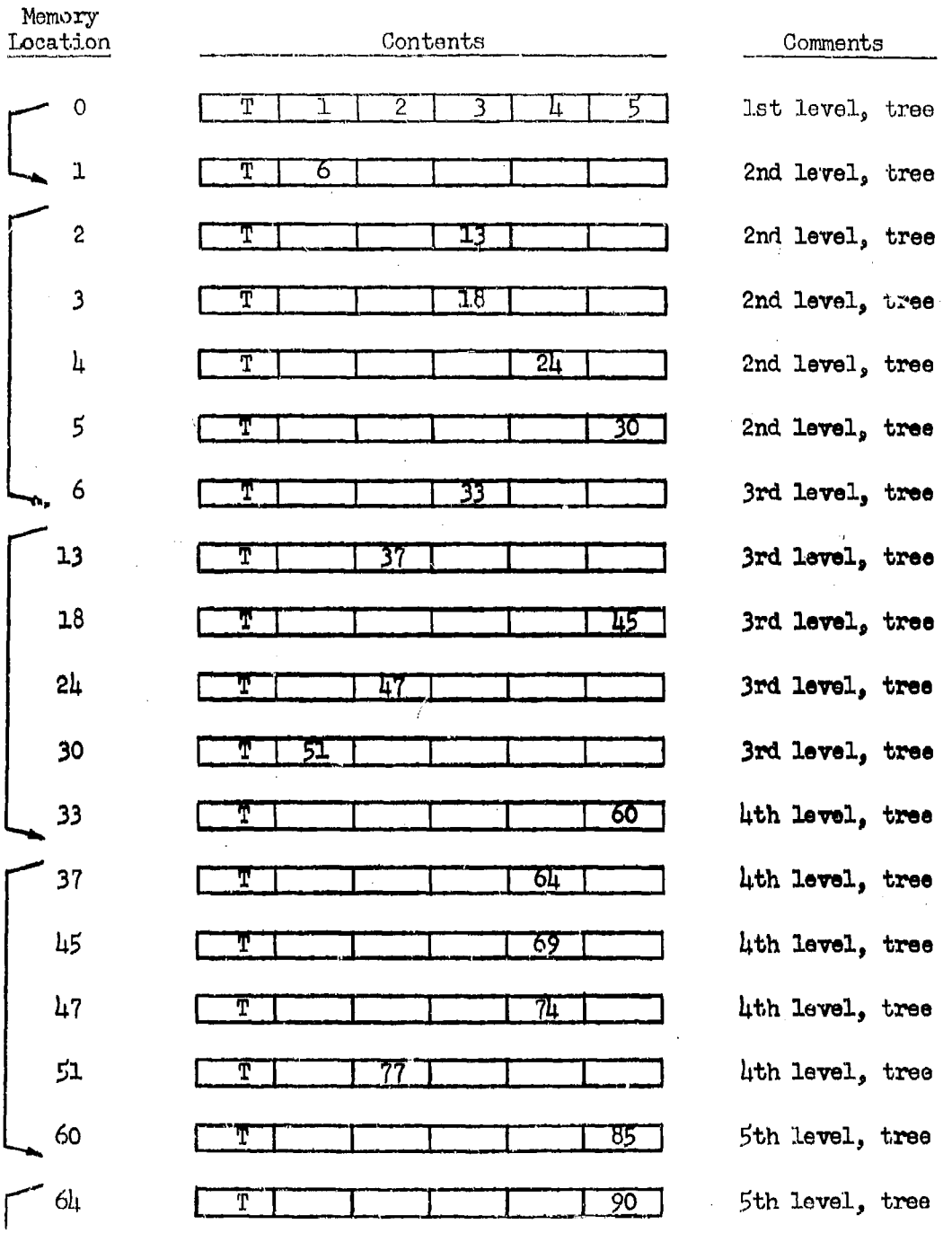


FIGURE 12. Example of Multi-List Memory Contents for Figure 11

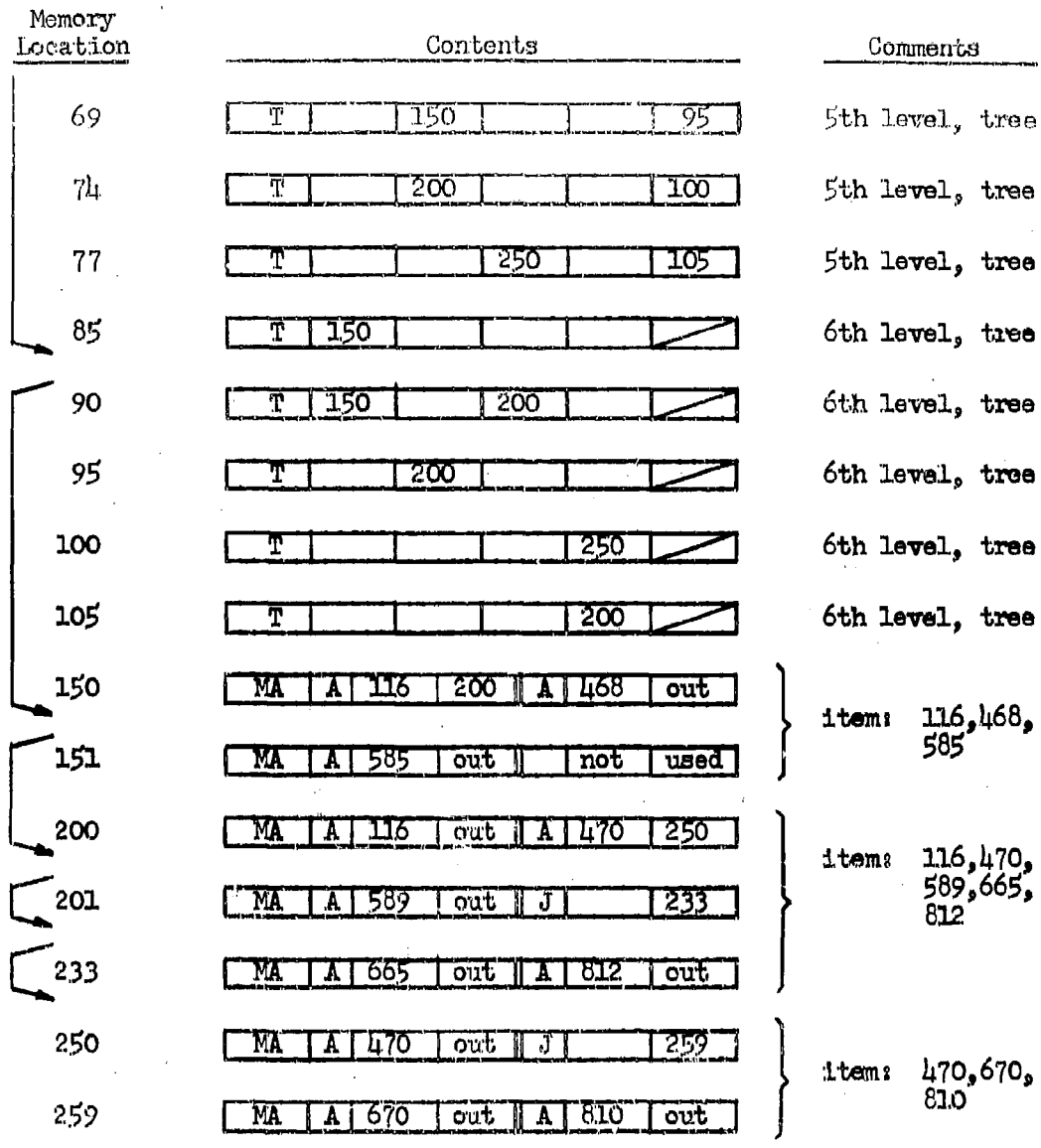


FIGURE 12. Example of Multi-List Memory Contents for Figure 11 (Continued)

limited to locating the documents that satisfy some level of Boolean concatenation of descriptors. Most of the early work on the project implicitly assumed essentially such a query capability. Since this approach is well understood, it will not be considered further.

For a descriptor-oriented retrieval system in which documents have probabilities attached to each descriptor, a mere matching procedure is not suitable for query processing. One method of treating this situation is to assign probability thresholds for the different descriptors; the assignment will be dependent upon the nature of the query.

One of the major problems in generating the appropriate response to a query is the existence of redundancy in the retrieved data. In certain applications, such as personnel file processing, this problem poses no appreciable difficulty. In literature retrieval or intelligence analysis, the problem may become acute. Therefore, analysis of the redundancy problem is a cogent necessity.

It was pointed out in Section 4.1.5.3 that query processing is a non-trivial problem in dealing with intelligence data, but much less of a problem in simpler situations. In the most difficult situation, the system must be designed around the concept of a dialogue between the system and the user. In addition, the full power of an implicit information system may be necessary. In this section, both of these aspects of query processing will be discussed.

4.5.1 Probabilistic Retrieval - The purpose of this section is to present a method for deciding which documents should be retrieved in

response to a query, given that a description consists of a list of non-exclusive category names, when documents are assigned to categories probabilistically rather than absolutely. The decision algorithm will be developed on the basis of maximizing a value function that measures the goodness of the set of retrieved documents. Before proceeding further, however, it will be helpful to examine some specific situations in which probabilistic retrieval would be appropriate.

(a) The Case of Many Users - A situation may occur where the views of users regarding membership of some documents in a certain category are divergent. Assume, for example, that there are 100 users, 5 categories, and 10 documents. Each user is asked to assign each document to one or more categories. Table 2 illustrates a possible set of choices. The numbers at the intersection of rows and columns indicate the probability of a document belonging to a certain category. Thus document No. 10 will belong to category D with probability 1, since all the users agree to place it there. On the other hand, the same document will have a probability of zero of belonging to category B; again, all the users agree to exclude it from this category. Since 45 percent of the users agreed to place document No. 10 in category A, it has been assigned a probability of .45.

(b) Automatic Category Formation - Documents may be assigned to categories in accordance with an automatic procedure. This procedure may be intrinsically probabilistic in nature; that is, a document is assigned to a category with probability p dependent upon the circumstances pertaining to the assignment.

TABLE 2. PROBABILISTIC ASSIGNMENT OF DOCUMENTS TO CATEGORIES BY USERS

	DOCUMENTS									
CATEGORIES	1	2	3	4	5	6	7	8	9	10
A	65	50	75	80	25	0	0	15	30	45
B	100	50	35	40	60	25	50	75	25	0
C	90	80	60	0	20	50	40	0	0	10
D	35	50	25	30	15	15	0	25	80	100

The specific response to a query will be determined through the use of one or more cutoff points. For retrieval on a single category, documents belonging to the category with a probability greater than or equal to the cutoff point will be included in the response; all others will be excluded. For queries specified as Boolean functions of categories, multiple cutoff points will be needed, one for each category involved in the query. The selection of cutoff points will be performed in such a way as to maximize the goodness of the response. The following questions must then be answered:

(a) How is the goodness of a response to be determined quantitatively?

(b) How is the cutoff point for a simple (i.e., one-category) query to be determined?

(c) How are the cutoff points for a compound query to be determined?

These questions will be considered in the sequel.

4.5.1.1 The Problem of Establishing Criteria for Determining User's Value of An Average Retrieval Procedure - With respect to any retrieval request the entire collection of documents may be divided into four subgroups:

(a) The retrieved documents that are relevant.

(b) The retrieved documents that are not relevant.

(c) The unretrieved documents that are relevant.

(d) The unretrieved documents that are not relevant.

Since it was assumed that the documents are assigned to categories on a probabilistic basis, all four subgroups will generally be represented in any retrieval process.

Regardless of any special assumptions, it is clearly permissible to assert that as the number of documents in categories (a) and (d) increases and as the number of documents in categories (b) and (c) decreases, the value of the retrieved collection to the user will increase. Thus,

$$V = f_1\{I\} - f_2\{II\} - f_3\{III\} + f_4\{IV\} + K \qquad (153)$$

where V is defined as the user value of the retrieved collection; f_1, f_2, f_3, and f_4 are unspecified, monotonically increasing functions; and {I}, {II}, {III}, and {IV} are the numbers of documents in the subclasses (a), (b), (c), and (d), respectively. K is defined as a constant that determines the minimal value for the user below which the retrieval is not justified under any circumstances.

For simplicity, replace f_1, f_2, f_3, and f_4 by the constants α, β, γ, and δ, and set $K = 0$. The results of this discussion are not essentially modified by this simplification. Equation (153) then becomes:

$$V = \alpha\{I\} - \beta\{II\} - \gamma\{III\} + \delta\{IV\} \qquad (154)$$

Since $K = 0$, the retrieval process should proceed as long as the increment

of V, dV, is positive. That is, the process may select a group of documents with common probability characteristics (in relation to the request profile) and then investigate the change of V by including some additional documents with lower probability characteristics. The question as to which documents will be retrieved is the problem of fixing the most advantageous values for the set $\{\sigma_i\}$ of cutoff points for the descriptor classes.

The appropriateness of replacing the functions f_1, f_2, f_3, and f_4 by the constants α, β, γ, and δ rests upon the understanding of what factors could be responsible for the non-linearity of the function V. Essentially there are two reasons why the function V should be non-linear. The first pertains to the economics of using documents; the other, to the problem of redundancy. In general, the efficiency with which the retrieved collection is used depends upon its size, even if the value of the individual documents in the collection is not prejudged. Nevertheless, since retrieval systems can be used in various ways, it is safe to assume that for many uses the relative emphasis placed upon the classes of retrieved and unretrieved documents remains unchanged. To the extent that this assumption is true, the fact that the function V depends upon class [IV], the class of correctly unretrieved documents, helps to remedy the situation.

The second reason for non-linearity is more serious. Among the retrieved documents there may be a high degree of redundancy; in some cases the same amount of information may be entirely covered by a smaller number of documents. It is difficult, however, to decide whether or not

redundancy is a linear function of the size of the retrieved collection. To answer this question adequately, it would be necessary to formalize the concept of redundancy among documents and then perhaps to formulate theoretical prescriptions for procedures that would permit the system to retrieve the most efficient covering of the topic specified in the request. (This problem is a difficult task in itself and merits separate investigation.) Pending a quantitative formulation of the theory of redundancy, this discussion will be confined to the simplest assumption of linearity. Therefore, given the function V in the form of Equation (154), the first task is to find the set of cutoff points that will maximize the user's value for an average retrieval process.

4.5.1.2 Determination of Cutoff Points for Simple Queues - We start by introducing some notation. We assume that there are s categories, denoted by the integers i = 1,2,...,s. To facilitate computation, the number of documents in each class are assumed to be large enough and the subdivision into the probability brackets fine enough to permit integration techniques to replace summation. Let:

$$
\left.
\begin{aligned}
&N_i(p) = \text{the number of documents in category } i \text{ with probability } p \text{ or less.}\\
&n_i(p) = \frac{dN_i(p)}{dp}.\\
&\bar{p}_i(\sigma) = \frac{1}{N}\int_0^{\sigma} n_i(p)\,p\,dp.\\
&\bar{p}_i = \bar{p}_i(1)\\
&f_i = \text{the frequency with which category } i \text{ is requested.}
\end{aligned}
\right\} \quad (155)
$$

σ_i = the cutoff point for category i.

N = the total number of documents in the collection.

If we assume that every document belongs to every category with at least some non-zero probability, then we have:

$$N_i(0) = 0$$

and;

$$N_i(1) = N$$

We also assume that $n_i(p)$ is non-zero throughout the interval [0,1], since its value can always be made sufficiently small to be statistically insignificant.

The quantity $\overline{p}_i(\sigma)$ represents the expected proportion of incorrectly unretrieved documents when retrieval is performed with cut-off point σ, that is:

$$\overline{p}_i(\sigma) = \frac{\text{incorrectly unretrieved documents}}{\text{total documents in the collection}}$$

To follow this point, note that for $0 \leq p < \sigma$, the expected number of documents in the interval $(p, p + dp)$ is $n_i(p)dp$, and that p of these documents will actually belong to category i. Thus the number of documents in the interval belonging to i will be $pn_i(p)dp$, and since $p < \sigma$, none of these documents will be retrieved. Since these documents do in fact belong to category i, they are incorrectly unretrieved. $\overline{p}_i(\sigma)$ is obtained by integrating $pn_i(p)dp$ over the interval from 0 to σ, thus covering all incorrectly unretrieved documents. Note also that $\overline{p}_i$ represents the expected proportion of documents in category i, since with a retrieval threshold of certainty no documents will be retrieved; hence

all documents in category i will be incorrectly unretrieved.

We note also that from Equation (155):

$$N_i(\sigma) = \int_0^{\sigma} n_i(p)dp \tag{156}$$

The procedure for calculating the set of σ_i's that will maximize V is:

(a) Calculate the numbers of documents for the four subclasses of documents that enter V for an unspecified σ_i.

(b) Obtain a general expression for V for a single category.

(c) Obtain an expression for the expected value for all V's.

(d) Differentiate the expression obtained under (c), and set the s coefficients of the differentials equal to zero in order to obtain a set of conditions for the maximum.

(e) Solve the equations to obtain the values of the σ_i's.

We will permit different σ_i for different categories.

We first calculate the number of documents in each subclass:

(a) Class I - The class of all correctly retrieved documents:

$$\{I\} = \int_{\sigma_i}^{1} pn_i(p)dp \tag{157}$$

(b) Class II - The class of all incorrectly retrieved documents:

$$\{II\} = \int_{\sigma_i}^{1} (1 - p)\, n_i(p)dp \tag{158}$$

(c) Class III - The class of all incorrectly unretrieved documents:

$$\{III\} = \int_0^{\sigma_i} pn_i(p)dp \tag{159}$$

(d) Class IV - The class of all correctly unretrieved documents:

$$[IV] = \int_0^{\sigma_i} (1 - p)\, n_i(p)dp \tag{160}$$

For a query on category i, then, we have:

$$V_i = \alpha \int_{\sigma_i}^{1} pn_i(p)dp - \beta \int_{\sigma_i}^{1} (1 - p)\, n_i(p)dp$$

$$- \gamma \int_0^{\sigma_i} pn_i(p)dp + \delta \int_0^{\sigma_i} (1 - p)\, n_i(p)dp \tag{161}$$

The expected value of V over all categories is obtained as a weighted sum:

$$V = \sum_{i=1}^{s} f_i V_i$$

$$= \sum_{i=1}^{s} f_i [\alpha \int_{\sigma_i}^{1} pn_i(p)dp - \beta \int_{\sigma_i}^{1} (1 - p)\, n_i(p)dp$$

$$- \gamma \int_0^{\sigma_i} pn_i(p)dp + \delta \int_0^{\sigma_i} (1 - p_i)\, n_i(p)dp] \tag{162}$$

The conditions for a maximum are obtained by setting the partial derivatives with respect to each σ_i to 0:

$$f_i[- \alpha \sigma_i\, n_i(\sigma_i) + \beta(1 - \sigma_i)\, n_i(\sigma_i) - \gamma \sigma_i\, n_i(\sigma_i)$$

$$+ \delta(1 - \sigma_i)\, n_i(\sigma_i)] = 0 \tag{163}$$

Dividing by $f_i\, n_i(\sigma_i)$ yields:

$$\left.\begin{aligned} &- \alpha \sigma_i + \beta - \beta \sigma_i - \gamma \sigma_i + \delta - \delta \sigma_i = 0 \\ &\sigma_i(\alpha + \beta + \gamma + \delta) = \beta + \delta \end{aligned}\right\} \tag{164}$$

so that;

$$\sigma_i = \frac{\beta + \delta}{\alpha + \beta + \gamma + \delta} \qquad (165)$$

The quantity (165), then, is the optimal cutoff point for single-descriptor queries. It is of interest to note that the cutoff point is the same for all categories and, in fact, does not even depend on the probability distribution of documents within the categories.

4.5.1.3 Determination of Cutoff Points for Compound Queries - We now consider queries that are of the form $c_i \bullet c_j$; that is, we seek documents that belong both to category i and to category j. In general, the thresholds to be used on the individual categories will be different for joint retrievals than for simple ones. We will initially assume that the distributions of documents within categories are independent; that is, that the membership of a document in category i does not affect the probability of its membership in category j. We will also require that a single cutoff point be established for each category given that the query is of the form $c_i \bullet c_j$. As part of our independence assumption we will assume that:

$$f_{ij} = f_i f_j \qquad (166)$$

Thus the frequency of retrieval on a joint category is the product of the frequencies on the individual categories. Under these assumptions, we can carry out the analysis in the same way that we did for simple queries.

4.5.1.3.1 Development of the Cutoff Point Equations - We let $N_{ij}(p_i,p_j)$ denote the cumulative joint distribution function for categories

i and j; therefore, $N_{ij}(p_i,p_j)$ represents the number of documents that belong both to category i with probability p_i or less and to category j with probability p_j or less. We let $n_{ij}(p_i,p_j)$ represent the corresponding density function, where:

$$n_{ij}(p_i,p_j) = \frac{\partial^2 N_{ij}(p_i,p_j)}{\partial p_i \, \partial p_j} \tag{167}$$

Similarly, we let $p_{ij}(p_i,p_j)$ denote the average probability of a document belonging to both category i and category j, given that the document belongs to category i with probability p_i and category j with probability p_j.

The assumption of independence of categories can be broken down into two separate mathematical statements:

$$\frac{N_{ij}(p_i,p_j)}{N} = \frac{N_i(p_i)}{N} \cdot \frac{N_j(p_j)}{N} \tag{168}$$

and;

$$p_{ij}(p_i,p_j) = p_i p_j \tag{169}$$

These statements are independent in the sense that neither can be derived from the other, and they represent two different aspects of independence of categories. As a consequence of Equations (167) and (168), we obtain:

$$n_{ij}(p_i,p_j) = \frac{n_i(p_i)n_j(p_j)}{N} \tag{170}$$

for independent categories.

We can write expressions giving the number of documents in each of the four classes involved in the value function V_{ij}:

(a) Class I - The class of all correctly retrieved documents:

$$\{I\} = \int_{\sigma_i}^{1} \int_{\sigma_j}^{1} p_{ij}(p_i,p_j)\ n_{ij}(p_i,p_j)dp_j\ dp_i \tag{171}$$

(b) Class II - The class of all incorrectly retrieved documents:

$$\{II\} = \int_{\sigma_i}^{1} \int_{\sigma_j}^{1} [1 - p_{ij}(p_i,p_j)]\ n_{ij}(p_i,p_j)dp_j\ dp_i \tag{172}$$

(c) Class III - The class of all incorrectly unretrieved documents:

$$\{III\} = \int_{o}^{\sigma_i} \int_{o}^{\sigma_j} p_{ij}(p_i,p_j)\ n_{ij}(p_i,p_j)dp_j\ dp_i \tag{173}$$

(d) Class IV - The class of all correctly unretrieved documents:

$$\{IV\} = \int_{o}^{\sigma_i} \int_{o}^{\sigma_j} [1 - p_{ij}(p_i,p_j)]\ n_{ij}(p_i,p_j)dp_j\ dp_i \tag{174}$$

Since we are assuming independence of categories, we can simplify Equations (171) through (174) by using Equations (169) and (170):

(a) $$\{I\} = \int_{\sigma_i}^{1} \int_{\sigma_j}^{1} \frac{n_i(p_i) \bullet n_j(p_j)}{N}\ p_i\ p_j\ dp_j\ dp_i \tag{175}$$

(b) $$\{II\} = \int_{\sigma_i}^{1} \int_{\sigma_j}^{1} \frac{n_i(p_i) \bullet n_j(p_j)}{N}\ (1 - p_i p_j)dp_j\ dp_i \tag{176}$$

(c) $$\{III\} = \int_{o}^{\sigma_i} \int_{o}^{\sigma_j} \frac{n_i(p_i) \bullet n_j(p_j)}{N}\ p_i\ p_j\ dp_j\ dp_i \tag{177}$$

(d) $$\{IV\} = \int_{o}^{\sigma_i} \int_{o}^{\sigma_j} \frac{n(p_i) \bullet n(p_j)}{N}\ (1 - p_i p_j)dp_j\ dp_i \tag{178}$$

The retrieval process proceeds until the predetermined cutoff point σ_i for descriptor i and σ_j for descriptor j has been reached. To

retrieve beyond this point will be detrimental, since on the average the increment in V caused by additional retrieval will be negative.

The four double integrals in Equations (175) through (178) can now be evaluated. For Equation (175):

$$[I] = \int_{\sigma_i}^{1} \int_{\sigma_j}^{1} \frac{n_i(p_i) \bullet n_j(p_j)}{N} p_i \, p_j \, dp_j \, dp_i$$

$$= \frac{1}{N} \int_{\sigma_i}^{1} n(p_i) p_i \, dp_i \int_{\sigma_j}^{1} n(p_j) p_j \, dp_j$$

$$= N[\bar{p}_i - \bar{p}_i(\sigma_i)] \, [\bar{p}_j - \bar{p}_j(\sigma_j)] \tag{179}$$

Similarly, Equations (176) through (178) become:

$$\left.\begin{aligned} [II] &= \int_{\sigma_i}^{1} \int_{\sigma_j}^{1} \frac{n_i(p_i) \bullet n_j(p_j)}{N} (1 - p_i p_j) dp_j \, dp_i \\ &= \frac{1}{N} \{[N - N_i(\sigma_i)] \, [N - N_j(\sigma_j)] \\ &\quad - N^2[\bar{p}_i - \bar{p}_i(\sigma_i)] \, [\bar{p}_j - \bar{p}_j(\sigma_j)]\} \end{aligned}\right\} \tag{180}$$

$$[III] = \int_{o}^{\sigma_i} \int_{o}^{\sigma_j} \frac{n_i(p_i) \bullet n_j(p_j) p_i \, p_j \, dp_j \, dp_i}{N}$$

$$= N \, \bar{p}_i(\sigma_i) \, \bar{p}_j(\sigma_j) \tag{181}$$

$$[IV] = \int_{o}^{\sigma_i} \int_{o}^{\sigma_j} \frac{n_i(p_i) \bullet n_j(p_j)}{N} (1 - p_i p_j) dp_j \, dp_i$$

$$= \frac{1}{N} [N_i(\sigma_i) \, N_j(\sigma_j) - N^2 \, \bar{p}_i(\sigma_i) \, \bar{p}_j(\sigma_j)] \tag{182}$$

By substituting Equations (179) through (182) into Equation (154),

the function V_{ij} for the value of a joint retrieval on categories i and j becomes

$$\left.\begin{aligned} V_{ij} = \frac{\alpha}{N} \{N^2 [\bar{p}_i - \bar{p}_i(\sigma_i)] [\bar{p}_j - \bar{p}_j(\sigma_j)]\} \\ - \frac{\beta}{N} \{[N - N_i(\sigma_i)] [N - N_j(\sigma_j)] \\ - N^2 [\bar{p}_i - \bar{p}_i(\sigma_i)] [\bar{p}_j - \bar{p}_j(\sigma_j)]\} \\ - \frac{\gamma}{N} N^2 \bar{p}_i(\sigma_i) \bar{p}_j(\sigma_j) \\ + \frac{\delta}{N} [N_i(\sigma_i) N_j(\sigma_j) - N^2 \bar{p}_i(\sigma_i) \bar{p}_j(\sigma_j)] \end{aligned}\right\} \quad (183)$$

By using Equation (183), it is possible to find the values of σ_i and σ_j that will maximize a <u>specific</u> V_{ij}. In general, however, the values σ_i' and σ_i'' obtained by solving the maxima in expressions V_{ij} and, say, V_{ik} will be different. Consequently we need a set of values $\{\sigma_i\}$ that will maximize an <u>average</u> V_{ij}.

The average value of V_{ij} is, of course, its expected value:

$$\begin{aligned} E(V) &= \sum_{i=1}^{s} \sum_{j=1}^{s} V_{ij} f_{ij} \\ &= \sum_{i=1}^{s} \sum_{j=1}^{s} V_{ij} f_i f_j \end{aligned} \quad (184)$$

since $f_{ij} = f_i f_j$ by Equation (166), and this function will have to be maximized. The differential of Equation (184) is:

$$dE = \sum_{i=1}^{s} \sum_{j=1}^{s} f_i f_j \left[\frac{\partial V_{ij}}{\partial \sigma_i} d\sigma_i + \frac{\partial V_{ij}}{\partial \sigma_j} d\sigma_j\right] \quad (185)$$

or

$$dE = \sum_i f_i \left[\sum_j f_j \frac{\partial V_{ij}}{\partial \sigma_i} \right] d\sigma_i$$

which implies the following condition for a maximum:

$$\sum_j f_j \frac{\partial V_{ij}}{\partial \sigma_i} = 0 \qquad (i = 1,2,\ldots,s) \qquad (186)$$

The partial derivatives $\partial V_{ij}/\partial \sigma_i$ in Equation (186) can be computed by using Equations (179) through (187):

$$\frac{\partial [I]}{\partial \sigma_i} = [\bar{p}_j - \bar{p}_j(\sigma_j)] [-\sigma_i \, n_i(\sigma_i)] \qquad (187)$$

$$\frac{\partial [II]}{\partial \sigma_i} = \frac{1}{N} [-N + N_j(\sigma_j)] [n_i(\sigma_i)] + [\bar{p}_j - \bar{p}_j(\sigma_j)] [\sigma_i \, n_i(\sigma_i)] \qquad (188)$$

$$\frac{\partial [III]}{\partial \sigma_i} = \bar{p}(\sigma_j)\sigma_i \, n_i(\sigma_i) \qquad (189)$$

$$\frac{\partial [IV]}{\partial \sigma_i} = \frac{1}{N} [N_j(\sigma_j) \, n_i(\sigma_i) - N \, \bar{p}(\sigma_j)\sigma_i \, n_i(\sigma_i)] \qquad (190)$$

Performing the summations in Equation (186) on Equations (187) through (190) results in:

$$\sum_j f_j \frac{\partial [I]}{\partial \sigma_i} = - \sigma_i \, n_i(\sigma_i) \sum_{j=1}^{s} f_j [\bar{p}_j - \bar{p}_j(\sigma_j)] \qquad (191)$$

$$\sum_j f_j \frac{\partial [II]}{\partial \sigma_i} = \frac{1}{N} n_i(\sigma_i) \sum_{j=1}^{s} [- N + N_j(\sigma_j)] f_j + \sigma_i n_i(\sigma_i) \sum_{j=1}^{s} [\bar{p}_j - \bar{p}_j(\sigma_j)] f_j \qquad (192)$$

$$\sum_j \frac{\partial\{III\}}{\partial\sigma_i} = \sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, \bar{p}(\sigma_j) \tag{193}$$

$$\sum_j \frac{\partial\{IV\}}{\partial\sigma_i} = \frac{n_i(\sigma_i)}{N} \sum_{j=1}^{s} f_j\, N_j(\sigma_j) - \sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, \bar{p}(\sigma_j) \tag{194}$$

Therefore, the condition for a maximum is given by the equations:

$$\left.\begin{aligned} &-\frac{\alpha}{N}\,\sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, N\,[\bar{p}_j - \bar{p}_j(\sigma_j)] \\ &+\frac{\beta}{N}\, n_i(\sigma_i) \sum_{j=1}^{s} [N - N_j(\sigma_j)] f_j \\ &-\frac{\beta}{N}\,\sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} [\bar{p}_j - \bar{p}_j(\sigma_j)] f_j\, N \\ &-\frac{\gamma}{N}\,\sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, N\, \bar{p}(\sigma) \\ &+\frac{\delta}{N}\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, N_j(\sigma_j) \\ &-\frac{\delta}{N}\,\sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, N_j\, \bar{p}(\sigma_j) = 0 \end{aligned}\right\} \tag{195}$$

for $i = 1,2,\ldots,s$. It remains to show that a solution actually exists, and to examine the properties of the solution.

4.5.1.3.2 Existence of Solutions to the Cutoff Point Equations - In order to get some insight into the situation, set $\gamma = \delta = 0$; i.e., assume that the function V depends only upon classes {I} and {II}. In this case, Equation (195) is simplified to:

$$\left.\begin{aligned} &-\frac{\alpha}{N}\sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} f_j\, N\,[\bar{p}_j - \bar{p}_j(\sigma_j)] \\ &+\frac{\beta}{N}\, n_i(\sigma_i) \sum_{j=1}^{s} [N - N_j(\sigma_j)] f_j \\ &-\frac{\beta}{N}\sigma_i\, n_i(\sigma_i) \sum_{j=1}^{s} [\bar{p}_j - \bar{p}_j(\sigma_j)] f_j\, N = 0 \end{aligned}\right\} \qquad (196)$$

for $i = 1,2,\ldots,s$. After rearranging and dividing by the common factor, $n_i(\sigma_i)/N$, Equation (196) becomes:

$$\sigma_i = \frac{\beta \sum_{j=1}^{s} f_j[N - N_j(\sigma_j)]}{(\alpha + \beta)\left[N \sum_{j=1}^{s} f_j\,[\bar{p}_j - \bar{p}_j(\sigma_j)]\right]} \qquad (197)$$

for $i = 1,2,\ldots,s$.

From Equation (197) it follows that if a solution exists at all, then it is the same for all i, since the right side of this equation does not depend on i. If we let:

$$\begin{aligned} h(\sigma) = &-\alpha\,\sigma\, N \sum_j f_j[\bar{p}_j - \bar{p}_j(\sigma)] \\ &+\beta \sum_j f_j[N - N_j(\sigma)] \\ &-\beta\,\sigma\, N \sum_j f_j[\bar{p}_j - \bar{p}_j(\sigma)] \end{aligned} \qquad (198)$$

then we can rewrite Equation (196) as;

$$\frac{n_i(\sigma)}{N} \cdot h(\sigma) = 0 \qquad (199)$$

We need to show that there exists a σ such that $0 < \sigma < 1$ and $h(\sigma) = 0$. Given this σ, then $\sigma_1 = \sigma_2 = \ldots = \sigma_s = \sigma$ will be a non-trivial solution

of Equation (196). We demonstrate this fact by showing the following:

$$h(0) > 0 \tag{200A}$$

$$h(1) = 0 \tag{200B}$$

$$h'(1) > 0 \tag{200C}$$

It is sufficient to show Equation (200A), since from Equations (200B) and (200C), $h(\sigma) < 0$ for $\sigma = 1 - \epsilon$, where ϵ is positive and sufficiently small. The result then follows from the Intermediate Value Theorem.

From Equation (198) we have:

$$h(0) = \beta \sum_j f_j[N - N_j(0)] = \beta N$$

which demonstrates Equation (200A). Also, since $\bar{p}_j = \bar{p}_j(1)$ and $N = N_j(1)$, clearly $h(1) = 0$; so Equation (200B) is true. Finally,

$$h'(\sigma) = -\alpha N \sum_j f_j[\bar{p}_j - \bar{p}_j(\sigma) - \frac{\sigma^2 n_j(\sigma)}{N}]$$

$$+ \beta \sum_j f_j[-n_j(\sigma) - N\bar{p}_j + N\bar{p}_j(\sigma) + \sigma^2 n_j(\sigma)]$$

so;

$$h'(1) = \alpha \sum_j f_j \, n_j(1) \tag{201}$$

Since $n_j(\sigma)$ has been assumed to be strictly positive in the unit interval, it follows from Equation (201) that $h'(1) > 0$, so that Equation (200C) holds. Hence a solution to Equation (196) does in fact exist. It can similarly be shown that a solution to Equation (195) exists, provided that δ is not too large. The details will not be given here.

4.5.1.3.3 Further Analysis of the Cutoff Point Equations for $\gamma = \delta = 0$ - We now let:

$$\left.\begin{aligned} g_N(\sigma) &= \sum_{j=1}^{s} f_j[N - N_j(\sigma)] \\ g_{\bar{p}}(\sigma) &= N \sum_{j=1}^{s} f_j[\bar{p}_j - \bar{p}_j(\sigma)] \end{aligned}\right\} \qquad (202)$$

Then Equation (197) becomes:

$$\sigma = \frac{\beta\, g_N(\sigma)}{(\alpha + \beta)\, g_{\bar{p}}(\sigma)} \qquad (203)$$

and this equation can be solved for σ, as we have shown.

Since $N(\sigma)$ is a monotonically increasing function of σ, it is now possible to interpret the value of σ established in Equation (203). It is apparent that $g_N(\sigma)$ represents the average or expected number of retrieved documents. On the other hand, each term of $g_{\bar{p}}(\sigma)$ represents a product of the average probability of retrieved documents times the size of the descriptor group normalized by the frequency of usage of this descriptor. Thus the $g_{\bar{p}}(\sigma)$ function expresses the average number of retrieved documents properly belonging to the average descriptor weighed by its frequency of occurrence. It is thus seen that the optimum σ, expressed by Equation (203), is a function of the constants α and β, which express the relative importance attached to the correctly and incorrectly retrieved documents; the optimum σ is also a function of two averages--namely, $g_N(\sigma)$ and $g_{\bar{p}}(\sigma)$.

It is evident that the higher the value of β--that is, the importance attached to incorrectly retrieved documents--the higher will be the value of σ. And as σ increases, fewer documents will be retrieved.

On the other hand, the higher the value of α--that is, the importance attached to the correctly retrieved documents--the lower will be the value of σ. For lower values of σ more documents will be retrieved. The function decreases with the increment of value of σ, and so does $g_{\bar{p}}(\sigma)$. When $\sigma = 0$:

$$\left.\begin{aligned} g_N(0) &= N \sum_j f_j = N \\ g_{\bar{p}}(0) &= N \sum_j f_j \bar{p}_j \end{aligned}\right\} \qquad (204)$$

and when $\sigma = 1$;

$$g_N(1) = g_{\bar{p}}(1) = 0 \qquad (205)$$

Thus at $\sigma = 0$:

$$\frac{\beta\, g_N(0)}{(\alpha + \beta)\, g_{\bar{p}}(0)} = \frac{\beta}{(\alpha + \beta) \sum_j f_j \bar{p}_j} \qquad (206)$$

To evaluate the expression for $\sigma = 1$, L'Hopital's rule must be used because of the indeterminacy of 0/0:

$$\left.\begin{aligned} &\frac{g_N(\sigma)}{g_{\bar{p}}(\sigma)} \to \frac{g_N'(1)}{g_{\bar{p}}'(1)} \quad \text{as } \sigma \to 1 \\ &g_N'(\sigma) = -\sum_j f_j\, n_j(\sigma) \\ &g_{\bar{p}}'(\sigma) = -\sigma \sum_j f_j\, n_j(\sigma) \end{aligned}\right\} \qquad (207)$$

Thus at $\sigma = 1$:

$$\frac{\beta\, g_N(\sigma)}{(\alpha + \beta)\, g_{\bar{p}}(\sigma)} = \frac{\beta}{\alpha + \beta} \neq 1 \qquad (208)$$

From Equations (206) and (208) it follows that the optimum σ never lies at the extrema of the unit interval.

For simple queries, it follows from Equation (165) that for $\gamma = \delta = 0$, the cutoff point is the same for all categories and is given by:

$$\sigma = \frac{\beta}{\alpha + \beta} \tag{209}$$

For joint retrievals, we have:

$$\sigma = \frac{\beta}{\alpha + \beta} \frac{g_N(\sigma)}{g_p(\sigma)} \tag{210}$$

Since

$$\frac{g_N(\sigma)}{g_p(\sigma)} = \frac{\sum_j f_j \int_\sigma^1 n_j(p)dp}{\sum_j f_j \int_\sigma^1 pn_j(p)dp} > \frac{\sum_j f_j \int_\sigma^1 pn_j(p)dp}{\sum_j f_j \int_\sigma^1 pn_j(p)dp} = 1 \tag{211}$$

we see that:

(a) The cutoff point for joint retrieval on two categories is always greater than the cutoff point for a single category;

(b) The cutoff point for joint retrieval does depend on the probability distribution of documents within the categories.

4.5.1.4 Possible Generalizations - Generalizations to the method of retrieval described here may proceed in either of two directions. The first direction is to extend the method to handle Boolean combinations of descriptors other than the conjunction of two descriptors; the second generalization is to consider the more realistic situation where the probability distributions of documents within different categories are not independent.

The extension of probabilistic retrieval to the more general Boolean functions appears to be a laborious but straightforward mathematical task. It has not appeared worthwhile actually to carry out this extension. However, on the basis of the results already presented it would seem reasonable to expect that the cutoff point for a more complicated retrieval would depend on the form of the retrieval and on the ensemble of distributions, but not on the particular descriptors involved.

A considerable amount of effort was expended in attempting to analyze the situation for the case of dependent categories. Unfortunately, it appears that this problem is insoluble. The remainder of this section will discuss the reasons for this conclusion.

The case of dependent categories is a generalization of the case of independent categories. One theoretically possible but impractical solution would be to compute the joint distributions $N_{ij}(p_i, p_j)$ for each (i, j) pair by actually counting the appropriate numbers of documents. If values of p_i and p_j are computed in increments of δ, then this would require keeping $\frac{s^2 - s}{\delta^2}$ separate statistics: $(\frac{1}{\delta})^2$ for each category and $(s^2 - s)$ times that number for all possible pairs of s distinct categories. Similar statistics would be required for $p_{ij}(p_i, p_j)$. Therefore, one would hope to find a single measure of relatedness between categories and to use this measure in two different relationships: one that would express $p_{ij}(p_i, p_j)$ in terms of p_i and p_j in a convenient functional form; and the other that would express $N_{ij}(p_i, p_j)$ in terms of $N_i(p_i)$

and $N_j(p_j)$. The assumption of independence led to the relations in Equations (168) and (169), which accomplished this aim.

It is possible, and perhaps even reasonable, to assume that $P_{ij}(p_i, p_j) = p_i p_j$ and to incorporate the effects of dependence between categories into the distribution function N_{ij} alone. The rationale for this procedure is as follows: suppose that the distribution statistics are based on the results of having documents assigned to categories by a panel of users. If two categories are highly dependent--for example, almost synonymous--then one would expect that those documents that have a high probability of belonging to one category also have a high probability of belonging to the other. A similar rationale holds for documents that have a low probability of belonging to one or the other of these categories. This effect would manifest itself as a skewness in $N_{ij}(p_i, p_j)$. However, consider a single document that had been assigned to category i by p_i of the users and to category j by p_j of the users. Even if the categories are closely related in the sense that documents belonging to one are likely to belong to the other, the judgments of a particular panel member with respect to the two categories may well be independent. For instance, suppose that two categories are closely related, and a particular document is assigned to each of them with a probability of 90 percent. It need not be true that the 90 percent of the users who assigned the document to the first category are the same 90 percent as those who assigned it to the second category. It could reasonably be assumed that the two groups are in fact selected independently, so that only 81 percent of the users assign the document to both

categories. If we make this assumption, then we can take $p_{ij}(p_i, p_j) = p_i p_j$. However, the problem of N_{ij} remains.

The type of relationship we are looking for should be of the form:

$$N_{ij}(p_i, p_j) = F[p_i, p_j, N_i(p_i), N_j(p_j), k_{ij}] \tag{212}$$

where k_{ij} is a parameter that measures the relatedness of category i to category j. If we do not assume that $k_{ij} = k_{ji}$, then k_{ij} would measure the tendency of items in j to belong to i also, and conversely for k_{ji}. That this situation can in fact arise is illustrated by the case of nested categories; every document that belongs to the subcategory also belongs to the larger category, but not conversely.

Let us consider the constraints on the expression for N_{ij} as given in Equation (212). Since N_{ij} represents a distribution function, we must have:

$$n_{ij}(p_i, p_j, k_{ij}) \geq 0 \tag{213}$$

where $n_{ij}(p_i, p_j, k_{ij})$ is, as before, the joint probability density defined by Equation (167), with k_{ij} as a parameter. Since every document belongs to category j with <u>some</u> probability between 0 and 1, we have:

$$\frac{1}{N}\int_0^1 n_{ij}(p_i, p_j, k_{ij})dp_i = n_j(p_j) \tag{214}$$

and similarly,

$$\frac{1}{N}\int_0^1 n_{ij}(p_i, p_j, k_{ij})dp_j = n_i(p_i) \tag{215}$$

If we define $k_{ij} = 0$ to be the case of independent categories, then we must have:

$$n_{ij}(p_i, p_j, 0) = \frac{n_i(p_i)\, n_j(p_j)}{N} \tag{216}$$

Finally, if we define $k_{ij} = 1$ to indicate synonymous categories, then we will want

$$n_{ij}(p_i, p_j, k_{ij}) \to \infty \text{ as } k_{ij} \to 1 \tag{217}$$

The reasoning behind this equation is that for synonymous categories the density function n_{ij} will be zero for $p_i \neq p_j$, since every document will be assigned to the two categories with the same probability. Since n_{ij} will be non-zero, only along the line $p_i = p_j$ and since this line has zero area, the density function on the line must be infinite if the integral of the density is to be non-zero. This situation is, however, approached only in the limit: hence we have Equation (217).

A careful examination of the forms that n_{ij} might take has led to the conclusion that there is no reasonably simple n_{ij} that can be found; and if n_{ij} is too complicated, it will be impossible to carry out the remainder of the analysis, which was difficult enough even in the independent case. The two most likely forms were

$$n_{ij}(p_i, p_j, k_{ij}) = n_i(p_i)\, n_j(p_j)\, f(p_i, p_j, k_{ij}) \tag{218}$$

and

$$n_{ij}(p_i, p_j, k_{ij}) = \frac{n_i(p_i)\, n_j(p_j)}{N} + k_{ij}\, f(p_i, p_j, k_{ij}) \tag{219}$$

We will consider Equation (218) first.

For Equation (218) the constraints, Equations (213) through (217), yield:

$$f(p_i, p_j, k_{ij}) \geq 0 \tag{220A}$$

$$\frac{1}{N}\int_0^1 n_i(p_i)\, n_j(p_j)\, f(p_i, p_j, k_{ij})dp_i = n_j(p_j) \tag{220B}$$

$$\frac{1}{N}\int_0^1 n_i(p_i)\, n_j(p_j)\, f(p_i, p_j, k_{ij})dp_j = n_i(p_i) \tag{220C}$$

$$f(p_i, p_j, 0) = 1 \tag{220D}$$

$$f(p_i, p_j, k_{ij}) \to \infty \text{ as } k_{ij} \to 1 \tag{220E}$$

It is apparent that f should be symmetric in p_i and p_j. Furthermore, from Equation (220B) we see that the quantity,

$$\int_0^1 n_i(p_i)\, f(p_i, p_j, k_{ij})dp_i \tag{221}$$

must be invariant for all possible $n_i(p_i)$. Since $n_i(p_i)$ is an arbitrary positive function of p_i, we must have:

$$f(p_i, p_j, k_{ij}) = \frac{f^*(p_i, p_j, k_{ij})}{n_i(p_i)} \tag{222}$$

to cancel out the effect of varying $n_i(p_i)$. By symmetry, however, we must also have:

$$f(p_i, p_j, k_{ij}) = \frac{f^*(p_i, p_j, k_{ij})}{n_j(p_j)} \tag{223}$$

Since f* must be the same in Equations (222) and (223), we have a contradiction and Equation (218) must be discarded.

If we try Equation (219), the constraint equations yield

$$k_{ij}\, f(p_i, p_j, k_{ij}) \leq \frac{n_i(p_i)\, n_j(p_j)}{N} \tag{224A}$$

$$\int_0^1 f(p_i, p_j, k_{ij})\, dp_i = 0 \tag{224B}$$

$$\int_0^1 f(p_i, p_j, k_{ij})\, dp_j = 0 \tag{224C}$$

$$f(p_i, p_j, k_{ij}) \to \infty \text{ as } k_{ij} \to 1 \tag{224D}$$

We have a similar difficulty. If f contains a multiplicative factor of $n_i(p_i)\, n_j(p_j)$, then we can remove $n_i(p_i)\, n_j(p_j)$ from n_{ij} and use the same argument as the one raised against Equation (218). Yet without this factor there does not appear to be any way to satisfy Equation (224B) in view of the arbitrary nature of $n_i(p_i)$ and $n_j(p_j)$.

What we have shown is that there does not appear to be any possibility of developing an analysis of probabilistic retrieval that will account for the relatedness of categories used in a query. However, for most retrieval requests encountered in practice it would be reasonable to expect that different categories mentioned in the request would be at worst slightly related. Furthermore, a well-chosen set of categories will probably have little correlation among its members since the existence of correlation degrades the utility of the categories. In summary, then, the use of the independence assumption should not unduly distort the results of probabilistic retrieval.

4.5.1.5 Conclusions - It is now possible to outline the general features of a probabilistic retrieval system. To each category

there will correspond a collection of classes of documents instead of a unique class of documents. Each class will be determined by a different cutoff point σ. For each document, there will be two types of cutoff points, disjunctive and conjunctive. Within each of these categories an individual σ will have its value determined in accordance with the type of joint retrieval it is scheduled to participate in. Thus there will be one cutoff point for the conjunction of two descriptors, another for conjunction of three, etc. The same principle holds for the cutoff points for disjunctive retrievals. Any incoming request will be transformed into convenient canonical form; for example, a disjunction of conjunctions. The appropriate cutoff points will then be selected and retrieval effected.

In order to calculate the cutoff points, certain parameters are required. These parameters can be obtained by requiring the system to perform bookkeeping operations that will supply the required data. Essentially, the kind of statistical data necessary for the calculation of the cutoff points is:

(a) $n_i(p)$ = the density of documents pertaining to a given descriptor for a given probability interval.

(b) $\bar{p}_i(\sigma)$ = the average probability value of a document belonging to the descriptor i as a function of a cutoff point.

(c) $N_i(\sigma)$ = the total number of documents belonging to the descriptor i as a function of σ

The most fundamental of the three types of data is (a), since (b) and (c) can be calculated from it.

4.5.2 The Problem of Redundancy

4.5.2.1 Introduction - Redundancy in the information retrieval

processes occurs whenever the retrieved data is duplicated. To avoid redundancy is important, not only for the rather obvious economic reason, but also for operational and logical reasons. Theoretical considerations pertaining to the nature of measures for removing redundancy will be best understood within the context of a more detailed discussion of the undesirability of duplication from these three points of view.

4.5.2.2 Economic Point of View - For some types of information retrieval systems the cost of retrieval may become prohibitively high, especially if all the data pertaining to the request profile is retrieved.

The use value of the information contained in the retrieved data may be drastically reduced by the existence of redundant material. Effectively the user of the data is swamped by repetitious information.

4.5.2.3 Operational Point of View - Many information retrieval systems enter into larger systems as component units. The retrieved data may form an input to other processes such as control, command and control, or real-time monitoring. The occurrence of redundant material may not only reduce the efficiency of the functioning of the system, but also affect the outcome of the processes to which the retrieved data forms an input. For example, imagine a system that is required to perform some statistical tabulations on the incidence of car accidents among various population groups. Furthermore, assume that the reports on automobile accidents are incoming from diverse sources so that some accidents may be reported more than once. Under such conditions it will be necessary, in order to obtain valid results, to introduce some filtering stage that

will prevent or eliminate duplication. Estimates of the reliability of the results obtained will in general depend upon the effectiveness of the filtering stage. The removal of data redundancy is thus vital to the satisfactory performance of the system as a whole.

4.5.2.4 Logical Point of View - In the process of decision making the origin of the data may be as relevant to the decision as its content. It is even conceivable that the existence of large amount of redundancy in the collected data may be one of the important factors influencing the nature of the decision. In other words, the decision process may be dependent on the manner in which the data is presented. As an example, imagine a system whose task it is to solve transportation-routing problems. The kind of solution employed may well depend upon the complexity of a particular problem. If the particular transportation network contains many nodes, the system will use one type of an algorithm; if it contains few nodes, then another.

Determining the nature of the problem may depend upon sampling of data; thus inaccuracies will arise if the data contains a large amount of redundancy. Such a situation is particularly prone to arise if the system schedules its own operations and batches many problems together.

4.5.2.5 Tentative Measures of Redundancy - Considering several ways in which the concept of redundancy is implicated in the information retrieval processes, a basic dichotomy becomes apparent:

(a) Some of the redundancy problems require the exact scrutiny of the individual data items. If data items are conventionally

thought of as documents, then a sort of redundancy map could be obtained by indicating the relationship with respect to redundancy of each document to every other document in the collection. The simplest kind of relation between documents with respect to redundancy is that of inclusion; that is, one document may express everything that another document expresses with respect to a given topic. Another possible relation, although a less simple one, is that of overlap. A document may partially express the content of another document with respect to a given topic with some numerical measure of the partial covering.

(b) It may be possible or desirable to handle the problem of reducing redundancy on an aggregate level. The distinguishing feature of this approach is the statistical handling of information contained in the documents. It is important to remember that, since the primary concern is redundancy, the basic measure of information must be relative rather than absolute. That is, such a measure when applied to a document should be able to determine the expected number of documents rendered superfluous by the document in question; alternatively, the measure should indicate how many documents render a given document superfluous.

Usually a document will cover a number of topics. In general, it must be expected that the redundancy measure will not be

evenly distributed among all the topics that a given document deals with. Thus with respect to one topic a document may be highly unique, whereas with respect to another, highly redundant. Whether or not it is advisable to average the redundancy measure over all topics or handle them separately is a question that may be decided only after a more detailed and rigorous study. It is also possible that this question admits no unique answer, since information retrieval systems are highly differentiated with respect to their functional characteristics.

It would be incorrect to assume that this dichotomy represents two alternative approaches. It is quite unrealistic to expect that an exhaustive redundancy map comprising the detailed breakdown of all relations among all documents individually is feasible. Practically, some sort of statistical approach is necessary. It is necessary, however, to demand that any statistical averages employed to reduce redundancy capture the true statistical properties of a system based upon the requirements for a redundancy map.

4.5.2.6 Conclusion - It is important to avoid redundancy for operational, logical, and economic reasons. Two tentative examples of redundancy measures are:

(a) Each document is characterized by a set of numbers expressing the percentage of documents containing more, or less, information concerning a given topic.

(b) Each document is characterized by a set of numbers expressing the additional contribution that the document would make to the given topic, assuming the average number of documents already retrieved.

4.5.3 Adaptation to User Requirements

4.5.3.1 User Orientation - The users of an information system are often conceived as a univocal mass that knows precisely what type of information it wants from the system. The problem of system design is then reduced to the simple expedient of devising means of access to the general body of stored information for this class of users.

In fact, however, the users are neither univocal nor certain; if they were, the problem of information retrieval would be vastly simplified. Any intermediary for gaining access to stored information would be superfluous, since the users by definition have a priori knowledge about the nature of the information they seek. The difficulty is that users approach any information system--even a library card catalogue--because their questions are vague and ill formed. Furthermore, each user wishes to fulfill a different need.

In confronting a new system, any user is wary at first; the mechanism of the system stands as a barrier (and possibly a threat) between his questions and whatever answers may be available. The first criterion for gaining the user's confidence, then, is simplicity; the mechanics of the system should be readily grasped after a few moments of study. The second criterion is that the user quickly gain confidence that the system can indeed produce reasonable responses to reasonably well formed queries.

This second factor poses the greatest difficulty. If a user has confidence in the system, he is willing to enter a tacit dialogue. A

simple question, however ill formed, produces sufficient information to lead to another, more cogent question. The dialogue continues from question to answer to question until the user eventually frames precisely the right question to gain access to the information he originally sought. This process with the familiar card catalogue is heuristic; the same process should occur with an automated system, but the interposition of a machine may easily restrain the facility of the dialogue.

An information system deals with the functional elements of information in such a way that a sequence of operations upon these elements or upon concatenations of these elements produces the requested information. What is desired is information explicitly or implicitly contained in the data received by the system. Thus, ultimately, logical implications, generalizations, correlations, and even logical appraisals of the original data (credulity measures and ordering relations) may be the results of these operations.

The requirements for performing operations upon the information parallel, at least in part, those for storing information. These operations should be defined so that information can be recombined into forms that are not explicitly formed in the original information. Such processing operations should be specified in relation to the storage operations. The retrieval processes may then gather relevant material from the stored data so that it may be operated upon and used to answer questions. Some of these operations are based upon statistical analyses of the data. Other operations are functions performed upon the question in order to

improve the formulation of a query. In this way the inherent difficulties in establishing a dialogue between the user and the system may be reduced, if not entirely eliminated.

Additional operations on information may be necessary. The system may be expected to derive logical relationships existing among data contained in its memory. In addition to logical inferences (deductions), the system may be expected to perform inferential processes (inductions). Such inductive inferences differ from deductive inferences in two important respects: the relationships derived are not necessarily valid; and not all the rules of inductive reasoning are explicitly formalized.

Implied relationship is a generic term for all relationships not explicitly contained in a system. Such relationships are derived by means of inferential processes; that is, inductions and statistical correlations. The term implied relationship includes relationships derived on the basis of inductive, or non-rigorous, inferential processes. Such relationships are by their nature not as well defined as relationships obtained deductively. The system must, therefore, be designed with the capacity to estimate the degree of credibility of such derived relations and the degree of relevance to other information. On the basis of such estimates the system may accept or reject the derived conclusions.

Since the set of implied relationships is not well defined, such a system will arbitrarily limit the range of derivable relationships.

cannot be expected that the system will attempt to derive all the implied relationships that lie within a specified range without being requested to do so, either directly or indirectly, in terms of a question. On the other hand, some of the implied relationships might be so important to the functioning of the system that they ought to be derived even without any initiating query. An information system would, therefore, be more powerful if it possessed a set of decision algorithms for determining at which point it must stop its inferential activities.

It is necessary to state the criteria employed to select the relationships the system will derive. While the set of explicit relationships stored in the memory of a system may be well defined, the corresponding set of implicit relationships may not be. The derived implicit relationships depend not only upon the set of explicit relationships, but also the nature of the formal or informal inferential methods as well as upon other factors--for example, the richness of association--less amenable to precise description. Because of these factors it may be questioned whether the notion of the set of all implicit relationships derivable from the information is meaningful. From a practical viewpoint, some limitations upon the range of implicit relationships must be imposed.

The criteria for the limitations that are to be imposed upon a system's ability to derive implicit relationships ought to include:

(a) Only implicit relationships possessing potential utility to the users of the system should be derived.

(b) The system should not try to derive implicit relationships of so complex a nature that the attempt is likely to end in failure.

(c) The limitations should be flexible enough to leave room for learning.

The system may be able to increase the range of derivable implicit relationships as it obtains more input information or elicits more information about a question from the user; again the importance of a dialogue is apparent. The criterion for the selection of derivable relationships, which includes all three of these characteristics is: the system is only concerned with those implied relationships that can be derived in response to a definite procedure specified by the user. This principle may be considered as the organizing principle of the system.

There are several points that will clarify the meaning of this principle. In addition, the adoption of this principle has certain implications for the learning processes that will take place in an information system. The phrase, ".....in response to a definite procedure specified by the user," does not mean that the user is obliged to supply the directives that could be directly translated into programs--that is, a sequence of action resulting in an output consisting of the appropriate implicit relationships. Neither does it mean that such a specification need be supplied to the system initially.

The principle simply states that the user knows <u>how to go about</u> solving the problem embodied in a query addressed to the system; he knows how to solve the problem in terms of human mental processes. Moreover, the principle does not require the user to state the procedure formally. The concept of knowing how to go about solving problems implies

no more than that the user know enough about his own procedures to answer questions about his approach to the problem.

4.5.3.2 A Concept of Questioning - In order to optimize the retrieval ability of a system, the user should question the system within the framework of a theory of questioning. The development of a concept of questioning has occasioned considerable scientific interest within the last decade. In part, such an interest is related to problems of retrieving information, for even a cursory examination of questioning indicates that it plays an important role in the retrieval of information. Every pragmatically important question has a correct answer associated with it. Such a correct answer is a statement that provides a person with information--knowledge that he did not possess at the time that he asked the question. The statement may be true or false and still fulfill this criterion. Given a framework of this kind, the concept of questions requires a development along two parallel lines: the semeiology and the methodology of questions.

The semeiology of questions pertains to the form and nature of queries. Questions are a type of linguistic structure. Composed as they are of signs--letters and words--questions have meaning. Such meaning may be even more complex than the meaning of declarative statements, since questions may also be logical functions of such meanings.

There are two possible ways to investigate the meaning of a question. A question may be correlated with a class of statements, any one of which is a correct answer to the question. In this sense, the

question defines the scope of possible answers; it is neither responsive nor meaningful to answer the question, "What time is it now?" with the statement, "The Parthenon is located in Athens, Greece." On the other hand, there are questions that do not define the kind of statement that is a correct answer. Consider the question, "How many horns does a unicorn have?" "There are no such things as unicorns," is as correct an answer as, "A unicorn has one horn." In other words, a question may pragmatically admit unclarity about the boundaries of a subject. Only procedurally correct questions request information within a framework of concepts and statements accepted as true by both the questioner and the informer.

The realization that a question is related to a given state of knowledge requires further exploration. It is clear that a question is meaningful only if the questioner refers to a set of interrelated concepts either explicitly or implicitly. When a questioner asks, "What time is it?" he knows that the answer is a set of numbers that have a certain order--for example, "later than." But it remains a problem whether some concept must be assumed explicitly or implicitly for any question to be meaningful. It may be that in order for a question to be meaningful, some restriction of its scope must be present.

The meaning of a complex term is not only determined by its relationship to non-linguistic factors, but also by its logical relationship to other terms. The meaning of questions is in part specified by their logical or syntactical relationship to other questions. What is required, then, is a formal logic of questions. Such a logic would rigorously formulate:

(a) The syntax of a formal language into which questions in natural language are translatable.

(b) The rules of deduction for such a language.

(c) The theorems concerning logical relations formulatable in such a system.

It seems that the language in which the logic is formulated may be constructed out of declarative sentences by the use of an undefined logical operator [28, 29]. Logical functions analogous to deduction can then be defined. In any system the correlation between questions and permissible answers must be formally modeled by mapping a question on a set of sentences. Semantically, at least, the range of variables should also be specified for answers that are specifiable for standard types of questions.

In addition to logical deducibility that would be studied by such a calculus, there is another dimension of logical analysis. This area pertains to the relative complexity of questions. It may be, for example, that in a certain context a Why question is translatable into a finite set of How questions. In this context, Why questions are more complex than How questions. But there are many types of questions. In addition, there are disjunctive and conjunctive questions as well as general and particular questions. This brief discussion indicates that a logical theory is necessary to consider problems of this kind systematically.

Once a formal analysis of questions has been developed, it will provide insight into the methodology of questions. If the questions that imply other questions are known or are reducible to other questions, then

it is easier to develop strategies for sequencing questions so as to obtain maximum information for a minimum set of questions. It is advantageous for any information processing system to allow this condition to be fulfilled.

Besides purely logical and formal considerations, there is a problem of methodology--the strategy or heuristic of interrogation. This problem centers on the problem of efficiency and purposefulness in interrogation. The main objective is to relate the formal characteristics of questioning to intentions that the questioner may have. From the nature of the problem it is evident that, unlike the inquiry into formal properties of questions, this discussion is mainly concerned with sequences of questions.

There are two types of goals that can be associated with the procedure of interrogation. The first is the desire to obtain more factual information. A simple example of this type of interrogation is: "How many people reside in Rome?" The second goal is to obtain a better understanding of a certain area of inquiry. This objective may be related to the interrogator's perception of gaps in the flow of information or to his lack of understanding of the information. Efficient and intelligent questioning depends upon the precision with which the interrogator can pinpoint the kind of information he wants as well as upon his ability to formulate the appropriate sequences of questions.

The objective of this concept of questioning is to establish procedures for an interrogator to discern the intention of his interogations.

The concept is not psychologically oriented. The problem is not to correlate subjective states of mind with the objective elements of the questioning process. The concept seeks to associate the properties of sets of information with the rational formulation of interrogative intentions. These intentions are then fulfilled if the sequence of questions is appropriate for its purpose.

The ordering and the retrieval of information depend upon initially specified rules for information handling. These rules may not be the only rules for data handling necessary for the proper and efficient operation of an information system. The system must be able to acquire new rules and modify old rules as it continues to process information. The acquisition of rules may be divided into two categories.

One category includes processes based upon success-failure criteria. In processes of this kind an information system attempts to improve its performance without an interchange of complex questions with the user. If the criteria for adequate performance are not satisfied, the system seeks to improve its performance solely on the basis of its store of data and its own experience.

The second category includes processes based upon a system's attempt to elicit information pertinent to the formation of adequate processing rules from the user. Such processes are more complex than those in the first category. In addition to being able to use its own experience, the system is able to question human beings and to use human guidance. In this way the essential dialogue between a user and a system

may lead to the necessary well formed questions that will elicit the required information for the user.

The implication of this discussion is that the user-system dialogue will necessarily span a range of questions over a period of time, however short the time. But this implied constraint need not follow. A simple question may be simply answered; yet in a simple question the necessary clues to the relevant information are almost apparent. Consider a slightly more difficult instance. If the system contains N categories of information, then N! question combinations are possible. The information may also be stored so that a relation (A,B,C...) holds. The query may be framed (C,B,A). A simple response would state: "If your request could also be (A,B,C), then your answer is..." This approach appears too easy, but it is not uncommon. And if these functions were automated, the demon of interrogation could be greatly simplified.

4.5.3.3 The Linguistic Problem - Given an appropriate formal representation of linguistic input, there still exist problems of equivocation in word use that would disrupt the functioning of an inferential processor. Consider the following true assertions:

(a) The number 2 is rational.

(b) Socrates is rational.

(c) Anything rational can reason.

These sample sentences have little inherent interest. Their purpose, however, is paradigmatic rather than practical.

The word rational in sentence (a) is being used in a different sense from the same word in sentences (b) and (c). Unfortunately, this difference is more than a mere linguistic difficulty. It is conceivable that at a purely linguistic level the equivocation is irrelevant. An example is translation to another language that has the same ambiguity in the use of the word rational. In the context of accurate inference, however, this kind of apparently insignificant linguistic difficulty can lead to serious logical problems. Thus, sentences (a) and (c) seem to lead to the conclusion that the number 2 can reason. This falsehood is directly attributable to the fallacy of the four-term syllogism produced by the equivocation in the use of the word rational.*

For any deductive inference processor an awareness of such equivocation is essential. Other research [24, 74] has developed a sense value theory that may be able to discover such distinctions in sense mechanically. For the purpose of inferential processing it would be desirable to establish whether sense value theory may be applied to

---

*It is possible to argue that the difficulty lies not in the equivocation in the use of "rational" but in the falsehood of sentence (c), given such equivocation. Perhaps the example is ill chosen, but we would ordinarily allow the use of generalizations such as (c) provided that the sense of the words involved is clear. Thus, that anything that is heavy (or light) has weight seems beyond question. The reason that there is no question is that it is clear that the terms heavy and light are being used in the sense of weight. We are not led to reject the generalization because colors, for example, may be said to be heavy (awkward, but possible) or light. We say rather that colors (as opposed to pigments) are not the sorts of things that have weight and that the sense in which "heavy" or "light" may be used to describe them is quite different from the sense in which these words describe relative weight--even though there is a metaphoric value in the analogy between weight and color demansions.

the mechanical discovery of sense equivocations in practice. Appendix D presents a discussion of the fundamental concepts of sense value theory and an account of possible approaches to the application of sense value theory to inferential processing.

4.5.3.4 The Logical Problem - This section considers the development of inferential capabilities, given a mass of initially linguistic data reduced to an appropriate unequivocal form suitable for further machine processing. Two kinds of inferential problems can be distinguished at this point:

(a) The relatively straightforward problem of checking whether a conclusion deductively follows from the information in the file.

(b) The more difficult problem of assessing the validity of a generalization inductively.

While the problem of inductive inference will be left for later development and will not receive much further consideration in this section, it should be noted that most linguistic information files are probably far too complex for simple deductive processing schemes to be effective in regard to the answering of many kinds of questions.

Among the difficulties we may expect to encounter in implementing automatic deductive processing, two are especially salient:

(a) A great deal of information that people use in developing valid inferences about practical matters is never explicitly stated in a textual account of the facts concerning some matter of interest.

(b) Textual sources may contain contradictory assertions that render successful deductive processing impossible because any conclusion may follow.

The second difficulty may be regarded as an instance of the kind of problem that only inductive systems that use probabilistic techniques for weighting and significance of observations or assertions can overcome. For the purpose of this discussion the second kind of difficulty is regarded as one that automatic deductive systems should be able to detect while leaving correction as a human function. The former difficulty, however, will be a serious limitation on deductive systems. It seems that it should be possible to work on this problem within a purely deductive framework. That is, the problem does not inherently require inductive techniques such as probabilistic weighting or generalization.

An example may help clarify the last conclusion. The human being has no difficulty concluding from the fact that X was in Chicago all day on a given day, that he was not in New York or Los Angeles or any other <u>different place</u> on the given occasion. He is further generally able to conclude that the individual in question <u>was</u> in Illinois rather than that he was not in Illinois. Our hypothetical cogitator is able to perform these feats of inference in essentially deductive fashion by appending to the assertion about X being in Chicago, appropriate assertions about naming conventions and spatio-temporal relations. Of course, the ordinary person is able to derive these conclusions <u>automatically</u> without explicitly stating the suppressed premises for the syllogisms leading to the appropriate conclusions from the fact that X was in Chicago. We are, however, ultimately assured of the validity of any argument because it can be reduced to a deduction from premises about which we do not entertain any doubts. The models of the real word that the human being possesses allow

him to draw accurate conclusions because the models are accurate and because the automatic conclusion-generating mechanisms he possesses are in accord with explicit deductive reasoning. To the extent that these conditions are not met, the human being's inference is bound to result in error--or, at best, be only fortuitously correct despite the invalidity of the underlying argument or the falsity of the implicit premises.

The task for information systems technology is not to simulate the inferential machinery the human being uses, but to reproduce its results reliably when they correspond with valid arguments from acceptable premises. To the extent that the simulation of human cognitive processes furthers this end, it should be pursued for wholly technological reasons. There have been several attempts to incorporate limited models of naming conventions or spatial relations into systems of deductive inference for a computer's answering of questions. Some examples of the former are Green's Baseball Program [24] and Lindsay's Sad Sam Program [43]. The former is able to deal with the logical relations implicit in the use of various baseball terms; the latter is directed to the analysis of kinship relations implicit in limited verbal statements about how one person is related to another--for example, that X is a brother of Y automatically tells Sad Sam that X is male and has a common ancestry with Y. Examples of inferential systems for computers that use models of spatial relations include Gelernter's geometry program [21] and Raphael's current research aimed at developing a conversational computer that can answer questions about assertions [66].*

*The last example also models non-spatial relations. Nor is the primary

4.6 REFERENCES

[1] "A Solution to the Information Retrieval Problem," Electrical Engineering (No author named, based on article by Holm, B. E.), August 1962, p 623.

[2] Atherton, P., "Indexing Requirements of Physicists," Proceedings of Conference on the Literature of Nuclear Sciences, Its Management and Use...September 4-13, 1962, Oak Ridge, Tennessee, December 1962, p 215.

[3] Baker, F. B., "Information Retrieval Based on Latent Class Analysis," Journal of the ACM, Vol. 9, No. 4, October 1962, pp 512-521.

[4] Baxendale, P. B., "Machine-Made Index for Technical Literature--An Experiment," IBM Journal of Research and Development, Vol. 2, No. 4, October 1958, pp 354-361.

[5] Birkhoff, G., Lattice Theory, The American Mathematical Society Colloquium Publications, 1948.

[6] Bodewig, E., Matrix Calculus, North-Holland Publishing Co., 1959.

[7] Borko, H., A Research Plan for Evaluating the Effectiveness of Various Indexing Systems, Systems Development Corporation, FN-5649/000/01, (AD 278-624), July 10, 1961.

[8] Borko, H., The Construction of An Empirically Based Mathematically Derived Classification System (AD 267-901), System Development Corporation (Report SP-585), October 1961.

....."The Construction of An Empirically Based Mathematical Derived Classification System," Proceedings of the Western Joint Computer Conference, May 1962.

[9] Borko, H., and Bernick, M. D., Automatic Document Classification, System Development Corporation, (Technical Memorandum TM-771), November 1962.

---

emphasis in Baseball on the analysis of naming conventions. These programs are, however, illustrative of the kinds of modeling processes that are essential in adding sufficient information of a general nature to textual input or explicit premises in order to allow more powerful inferences. An Automatic inference processor may be considered more powerful if it can draw a larger number of valid conclusions than a human being from a set of premises or if it can draw a given conclusion more rapidly or in a fewer steps.

....."Automatic Document Classification," Journal of the ACM, Vol. 10, No. 2, April 1963, pp 151-162.

[10] Bornstein, H., A Paradigm for a Retrieval Effectiveness Experiment, General Electric Company, Information Systems Section, Washington 4, D. C., March 1961.

[11] Brandenberg, W., "Write Titles for Machine Index Information Retrieval Systems," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part I, Chicago, October 1963, p 57.

[12] Connolly, T. F., "Author Participation in Indexing--From Primary Publication to Information Center," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part I, Chicago, October 1963, p 35.

[13] Current Research and Development in Scientific Documentation, National Science Foundation, Office of Science Information Service, November 1962.

[14] Darmstadt, Q., A Formal Development and Application of the Theory of Sense Values and Sense-Value Trees for Natural Languages, Parts I, II, and III, ITT DISD, Paramus, New Jersey, 1962.

[15] Dubinin, M. M., "Exchanging Scientific Information," Bulletin of the Atomic Scientists, Vol. 18, October 1962, p 13.

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

[17] Faddeeva, V. N., Computational Methods of Linear Algebra, Dover Publications, Inc., 1959.

[18] Fairthorne, R. A., "Delegation of Classification," American Documentation, Vol. 9, March 1953.

[19] Farradane, J., "Relational Indexing and New Methods of Concept Organization for Information Retrieval," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 135.

[20] Feller, W., An Introduction to Probability Theory and Its Applications, Vol. I, John Wiley and Sons, Inc., New York, 1957.

[21] Gelernter, N., "Realization of a Geometry Theorem-Proving Machine," Proceedings of the International Conference on Information Processing, Paris, 1959.

[22] Giuliano, Vincent E., Studies for the Design of An English Command and Control Language System, Arthur D. Little, Inc., Cambridge, Massachusetts, June 1962.

[23] Goldman, S., Information Theory, Prentice-Hall, Inc., New York, 1955.

[24] Green, B. F., Wolf, A. K., Chomsky, Carol, and Laughery, K., "Baseball, An Automatic Question-Answerer," Proceedings of the Western Joint Computer Conference, Los Angeles, May 1961.

[25] Hake, D. L., "Improving the Information Flow," Bulletin of the Atomic Scientists, Vol. 18, November 1962, p 21.

[26] Hamming, R., "The Mechanization of Science," Preprints of Summaries of Papers, Association for Computing Machinery, 16th Annual Meeting, September 5, 1961.

[27] Harmon, H. H., Modern Factor Analysis, University of Chicago Press, 1960.

[28] Harrah, D., "The Logic of Questions and Answers," Philosophy of Science.

[29] Harrah, D., "The Logic of Questions," Proceedings of Congress of Philosophy.

[30] Hayes, R. M., Information Storage and Retrieval, UCLA, 1962.

[31] Hayes, R. M., Mathematical Models for Information System Design and A Calculus of Operations, Final Report, Air Force Contract AF 30(602)-2111, Advanced Information Systems Co., 1961.

[32] Hayes, R. M., Report on the Organization of Large Files with Self-Organizing Capabilities, Advanced Information Systems Co., NSF Contract C-162, 1961.

[33] Hestenes, M. R., "Inversion of Matrices by Biorthogonalization and Related Results," Journal of Society for Industrial and Applied Mathematics, March 1958.

[34] Hilf, J., "Matching of Descriptors in a Selective Dissemination System," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part I, Chicago, October 1963, p 65.

[35] Hooper, R. S., "A Facet Analysis System," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 253.

[36] Institute of Radio Engineers, "Abstracts of Current Computer Literature," IRE Transactions, EC-8, Nos. 1, 2, and 3, 1959.

[37] Jacobson, S. N., "Paragraph Analysis Naval Technique for Retrieval of Portions of Documents," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II Chicago, October 1963, p 191.

[38] Jonker, F., The Descriptive Continuum - A Generalized Theory of Indexing, Air Force Office of Scientific Research, June 1957.

[39] Kennedy, R. A., "Writing Informative Titles for Technical Papers--A Guide to Authors," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 133.

[40] Kent, Allen, and Perry, J. W., Technical Notes (series), Center for Documentation and Communication Research, School of Library Science, Western Reserve University.

[41] Klingbiel, P. H., Language Oriented Retrieval Systems, (AD 271-600), February 1962.

[42] Lefkovitz, D., and Prywes, N. S., "Automatic Stratification of Information," Proceedings of the Spring Joint Computer Conference, May 1963.

[43] Lindsay, R. K., "A Program for Parsing Sentences and Making Inferences about Inference Relations," Proceedings of the Western Management Conference on Simulation, (A. Hoggatt, Ed.), 1962.

[44] Luhn, H. P., "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," IBM Journal of Research and Development, Vol. 1, No. 4, October 1957, pp 309-317.

[45] Luhn, H. P., Auto-Encoding of Documents for Information Retrieval Systems, IBM Research Center, Yorktown Heights, New York, 1958.

[46] Luhn, H. P., "The Automatic Creation of Literature Abstracts," IBM Journal of Research and Development, Vol. 2, No. 2, April 1958, pp 159-165.

[47] Maron, M. E., Automatic Indexing: An Experimental Inquiry, (AD 245-175), RAND Corporation. Santa Monica, California, 10 August 1960.

....."Automatic Indexing: An Experimental Inquiry," Journal of the ACM, Vol. 8, No. 3, July 1961, pp 404-417.

[48] Maron, M. E., and Kuhns, J. L., "On Relevance, Probabilistic Indexing and Information Retrieval," Journal of the ACM, Vol. 7, No. 2, July 1960, pp 216-244.

[49] Meyer-Uhtenried, K. H., Lustig, G., "Analysis, Indexing, and Correlation of Information," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 229.

[50] Mooers, C. N., The Use of Symbols in Information Retrieval, RADC-TN-59-133, (AD 213-781), April 1959.

[51] Newbaker, H. R., and Savage, T. R., "Selected Words in Full Title (SWIFT): A New Program for Computer Indexing," American Documentation Institute (ADI), Annual Meeting, Short Papers, Part I, Chicago, October 1963, p 87.

[52] Newell, A., Shaw, J. C., and Simon, H. A., "Report on a General Problem Solving Program," Proceedings of the International Conference on Information Processing, 1959, pp 256-264.

[53] Newell, A., and Simon, H. A., The Simulation of Human Thought (AD 235-801), RAND Corporation, RM-2506, December 28, 1959.

[54] Oswald, V. A., Jr., et al, Automatic Indexing and Abstracting of the Contents of Documents, (RADC-TR-59-208), Rome Air Development Center, Air Research and Development Command, United Stated Air Force, 31 October 1959, pp 5-34, 59-133.

[55] Parzen, E., Modern Probability Theory and Its Applications, New York, John Wiley and Sons, Inc., 1960.

[56] Parker-Rhodes, A. F., and Needham, R. M., The Theory of Clumps, Cambridge Language Research Unit, Cambridge, England, February 1960.

[57] Penny, S. K., Trubey, D. K., and Emmett, M. B., "Radiation Shielding Information Center Information Retrieval System," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 251.

[58] Perry, J. W., Kent, A., and Berry, M. M., Machine Literature Searching, New York, 1956.

[59] Proceedings of the Engineering Information Symposium, Engineering Joint Council, New York, 17 January 1962.

[60] Proceedings of the 26th Annual Meeting of the American Documentation Institute, Chicago, Illinois, October 1963.

[61] Prywes, Noah S., Gray, H. J., et al, Information Retrieval and the Design of More Intelligent Machines, U. S. Signal Corps, Final Report No. AD59URI, Moore School of Electrical Engineering, University of Pennsylvania, July 1959.

[62] Prywes, Noah S., Gray, H. J., et al, The Multi-List System, Office of Naval Research, Technical Report No. 1, Volumes I and II, Moore School of Electrical Engineering, University of Pennsylvania, 30 November 1961.

[63] Ranganathan, S. R., Classifying, Indexing, Coding, Western Reserve University, September 1959.

[64] Ranganathan, S. R., Classification and Retrieval - Problems for Pursuit, Western Reserve University, September 1959.

[65] Ranganathan, S. R., Natural, Classificatory, and Machine Languages, Western Reserve University, September 1959.

[66] Raphael, B., Personal Communication to G. Greenberg at the 1963 RAND Institute on Cognitive Simulation.

[67] Rath, G. J., Resnick, A., and Savage, T. R., Comparisons of Four Types of Lexical Indicators of Contents, (Research Report RC-187), IBM Research Center, Yorktown Heights, New York, 14 August 1959.

[68] Research in Information Retrieval, Report No. 1, ITT: IEC, 30 October 1962.

Research in Information Retrieval, Report No. 2, ITT: IEC, 31 January 1963.

Research in Information Retrieval, Report No. 3, ITT: IEC, 30 April 1963.

Research in Information Retrieval, Report No. 4, ITT: IEC, 31 July 1963.

Research in Information Retrieval, Report No. 5, ITT: DISD, 31 September 1963.

Research in Information Retrieval, Report No. 6, ITT: DISD, 30 January 1964.

Research in Information Retrieval, Report No. 7, ITT: DISD, 30 April 1964.

[69] Richmond, P. A., "Review of the Cranfield Project," American Documentation, Fol. 14, No. 4, October 1963, p 307.

[70] Science, Government, and Information, The White House, Washington, D. C., 10 January 1963.

[71] Shiloh, A., "The Plague of Print," New Scientist, Vol. 14, April 26, 1962, p 169.

[72] Slamecka, V., "Classificatory, Alphabetical and Associative Schedules as Aids in Coordinate Indexing," American Documentation, Vol. 14, No. 3, July 1963, p 223.

[73] Slamecka, V., and Zunde, P., "Automatic Subject Indexing from Textual Condensations," American Documentation Institute (ADI) Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 139.

[74] Sommers, F., Semantic Structures and the Automatic Clarification of Linguistic Ambiguity, ITT DISD, 1961.

[75] Stiles, H. E., "The Association Factor in Information Retrieval," Journal of the ACM, Vol. 8, No. 2, April 1961, pp 271-279.

[76] Swanson, D. R., "Searching Natural Language Text by Computer," Science, No. 130, 1960, pp 1099-1104.

[77] Taube, M., et al, Studies in Coordinate Indexing, Documentation Incorporated, 1953-57.

[78] Thurstone, L. I., Multiple Factor Analysis, University of Chicago Press, Chicago, 1947.

[79] Trachtenberg, Alfred, "Automatic Document Classification Using Information Theoretical Methods," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 349.

[80] Vickery, B. C., Journal of the American Documentation Institute, Vol. X, 1959, pp 234-241.

[81] Vickery, B. C., On Retrieval System Theory, Butterworths and Company, Ltd., London, 1961.

[82] Waldo, W. H., "Searchers Want Facts Not Fiction--Retrieve Data Not Documents--The Needle is Dull--Sharpen it With Automation," American Documentation Institute (ADI), Annual Meeting-1963, Short Papers, Part II, Chicago, October 1963, p 207.

[83] Watanabe, S., A Probabilistic View of the Formation of Concept and of Association, IBM Research Laboratory, Yorktown Heights, New York, presented at the annual meeting of the AAAS, 26-30 December 1961.

[84] Watanabe, S., Inference and Information, John Wiley and Sons, Inc., New York, 1964.

[85] White, S. P., and Walsh, J., "A Computer Library's Approach to Information Retrieval," Special Libraries, July-August 1963, p 345.

5. CONCLUSIONS

This section presents some ad hoc conclusions pertaining to the specific areas investigated during the course of this project. The over-all conclusions are presented in Section 6.

These conclusions are ad hoc because they represent only the first stages of research into a complex problem. The results, therefore, are tentative. Continued research could lead either to more definite results or to an entirely different set of conclusions based upon problems that are only now being defined. The conclusions are organized in terms of the basic questions discussed in the specification of retrieval systems.

5.1 DESCRIPTIVE STRUCTURE OF RETRIEVAL SYSTEMS

The most popular form of description in existing retrieval systems is the descriptor list. Although other forms of description have been considered, they have not been developed to any significant degree of effectiveness. The considerations presented regarding economy of descriptions can serve as a basis for further development, but this development remains to be implemented.

Given that the descriptor list is in fact used as the mode of description, analytic methods can be helpful in selecting the particular set of descriptors to be used. These methods are based both on the logical structure of any given document collection and on the use of that collection. Since dynamic retrieval systems change as the demands on them change and as their contents shift, corrective methods must be used to keep the descriptor set updated. The invariants that are

associated with relatedness can be providently used to keep the set updated by constantly bringing the system classification scheme into conformity with the users' classification scheme.

5.2 ASSIGNMENT OF DESCRIPTORS TO DOCUMENTS

The rationale for assigning descriptors to documents automatically--that is, with computational techniques--is that a greater degree of consistency will be achieved. Human beings are subject to numerous vagaries and inconsistencies, while a machine is invariant. Since automatic techniques depend upon the information contained in a document, the problem is to develop computational methods that will enable a machine to categorize documents accurately on the basis of both the explicit and the implicit information--or, more precisely, words--in those documents.

Two complementary techniques were analyzed during the course of this project; these techniques were based on information theory and game theory. The information theoretic formulation is a method for assessing the individual validity of descriptors on the basis of clue words occurring in documents. The game theoretic formulation provides a method for selecting an optimal set of clue words.

The use of information theoretic techniques to select clue words appears to be a promising method of document categorization. From the purely heuristic viewpoint this technique seems to be valuable and to represent an improvement over existing techniques. The use of this technique as a means of categorizing documents is easily mechanized. To the extent that the occurrences of clue words are relatively

independent of each other, this computationally simpler approach should adequately suffice for selecting clue words and is an attractive solution to the problem. However, the over-all reliability of this technique remains in doubt because it is not at all certain that clue words per se convey both the necessary and sufficient information for correct categorization and because the methods for selecting the best clue words are not ideal. Ultimately, the validity of this technique, particularly in comparison with existing methods, warrants empirical verification.

The game theoretic approach to selecting clue words is theoretically more appealling but more difficult to execute in practice. In theory this technique will in fact select the best possible set of clue words. But in practice it is still impossible to develop sufficient statistics to predict the best possible set. As yet no good techniques for approximating these statistics have been developed, but further research along these lines should be undertaken.

5.3 FILE STRUCTURE

The quantitative results obtained in the analysis of certain basic types of file structures demonstrate the value of trees and lists in information retrieval systems. These results must be tempered by a consideration of the time required for indexing operations in list-oriented file structures; in particular, for small files the standard linear methods appear to be the best because of the bookkeeping costs associated with lists. The standard deviation of the search times required for indexed trees is small, so that search times for this type

of structure can be reliably predicted. Linear forms of storage, on the other hand, tend to have high standard deviations and highly variable search times.

The Multi-List structure cannot be directly compared with the basic types of file structures because it is based upon retrieval on more than one criterion at a time. The Multi-List technique appears to be an effective way of performing retrieval of the kinds for which it was designed; however, although adding items to the file or altering items is fairly easy, deleting items is a complicated process. The value of the Multi-List system probably cannot be suitably appraised until the system is used in a practical application, since its approach is sufficiently distinctive to make it difficult analytically to compare Multi-List against other methods.

5.4 QUERY PROCESSING

The type of query processing appropriate to a given information retrieval task is highly dependent on the nature of the task. For personnel files, for instance, the problem is virtually trivial. For literature retrieval, the problem becomes more difficult and techniques such as probabilistic retrieval becomes useful. For intelligence data, quite sophisticated search and inference strategies become necessary. In both literature and intelligence information, it is important to bear in mind the amorphous nature of the user's question as contracted with his query.

Probabilistic retrieval should be a useful method for increasing the effectiveness of literature retrieval through the use of additional

information--namely, the probability that a given categorization of a document is correct. The distributional statistics needed for compound retrievals require a significant amount of bookkeeping, but this cost may well be repaid in terms of system effectiveness. For single-category retrieval, of course, no statistics are needed. The effect of raising or lowering the retrieval cutoff point permits a trade-off of false drops against missing information. However, there may be room for improvement in the particular parameters used in the optimization of the goodness of retrieval; parameters based on ratios rather than on absolute numbers of documents might possibly be more effective.

It is apparent that in any attempt to perform content retrieval rather than document retrieval, query processing lies at the heart of the problem. The system will need to perform a great deal of inference, and the ways that this inferential process can be performed are not at all clear as yet. In addition, severe problems exist with respect to the semantics of the data and the resolution of ambiguity, although there are some promising approaches in this area, particularly the application of sense-value theory. The work on the theory of questioning is still embryonic; however, some progress has been made in this area by other investigators.

6. OVER-ALL CONCLUSIONS

The state-of-the-art in information retrieval is characterized by two different approaches:

(a) Ad hoc methods for solving logically straightforward problems with the greatest possible efficiency.

(b) Theoretical efforts to resolve the difficult problems associated with descriptive structures, assigning descriptions to documents, file structure and memory organization, and query processing.

This project has been oriented toward the second approach. The appropriate approach is strictly a function of the particular application being dealt with. For retrieval on personnel files and similar applications, a highly coordinated approach to develop a complete specialized system is sufficient. The primary question then is one of application. For problems such as general documentation and intelligence analysis, there does not appear to be any way to short-cut the truly difficult problems. This study has highlighted some of these problems and developed a few tentative steps towards solving them.

The frame of reference for the research performed during the course of this project was a general system model in which two processes occur simultaneously and independently: entering documents or information about documents into the system; and responding to queries related to specific requirements for information. Although four general research tasks were isolated and analyzed, the content of these tasks was interrelated. Thus the descriptive structure of retrieval systems and the assignment of descriptors are interdependent and both are intrinsically related to the ultimate problem of query processing. These

factors also impinge upon the correlated functions of storage and retrieval. In storage devices or memories neither size nor speed are the important problem; rather, it is a question of organization, the structure of information as it pertains to the essential requirements of serving a user's demand for information.

This report has emphasized possible techniques for automating all storage and retrieval processes. A tacit assumption underlying this stress has been the problems of large information systems. Manual techniques are still suitable for relatively small collections of information. But, granting the assumption of magnitude, it is essential to develop techniques for the analysis of information by machines, primarily because human beings are notoriously inconsistent and prone to error. Only in large systems do these human tendencies lead to inefficiency and ineffectiveness.

At this stage of the research process knowledge about the nature of the total problem is insufficient. For this reason the conclusions about the research performed are tentative. Each area could be studied further with more definitive results; alternatively techniques that are potentially more beneficial could evolve. Any future research would also benefit from a test bed of data that could be used empirically to test theoretical concepts.

One fact is clear: it is still premature to develop special purpose equipment for information storage and retrieval. Such a step should be deferred until the requisite research and empirical verification has

produced reasonably complete knowledge about the problem and a comprehensive description of the requirements.

7. RECOMMENDATIONS

The concept of information retrieval has degenerated from a rigorously defined problem to a general catch-all for a variety of problems. The range of the popular description includes both the difficult and the mundane. This study has attempted to limit the definition and the scope of information retrieval to the difficult problems related either to scientific and technical documentation or to intelligence analysis.

Both documentation and intelligence analysis systems are characterized by a particular attribute: their content and nature cannot be defined a priori. Both are dependent upon their information content for their descriptions. Unless these descriptions are satisfactorily specified, and no existing method permits adequate specification, the retrieval systems will be virtually useless.

The first recommendation may, therefore, be startling. If the contemplated system is definable a priori and if the information content is well structured, no further research is required to describe a suitable retrieval system. Personnel files are the ubiquitous example. The appropriate subject in this case is not research but either systems or applications analysis. If the objective is to develop equipment, then the nature of the information system must be described, and operational characteristics must be specified for speed, accuracy, efficiency, and effectiveness.

The second recommendation has evolved from the difficulty of adhering

to a pure definition of information retrieval. This recommendation also follows from the current state of knowledge about the subject. The subject of information retrieval has become too broad, while specific problems confronted in information retrieval have been either roughly or specifically defined during the course of several research programs, including this one sponsored by USAEL. Further research in information retrieval *per se* would result in an indefinitely structured project. Funds would be more fruitfully expended on research projects related to specific problem areas encompassed by information retrieval.

The need for special studies, defined and specified as such, is urgent. The research conducted during this project, for example, constitutes only a beginning. This recommendation, therefore, is presented as a necessary next step in advancing the state-of-the-art and in enhancing the use of automated techniques, specifically computer-oriented techniques.

The principal recommendation for future work is that it be directed more towards specific types of problems. For applications where the problems of developing a descriptive structure and assigning descriptions to documents are trivial, it is advisable to develop an *ad hoc* system that is highly coordinated internally and specialized for a particular problem. Such systems need not be completely specialized because a system that is appropriate for personnel records may also be appropriate for parts listings or for literature with an existing fixed set of categories and manual categorization. However, it is inadvisable to

try to attack problems such as intelligence analysis with a similar system.

The importance of the more difficult problems is sufficiently great so that a long-term and continuing research program is thoroughly warranted. This program would require the extension of some of the ideas developed in this project within a more rigorous theoretical framework. The studies should consider the following problems as well as others:

(a) Descriptive Structure - The work performed during this project has only begun to attack this problem. It is necessary to develop a formal, perhaps mathematical, theory of the structure of knowledge and to base the descriptive scheme on this structure. The development of a formal theory has been attempted, but as yet the efforts have been inadequate to the task. A solid theory of descriptive structure is the essential underpinning of any content retrieval system; until this theory has been completed, all other conclusions are at best tentative.

(b) Linguistic Analysis - It is recommended that existing work in mechanical translation of languages be applied to the transformation of natural languages to formal languages suitable for deductive reasoning. Many of the problems of natural language translation can be sidestepped in this effort, since the translational defects will not seriously impair the effectiveness

of a retrieval system. For instance, the problem of translating a word with several alternative meanings can be considerably simplified, since for most purposes the mere identity of words will be sufficient for the kinds of deductions to be performed. It should be emphasized that this recommendation is for the application of existing work in a different area rather than for totally new investigations.

(c) Methods of Inference - Given a large body of formal statements, methods are needed for obtaining the desired logical consequences of these statements. The problem resembles, but is not identical to, the problem of developing formal proof procedures for symbolic logic. The major difference is that relatively immediate inferences are to be drawn from a large base of information rather than quite deep inferences from a small base of information. The solution of this problem is also essential for an effective content retrieval system.

(d) Development of Query Languages - The particular mode of communication between the user and the retrieval system must be studied in detail. It is recommended that work should be performed in this area, but not until the other areas have been more thoroughly developed.

8. IDENTIFICATION OF PERSONNEL

8.1 PERSONNEL ASSIGNMENTS

The following personnel were assigned to this project during the course of the contract:

Jacques Harlow*	Principal Investigator
Paul W. Abrahams, Sc.D.*	Research Specialist
George Greenberg, Ph.D.*	Research Specialist
Quentin A. Darmstadt	Research Specialist
Alexander Szejman	Senior Specialist
Alfred Trachtenberg	Senior Program Analyst
Maralyn W. Lindenlaub	Senior Program Analyst

The asterisk (*) indicates those personnel who contributed to the project during the final quarter. Both Drs. Greenberg and Abrahams acted as associate investigators at different times in the course of this research program; particularly, Dr. Abrahams filled this role during the last two quarters and contributed significantly to the integration of the several research tasks.

The approximate number of man-hours by title expended during the total contractual period was:

Management and Supervision	700
Research Specialist	1500
Senior Specialist	4500
Senior Program Analyst	4500
Clerical	300

The titles in the previous paragraph reflect each person's position

during the last quarter of his participation in the project. Therefore, the distribution of man-hours differs from the distribution of present titles.

8.2 BACKGROUND OF PERSONNEL

The background of each person assigned to this project was summarized in the quarterly reports.

9. APPENDICES

9.1 APPENDIX A - Maxima and Minima of the Measures

In this appendix the behavior of the measures of goodness and the various entropy functions will be examined. Maxima and minima in terms of the p_j and p_{ij} are summarized in Tables 3 and 4.

For these tables it is assumed that A is chosen such that $A = 1/p_e$ where p_e is the smallest p_j; that is, $p_e \leq p_j$ for all j. For the functions of Table 3--H, H_i, H_A, and S_i--the pertinent values are the maximum and minimum values in terms of a given p_e and the absolute maximum and minimum values of each function.

For H and H_i, maxima are reached when the probabilities are equal or, for a particular p_e, when the other p_j are equal, minima are reached when one probability becomes a maximum and the rest are minima.

While H_A does <u>not</u> reach an absolute maximum when H does, since it was assumed that $A = 1/p_e$, it does reach a maximum together with H for a particular p_e. Then:

$$H_A = -\sum_j p_j \log p_j + \log A = -\sum_j p_j \log p_j - \log p_e$$

$$= -\sum_{j \neq e} p_j \log p_j - (1 + p_e) \log p_e \qquad \text{(A-1)}$$

Therefore, H_A becomes a maximum for a <u>particular</u> p_e when $p_j = \frac{1 - p_e}{k - 1}$ for $j \neq e$. Then:

$$H_{Amax} = (1 - p_e) \log \left(\frac{k - 1}{1 - p_e}\right) - (1 + p_e) \log p_e \qquad \text{(A-2)}$$

TABLE 3. Maxima and Minima of Entropy Functions

Function	Maximum occurs at:	Maximum is:
H	max when: $p_j = \frac{1-p_e}{k-1}$ for $j \neq e$; $p_e \leq p_j$ for all j	$H_{max} = (1 - p_e) \log (\frac{k-1}{1-p_e}) - p_e \log p_e$
	abs max when all p_j are equal: $p_j = p_e = \frac{1}{k}$	$H_{absmax} = \log k$
H_i	$p_{ij} = \frac{1}{k}$ for all j	$H_{imax} = \log k$
H_A	max when: $p_j = \frac{1-p_e}{k-1}$ for $j \neq e$; $p_e \leq p_j$ for all j	$H_{Amax} = (1 - p_e) \log (\frac{k-1}{1-p_e}) - (1 + p_e) \log p_e$
	abs max when: $p_e = \frac{1}{N}$	$H_{Aabsmax} = (1 + \frac{1}{N}) \log N + (1 - \frac{1}{N}) \log (\frac{k-1}{1-\frac{1}{N}})$
S_i	max when: $p_{ij} = p_j$ for all j	$S_{imax} = - \log p_e$
	abs max when: $p_e = \frac{1}{N}$	$S_{iabsmax} = \log N$

TABLE 3 (Continued). Maxima and Minima of Entropy Functions

Function	Minimum occurs at:	Minimum is:
H	min when: $p_t = 1 - (k-1)p_e$, $p_j = p_e$ $(j \neq t)$ $p_e \leq p_j$ for all j	$H_{min} = -(k-1)p_e \log p_e - [1-(k-1)p_e] \log[1-(k-1)p_e]$
	abs min when: $p_j = p_e = \frac{1}{N}$ $(j \neq t)$ $p_t = 1 - \frac{k-1}{N}$	$H_{absmin} = \frac{k-1}{N} \log N - (1 - \frac{k-1}{N}) \log (1 - \frac{k-1}{N})$
H_i	$p_{ie} = 1$, $p_{ij} = 0$ $(j \neq e)$	$H_{imin} = 0$
H_A	min when: $p_t = 1-(k-1)p_e$ $p_j = p_e$ $(j \neq t)$ $p_e \leq p_j$ (for all j)	$H_{Amin} = -[1+(k-1)p_e] \log p_e - [1-(k-1)p_e] \log[1-(k-1)p_e]$
	abs min when: $p_e = p_j = \frac{1}{k}$	$H_{Aabsmin} = 2 \log k$
S_i	$p_{ie} = 1$, $p_{ij} = 0$ for $j \neq e$ $p_e \leq p_j$ for all j	$S_{imin} = 0$

The largest H_{Amax} occurs when $p_o = 1/N$. Then:

$$H_{Aabsmax} = (1 + \frac{1}{N}) \log N + (1 - \frac{1}{N}) \log (\frac{k - 1}{1 - \frac{1}{N}}) \qquad (A-3)$$

H_A becomes a minimum for a particular p_e when H does; that is, when the maximum p_j, $p_t = 1 - (k - 1)\, p_e$, and $p_j = p_e$ for $j \neq t$, where $p_e \leq p_j$ for all j. Then:

$$H_{Amin} = - [1 - (k - 1)\, p_o] \log[1 - (k - 1)\, p_e] - [1 + (k - 1)\, p_e] \log p_e \qquad (A-4)$$

The smallest H_{Amin} occurs when $p_e = 1/k$. Then:

$$H_{Aabsmin} = 2 \log k \qquad (A-5)$$

S_i becomes a maximum when $p_{ij} = p_j$ for all j. This maximum can be derived by using Gibbs' theorem, as in Watanabe [84]:

$$S_{imax} = \log A = - \log p_e \qquad (A-6)$$

The largest S_{imax} occurs when $p_e = 1/N$.

$$S_{iabsmax} = \log N \qquad (A-7)$$

S_i becomes a minimum when p_{ij} becomes one for the particular j for which p_j is smallest. Then:

$$S_{imin} = - \log \frac{1}{Ap_e} \qquad (A-8)$$

But; $$A = 1/p_e \qquad (A-9)$$

So: $$S_{imin} = 0 \qquad (A-10)$$

For the functions of Table 4--M_1, M_2, M_3, and M_4--there are three maximum and minimum values: the maxima and minima for a given p_j distribution; the maxima and minima when only p_e is given; and the absolute maxima and minima. To keep the notation consistent with that of Table 3, these maxima and minima will be indicated as follows:

$$M_{1maxj},\ M_{2maxj},\ \text{etc.}$$

are the maxima for a given p_j distribution. Similarly,

$$M_{1minj},\ M_{2minj},\ \text{etc.}$$

are the minima for a given p_j distribution.

M_{1max}, M_{2max}, M_{1min}, M_{2min}, etc., are the maxima and minima when only p_e is given, and $M_{1absmax}$, $M_{2absmax}$, $M_{1absmin}$, $M_{2absmin}$, etc., are the absolute maxima and minima.

$M_1 = H - H_i$ is maximized for a particular p_j distribution when H_i is a minimum ($H_{imin} = 0$). Then M_{1maxj} is simply the a priori entropy H. M_{1max}, which is M_1 maximized for a particular p_e, is simply the a priori entropy maximized, H_{max}. $M_{1absmax}$ is the absolute maximum of the a priori entropy.

Similarly the minima of M_1 are obtained when H_i is set equal to H_{imax} ($H_{imax} = \log k$) by minimizing the a priori entropy.

$M_2 = H - S_i$ is maximized when S_i is a minimum ($S_{imin} = 0$); the maxima are simply the maxima of the a priori entropy. M_2 is minimized when $S_i = S_{imax} = -\log p_e$; $M_{2min} = H_{min} - S_{imax}$ when $H = H_{min}$ in addition.

TABLE 4. Maxima and Minima of Measures of Goodness

Function	Maximum occurs at:	Maximum is:
$M_1 = H = H_i$	max for p_j when: $H_i = H_{imin}$	$M_{1maxj} = H$
	max when: $H = H_{max}$, $H_i = H_{imin}$	$M_{1max} = H_{max}$
	abs max when: $H = H_{absmax}$, $H_i = H_{imin}$	$M_{1absmax} = H_{absmax} = \log k$
$M_2 = H - S_i$	max for p_j when: $S_i = S_{imin}$	$M_{2maxj} = H$
	max when: $H = H_{max}$, $S_i = S_{imin}$	$M_{2max} = H_{max}$
	abs max when: $H = H_{absmax}$, $S_i = S_{imin}$	$M_{2absmax} = H_{absmax} = \log k$
$M_3 = H_A - S_i$	max for p_j when: $S_i = S_{imin}$	$M_{3maxj} = H_A$
	max when: $H_A = H_{Amax}$, $S_i = S_{imin}$	$M_{3max} = H_{Amax}$
	abs max when: $H_A = H_{Aabsmax}$, $S_i = S_{imin}$	$M_{3absmax} = (1 + \frac{1}{N}) \log N + (1 - \frac{1}{N}) \log(\frac{k-1}{1-\frac{1}{N}})$
$M_4 = \log A - S_i$	max for p_j when: $S_i = S_{imin}$	$M_{4maxj} = \log A = -\log p_e$
	max when: $S_i = S_{imin}$	$M_{4max} = \log A = -\log p_e$
	abs max when: $S_i = S_{imin}$, $p_e = \frac{1}{N}$	$M_{4absmax} = \log N$

TABLE 4 (Continued). Maxima and Minima of Measures of Goodness

Function	Minimum occurs at:	Minimum is:
$M_1 = H - H_i$	min for p_j when: $H_i = H_{imax}$	$M_{1minj} = H - H_{imax} = H - \log k$
	min when: $H = H_{min}$, $H_i = H_{imax}$	$M_{1min} = H_{min} - H_{imax} = H_{min} - \log k$
	abs min when: $H = H_{absmin}$, $H_i = H_{imax}$	$M_{1absmin} = -\log k + \frac{k-1}{N} \log N - (1 - \frac{k-1}{N}) \log(1 - \frac{k-1}{N})$
$M_2 = H - S_i$	min for p_j when: $S_i = S_{imax}$	$M_{2minj} = H - S_{imax} = H + \log p_e$
	min when: $H = H_{min}$, $S_i = S_{imax}$	$M_{2min} = H_{min} - S_{imax} = H_{min} + \log p_e$
	abs min when: $H = H_{absmin}$, $S_i = S_{iabsmax}$	$M_{2absmin} = (\frac{k-1}{N} - 1) \log N - (1 - \frac{k-1}{N}) \log(1 - \frac{k-1}{N})$
$M_3 = H_A - S_i$	min for p_j when: $S_i = S_{imax}$	$M_{3minj} = H$
	min when: $H = H_{min}$, $S_i = S_{imax}$	$M_{3min} = H_{min}$
	abs min when: $H = H_{absmin}$, $S_i = S_{imax}$	$M_{3absmin} = \frac{k-1}{N} \log N - (1 - \frac{k-1}{N}) \log(1 - \frac{k-1}{N})$
$M_4 = \log A - S_i$	min for p_j when: $S_i = S_{imax}$	$M_{4minj} = 0$
	min when: $S_i = S_{imax}$	$M_{4min} = 0$
	abs min when: $S_i = S_{imax}$	$M_{4absmin} = 0$

$M_{2absmin}$ occurs when $H = H_{absmin}$. $M_3 = H_A - S_i$ is maximized when $S_i = S_{imin}$; the maxima are H_A, H_{Amax}, and $H_{Aabsmax}$, respectively. The minima of M_3 are not as obvious, for the conditions of maximizing S_i and minimizing H_A can be contradictory. It is best to analyze the minima of M_3 as follows:

$$M_3 = H_A - S_i = -\sum_j p_j \log p_j + \log A + \sum_j p_{ij} \log \frac{p_{ij}}{Ap_j}$$

$$= -\sum_j p_j \log p_j + \sum_j p_{ij} \log \frac{p_{ij}}{p_j} \qquad \text{(A-11)}$$

For a particular p_j distribution, M_{3minj} occurs when $p_{ij} = p_j$ for all j. Therefore:

$$M_{3minj} = -\sum_j p_j \log p_j = H \qquad \text{(A-12)}$$

Then for a particular p_e:

$$M_{3min} = H_{min} \qquad \text{(A-13)}$$

and the absolute minimum is simply:

$$M_{3absmin} = H_{absmin} \qquad \text{(A-14)}$$

M_4 is the simplest measure of them all, reaching a maximum when S_i is minimum, and a minimum when S_i is maximum.

$$M_4 = \log A - S_i = +\sum_j p_{ij} \log \frac{p_{ij}}{p_j} \qquad \text{(A-15)}$$

That this measure is always greater than or equal to zero can be shown by applying Gibbs' theorem:

$$M_4 = \sum_j p_{ij} \log p_{ij} - \sum_j p_{ij} \log p_j \qquad \text{(A-16)}$$

But:

$$\sum_j p_{ij} \log p_{ij} - \sum_j p_{ij} \log p_j \geq 0 \qquad \text{(Gibbs' theorem)} \qquad \text{(A-17)}$$

Therefore;

$$M_4 \geq 0. \qquad \text{(A-18)}$$

The maximum of M_4 is:

$$M_{4maxj} = M_{4max} = \log A \qquad \text{(A-19)}$$

The absolute maximum occurs when $p_e = 1/N$; then, $A = N$ and:

$$M_{4absmax} = \log N \qquad \text{(A-20)}$$

9.2 APPENDIX B - Derivation of the Predictor Effectiveness Measure M_4 From Some Fundamental Definitions of Information

The information, I, supplied by an event is usually defined as the difference between the a priori and a posteriori entropies. In this case:

$$I = H - H_1 \tag{B-1}$$

Where;

$$H = - \sum_j p_j \log p_j \tag{B-2}$$

And;

$$H_1 = - \sum_j p_{ij} \log p_{ij} \tag{B-3}$$

To overcome the difficulty of having a negative information quantity at times, which does not concur with our intuitive notions of information, Watanabe [84] suggests that relative entropy functions should be used instead of the usual entropy functions H and H_1. The relative entropy, S, in general is:

$$S = - \sum_j \Pi_j \log \frac{\Pi_j}{Bq_j} \tag{B-4}$$

Where;

Π_j = the probability distribution under study

q_j = the a priori, or reference, probability distribution

B = a positive constant.

Then, by using the standard definition of information, the difference between the two entropies, except for substituting relative entropies this time, we obtain:

$$I_r = S(p_j) - S(p_{ij}) \tag{B-5}$$

To evaluate $S(p_j)$, let:

$$\left.\begin{aligned} \Pi_j &= p_j \\ q_j &= p_j \\ B &= A \end{aligned}\right\} \tag{B-6}$$

To evaluate $S(p_{ij})$, q_j remains equal to p_j and $B = A$, but:

$$\Pi_j = p_{ij} \tag{B-7}$$

Then:

$$S(p_j) = \log A \tag{B-8}$$

And:

$$S(p_{ij}) = -\sum_j p_{ij} \log \frac{p_{ij}}{Ap_j} = S_i \tag{B-9}$$

Then:

$$I_r = \log A - S_i = \sum_j p_{ij} \log \frac{p_{ij}}{p_j} \tag{B-10}$$

Therefore:

$$I_r = M_4 \tag{B-11}$$

and, M_4 then measures the amount of information supplied by the occurrence of word W_i.

M_4 can also be derived by using the definition of information used by Goldman [23]: the log of the ratio of the a posteriori to the a priori probability. Symbolically, for this case:

$$I' = \log \frac{p_{ij}}{p_j} \tag{B-12}$$

If this quantity is averaged over all i and j, then the usual information quantity results. However, this quantity should be averaged over j only; and this averaging must be done for a particular i. The quantity desired is:

$$\langle I' \rangle_{j|i} = \left\langle \log \frac{p_{ij}}{p_j} \right\rangle_{j|i} \tag{B-13}$$

It is necessary, then, to use the conditional probability distribution p_{ij} to obtain the correct average. Then:

$$\langle I' \rangle_{j|i} = \sum_j p_{ij} \log \frac{p_{ij}}{p_j} \tag{B-14}$$

And;

$$\langle I' \rangle_{j|i} = M_4 \tag{B-15}$$

9.3 APPENDIX C - Existing Methods of Document Description

9.3.1 Indexing and Automation - A fundamental aspect of today's indexing schemes is their ultimate adaptibility to automated procedures. These procedures have been used to produce many different types of indexes, including author, citation, report number, conventional subject-heading, and coordinate indexes. Coordinate indexing, which may be considered as one of the first steps beyond the traditional manual indexing systems, consists of the description of information contained in documents by the use of unit-concepts. These unit-concepts are called by many names: Uniterms (Taube), keywords (Luhn), and descriptors (Mooers). Unit-concepts can be characterized by the controls placed upon them. For example, if we extract words directly from documents and use these words without further controls of any kind (such unit-concepts have been called Uniterms), we have the basis of a permuted or KWIC indexing scheme. We shall review this indexing method in some detail and analyse some of the effects that such a control-free word system appears to be having on indexers and authors alike.

The use of a Uniterm system can inflict a large number of synonyms upon a user. For example, if we use Roget's Thesaurus as an authority, the word "hardness" has such synonyms as: rigidity, firmness, stiffness, inflexibility, temper, toughness, etc. Such a system of Uniterms needs cross-referencing from one word to synonyms or related words. The Chemical Engineering Thesaurus and the ASTIA Thesaurus of Descriptors (2nd edition) are examples of such referencing. Such a free vocabulary may be transformed into a formal descriptor language that will be

synonym free since explicit definitions or scope notes will exist for each descriptor. If the number of descriptors to be used is not fixed, then at least the rate of growth should be subject to careful regulations. Since only a limited number of descriptors can efficiently be assigned to a text, Jacobson [37] has assumed that only a limited amount of text can be efficiently indexed. He further suggests the need to divide the text of documents into distinct portions and to subject each portion to certain indexing regulations.

As more descriptors are assigned to a document in an effort to anticipate novel requests for information, the possibility of increasing the noise, or non-relevant information, is increased. Several devices have been incorporated into descriptor schemes to reduce this noise. Maron and Kuhns [48] suggests that each descriptor may be weighted according to its relevance for the particular document involved. Hilf [34] reports a practical approach to weighting by the use of an asterisk to indicate those descriptors of major interest.

9.3.2 Facet Analysis and Role Indicators - One technique for organizing the proliferation of descriptors is known as facet analysis. The entire set of descriptors is grouped into facets. The descriptors within a facet can be viewed as the possible answers to a question concerning the contents of a document to be classified. Thus a facet represents the question itself; ideally, facets should be chosen so that their corresponding questions exhaust the information on how to classify the document and, at the same time, so that there is a minimal overlap

(hopefully, none) of the informational content in the answers to the questions. If, for a particular document, the question represented by a facet, is meaningless, no descriptor from this facet will be assigned to the document.

In terms of the Multi-List system discussed in Section 4.4.2, attributes may be viewed as facets and values of attributes, as descriptors within facets. If attributes are set up by human beings, they may correspond to natural questions; but if they are set up mechanically, they may correspond to quite complicated and artificial questions.

A discussion of facet analysis appears in Vickery [81]. Vickery specifies the product of a facet analysis to be a set of schedules in which terms are first grouped into well-defined facets and then--within each facet--arranged in a class order. The classifier using these schedules is aided because the structure of each subject is displayed. The selection of facets is dictated by the user's requirements. As an example, a survey of 1000 research physicists identified some of the following performance characteristics of a reference retrieval system: it should specify type of research (whether experimental or theoretical); it should specify aspect of research (property, object, method) [2]. An example of a working system is an engineering field consists of a descriptor vocabulary of 600 words within a framework of nine facets [35]. Hayes [30] has also pointed out the advantages of facet analysis from the automation point of view. Slamecka [72], however, feels it is conjectural whether facet analysis helps to improve the quality of indexing.

Closely related to facet analysis is another method known as role indication. When this method is used, each descriptor has appended to it a suffix that says what sort of descriptor it is; or, in terms of facet analysis, what facet does it belong to. These suffixes are known as role indicators. For example, in the Western Reserve University system, which utilizes twenty-four role indicators, the suffix KAM indicates a descriptor referring to a process and the suffix KIT, a descriptor of time or place. Costello and Wall use eleven role indicators, Farradane [19] has proposed the use of nine, and the Engineering Joint Council [59] recommends the use of ten.

It is difficult to ascertain the relative effectiveness of the various descriptor organizations used in indexing. The Cranfield Project [69] was designed as an investigation into the relative retrieval efficiency of four forms of indexing: universal decimal classification, a subject-heading system, a faceted classification, and the Uniterm system. The results of this project are now available, but must be interpreted only in the light of a thorough knowledge of the project.

9.3.3 KWIC Indexing - The procedure commonly known as permuted indexing or KWIC indexing--that is, Key-Word-In-Context indexes--is the most sophisticated of today's operational automated indexing schemes. Yet it is not without its critics, and certainly not without inherent limitations. We shall briefly review the nature of this system as well as some present thoughts on making such indexing more effective.

KWIC indexing may be carried out on various levels; the process

may be applied to the title, the abstract, portions of the text, or, indeed, the entire text. Thus far the method has had its greatest reported use in connection with titles. KWIC indexing uses the content words in the title of an article as index terms. A list of non-significant words is prepared for use in processing a KWIC index. This list would include words such as "an", "of", "in", "the", "at", "are", etc. Each word within the title that is not on the non-significant word list is cyclically permuted in such a way that the word is aligned on a particular column so that alphabetical sequence is observable. For example, consider the title:

"An Evaluation of KWIC Indexing Methods in Chemistry."

This title would be arranged as follows in a KWIC index:

INDEXING METHODS IN CHEMISTRY. AN EVALUATION OF KWIC
IN CHEMISTRY. AN EVALUATION OF KWIC INDEXING METHODS
EVALUATION OF KWIC INDEXING METHODS IN CHEMISTRY. AN
AN EVALUATION OF KWIC INDEXING METHODS IN CHEMISTRY.
OF KWIC INDEXING METHODS IN CHEMISTRY. AN EVALUATION

The first use of KWIC indexing was reported at the International Conference on Scientific Information in Washington, D. C., in 1959 [7]. Since that time the KWIC technique has been used to index the literatures of chemistry, biology, aerospace, and a score of other fields.

9.3.3.1 The Descriptive Power of Titles - The KWIC indexing procedure is based upon the assumption that the title of an article is descriptive of the information content of the article and significantly related to it. Some of the reported problems with the system have been

based on the simple fact that most of these indexes have used a single line (60-character) print format that does not effectively handle the longer titles. More significant, however, are those problems that seem to attack the fundamental assumption of this indexing method. The problem is described in various ways. Meyer-Uhlenried [49] states that an analysis of different KWIC indexes has shown that titles are often not significant enough for the publication; and Perry, et al, [57] have said that the literature must be examined thoroughly in order to determine content because the content is not always obvious from the abstract. Newbaker [51], on the other hand, claims that titles contain sufficient indexing information for most retrieval applications.

Data are occasionally presented to substantiate a position on the matter. For example, Slamecka and Zunde [73] report that, when evaluated for use in permuted and KWIC indexes, between 50 and 90 percent of author-prepared document titles (depending on subject field and other factors) were found fully to reflect the subject terms to which their documents were assigned by human indexers. In a preliminary examination of various legal information problems by the American Bar Foundation [13] an experiment was conducted in which KWIC indexing of titles was compared with indexing by the subject-heading classification system. The results showed that 64.4 percent of the title entries contained as keywords compared to one or more of the subject-heading words under which they had been indexed and 25.1 percent contained logical equivalents. In a report by White [85] of experiments on methods of indexing the 1967 issues of the Abstracts of Computer Literature, the

permuted-title-indexing retrieved only 52 percent of the information.

Data from comparative tests of this kind will vary depending on such items as test criteria and definitions, indexing systems being compared, and subject field being indexed. Bornstein [10] states that the conflict in Swanson's [76] results can be traced to the different experimental methods used and the definition of the criteria of success. For example, we recently compared the descriptors (part of a faceted classification scheme) used to index 162 papers in the field of scientific communication [60] and the terms in a KWIC index of the same papers. Only 13 percent of the papers had titles that reflected fully the descriptors used to index the same documents.

9.3.3.2 Querying Problems Using KWIC - A quite different difficulty arising with KWIC indexing lies in the fact that querying is done manually by scanning an output list. Once the output exceeds a size such that it can be scanned by a human being in a reasonable time, its value decreases significantly. One reason for this change in value is the problem of synonymy. As long as the output is manageably small, a user of a KWIC index can simply read through the entire index and note the associated documents whenever he encounters a synonym of the descriptor that concerns him. He need not think of the synonyms beforehand, since he will recognize them when he sees them. Once the output becomes too bulky to be scanned in its entirety, the user must resort to a thesaurus of synonyms. Even with such a thesaurus the large number of synonyms may make retrieval extremely awkward.

A further difficulty arises when the desired documents belong to the intersection of two or more descriptive categories. Each of the categories may be quite large; yet their intersection may be small. The user must scan each of the categories in full to find that small set of documents lying in the intersection.

9.3.3.3 Improving the KWIC System - KWIC is now an operational automated indexing system. The problems that have been noted seem real, but solutions to these problems are being advanced and some are themselves becoming operational. The solutions that we shall enumerate run the gamut of possible controls and procedures that would affect an indexer, an author, and a user.

At the Scientific and Technical Information Facility titles of documents are expanded and elaborated into a notation of content for publication in STAR. This notation of content can be considered either as an expanded title or as a highly condensed abstract. This technique might be considered the first step, from the indexing point of view, towards improving the effectiveness of titles for deriving indexing terms. It is a fundamental assumption of an indexing system proposed by the Engineering Joint Council [1, 59] that the author of a technical article can be the most instrumental in the one-time indexing of his article. As an ideal situation this indexing would satisfy all future indexing requirements for that article.

Connolly [12] discusses his experience with key-terms or content analysis appearing in Applied Physics Letters. The terms were

originally assigned in the editorial office of the Jet Propulsion Laboratory, but they are now generally provided by the author's filling out a form that is sent to him when his paper is received. A combination of the terms drawn from the author's completed forms and from the title of the articles might well overcome one major objection to KWIC indexing: that titles alone may be inadequate as descriptors of the content of a paper.

Another point of view for including the author in the indexing problem is noted by Brandenburg [11], who states that title writing must balance machine requirements against human scanning habits. Man-machine requirements may conflict with acceptable title length, significant words, attention-getting devices, and work forms for retrieval. Similarly, Kennedy [39] has enumerated nine steps for the construction of good titles for ultimate KWIC indexing.

9.3.4 Other Problems in Scientific Documentation - Some of the difficulties in retrieving scientific information lie in the nature of the documents themselves rather than in the descriptiveness of titles and index terms. This conclusion is reached by the Weinberg Report [70; see also 68, No. 4], which laments the failure of scientists and engineers to express themselves clearly. It is reported [13] that Tufts University is critically reviewing the literature and past research on the effectiveness of technical writing as a means of communication. The study is concentrated on the variables in the writing and graphic processes that have some measureable communication effect upon the reader.

Of interest to the author of scientific communication are the many comments [25, 82] that suggest that part of today's problem of information retrieval from the sheer volume of literature and a certain carelessness with which scientists stuff the literature with their reports. Waldo [82] refers to a system that replaces report writing, indexing, and file storage by storing data on magnetic tape and by retrieving as necessary through appropriate questions to a computer. A similar notion was viewed by Hamming [26] as information regeneration. He gives the example: rather than retrieve the values of trigonometric functions, regenerate them as needed. Dubinin [15] of the USSR also suggests the use of computing machines for storing information and for retrieving available information only upon demand. And, perhaps as a final extreme, Shiloh [71] has suggested that the burden of reading should be lightened by using other techniques of communication, in particular the use of international seminars.

9.3.5 Summary - A great deal of existing literature has been examined in order to discover existing systems for organizing descriptors. These systems have included thesauri for treating synonyms, various unit-concept systems, facet analysis, and role indicators. In addition, the KWIC system has been investigated. This system is significant chiefly because it is by far the most popular system in use today, and for many applications it fulfills the user's needs at a low cost. Nevertheless, it has significant drawbacks. Titles are often created without anticipating their use in KWIC indexing, and these titles are not always a good reflection of the content of the articles to which they are attached,

although this point is still actively disputed. In addition, when a list of documents is too long to be scanned conveniently by a human being, difficulties arise both in searching for synonyms of a given descriptor and in retrieving documents from the intersection of two or more large categories.

Some of the problems in descriptor organization and information retrieval generally stem from the failure of authors to express themselves clearly. This difficulty appears in the form of meaningless titles and in the form of articles that are difficult to index even manually. Requiring the author to attach descriptors to his work may help to solve this problem, but the probability of effective help is low.

9.4 APPENDIX D - Sense Value Theory and Equivocation in Relation to Inferential Information Systems

This appendix is an illustrative exposition of sense value theory. Its primary intent is to clarify the applicability of sense value theory to the problem of equivocation and to outline necessary further research and development on sense value theory in order to render it applicable to problems in inferential information processing. A formal exposition of sense value theory is contained in Sommers [74] and Darmstadt [14].

In order to appreciate the relevance of sense value it is necessary to understand the level of language to which sense value theory is addressed. For the sake of this discussion, five levels of language may be discriminated:

(a) Morphology, orthography, or spelling.

(b) Syntax or grammar.

(c) Sense.

(d) Logic, consistency, or inference.

(e) Fact, truth, or reference.

This description of language levels is suggestive rather than precise. In general, information systems are ultimately concerned with language at level five; that is, someone needs to know the facts in a given field of knowledge. But an automated information system cannot, at present, conceivably perform any empirical tests on the truth of its assertions. Such verification is still best left to human performance. At present we are most interested in developing processing capabilities at the fourth level of language, and sense value theory is primarily

addressed to the third level of language. It is important to note, however, that valid conclusions at higher levels of language depend upon the organization of assertions at lower levels of language.

The last conclusion, as well as the language level classification, is perhaps best understood in terms of a specific example. Consider the assertion, "John Smith is the Prime Minister of England." At the factual level we are interested in the truth of this assertion. If, however, we amend the assertion to read "...and so is John Jones," then we can conclude from considerations at the fourth (logical) level that the statement need not be evaluated at the fifth (factual) level.*

A statement becomes inappropriate for evaluation at the fourth level if a failure or error occurs at an earlier level. Thus, if we change the statement to say, "John Smith is a prime number," then in the ordinary sense of the use of proper names and of prime number the statement simply does not make sense. It is not a matter of empirical test that people are not prime numbers nor even a function of arbitrary definition such as that there is only one prime minister. People just are not the sorts of things that can be prime numbers, nor are numbers the sorts of things that can be prime ministers.

The last example, while failing at the third level of language--that is failing to make sense--still is adequately formed at lower levels of

---

*Given the knowledge that only one person may, by definition, be prime minister and that Smith and Jones are not the same person, then the sentence is logically incorrect.

language. Thus the grammar and orthography of the example are impeccable. It is not necessary to give examples of failures at the syntactic or morphological level; they are both obvious and outside the scope of this discussion. It is apparent, however, that the progression of criteria applies to the lower levels of language. Thus it is pointless to determine whether a combination of letters that do not form words in the language is grammatical or whether a combination of words that is not a sentence meets the sense criterion of level three.

The observation that it makes no sense to say of some sorts of things--for example, people--that they are other sorts of things--for example, prime numbers--is central to the theoretical treatment of the sense level. To say that a thing is a particular sort of thing is to predicate something of it. Some predicates may be applied to the same things and thus may be called copredicable. The fundamental hypothesis of sense value theory is that if two predicates, say A and B, are copredicable, then either A is predicable of all the things of which B is predicable, or else B is predicable of all the things of which A is predicable, or both. The last statement implies that for two predicates, A and B, either all the individuals or things of which A is predicable may also be described by B, or else A may be predicated of all the things of which B is predicable, or else there are no individuals of which both A and B may be predicated.

The predicability relations between predicates is perhaps best illustrated graphically with a specific example. Figure 13 shows some

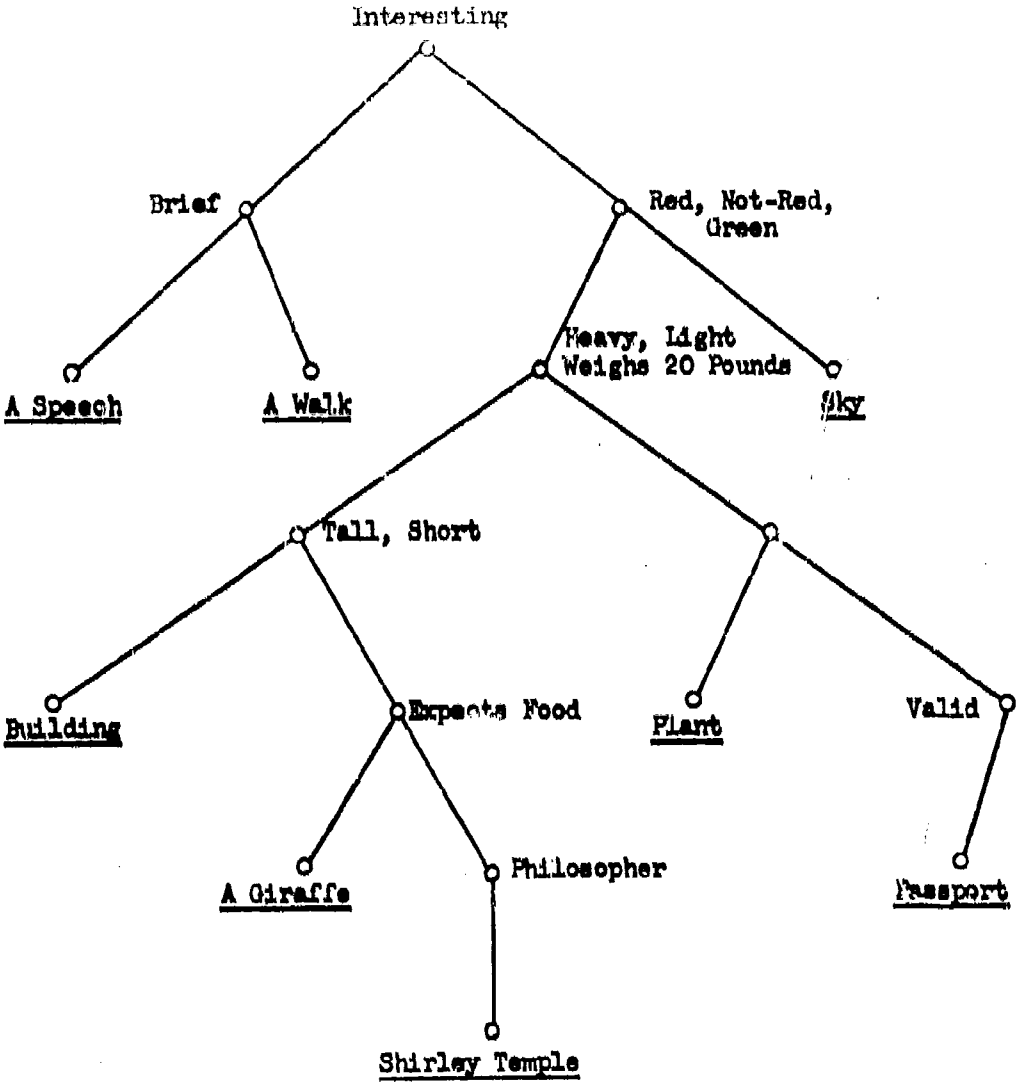


FIGURE 13. Terms of a Language Disposed in Hierarchical Tree

of the terms in a language disposed in a hierarchical tree. The individuals or things are underlined and are located in the lowest nodes. The predicates are in the higher nodes. If a predicate is connected to an individual by descending lines without any ascending lines intervening, then it is predicable of that individual. If follows that any pair of predicates connected by a series of lines without reversals from ascending to descending, or that are at the same node, are copredicable. For those at the same node, the same set of individuals may be described. If one predicate is higher, its scope is greater and it applies to more individuals than the lower predicate, but its scope includes the scope of the lower predicate. If two predicates cannot be connected by a series of lines without reversing direction, then their scopes have no individuals in common. The latter condition requires that no more than one descending line enter a node; generally, more than one will leave it if it is not a bottom node.

Some specific examples from the tree may clarify these generalizations. The top node of the tree in this case is filled by "interesting." One of the theorems in the formal development of sense value theory demonstrates that there must always be a single upper node for any given language. This theorem means that there are always some predicates that are predicable of all individuals. The right hand node below "interesting" contains all color terms. It is worth noting that both "red" and "not-red" have the same scope in sense value terms, even though they will be mutually exclusive at the factual level. This correspondence occurs because it makes sense to describe a sky that happens not to be blue as

blue or a book that happens not to be green as green. Notice, however, that the tree has already bifurcated and that there are some individuals that cannot be described by color predicates--for example, "a speech" and "a walk." Thus "brief" and "red" are not copredicable while "red" and "heavy" are.

For the purpose of this discussion, however, we are not primarily interested in mapping predicability relations but in the contribution of sense value theory to automatic inferential processing via the detection of equivocation. But it is precisely the mapping of predicability relation that allows us to detect equivocation automatically. Thus, there is a sense in which "a speech" might be referred to as colored or even as "red." Yet there does not seem to be any obvious sense in which "a giraffe" would be described as "brief." If we accepted the sensibility of a "red speech," without taking into account the new sense in which "red" was being used, then it would be necessary to place a descending line from the "red" node to the "speech" node in the graphic representation. But this step violates the fundamental hypothesis of sense value theory. In this case it is easy to see that the hypothesis is correct and that it is only apparently violated because "red" is being used in two senses. The resolution of the apparent difficulty in sense value terms is to say that there are at least two senses of "red"--"red 1" (color) and "red 2" (politics). Each of these predicates could then be placed in its appropriate tree location.

Let us consider a specific set of assertions, their possible tree

representations, the automatic detection of an equivocation, and an approach to the automatic resolution of the equivocation. Some of the terms in the example on page will be used to show how the problem of equivocation may result in invalid inference. The individuals to be considered are:

Socrates = S

The number 2 = N

A building = B

The predicates are:

Interesting = I

Rational = R

Tall = T

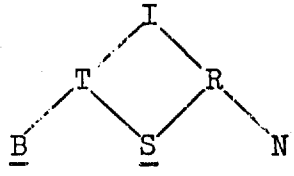
The possible predications, the only ones we are likely to encounter in sensible text, are:

S - I S - T S - R

N - I N - R

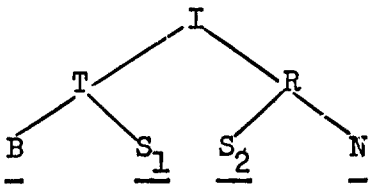
B - I B - T

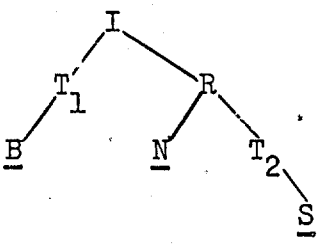
A graphic representation of the sense relationships, ignoring equivocation, is:

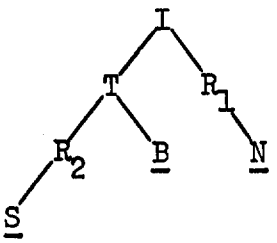


But this representation violates the basic assumption of sense value theory; two descending lines enter node S. Therefore, we automatically have evidence of an equivocation. There are three terms that, if regarded

as equivocal, can resolve the difficulty. These terms--S, R, and T--lead to three possible graphic solutions consistent with sense value theory:







It is intuitively obvious that the first two representations are incorrect because "Socrates" and "Tall" have not been used equivocally in these assertions. That the third representation, which regards "rational" as equivocal, is indeed correct can, however, be concluded on non-intuitive grounds. There exist both economic and aesthetic criteria that lead to a correct conclusion about which term is equivocal, and these criteria can be automated. Thus, consider the problem of adding new terms to each of the structures. If we wanted to add "Aristotle" or any other person to the first representation, it too would have to be regarded as equivocal since both "rational" and "tall" may be predicated of "Aristotle." If, on the other hand, we wanted to add a predicate such as "heavy" or "colored" to the second representation, then both of these terms would have to be made equivocal. It is only

the third representation that can accommodate both additions without increasing the number of theoretically necessary equivocations.

It is possible to formulate appropriate algorithms for automatically detecting and resolving equivocation in a corpus. The algorithm would assume that all linguistic work at levels of language lower than the level of sense would be supplied--that is, at the levels of syntax and spelling. Thus a computer program for detecting and resolving equivocation on the basis of sense value theory would assume an input of individual predicate pairs distilled from the sentences of a corpus by a previous syntactic processor.* The program would then detect any violations of the sense value hypothesis. This function could be done by producing a machine structure analogous to the graphic representations and checking for multiple descending entries into a node. Such a representation is perhaps most conveniently developed in a list processing system. Once a tree violation had been detected, the rule of economy could be used for the resolution of equivocation. That is, the term that produces the smallest number of entries or equivocations in the tree representation when regarded as two terms is interpreted as equivocal.

In addition to developing a partial model of an automatic system that can detect and correct equivocations by using sense value theory,

---

*Schemes for handling multitermed predicates have also been developed. One way to deal with an N-termed predicate is as N single-termed predicates. Thus the analysis of relations involving any number of individuals would be possible in principle with a system using individual predicate pairs as input.

it would also be desirable to verify the theory and its applicability empirically. The essential questions are whether the basic hypothesis of the theory as outlined is correct for a substantial corpus of text or sense value judgments and whether the economy criterion for resolving equivocation produces accurate results. Since syntactic preprocessing is assumed for this partial model, experimental inputs can as well be developed from judgments about the sensibility of individual predicate pairs rather than from an extensive search of an information corpus.

DISTRIBUTION LIST

Receipient	Copies
Office of the Assistant Secretary of Defense (Research and Engineering) ATTN: Technical Library Room 3E1065, The Pentagon Washington 25, D. C.	1
Chief of Research and Development Department of the Army Washington 25, D. C.	2
Chief, U. S. Army Security Agency ATTN: ACofS, G4 (Technical Library) Arlington Hall Station Arlington 12, Virginia	2
Deputy President, U. S. Army Security Agency Board Arlington Hall Station Arlington 12, Virginia	1
Commanding General, U. S. Army Materiel Command ATTN: R&D Directorate Washington, D. C. 20315	2
Commanding Officer U. S. Army Combat Developments Command ATTN: CDCMR-E Fort Belvoir, Virginia	1
Commanding General U. S. Army Combat Developments Command Communications-Electronics Agency Fort Huachuca, Arizona	1
Commanding Officer U. S. Army Engineer Research & Development Laboratories ATTN: STINFO Branch Fort Belvoir, Virginia	2

Receipient	Copies
Commanding Officer U. S. Army Nuclear Defense Laboratory ATTN: Library Edgewood Arsenal, Maryland 21010	2
Commandant, U. S. Army Air Defense School ATTN: Command & Staff Department Fort Bliss, Texas	1
Rome Air Development Center ATTN: RAALD Griffiss Air Force Base New York	1
Systems Engineering Group (SEPIR) Wright-Patterson Air Force Base, Ohio 45433	1
Air Force Cambridge Research Laboratories ATTN: CRXL-R L. G. Hanscom Field Bedford, Massachusetts	2
Electronic Systems Division (AFSC) Scientific & Technical Information Division (ESTI) L. G. Hanscom Field Bedford, Massachusetts 01731	2
USAEL Liaison Officer Rome Air Development Center ATTN: RAOL Griffiss Air Force Base, New York 13442	1
Director, U. S. Naval Research Laboratory ATTN: Code 2027 Washington, D. C. 20390	1
Commanding Officer & Director U. S. Navy Electronics Laboratory ATTN: Library San Diego 52, California	1

Receipient	Copies
AFSC Scientific/Technical Liaison Office U. S. Naval Air Development Center Johnsville, Pennsylvania	1
Commanding Officer U. S. Army Electronics Research & Development Activity ATTN: AMSEL-RD-WS-A White Sands, New Mexico 88002	1
Commanding Officer U. S. Army Electronics Research & Development Activity ATTN: Technical Library Fort Huachuca, Arizona	1
Director, Monmouth Office U. S. Army Combat Developments Command Communications-Electronics Agency Fort Monmouth, New Jersey	1
Commanding General U. S. Army Electronics Command ATTN: AMSEL-CM Fort Monmouth, New Jersey	1
Director, Material Readiness Directorate Headquarters, U. S. Army Electronics Command ATTN: AMSEL-MR Fort Monmouth, New Jersey	1
Marine Corps Liaison Office U. S. Army Electronics Laboratories ATTN: AMSEL-RD-LNR Fort Monmouth, New Jersey	1
AFSC Scientific/Technical Liaison Office U. S. Army Electronics Laboratories ATTN: AMSEL-RD-LNA Fort Monmouth, New Jersey	1

Receipient	Copies
Director U. S. Army Electronics Laboratories ATTN: Logistics Division Fort Monmouth, New Jersey 07703 MARKED FOR: Mr. Lorenz Sarlo	9
Director U. S. Army Electronics Laboratories ATTN: AMSEL-RD-DR Fort Monmouth, New Jersey	1
Director U. S. Army Electronics Laboratories ATTN: Technical Documents Center (AMSEL-RD-ADT) Fort Monmouth, New Jersey	1
Director (Record Copy) U. S. Army Electronics Laboratories ATTN: AMSEL-RD-ADO-RHA Fort Monmouth, New Jersey	1
Director U. S. Army Electronics Laboratories ATTN: AMSEL-RD-NP-2 Fort Monmouth, New Jersey	1
Commander, Defense Documentation Center ATTN: TISIA Cameron Station, Building 5 Alexandria, Virginia 22314	10
NASA Representative Scientific and Technical Information Facility P. O. Box 5700 Bethesda, Maryland 20014	1
Director U. S. Army Electronics Laboratories ATTN: AMSEL-RD-X Fort Monmouth, New Jersey	1

Receipient	Copies
Director U. S. Army Electronics Laboratories ATTN: AMSEL-RD-G (Mr. Hennessy) Fort Monmouth, New Jersey	1
Director U. S. Army Electronics Laboratories ATTN: AMSEL-RD-X (Mr. Jack Benson) Fort Monmouth, New Jersey	1
Headquarters, Aeronautical Systems Division Air Force Systems Command, USAF ATTN: ASRCEM-1 (Mr. Thompson) Wright-Patterson Air Force Base, Ohio	1