AD 636275

MAC-TR-29 (THESIS)

SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS BETWEEN DOCUMENTS

by

Evan Leon Ivie

CLEARINGHOUSE
FOR FEDERAL SCIENTIFIC AND
TECHNICAL INFORMATION
Hardcopy | Microfiche
$6.00 | $1.25 | 243 pp
ARCHIVE COPY

DDC
AUG 8 1966
B

June 1966

Project MAC
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Massachusetts Institute of Technology
Project MAC
545 Technology Square
Cambridge, Massachusetts
02139

Work reported herein was supported in part by Project MAC, an M.I.T. research project sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Nonr-4102(01). Reproduction of this report, in whole or in part, is permitted for any purpose of the United States Government.

Government contractors may obtain copies of this report from the Defense Documentation Center, Document Service Center, Cameron Station, Alexandria, Virginia 22314. Orders will be expedited from DDC if placed by your librarian, or some other person authorized to request documents.

Other U.S. citizens and organizations may obtain copies of this report from the Clearinghouse for Federal Scientific and Technical Information (CFSTI), Sills Building, 5285 Port Royal Road, Springfield, Virginia 22151.

ACCESSION for
CFSTI WHITE SECTION ☑
DDC BUFF SECTION ☐
UNANNOUNCED ☐
JUSTIFICATION .....
.....
BY .....
DISTRIBUTION/AVAILABILITY CODES
DIST. AVAIL. and/or SPECIAL
1

SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS BETWEEN DOCUMENTS

by

Evan Leon Ivie

B.S., Brigham Young University
(1956)

B.E.S., Brigham Young University
(1956)

M.S., Stanford University
(1957)

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May, 1966

Signature of Author Evan L. Ivie
Department of Electrical Engineering, May 20, 1966

Certified by Robert M. Fano
Thesis Supervisor

Accepted by Thomas S. Gray
Chairman, Departmental Committee on Graduate Students

SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS BETWEEN DOCUMENTS

by

Evan Leon Ivie

Submitted to the Department of Electrical Engineering on
20 May 1966, in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

ABSTRACT

In this thesis a new type of information retrieval system is suggested which utilizes data of the type generated by the users of the system instead of data generated by indexers.

The theoretical model on which the system is based consists of three basic elements. The first element is a measure of the relatedness between document-pairs. It is derived from information theory. The second element is a definition of what constitutes a set (cluster) of inter-related documents. This definition is based on the measure of relatedness. The last element is a procedure which transforms a request for information into a cluster of answer documents.

Requests are made by designating one or more documents to be of interest and perhaps some to be of no interest. The requestor can continue to interact with the procedure as it locates the answer cluster by specifying as interesting or not interesting other documents which are presented to him. The answer cluster which is generated is automatically made as small (specific) or as large (general) as is desired, depending on the initial request and the subsequent interactions.

An experimental system was developed to test the model in a realistic environment. It was programmed for the Project MAC time-sharing system and utilized the physics data file of the Technical Information Project. Citations were used as the data base for the measure of relatedness. A file structure and retrieval language were designed which allowed close man-machine coupling.

Experiments were conducted which compared the clusters of documents produced by the experimental system with various sets of documents of known mutual pertinence. These sets included bibliographies from review articles, subject categories, and sets of documents found to be of interest to selected users of the system. It was found that between 60-90% of the documents of known pertinence were included in the corresponding clusters. Ways of improving this retrieval efficiency even further are suggested.

Thesis Supervisor: Robert M. Fano

Title: Ford Professor of Engineering

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to Professor Robert M. Fano who laid the foundation for this thesis, and who, as thesis supervisor, supplied continuing guidance and enthusiasm throughout the course of the research.

I also wish to thank Dr. Myer M. Kessler and Professor Joseph Weizenbaum for their assistance and suggestions as readers. Dr. Kessler was also Director of the Technical Information Project where I worked as a research assistant and supplied much help and encouragement in that capacity.

Members of the staff of Project MAC and the Technical Information Project were also of significant help, particularly William D. Mathews, who assisted with the programming.

Thanks are extended to Dr. Robert E. Richardson and Dr. Wallace G. Clay of Lincoln Laboratory who aided in the evaluation of the experimental system.

Finally and most gratefully I am indebted to my wife for typing the drafts and final manuscript of the thesis and for her unfailing support throughout the doctoral program. Also assisting in proofreading and in less tangible ways were our children.

TABLE OF CONTENTS

PART I - INTRODUCTION

Chapter I - Background 13
1.1 Introduction 13
1.2 Areas Needing Improvement. 13
1.21 Closer Man-System Coupling 14
1.22 More Flexibility in Requests 15
1.23 Physical Barriers. 15
1.24 Quality of Selection Information 16
1.25 Restrictive Classification Model 17
1.26 Need for Dynamic Indexing. 18
1.3 Evaluation of Previous Efforts 19
1.31 Hardware Developments. 19
1.32 Indexing Methods and Models. 21
1.33 New Bases for Selection Information. 22
1.34 Measures of Relevance. 24
1.35 Automatic Classification and Clumping Experiments.25
1.36 Systems Evaluation 27

Chapter II - Objective of This Project 29
2.1 Brief Description of Project Objective 29
2.2 Value for Usage Information. 31
2.21 Objections 31
2.22 Supporting Arguments 34
2.23 Collecting Usage Information 36
2.3 The Purpose of Measures of Relatedness 37

PART TWO: THEORETICAL DEVELOPMENT

Chapter III - Measures of Relatedness. 40
3.1 Sample Space 40

3.2 Criteria for Selecting a Measure of Relatedness. 40
3.3 Selection of a Measure 44
3.4 Practical Considerations 46
3.5 Characteristics of the Measure for Document Pairs. . . .47
3.6 Document Networks. 50

Chapter IV - Document Clusters 56
4.1 Local Maximum Clusters 56
4.2 Subset Clusters. 61
4.3 Finding Subset Clusters. 65
4.31 Locating Splits. 66
4.32 Forming Kernels. 69
4.4 Biased Clusters. 70
4.5 Final Cluster Decision 73

Chapter V - Search Procedure 77
5.1 Definitions. 77
5.2 Attributes of a Good Clustering Procedure. 79
5.3 Description of Procedure 80
5.4 Earlier Procedures 85
5.5 Analysis of Procedure. 91
5.51 Request Satisfaction 91
5.52 Request Modification 98
5.53 Convergence. 103
5.54 Minimum Number of Iterations 105

PART THREE: EXPERIMENTAL SYSTEM

Chapter VI - Computational Facilities and Data Base. 108
6.1 Computational Facilities 108
6.2 Data Base. 111
6.21 Document Collection. 111
6.22 Partitions 114

Chapter VII - File Structure 120
7.1 Description and Arrangement of Data. 120
7.2 Types of Files 124
7.21 Raw Data File. 124
7.22 Inverted Files 125
7.23 Linkage Files. 128
7.24 Request - Answer File. 130
7.3 Storage Systems. 130
7.31 Storage Space Required 131
7.32 Processing Time. 132
7.33 Updating and Editing 133
7.34 Flexibility and Compatability. 135
7.4 Selection of Storage System. 136
7.5 High Speed Storage Structure 137

Chapter VIII - Interaction Language. 140
8.1 Background to Language 140
8.11 Design Objectives of Language. 140
8.12 Example of Language. 142
8.2 Description of Language. 144
8.21 Finite State Machine Description 144
8.22 Backus Normal Description. 148
8.23 Equivalence of Descriptions. 152
8.3 Interpretive Algorithm 152
8.31 Vocabulary and Literals. 154
8.32 Available Functions. 154
8.33 Data Generated 157
8.34 Request Structure. 157

PART FOUR: RESULTS AND CONCLUSIONS

Chapter IX - Experimental Results. 166
9.1 Cluster Parameters 166
9.2 Cluster Composition. 173

9.3 Comparison to Bibliographies 176
9.31 Bibliography 1 177
9.32 Bibliography 2 181
9.33 Bibliography 3 185
9.4 Comparison to Categories 190
9.41 Physical Review Category 190
9.42 Physics Abstracts Category 196
9.5 User Experience. 196
9.51 Simple Request 200
9.52 Expand Extensive Bibliography. 204
9.6 Summary of Results 215

Chapter X - Conclusions. 218
10.1 Evaluation of Experimental System 218
10.11 MAC Time-Sharing System 218
10.12 T.I.P. Document Collection. 221
10.13 Partitions. 222
10.14 Storage Structure 222
10.15 Retrieval Language. 223
10.2 Evaluation of Procedure 224
10.3 Evaluation of System. 225
10.4 Suggestions for Further Research. 228
10.41 Data Base and Structure 228
10.42 Procedure and Language. 229
10.43 Theoretical Problems. 231

Appendix A - Measures of Relatedness 234

Bibliography 236

LIST OF FIGURES

3.1 Range of measure of relatedness 48

3.2 Range of approximation to measure of relatedness. 48

3.3 Range of measure of relatedness for test data 49

3.4 Revised range of measure for test data. 50

4.1 Network with overlapping clusters 58

4.2 Network with a document (x_1) in no cluster. 59

4.3 Network with hierarchal cluster structure. 60

4.4 Cluster containing unrelated subsets. 61

4.5 Network containing no subset clusters 64

5.1 Ambiguous request 79

5.2 Overall flow chart. 81

5.3 Initialization. 86

5.4 Condition 1 and deletions 86

5.5 Condition 2 and additions 87

5.6 Phase III and other tests 88

5.7 Example showing why non-pertinent documents should not all be grouped into one cluster. 90

5.8 Example of difficulty with forming clusters around non-pertinent documents. 91

5.9 Local maximum cluster not accessible to procedure. 92

5.10 Local maximum cluster not accessible to procedure 92

5.11 Possible additions to S 93

5.12 Network where it does not matter which document is added to S first 94

5.13 Network showing that the procedure must be allowed to delete as well as add. 95

5.14 Network illustrating the difficulties involved in knowing which document to add to S on a given iteration. . . . 96

5.15 Tree illustrating the possible additions to S for the network and request of Fig. 5.14. 97

5.16 Network with x_1 and x_2 not in the same cluster. 100

5.17 Network with x_1 and x_2 in the same cluster. 100

5.18 Network for proof of theorem 101
5.19 Network which may cause a procedure to cycle 104
6.1 Project MAC equipment configuration. 110
6.2 Significant parameters of MAC system 110
6.3 Journals covered by the physics periodical file of the Technical Information Project 112
6.4 Example of the information available on a given article. . .113
6.5 Parameters of T.I.P. data file 113
7.1 Example of tree-like structure of data 121
7.2 Example used to show physical order given the data 123
7.3 Linear arrangement of data in Fig. 7.2 123
7.4 Polish prefix notation 124
7.5 Structure of raw data file 124
7.6 Percent of storage occupied by each information type 125
7.7 Structure of inverted author file. 125
7.8 Storage requirements for inverted files 126
7.9 Storage required for inverted author file. 127
7.10 Structure of linkage file. 128
7.11 Table of linkage file sizes for vol. 128 of the Physical Review. 129
7.12 Structure of request-answer file 130
7.13 Comparison of storage requirements for the four types of data systems 132
7.14 Average processing time required to find a cluster of 20 documents for the four types of storage systems 133
7.15 Processing time required to update a file of 2000 articles with 335 new articles for each of the four storage systems 135
8.1 Example of possible user interaction with data using retrieval language. 143
8.2 Finite state machine description of syntax of retrieval language. 145
8.3 Finite state diagram for the table of Fig. 8.2. 146
8.4 Classes of input symbols 146
8.5 Example of statement with acceptable syntax and statement with unacceptable syntax. 149

8.6 Backus normal statements describing syntax of language . . .149
8.7 Backus normal statements describing vocabulary of language .150
8.8 Backus normal description of literals. 151
8.9 Rules for transforming Backus normal statements to finite state diagram 152
8.10 Outline of steps proving equivalence of Backus-normal and finite state description. 153
8.11 File structure of data in scratchpad storage 157
8.12 Valid types of objects for each preposition class. 159
8.13 Types of nouns that each class of prepositions can modify. .164
9.1 Distribution of cluster size for 490 clusters. 167
9.2 Distribution of clusters by bias for 275 clusters. 168
9.3 Plot of average cluster size versus bias for 340 clusters. .168
9.4 Percent of deletions occurring in each quartile of the clustering process. 169
9.5 Example of clusters which result when documents are specified as non-pertinent. 171
9.6 Diagram of relationship of clusters of Fig. 9.5. 172
9.7 Test of request ambiguity. 172
9.8 Title-word frequency counts for six clusters 174
9.9 Author frequency counts for three clusters 175
9.10 Citation frequency counts for three clusters 178
9.11 Articles in the October 1965 Issue of the IEEE Proceedings that have 10 or more references to the T.I.P. file. . .177
9.12 The sets of articles included in the clusters for Bibliography 1. 178
9.13 List of the answer clusters formed for Bibliography 1. . . .178
9.14 Generalizations suggested by the results of Fig. 9.13. . . .179
9.15 Sketch showing the relationship of the answer clusters of Bibliography 1. 179
9.16 Titles of articles in the A_1 cluster 182
9.17 The sets of articles included in the clusters for Bibliography 2. 183
9.18 List of the answer clusters formed for Bibliography 2. . . .183
9.19 Generalizations suggested by the results of Fig. 9.19. . . .184

9.20 Relationship of answer clusters of Bibliography 2. 184
9.21 The sets of articles included in the clusters for Bibilography 3. 186
9.22 List of answer clusters formed for Bibliography 3. 187
9.23 Relationship of answer clusters of Bibliography 3. 188
9.24 The sets of articles included in the clusters for Category 1. 192
9.25 Answers to selected requests for Category 1. 193
9.26 Relationship of answer clusters for Category 1 194
9.27 Comparison of the three clusters formed for Category 1 . . .195
9.28 The sets of articles included in the clusters for Category 2. 197
9.29 Answers to selected requests for Category 2. 198
9.30 Relationship of answer clusters for Category 2 199
9.31 Sets of articles included in the clusters for Physicist 1. .201
9.32 Answers to selected requests for Physicist 1 202
9.33 Relationship of answer clusters for Physicist 1. 203
9.34 21 articles in Langmuir Probe that are in T.I.P. file. . . .205
9.35 Publication year distribution of initial Langmuir Probe bibliography. 205
9.36 Title word distribution for the 112 titles of the initial Langmuir Probe bibliography 206
9.37 Distribution of articles with various bibliographic coupling strengths. 208
9.38 The sets of articles included in the cluster for Langmuir Probe Bibliography 208
9.39 Answers to selected requests for Langmuir Probe Bibliography. 209
9.40 Relationship of clusters for Langmuir Probe Bibliography . .210
9.41 Langmuir Probe papers evaluated by physicist 212
9.42 Comparison of results of seven search strategies 214
9.43 Summary of the experimental results of Sections 9.3-5. . . .216
9.44 Average percent of bibliography articles added during each quartile of the clustering process 217

PART ONE: INTRODUCTION

This thesis is divided into four parts. In this part we introduce the project by describing results of related work and by discussing the objectives of the research. In Part Two the theoretical model on which the project is based is presented. Part Three contains a description of the experimental system which was developed to test the model. In the final part we present the experimental results and the conclusions about the theoretical model that can be drawn from them.

CHAPTER I

BACKGROUND

1.1 Introduction

In a pioneering article written at the close of World War II, Dr. Vannevar Bush, Director of the Office of Scientific Research and Development, called on scientists to redirect their energies to creating "a new relationship between thinking man and the sum of our knowledge." He noted that "our methods of transmitting and reviewing the results of research are generations old and by now are totally inadequate."[10]

His challenge to mechanize and streamline the library process has been accepted by numerous groups in the intervening twenty years. A large number of devices have been developed which mechanically or electronically select information from a store. Methods of automatically indexing, classifying, and abstracting documents have been devised. A myriad of other disciplines have been called in for assistance.

Before attempting to review and evaluate this activity, it is extremely important that the implied "inadequacies" of traditional library methods be clearly defined. Only then can one hope to determine the effectiveness of any given approach in resolving these problems.

1.2 Areas Needing Improvement

Six general aspects of library systems have been chosen as important areas which need improvement and which appear to be amenable to improvement through some type of mechanization. Most information

storage and retrieval projects have had as their stated or implied goals one or more of these objectives.

1.21 Closer Man-System Coupling

In many cases a user who comes to an information system cannot state precisely what he wants. He has a very real need for information, but he cannot define exactly what that need is verbally. In other cases a user can accurately specify his interests but changes his mind as to what he wants when he finds that there are too many or too few articles which satisfy the request.

Unfortunately most systems (automatic and manual) are designed for that rare individual who knows exactly what he wants and what the stack contains. In these systems there is a clear demarkation between request specification by the user and answer presentation by the system.

A much closer coupling of man and system is generally needed so that each can contribute to the best of his (its) ability at each step in the search. For example, the system might help the user in formulating the request by noting with each change in the request the probable number of documents in the final answer, by presenting representative documents for evaluation, and by ranking the output according to degree of relatedness. The user, on the other hand, could help the system find the desired answer by catching and correcting possible misunderstandings of the request as early in the search as possible, by narrowing or broadening the request if the size of the expected answer becomes too large or too small, and by continually refining the request based on the information supplied by the system.

1.22 More Flexibility in Requests

Even if it is assumed that a user can adequately specify his interests, there is still the difficulty of matching his request vocabulary with the vocabulary of the indexer. Perhaps the user is looking for books on "information retrieval" but fails to realize that the classifier posted such books under "documentation". Of course, the classifier may have foreseen this difficulty and placed a "see" card under information retrieval. However, this does not always occur.

Another basic problem is faced by the person who knows a given paper or a given author of interest but is forced to translate this knowledge into a set of descriptors instead of being able to feed it in directly as a request.

More flexibility is needed in the allowable vocabulary, language structure, and type of information which can be specified in a request.

1.23 Physical Barriers

The mere physical separation of the user from the library presents a barrier that has a greater impact than we may realize. This is also true of the separation of the card file from the stacks. Evidence of the importance of this factor is found in the popularity of small special collections distributed throughout a large organization and in the personal libraries maintained by most research workers.

There is also the time barrier. If a person could get an answer to his problem in five minutes, he might be interested. Whereas he might decide to bypass the problem if it takes one-half hour or more. A third barrier is cost. This factor is not a direct consideration to the user in most cases because no direct fee is levied for use of a library.

1.24 Quality of Selection Information

All libraries provide the user with certain types of information which help him to select from the total store those books which are of interest to him without having to scan the text of each book. Even those libraries which cater to the browser generally arrange books by content on the shelves and place the spine out so that the title and author can be seen at a glance.

There are at least three important factors which must be considered in the generation of selection information for a given document.

1. The actual contents of the document.
2. The collection in which the document will reside.
3. The needs and characteristics of the user population serviced by the collection.

If the only factor to be considered in indexing were the contents of the document, then a valid method for indexing would be to have each author, as the final authority on what the document contains, index it. However, libraries have found that the other two factors are also important and that an author cannot be expected to be familiar with each library and each user population that might have his book or article.

The approach used by conventional libraries is to rely on an indexer or classifier to generate the selection information needed. This type of individual is usually an expert on the contents of the library collection, but knows much less about the first and third factors. He usually has about 10-15 minutes' time to determine what the author of the document has said and predict the types of users this information will be of interest to (through the categories selected);

all this with little direct involvement in the field or area in question. The amazing part about the whole process is that an indexer can sometimes come up with a sketchy, but fairly useful portrayal of the document.

An additional problem is that much of the literature (periodicals, technical reports, etc.) never even receives the attention of an indexer.

1.25 Restrictive Classification Model

Even if the classifier were able to determine the exact contents of a document, he would still find difficulty in fitting his findings into the rigid classification systems currently in use (Dewey Decimal, Library of Congress, etc.).

First, the classifier is allowed only a yes-no type of response. Either the document is placed in a given category or it is not--there is no middle ground, no partial relationship.

Next there is the "broken relationship" problem inherent in hierarchal classification structures. No matter where a category is placed in the hierarchy tree, there are related fields to which it cannot be adjacent. For example, if the history of physics is placed in the science area, it loses its connection to history and vice-versa. This problem is only partially alleviated by the "see" and "see also" artifices.

Third, there is the difficulty encountered in changing a classification structure to fit with our current body of knowledge. This involves considerable expansion and contraction of areas along with insertion of entirely new fields and the deletion of obsolete ones. The old classification frame-ork eventually becomes so strained in certain areas that

there is danger of collapse.

Each of these difficulties encountered in the classification of documents generates a corresponding difficulty for the user. V. Bush described the use of a classification system in this way.

> "...information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome. Having found one item, moreover, one has to emerge, from the system and re-enter on a new path."[10]

1.26 Need for Dynamic Indexing

Consideration of the problem of indexing leads one to the conclusion that there is no intrinsic content to a document which, when once properly characterized by an appropriate set of words or phrases, is then adequately indexed for all situations and all users. In reality the depth and type of indexing needed depends both on the characteristics of the collection in which the document is imbedded and on the interests of the user population to be serviced by the collection at the time.

Once this point is conceded then it becomes apparent that the way a document is indexed must change as the collection and user population vary. One of the major drawbacks of conventional indexing methods is that in practice they are static. A document, once indexed, is almost never re-indexed. Indeed some people believe that a properly indexed document should never need re-indexing. R. A. Fairthorne claims the following--

"We have to assume that a classifier can decide that a text is relevant to a topic in such a way that, apart from blunders, neither future development nor decisions elsewhere shall compel revision. Future developments certainly should not upset any decision about relevance; if an item is relevant to some topic, it will always be relevant, though the relevance may become unimportant and new relevancies may be added."[17]

The case for dynamic indexing was clearly presented by M. M. Kessler:

"Indexing must be fluid and dynamic, reflecting the changing needs of society and the contributions of new insights. It is most unlikely that anybody, be he expert scientist or expert indexer, can read a given paper at a given time and see enough of its implications to classify it once and for all. If this philosophy of classification were accepted, as it now is, the resulting system would impose such a rigidity upon the flow of information that the working scientist would be forced to ignore it."[26]

1.3 Evaluation of Previous Efforts

It would be impossible to describe all of the work which has been undertaken in the field of information retrieval and documentation in the last 20 years. What will be attempted here is an analysis of certain representative efforts in each of six broad areas.

1.31 Hardware Developments

Many interesting machines have been developed for use in information processing (Rapid Selector, Peekaboo, Zator, Walnut, Minicard, general purpose computers, etc.). Instead of discussing the specific capabilities of these machines, let us note some of the general trends in hardware development which promise to have the greatest impact on

information retrieval.

The first would be the development of multiply-accessed (time-sharing) computers.[21] A research worker with a connection to such a computer would be able to query a large central store of information directly from his office, laboratory, or home and receive an almost immediate response. This is in contrast to the batch-processing computer which processes requests in groups at a central location and usually involves delays in response of from several hours to several days. A brief description of a particular time-sharing system (the one used by this research project) can be found in Sec. 6.1.

A system of users interacting with a large central information store through a time-shared computer offers another important capability that might be overlooked. Not only can the user obtain information from the system, but the system can also monitor the user. This monitored usage data could be collected at little or no inconvenience to the user. It would complete the information loop with feedback from the user continually modifying and improving system performance.

Another significant hardware advancement is the development of larger and larger mass memories. It is estimated that all of the textual information in the 20 million documents in the Library of Congress could be stored in a 10 trillion-bit (10^{13}) memory. Current random access devices store 10^9 - 10^{10} bits, while large magnetic tape installations have a capacity of 10^{11} bits. Random access storage devices have been announced in the 10^{12} bit range. It would appear that continued progress may soon eliminate storage capacity as a limiting factor in the mechanization of large information retrieval systems.

A parameter closely related to memory size is access time. Typical access times to any part of a 10^9-bit file on a random access disc are currently 100 ms. The real problem is in knowing which part of the file to read. Perhaps associative memories, complete file inversion, or some other artifice will resolve this problem.

1.32 Indexing Methods and Models

As important as hardware developments are, V. Bush pointed out an even more basic problem.

> "The real heart of the matter of selection, however, goes deeper than a lag in the adoption of mechanisms by libraries, or a lack of development of devices for their use. Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing."[10]

The 'systems of indexing' to which Bush referred are, of course, the traditional subject catalog and classification schemes still in use (Universal Decimal, Library of Congress, etc.). Some of the drawbacks of these classification systems were discussed in Section 1.25.

Beginning about 1950 efforts were made to replace these conventional classification methods. One result was "coordinate indexing."[47] In coordinate indexing documents are assigned Uniterms or descriptors (usually single words). These descriptors are given no hierarchal or other structure. A request consists of certain descriptors connected by the logical and-or-not operations.

Coordinate indexing eliminated many of the difficulties encountered in hierarchal classifications and subject catalogs. However, its strength was also its shortcoming. The elimination of all order and structure from the descriptors introduced many 'false drops'. For

example, a hypothetical user looking for papers on the causes of blindness in Venice might also retrieve articles on the design of Venetian blinds. To reintroduce that which was lost by eliminating descriptor context and order, such features as role indicators were used.

Currently some workers in the field seem to be disenchanted with coordinate indexing and have shifted reluctantly back to the conventional classification methods.[16]

Another field of endeavor was in the modeling area. A number of models were proposed which described the indexing and retrieval functions. Unfortunately that was all that these models did - they provided an alternate way of describing an already familiar problem. No new insights were gained and no helpful procedures resulted.

1.33 New Bases for Selection Information

It has already been noted that all library systems depend on selection information (classification categories, subject headings, author indexes, etc.) to locate documents relevant to a particular request. Customary library practice is to depend on the indexer to produce this information. Section 1.24 outlines some of the difficulties inherent to this dependence.

Studies during the past eight years have been undertaken to see if selection information generated by indexers can be supplemented and perhaps replaced by that generated by the automatic processing of a document's contents.

At first simple methods of exploiting the information found in a document were tried. Permuted title indexes and citation indexes met with some success. In 1958 Luhn proposed automatic abstracting.[31]

This consisted of the selection of certain words as the keywords of a document based on their frequencies of occurrence. The sentences and/or phrases which contained these words were then extracted to form the auto-abstract of the document. The idea was then extended by Maron in 1961 to the automatic indexing of documents with the keywords extracted becoming the descriptors.[32,33]

Automatic indexing was about 50 % successful in assigning documents to the same categories that the human indexer did.[16] This mediocre showing can be attributed to the fact that machine indexing did not make use of the order, context, syntax and synonyms of the words extracted. This in essence is the same difficulty found in coordinate indexing. Some of the subsequent efforts at automatic indexing attempted to account for syntax, but this trail encountered the same massive obstacles that had already slowed progress in automatic language translation.

Thus after some initial success, the automatic generation of selection information based on document contents ran aground. One cannot dispute the fact that a description of the subject covered by the article is contained within the article. Just how one can capitalize on that knowledge is the problem. The needed information is there, but machines and indexers currently can extract only a part of it.

There is one notable exception to the above comments. The citations found in articles do not have the same type of synonym and syntax problems that textual material does. Thus selection information generated from citations has had considerable success for those bodies of literature which have a good citation base.[28]

A discussion of the user of a library as a source of selection

information will be postponed until Chapter II, since little, if any, prior experimental work has been done in this area.

1.34 Measures of Relevance

In conventional library systems documents are assigned to categories and subject headings on a yes-no sort of basis. Either the document is in the category or it is not--there is no middle ground. The restrictive nature of this type of arrangement was pointed out by Maron and Kuhns in 1960.[33] They proposed that an 8-value weighted indexing scheme be used to represent the degree to which a document is related to a term.

This idea was extended to thesauri by Stiles in 1961.[43] A traditional thesaurus allows terms to be listed as synonyms or antonyms but the degree of synonymity is left unspecified. Stiles proposed an association factor to represent the amount of synonymity between terms.

Numerous other 'measures of relevance' between the various entities of libraries have been proposed since. Some of the better known of these measures are tabulated in Appendix A. Unfortunately, there appears to be considerable confusion over exactly what these measures represent, and the use of the term 'relevance' would seem to add to this confusion.

Many documentalists now speak with some assurance about the amount (to 3 or 4 significant figures) of 'relevance' of a document to a category or to a request. The 'relevance ratio' is an accepted way to measure information retrieval system efficiency. All too often these comments leave one with the impression that there is some intrinsic meaning to a word or document which has now been quantitatively described,

when in reality all that has been accomplished is the invention of some type of frequency ratio.

In traditional library work confusion also appears to exist. Indeed the very idea of classification implies to some that there is some inherent content of a document which must be indexed. The already quoted comment by R. A. Fairthoren can be cited as an expression of the attitude of some classifiers.

> "Future developments certainly should not upset any decision about relevance; if an item is relevant to some topic, it will always be relevant, though the relevance may become unimportant and new relevancies may be added."[17]

Let us suggest that the intrinsic meaning or concept behind a word is a philosophical problem and cannot be dealt with operationally. Those aspects of a document which do not influence its environment (i.e. the library and the user) are of no practical significance because they cannot be observed, measured, or even proved to exist.

To avoid adding further to this misunderstanding we shall avoid the use of the word 'relevance' in the rest of this paper. The frequency ratios used by this project will be termed 'measures of relatedness'. It is hoped that this term is less loaded with connotations of intrinsic meaning.

1.35 Automatic Classification and Clumping Experiments

After automatic indexing was proposed for the assignment of documents to categories, it was only natural that the automatic determination of the categories themselves should be tried also. This was done initially by borrowing two techniques from mathematical psychology--factor analysis and latent class analysis. Factor analysis is used to

discover the underlying factors which account for the performance of a group of people to a battery of tests. Latent class analysis is a procedure used to divide a group of people into disjoint sub-groups on the basis of their responses to a questionnaire.

Latent class analysis for information retrieval has not yet been experimentally tested.[1,52] Borko's work with factor analysis was based on the occurrence of keywords in document abstracts.[6-8] A correlation matrix of keywords versus keywords was formed and was factor analyzed, resulting in categories which had some resemblance to those manually selected for the same corpus.

An even earlier attempt at automatic classification was tried by Needham and Parker-Rhodes in England.[38,39,41] They called it clumping and produced a heuristic procedure which selected clumps of documents from a file. Their work has been extended in this country by Dale[13] and also by Bonner.[5]

Since clumping is the most closely related endeavor to the objectives of this project of any to date, a slightly more extended description of the results will be given. A library collection is thought of as a network with the nodes representing documents and values assigned to the links (usually 0 or 1 only). This collection is partitioned into two subsets, A and B. The sum of the links internal to A is denoted by AA and the sum of the links internal to B is denoted by BB. The only other links in the network are those which cross from set A to set B. The sum of these links is designated AB.

A GR clump is defined as any set A which produces a local minimum of the function F(A).[13]

$$F(A) = \frac{AB}{AA + BB}$$

A more recent type of clump, the D clump, is defined as any set A which produces a local minimum of the function G(A).[12]

$$G(A) = \frac{AB}{\sqrt{(AA)(BB)}}$$

GR clumps are fairly easy to locate. Some additional restrictions must be placed on D clumps to make the definition useful since local minima of G(A) occur for quite unrelated sets of documents. The latest effort has been to find an initial set of items by some other method and then use the D-clump method to complete the set.

Both the automatic classification and the clumping experiments are designed so that all of the classifying and indexing would be completed before the requests are processed.

1.36 Systems Evaluation

The most widely accepted method of evaluating the performance of information retrieval systems is currently through the recall and relevance ratios.[45] The recall ratio is the percentage of relevant items that are actually retrieved and the relevance ratio is the percentage of retrieved items that are relevant.

In determining what is or is not relevant, recourse is usually made to an indexer or a user. Recent studies have shown that these people are able to agree among themselves as to how documents should be classified in at most 80% of the cases. This "failure" of humans to index consistently has led some to try to find better automatic "non-judgemental" standards on which to validate relevance.[16]

If the primary objective of a library is in serving a given user population, then it is difficult to imagine that there could be any

criteria for relevance other than one based on those users. If, on the other hand, the function of a library is to set up a universal classification system, then the user should certainly be eliminated as the standard on which system efficiency is evaluated.

The idea that the users of a system can "fail" in classifying a document implies an intrinsic content in documents which one or more of the users has not recognized. A more practical outlook in keeping with the arguments of Sec. 1.34 is that these differences in indexing are only the normal result of individual backgrounds and interests.

CHAPTER II

OBJECTIVE OF THIS PROJECT

2.1 Brief Description of Project Objective

Let us assume for a moment that we wish to design an information storage and retrieval system which is based on feedback from users. In this system each request for information is to consist of a set of one or more documents that the user has already found to be of interest and a second (possible empty) set of documents that he knows are not of interest.

The purpose of each interaction of a user with the system is to transform a request of this type into a partitioning of the total collection into two disjoint subsets--one containing all documents that are of interest to the user and the other containing those not of interest (the rest of the stack). This process is to be accomplished jointly by the user and the system.

The feedback which the system stores for use in answering future requests is to consist of these file partitionings. A measure of the relatedness between any two documents based on their usage and co-usage patterns as found in the partitionings is to be utilized to facilitate the request-to-answer transformation.

The document collection of such a system can be thought of as a network where each node represents a document and each link is given a value corresponding to the measures of relatedness between the two linked documents.

The objective of this research endeavor is to devise, test, and evaluate a procedure which will perform the transformation of request to answer partition for this type of retrieval system.

In the above discussion we suggested for purposes of illustration a retrieval system based on file partitionings which are generated by the users of the system. Partitioning information of this sort would not be available for documents that have just been added to a file. Indeed, such information is not readily available for any file of documents at the present time.

There are, however, some types of partitionings which are available. Take, for example, the citations in an article. The author of an article selects for citation certain documents that he feels are pertinent to the article he has written. In a sense he is a special type of user of the library and has created a meaningful partition of the file. Other types of partitionings of the file could also be suggested.

Usage information was selected for discussion here because it is an interesting and representative example of the larger class of partitioning information for which we propose to design a retrieval system.

In the remainder of this chapter and in the next chapter we will, therefore, continue to talk in terms of the partitionings generated by users. It should be understood, however, that the type of retrieval system to be developed need not be restricted to this single type of partitioning data.

In the next section we will present some arguments for and against information retrieval based on usage information. We will then discuss how usage information can best be represented and utilized.

2.2 Value of Usage Information

In the article already cited at the beginning of Chapter I, V. Bush suggested that an individual's personal information storage and selection system could be based on direct connections between documents instead of the usual connections between index terms and documents. These direct connections were to be stored in the form of trails through the literature. Then at any future time the individual himself or one of his friends could retrace this trail from document to document without the necessity of describing each document with a set of descriptors or tracing it down through a classification tree.[10]

In 1956 R. M. Fano suggested that a similar approach might prove useful to a general library. He proposed that "the concomitant use of documents by experts as evidenced by library records, and other similar joint events" might be a useful basis for document retrieval.[19,19] His proposal evoked a number of adverse comments, two of which will be quoted here.

2.21 Objections

A theoretical objection to basing retrieval on usage was raised by Y. Bar-Hillel.

> "A colleague of mine, a well-known expert on information theory, proposed recently, as a useful tool for literature search, the compiling of pair-lists of documents that are requested together by users of libraries. He even suggested, if I understood him rightly, that the frequency of such co-requests might conceivably serve as an indicator of the degree of relatedness of the topics treated in these documents.
>
> "I believe that this proposal should be treated with the greatest reserve. Although much less ambitious

than Taube's proposal of an association dictionary, it is in many respects strikingly analogous to it and shares its short comings. The fact that a co-requestedness chain of documents can be easily followed up by a machine is not in itself a sufficient reason for making the assumption that this relation might be a useful approximation to the important relation of dealing-with-related-topics between documents. And one can think of many other easily establishable relationships between documents that stand a better chance of being a useful approximation, e.g. co-occurrence of their references in reference lists printed at the end of many documents, co-quotation, and so on."[2]

The shortcoming of 'Taube's proposal' referred to in this quote is the familiar triangle argument.

"Knowing that 'a' and 'b' co-occur...and that 'b' and 'c' co-occur...what do we know about the connection between the 'ideas' 'a' and 'c'? Clearly, nothing definite whatsoever..."[2]

What Bar-Hillel says is true also of hierarchal classification systems where the adjacency of categories a and b and of categories b and c proves nothing about the relationship of a and c. It is true of any system consisting of a set of items and characteristics that cannot be described by some type of metric space.

On the other hand the fact that documents a and c are not related in every case when linked through a third document b is more of a hypothetical objection than a practical one. If, in fact, items with the a-c type connection are found to be related on the average much more frequently than items chosen at random, then the usefulness of this type of connection in document selection should not be overlooked.

A second objection to Fano's suggestion was raised by C. N. Mooers. It is a practical instead of a theoretical objection.

"To provide feedback for improving machine performance Fano and others have suggested the use of statistics of the way which people use the library collection. Though the suggestion points in the right direction, I think this kind of feedback would be a rather erratic source of information on equivalence classes, because people might borrow books on Jack London and Albert Einstein at the same time. Although this difficulty can be overcome, there is a more severe problem. Any computation of the number of people entering a library and the books borrowed per day, compared with the size of the collection shows, I think, that the rate of accumulation of such feedback information would be too slow for the library machine to catch up to and get ahead of an expanding technology."[34]

Mooers' objection assumes that the capability of accepting feedback from the user is to be superimposed on a conventional library structure and that it will have little net effect on the frequency of use of that library. Let us accept these assumptions for the moment and suggest some reasons why usage information would still prove profitable.

First, libraries might well find it helpful to share usage patterns and thereby increase the total information available to any one library. Second, the well used documents will have plenty of usage statistics and be well 'indexed', while unused books will have no statistics--a seemingly equitable arrangement. Third, even the information on one usage of a document may prove more valuable than the information supplied by the indexer of that document. Fourth, usage information is not purported to be a cure-all which will replace all of the current types of selection information. It is felt to be a supplemental source of selection clues which should grow in importance as more user feedback is collected.

Now let us return to the initial assumptions and note that the number of people who enter a library is by no means an indication of the amount of time spent in the study of printed material. It is merely an indictment of current library practices. If, in fact, information were made available to research workers right in their offices through the type of computer time-sharing system described in Section 1.31, then the amount of feedback available from users should radically change.

2.22 Supporting Arguments

Thus far in this section we have cited two early proposals that document selection be based on user feedback. We have quoted both a theoretical and a practical objection to such an approach and have attempted to answer these objections. Let us now turn to some of the positive arguments favoring user feedback which, to this author at least, are compelling reasons why document retrieval should be based on information from the user.

The first argument has already been alluded to in Section 1.26. In this section the need for dynamic indexing was observed. It was noted that it is impossible for an indexer to foresee all of the possible applications of a paper at any given point in that paper's history and especially not just after it is written.

To account for the changing relationships and new applications of papers in a collection, a library must be supplied with information. Such information regarding the changing nature of the corpus must come from the three participants in the library process--author, indexer, and user.

To require indexers to periodically re-index the collection would be financially impossible. Many libraries find it difficult to even initially index each incoming document.

The textual information placed in the document by the authors offers little help also. Take, for example, a research worker who publishes a new discovery. A terminology which eventually evolves to describe that discovery may be markedly different from the language of the initial paper. And it would be a rather momentous task to develop a thesaurus which could connect the groping language of the basic paper with the codified terminology which eventually results.

Thus, the user is left as the one participant in the library system who is continually interacting with the collection and could introduce dynamic indexing into the system.

Let us note at this point that citation information in newly added documents represents a specialized type of user information (the author acting as a user of the old file), and as such can act in the same way as usage information to give the system a changing indexing structure. Some other advantages of this source of indexing information were noted in Sec. 1.33.

The second argument in support of the utilization of user feedback concerns the quality of the indexing which results thereby. The advantage of having the indexing done by people actually immersed in a given research area can hardly be overemphasized. Hitherto neglected refinements and distinctions can be made, the structure of the field as the actual worker sees it can be established, and many unintentional blunders can be avoided.

It should be noted that the quality of indexing by usage is a controllable parameter. Take , for example, the users of articles in the Physical Review. This group of people represents a highly knowledgeable and motivated segment of the population which should be able to form valid links between documents. If, however, the quality of the resulting indexing is still insufficient, the system could be designed to accept feedback from only a segment of the population--say the faculty but not the students. This could even be made a parameter specifiable by the user so that he could use the feedback from that segment of the population which most closely fitted his own background.

A third reason for indexing by user feedback is that it may be possible to do it as a by-product of normal library use and thus avoid, to some extent, the high cost of indexing which currently burdens a library.

2.23 Collecting Usage Information

Let us now discuss the problem of how the intellectual decisions needed from the user can best be obtained. The sets of citations found in articles form one readily available source of sets of documents that have been judged mutually pertinent. The data used by the experimental portion of this project was taken from this source. (See Sec. 6.22)

Let us consider for a moment whether a retrieval system could be designed which was based on usage data of the type described in Sec. 2.1. One major difficulty would be to devise some way of encouraging the user to supply the system with the data needed. Some possible ways this might be accomplished are the folowing:

1. The user finds that the system automatically disseminates to him new articles of interest if he has provided profiles of his interests in the form of sets of papers of known interest.
2. The user finds that in interacting with the retrieval program he converges on papers of interest more rapidly if he tells the system whether each paper presented is of interest or not.
3. The user contributes sets of related papers to the system because he wishes to improve its usefulness to himself and others.
4. Certain users are provided monetary remuneration for supplying the system with sets of related documents.

2.3 The Purpose of Measures of Relatedness

The next question that arises after one has accepted the idea that information selection might appropriately be based on some type of usage data concerns the form that this data should be expressed in. One might propose that each usage set be treated the same way as a subject heading or descriptor set with its label being the name of the user that generated the set. Under this scheme one might retrieve all of the papers of interest to a given user or all of the papers which have been found of mutual interest with a selected paper. Indeed the ability to answer these types of questions is a valid capability to equip a retrieval system with.

However, there are some significant differences between the sets of papers generated by users and the sets of papers generated by some type of indexing scheme. First, there is the fact that any given paper occurs in, at most, only a handful of indexing categories,while it might

possibly occur in a very large number of user sets. Second, there can be any number of user sets centering around a given area of research, but this area would be normally covered by only one subject category. Third, usage sets would be continually added to the system, but new categories would be added infrequently.

All this adds up to the fact that users who attempt to extract information from usage files with normal matching techniques will probably be overwhelmed with the non-uniform, massive, fluctuating nature of this type of data.

Some type of statistical measure is needed which will combine and summarize the results of many user interactions. The specific characteristics which this measure should have are discussed in Chapter III.

PART TWO: THEORETICAL DEVELOPMENT

The three chapters of this part describe the theoretical model on which the research project is based. There are three closely related components of the model.

Chapter III: Measure of Relatedness

Chapter IV: Cluster Definition

Chapter V: Search Procedure

The experimental system which was devised to test the applicability of the model to a real world situation will be described in Part Three. It is hoped that this organization will help in keeping the abstract ideas of the model separate from the particular physical implementation which was developed to test them. It may be somewhat misleading, however. In actuality the model was not completely developed before the implementation began. It was continually revised and improved as various versions of experimental systems were programmed, tested and then discarded. What is described in this and the next part is the current model and test program.

CHAPTER III

MEASURE OF RELATEDNESS

The first step in establishing the conceptual basis of the research project is the selection of a measure of the relatedness between documents. To this end a sample space will be defined and a probability distribution assigned to it. Then a measure based on these probabilities will be selected and some of its characteristics noted. Finally the document network generated by the measure will be described.

3.1 Sample Space

In order to motivate the choice of our mathematical model, we regard each interaction of a user with a library as a partitioning of the stack into two disjoint subsets of documents: one containing all the documents of interest to the user and the other containing the rest of the documents. Each interaction is assumed to have a single purpose in the sense that all documents of interest are of interest for the same purpose.

There are theoretically 2^n such partitionings possible for a stack of n documents. Now let us think of a discrete collection of 2^n points (a sample space[22]), each representing one of the possible partitionings. These points can be identified by n-bit binary numbers, $x_1 \ldots x_n$, where x_i is 1 if the ith document is in the subset of interest and 0 if it is in the subset of no interest for the partition in question. (A superscript will be used to denote the value of a variable: $x_i^1 \equiv x_i = 1$.)

For a given user population and document collection a probability distribution $p(x_1 \ldots x_n)$ can be assigned to the sample space. Each $p(x_1 \ldots x_n)$ may be regarded as the probability that a user chosen at random from the population will partition the document collection with the partition $x_1 \ldots x_n$.

Compound events can be defined in terms of the simple events represented by the sample points. For example, $p(x_1^1)$, the probability that document 1 will be of interest to some user can be obtained by summing the probabilities of all points for which $x_1=1$.

$$p(x_1^1)=\sum_{x_2 \ldots x_n} p(x_1^1 x_2 \ldots x_n)$$

Similarly $p(x_1^1 x_2^1)$, the probability that documents 1 and 2 will be found to be of interest jointly, can be obtained by summing up the probabilities of all points for which $x_1=1$ and $x_2=1$.

$$p(x_1^1 x_2^1)=\sum_{x_3 \ldots x_n} p(x_1^1 x_2^1 x_3 \ldots x_n)$$

In the sections that follow we will want to talk not only about the abstract theoretical values of these probabilities, but also about their estimated values as obtained from experimental data. Suppose that there is information available on a large number of partitionings of a library. Let us make the following definitions.

N: Total number of partitionings of the library that are available.

N_i: Number of partitionings in which document i occurs in the subset of interest.

N_{ij}: Number of partitionings in which both documents i and j occur in the subset of interest.

Based on these N's estimates of the probabilities can be made as

follows:

$$p(x_i^1) \approx \frac{N_i}{N}$$

$$p(x_i^1 x_j^1) \approx \frac{N_{ij}}{N}$$

etc.

The partitioning data employed in these estimates may result from experimental evidence other than actual user interactions with the stack of documents in question. For instance, one might partition the stack on the basis of whether or not the documents cite a given document, or on the basis of whether or not they contain a particular word in their titles. As a matter of fact, the experimental system described in Chapter VI uses partitionings based on whether or not the documents cite a given document because these were readily available while actual usage data were not.

This use of another type of partitioning data (other than usage data) by the experimental system is considered acceptable here since the purpose of the experimental portion of the project is to permit an investigation of general properties of the theoretical model that should be largely independent of the precise values of the probability estimates.

3.2 Criteria for Selecting a Measure of Relatedness

We have already noted in Sec. 1.34 that a number of measures of 'relevance' have been suggested for us in information retrieval. Some of the more widely known of these measures are tabulated in Appendix A. The differences between them are partially due to the fact that they were designed for different purposes and partially due to the varied

backgrounds of the people who proposed them. Some of them have a theoretical basis in probability, statistics, or information theory; others are of an ad hoc nature.

In Sec. 2.3 we discussed why a measure of relatedness was needed for this project. The purpose of such a measure is not to rate the individual or joint merit of the documents in the stack, but rather to represent their relationship in terms of frequency of use and co-use. To this end it was decided that the measure selected should have the seven characteristics listed below.

Not all of the measures of Appendix A are expressible in terms of the theoretical probabilities of the last section. Therefore, for purposes of comparison we shall express these seven criteria in terms of the frequency counts on which the estimated probabilities are based. The N's are as defined in the last section, C is the measure of relatedness between documents i and j, and $R \approx S|_T$ means that R monotonically increases with S as T is held constant.

1. Co-occurrence Factor $\qquad C \approx N_{ij}\Big|_{N, N_i, N_j}$

The measure should monotonically increase with the number of co-occurrences in the subset of interest of the documents in question if all other factors are held constant. Consider, for example, a pair of documents (i,j) and another pair (r,s). If the N's are the same for both pairs except that $N_{ij} > N_{rs}$, then the relatedness between i and j should be greater than the relatedness between r and s.

2. Other Usage Penalty Factor $\qquad C \approx 1/N_i\Big|_{N, N_i, N_j}$

The measure should monotonically decrease as the number of occurrences of one of the documents increases- all other factors being

held constant. That is, if document i is used a larger number of times but not in conjunction with document j, then the relatedness between i and j should decrease.

3. Co-occurrence Ratio Factor $C \simeq N_{ij}/N_i \Big|_{N, N_j}$

If the ratio or fraction of the number of co-occurrences of document i with document j to the total occurrences of document i increases, the measure should increase also. Note that this criterion is not a consequence of 1 and 2.

4. Function of Probability Estimates Only $C(N_i/N,\ N_j/N,\ N_{ij}/N)$

The measure should depend only on the ratios of frequency counts which are used to estimate the probabilities. As long as these ratios remain constant the measure should not change.

5. Statistical Independence

The one bench mark that is available for measures is the statistical independence of the events in question. It would seem logical that if the occurrence of two documents are statistically independent, their measure of relatedness should have the value 0.

6. Theoretical Basis

A measure that has a solid theoretical basis is to be preferred over one which has been developed by trial and error.

7. Ease of Use

The best measure is a simple one that is easy to calculate and manipulate.

3.3 Selection of a Measure

Let us now evaluate the measures of Appendix A in terms of the criteria of the last section. Measures (1) and (2) have no theoretical

basis (Criterion 6) and are not 0 for statistically independent events (Criterion 5). The Chi Square Formula (5) is not expressible in terms of the probability estimates (Criterion 4). The value of the Cosine Formula (6) for statistically independent events is $\sqrt{p(x_i^1 x_j^1)}$ which is neither 0 nor even constant. The Average Correlation Coefficient (7) does not satisfy Criteria 1, 2, or 3.

This leaves Measures 3, 4, and 8 which meet (at least partially) all of the criteria listed. Measure 8 was selected for this research project because its foundation in information theory has led to some very interesting and useful results.

The use of Measure (8) in document retrieval was first proposed by R. M. Fano[19]. In its more general form it expresses the degree to which a set of events $x_1^1,\ldots,x_r^1$, are correlated in terms of their individual and joint probabilities.

$$C(x_1^1\ldots x_r^1) = \log \frac{p(x_1^1\ldots x_r^1)}{p(x_1^1\ldots p(x_r^1)} \qquad (1)$$

The base of the logarithm function used in the formula and throughout the remainder of this paper will be assumed to be 2. This will mean that the unit of correlation will be the "bit".

If only 2 events, i and j, are considered, then the coefficient is equal to the mutual information, $I(x_i^1;x_j^1)$, between the 2 events as defined in information theory[20].

$$C(x_i^1 x_j^1) = I(x_i^1;x_j^1) = \log \frac{p(x_i^1 x_j^1)}{p(x_i^1)p(x_j^1)} \qquad (2)$$

Let us relate the probabilities of formulae (1) and (2) to the probabilities of document usage defined over the sample space of the preceding section. The event x_i^1 is now the occurrence of document i in

a user's set of interest. The correlation $C(x_i^1 x_j^1)$ is the degree to which the two documents, i and j, are taken to be mutually pertinent.

The approximation to C in terms of the estimated probabilities will be denoted by the symbol $\tilde{C}$.

$$C(x_i^1 x_j^1) = \log \frac{P(x_i^1 x_j^1)}{P(x_i^1)p(x_j^1)} \approx \log \frac{N\,N_{ij}}{N_i N_j} = \tilde{C}\,(x_i^1 x_j^1)$$

3.4 Practical Considerations

In order to calculate the measure of relatedness C for any arbitrary set of documents selected from a collection of n documents, one would have to estimate and perhaps store at least 2^{n-1} probabilities. This is, of course, out of the question for any reasonably-sized document file. If C is to be used, some approximating simplification must be made.

Let us now note that this correlation coefficient C can be expanded in terms of mutual information terms as follows[20]:

$$C(x_1^1 \ldots x_r^1) = \sum_{\substack{i,j=1 \\ (i \neq j)}}^{r} I(x_i^1;x_j^1) \cdot \sum_{i,j,k=1}^{r} I(x_i^1;x_j^1;x_k^1) + \ldots$$

where

$$I(x_1;x_2) = \log \frac{p(x_1 x_2)}{p(x_1)p(x_2)}$$

$$I(x_1;x_2;x_3) = \log \frac{p(x_1 x_2)p(x_1 x_3)p(x_2 x_3)}{p(x_1)p(x_2)p(x_3)p(x_1 x_2 x_3)}$$

etc.

It has been proposed that C be approximated by the first summation in this series, and that the other summations be dropped as higher-order effects. There are some theoretical reasons which would lead one

to believe that this would result in a good approximation to C^{20}. However, we shall rest our case here on practical necessity and not go into the details of these theoretical arguments.

$$C(x_1^1 \ldots x_r^1) \approx \sum_{\substack{i,j=1 \\ (i \neq j)}}^{r} I(x_i^1; x_j^1) = \sum_{\substack{i,j=1 \\ (i \neq j)}}^{r} \log \frac{p(x_i^1 x_j^1)}{p(x_i^1)p(x_j^1)}$$

For this approximation one need only estimate and store n univariate and $\binom{n}{2}$ bivariate probabilities in order to obtain the correlation between events and subsets of events.

Through the same approach one can obtain an approximation to the correlation between any two subsets of events--

$$C[(x_1^1 \ldots x_r^1)(y_1^1 \ldots y_r^1)] \approx \sum_{i,j=1} I(x_i^1; y_j^1)$$

If these subsets overlap then one or more of the terms in the series becomes the self correlation of the event.

$$C(x_i^1 x_i^1) = \log \frac{p(x_i^1 x_i^1)}{p(x_i^1)p(x_i^1)} = \log \frac{1}{p(x_i^1)}$$

3.5 Characteristics of the Measure for Document Pairs

The measure of relatedness is 0 for two statistically independent events:

$$p(x_1^1 x_j^1) = p(x_1^1)p(x_j^1)$$

For events occurring together less often than if they were statistically independent, C is negative and for events occurring together more often C is positive.

Theoretically the range of C is from $-\infty$ to $+\infty$. However, there is

a statement that can be made about the upper bound. Since $p(x_i^1 x_j^1)$ cannot be larger than $p(x_i^1)$ or $p(x_j^1)$ the following inequalities hold:

$$C(x_i^1 x_j^1) = \log \frac{p(x_i^1 x_j^1)}{p(x_i^1)p(x_j^1)} \begin{cases} \leq \log \dfrac{1}{p(x_i^1)} \\ \leq \log \dfrac{1}{p(x_j^1)} \end{cases}$$

The quantity $\log[1/p(x_i^1)]$ is termed the self information of x_i^1 in information theory[20]. Thus, the correlation between two events is always less than or equal to the self information of either event. Let us indicate this range on the simple graph of Fig. 3.1.

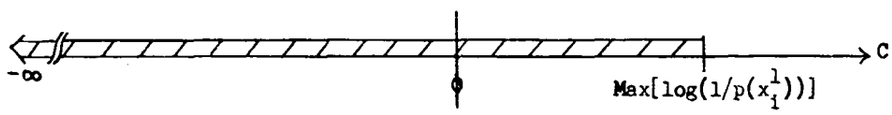


Fig. 3.1. Range of measure of relatedness.

Some additional comments about the range of the measure can be made if we consider $\tilde{C}$, the approximation to C based on the estimated probabilities. The maximum positive value of $\tilde{C}$ is (log N) and occurs when N_i, N_j, and N_{ij} all equal 1. Its minimum value other than $-\infty$ is (2-log N) and occurs when N_{ij} is 1 and N_i and N_j are N/2. This range is shown in Fig. 3.2.

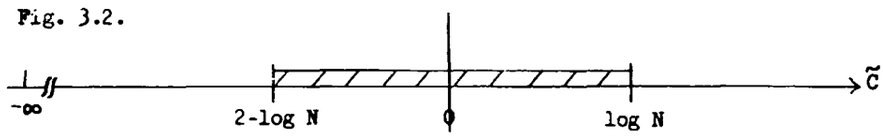


Fig. 3.2. Range of approximation to measure of relatedness.

For the test data utilized in the experimental portion of this project (see Sec. 6.1) it was found that the $\tilde{C}$'s were either $-\infty$ or had some positive value (see Fig. 3.3). The lower limit of (2-log N) in Fig. 3.2 is changed in Fig. 3.3 since all of the N_i's of the test data are much less than N/2. The new minimum of $\tilde{C}$ occurs when N_{ij}=1 and N_i

and N_j are maximum (called $(N_i)_{max}$).

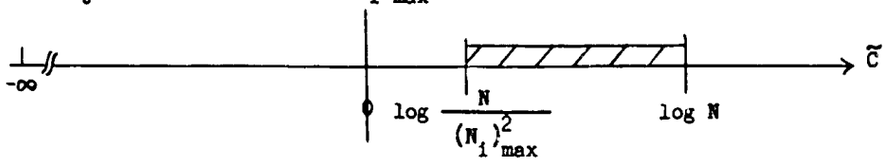


Fig. 3.3. Range of measure of relatedness for test data.

The range for the test data is due not so much to the fact that the occurrence of the documents in the test file are never statistically independent as to the fact that such statistical independence can only be detected with a very large data base. Consider documents i and j with $p(x_i^1)$, $p(x_j^1) = 0.0001$. If x_i^1 and x_j^1 are statistically independent, then $p(x_i^1 x_j^1) = 10^{-6}$. In order for any of the probability estimates to be this small we would need at least 10^6 partitionings. Many, many more partitionings than this would be needed if one wanted to have accurate estimates of the occurrences of such rare events. With fewer partitionings these events either never occur, resulting in $p(x_i^1 x_j^1) = 0$, or do occur with the estimate for $p(x_i^1 x_j^1)$ being larger than it should be. This is the phenomenon observed for the test data. Even if there were correlations that were 0 or slightly negative they would be pushed to $-\infty$ or to some positive value because of the limited number of partitionings available.

It is conjectured that this will be the situation in most practical cases for some time to come. In a very large document collection (10^5-10^7 items) the probability of occurrence of any one document is probably small, say 10^{-3} or 10^{-4}. This would require a file of 10^6 to 10^8 partitionings to measure statistical independence which would take considerable time and effort to collect. In a small document collection the probability of occurrence of any one document could be larger but the

number of partitionings available would undoubtedly be less also.

It should be pointed out that this measure will assume some value for every pair of documents in the stack (except perhaps documents that have never been used). Even two documents that have never co-occurred together ($N_{ij}=0$) are related by the value $-\infty$.

A few comments should be made about the value $-\infty$. It is not a realistic value for the correlation between most documents because it implies that there is absolutely no chance of two documents co-occurring. As has already been pointed out this arises because the probabilities may end up exactly zero. A much more practical and reasonable approach to the problem would be to make all correlations between document pairs for which $N_{ij}=0$ equal to some finite negative value instead of $-\infty$. More will be said on the choice of this negative value (K) later (Sec. 4.5).

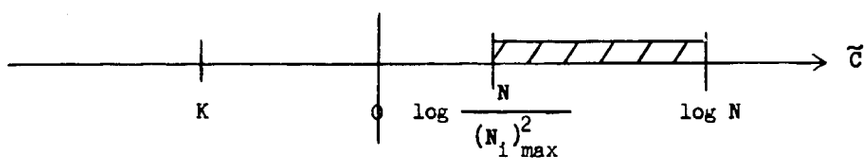


Fig. 3.4. Revised range of measure for test data.

Another feature of the selected measure is that it is non-directional. That is, the value of the measure from document i to j is the same as from j to i.

3.6 Document Networks

It has been suggested that measures of the relatedness between documents should be metrics[24]. This would require that a measure C exhibit the following properties:

(1) $C(x,x)=0$

(2) $C(x,y)>0$ (if $x \neq y$)

(3) $C(x,y)=C(y,x)$

(4) $C(x,y)+C(y,z) \geq C(x,z)$

The measure under consideration does meet property (3). It might conceivably be made to fit properties (1) and (2) through some type of normalization or restriction. There appears to be no way to make it have property (4), the triangle inequality. Indeed, it would be rather disturbing to this author if it did have property (4).

Bar-Hillel has pointed out in the comment cited in Sec. 2.21 that many of the important aspects of a document collection (except physical location) cannot be made to satisfy the triangle inequality and cannot, therefore, be represented by metrics. His conclusion was that measures derived from these features (joint usage, common citation, etc.) are useless. Our conclusion is that such measures should not be required to be metrics.

The idea that a metric space is the appropriate model for a document collection is rejected here. If one desires a model to aid in his mental picture of a document collection, a simple network is suggested. Each document can be considered a node and the link between two nodes can be assigned the value of the measure of relatedness between the corresponding documents. It has already been pointed out that the measure of relatedness chosen links every node (document) to every other node. It might, therefore, be easier to visualize the sub-network consisting of only positive links. This is the visual picture found most helpful to the author.

Thus far we have considered the problem of generating a document network from a set of probabilities. Let us now consider the reverse process. If one draws a document network and arbitrarily chooses the values to be assigned to the links, can a set of probabilities be found

which could have generated the network? This question is of interest because if there is only a certain class of networks that are realizable from sets of probabilities, then we need focus our attention only on that class.

Theorem. For every document network (with the restriction that the values of the positive links be finite) there is at least one set of probabilities which could have generated it.

Proof. The first step in proving this theorem will be to select a set of values for the elementary probabilities, $p(x_1 \ldots x_n)$. It will then be shown that the set selected yields the correct values for the links of the network in question and forms a valid set of probabilities (i.e. each value is in the range 0 to 1 and their sum is 1).

Before proceeding let us define the following symbols.

n: number of documents in the network ($n \geq 2$).

$C(x_i^1 x_j^1)$: value of the network link between documents x_i and x_j.

C_{max}: maximum value of $C(x_i^1 x_j^1)$.

k: the lesser of the two quantities: $(1/n)$ and $(1/n)2^{-C_{max}}$.

It will also be convenient to introduce at this point one additional notation convention. Let us allow the values of the variables in the $p(x_1 \ldots x_n)$'s which differ from 0 to be specified by a statement following a colon as well as by superscripting. For example:

$$p(x_1 \ldots x_n : x_i = 1) = p(x_1^0 \ldots x_{i-1}^0 \; x_i^1 x_{i+1}^0 \ldots x_n^0)$$

We are now ready to state the values for the elementary probabilities, $p(x_1 \ldots x_n)$. Four possible classes will be considered.

(1) All $p(x_1,,,x_n)$ for which three or more x's are 1.

$$p(x_1 \ldots x_n : \text{at least } 3 \text{ x's} = 1) = 0$$

(2) All $p(x_1 \ldots x_n)$ for which two x's are 1:

$$p(x_1 \ldots x_n : x_i, x_j = 1) = k^2 \, 2^{C(x_i^1 x_j^1)} \qquad \text{for all } i,j \ (i \neq j).$$

(3) All $p(x_1 \ldots x_n)$ for which one x is 1:

$$p(x_1 \ldots x_n : x_i = 1) = k - k^2 \sum_{\substack{j=1 \\ j \neq i}} 2^{C(x_i^1 x_j^1)} \qquad \text{for all } i.$$

(4) The $p(x_1 \ldots x_n)$ for which no x is 1.

$$p(x_1^0 \ldots x_n^0) = 1 - nk + (k^2/2) \sum_{\substack{i,j=1 \\ i \neq j}} 2^{C(x_i^1 x_j^1)}$$

The motivation behind the selection of these values will become clearer as the discussion proceeds. It may be helpful, however, to note three of the underlying ideas at this point.

(1) Each $p(x_i^1)$ is to have the same value.

$$p(x_i^1) = k$$

(2) The value of the $p(x_i^1)$'s is to be chosen so that the $p(x_i^1 x_j^1)$'s can be adjusted to give the desired $C(x_i^1 x_j^1)$'s.

$$p(x_i^1 x_j^1) = k^2 \, 2^{C(x_i^1 x_j^1)}$$

(3) The only elementary events that are allowed to occur are those with zero, one or two documents in the subset of interest.

Let us prove that the elementary probabilities as selected above generate the correct values for the links of the document network. Preliminary to doing this we will determine the values of the $p(x_i^1)$'s and $p(x_i^1 x_j^1)$'s.

$$p(x_i^1) = \sum_{\substack{\text{all p's for} \\ \text{which } x_i = 1}} p(x_1 \ldots x_n)$$

$$= p(x_1 \ldots x_n : x_i = 1) + \sum_{\substack{j=1 \\ j \neq i}}^{n} p(x_1 \ldots x_n : x_i, x_j = 1)$$

$$= k - k^2 \sum_{\substack{j=1 \\ j \neq i}}^{n} 2^{C(x_i^1 x_j^1)} + k^2 \sum_{\substack{j=1 \\ j \neq i}}^{n} 2^{C(x_i^1 x_j^1)}$$

$$p(x_i^1) = k \qquad \text{for all } i.$$

$$p(x_i^1 x_j^1) = \sum_{\substack{\text{all } p\text{'s for} \\ \text{which } x_i, x_j = 1}} p(x_1 \ldots x_n)$$

$$= p(x_1 \ldots x_n : x_i, x_j = 1)$$

$$p(x_i^1 x_j^1) = k^2 2^{C(x_i^1 x_j^1)} \qquad \text{for all } i,j \ (i \neq j).$$

$$C(x_i^1 x_j^1) = \log \frac{p(x_i^1 x_j^1)}{p(x_i^1) p(x_j^1)}$$

$$= \log \frac{k^2 2^{C(x_i^1 x_j^1)}}{(k)\,(k)}$$

$$= C(x_i^1 x_j^1) \qquad \text{for all } i,j \ (i \neq j).$$

In order for the set of values selected for the $p(x_1 \ldots x_n)$'s to form a valid set of probabilities, their sum must be 1.

$$S = \sum_{\text{over all } x\text{'s}} p(x_1 \ldots x_n)$$

$$= 1/2 \sum_{\substack{i,j=1 \\ i \neq j}}^{n} p(x_1 \ldots x_n : x_i, x_j = 1) + \sum_{i=1}^{n} p(x_1 \ldots x_n : x_i = 1 + p(x_1^0 \ldots x_n^0)$$

$$= (k^2/2) \sum_{\substack{i,j=1 \\ i \neq j}}^{n} 2^{C(x_i^1 x_j^1)} + nk - k^2 \sum_{\substack{i,j=1 \\ i \neq j}}^{n} 2^{C(x_i^1 x_j^1)} + 1 - nk + (k^2/2) \sum_{\substack{i,j=1 \\ i \neq j}}^{n} 2^{C(x_i^1 x_j^1)}$$

$$S = 1$$

We must also prove that the values selected for the $p(x_1 \ldots x_n)$'s

are in the range 0 to 1. The values for the first class of probabilities, $p(x_1 \ldots x_n$:at least 3 x's $=1)$, are all 0 and thus automatically in the range. The values assigned to the probabilities of the second class, $p(x_1 \ldots x_n : x_i, x_j = 1)$, can be shown to be in the range by the following argument.

$$k \leq (1/n)2^{-C_{max}} \leq (1/n)2^{-C(x_i^1 x_j^1)}$$

$$k2^{C(x_i^1 x_j^1)} \leq (1/n) \text{ and } k \leq (1/n)$$

$$\therefore \quad k^2 2^{C(x_i^1 x_j^1)} \leq (1/n)^2$$

$$0 \leq k^2 2^{C(x_i^1 x_j^1)} < 1$$

Next let us show that the values assigned to the probabilities of the third class, $p(x_1 \ldots x_n : x_i = 1)$, are in the correct range.

$$k - k^2 \sum_{\substack{j=1 \\ j \neq i}}^{n} 2^{C(x_i^1 x_j^1)} \leq k \leq 1/n < 1$$

$$k - k \sum_{\substack{j=1 \\ j \neq i}}^{n} 2^{C(x_i^1 x_j^1)} \geq k - k(n-1)(1/n) > 0$$

Finally let us check the range of $p(x_1^0 \ldots x_n^0)$.

$$1 - nk + (k^2/2) \sum_{\substack{i,j=1 \\ i \neq j}}^{n} 2^{C(x_i^1 x_j^1)} \leq 1 - nk + (1/2)(n)(n-1)(1/n) = 1 - \frac{nk}{2} - \frac{k}{2} < 1$$

$$1 - nk + (k^2/2) \sum_{\substack{i,j=1 \\ i \neq j}}^{n} 2^{C(x_i^1 x_j^1)} \geq 1 - nk \geq 1 - n(1/n) = 0$$

QED

CHAPTER IV

DOCUMENT CLUSTERS

In the last chapter a measure of relatedness between documents was defined and a document network based on the measure was described. The next step to be taken is to formulate a definition for what constitutes a subset (cluster) of highly inter-related documents based on this measure. The purpose of such a definition is to provide the user who has requested information from the system with a set (cluster) of papers which is judged to be related to his interest.

The exact form that a request for information can take and the procedure used to translate a request into an answer cluster will be described in Chapter V. The way a cluster is obtained, modified, and stored in the experimental system devised for this project will be covered in Chapter VI. In this chapter we shall confine our attention to what constitutes an appropriate cluster of documents. Two types of clusters will be defined and analyzed, and certain modifications will be described which make one of the definitions acceptable.

4.1 Local Maximum Clusters

The cluster definition which was first proposed and tested turned out to be the one which was eventually selected for this project. Let us formally define it and then discuss its characteristics.

In this definition and in the remainder of this thesis we will find use for the following set operators.

$\cup$: Set union--($A \cup B$) is the set of all documents in set A or in set B.

$\cap$: Set intersection--($A \cap B$) is the set of documents in both set A and set B.

$\subset$: Set inclusion--($A \subset B$) means that the set A is included in the set B.

$\overline{X}$: Set complementation--$\overline{X}$ is the set of all documents not in X.

Definition: Local Maximum Cluster

A local maximum cluster is defined to be any subset of documents $X_\alpha = (x_{\alpha_1}, \ldots, x_{\alpha_r})$ for which both of the following conditions hold.

1. Every document x_i in X is positively correlated to the remainder of X.

$$C[x_i(X_\alpha \cap \overline{x_i})] > 0 \qquad \text{for all } x_i \subset X_\alpha.$$

2. Every document x_j not in X_α is negatively correlated to X_α.

$$C(x_j X_\alpha) \leq 0 \qquad \text{for all } x_j \subset \overline{X}_\alpha.$$

(Note that zero is arbitrarily classed as a negative value.)

A local maximum cluster is so named because every possible single change (addition or deletion) to the cluster will result in a decrease in its internal correlation. The internal correlation C(X) of a subset X is defined to be the sum of the links whose ends both terminate in the subset. If X_α is a cluster, then

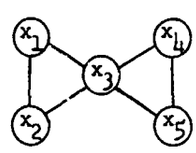
$$C(X_\alpha) > C(X_\beta) \qquad \text{for all } X_\beta \text{ which differ from } X_\alpha \text{ by a single document.}$$

Five specific characteristics of local maximum clusters have been selected for discussion below.

Size. The average size of the clusters produced by the local

maximum definition is very much a function of the correlation assigned to document pairs that have not co-occurred together ($N_{ij}=0$). It has already been noted that although this correlation, K, is $-\infty$ by the formula, some finite value is more appropriate (Sec. 3.14). If K is made positive, then there will be only one cluster consisting of the total file. If K is made just slightly negative, then the clusters formed will be disjoint and consist of all documents connected by one or more paths of positive links. If K is made very negative, the only clusters will be those sets of documents wherein every document has co-occurred with every other document.

Overlap. It is fairly obvious that local maximum clusters can overlap. Consider the network of Fig. 4.1 in which all the links shown have the value +5 and all the links not shown have the value -6. The two local maximum clusters, $(x_1x_2x_3)$ and $(x_3x_4x_5)$ overlap through x_3.



Links shown are +5

Links not shown are -6.

Fig. 4.1. Network with overlapping clusters.

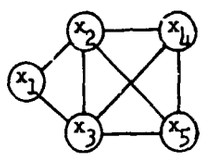
Coverage. The following simple theorem shows that local maximum clusters may not cover all the documents in the network.

Theorem. Document networks exist which have documents that are not included in any local maximum cluster.

Proof. First consider a document that has never co-occurred with any other document. Such a document does not prove the theorem because it is included in a cluster which consists of only the document itself.

Now consider the network of Fig. 4.2. The only cluster is

$(x_2x_3x_4x_5)$. The document x_1 cannot form a cluster by itself since x_2 and x_3 are positively correlated to it. It cannot form a cluster with x_2 and x_3 since x_4 and x_5 are positively correlated to the set $(x_1x_2x_3)$ with the value 5+5-6=4. Thus x_1 occurs in no cluster. QED



Links shown are +5.

Links not shown are -6.

Fig. 4.2. Network with a document (x_1) in no cluster.

Although local maximum clusters do not cover all possible documents in a network, one is at least assured of the following--

Theorem. Every document network contains at least one local maximum cluster.

Proof. The proof will be constructive. A local maximum cluster can be formed by successively making single changes (additions or deletions) to a subset of documents as outlined in the following 3-step procedure.

1. Pick a document at random as the initial member of the subset.

2. If every document outside the subset is negatively correlated to the subset and every document inside the subset is positvel correlated to the subset, then quit. The local maximum cluster has been found.

3. Otherwise either add a positively correlated document that is not in the subset or delete a negatively correlated document that is in the subset. It doesn't matter which is done, but only one change must be made. Now return to step 2.

This procedure is assured of termination if the document set is

finite because step 3 always increases the internal correlation (sum of the internal links) of the subset being formed. There is, of course, an upper limit to the internal correlation of any finite set of documents.

QED

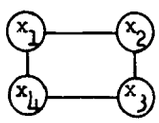
Structure. Local maximum clusters can form the type of hierarchal structure indicated by the following theorem.

Theorem. A local maximum cluster can be a subset of another local maximum cluster.

Proof. Again we can use an example to prove the theorem. In the document network of Fig. 4.3 there are five local maxima:

(x_1x_2), (x_2x_3), (x_3x_4), (x_1x_4), $(x_1x_2x_3x_4)$.

The first four of these are subsets of the fifth. QED



Links shown are +5.

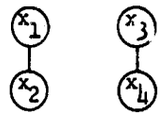
Links not shown are -6.

Fig. 4.3. Network with hierarchal cluster structure.

Relatedness. Now consider the problem of whether local maximum clusters form well related sets.

Theorem. Totally unrelated subsets of documents can occur together in a local maximum cluster. By totally unrelated we mean that no document in one set is positively correlated to a document in the other set.

Proof. This theorem can be proved by another simple example. The set $(x_1x_2x_3x_4)$ of Fig. 4.4 forms a cluster and yet there are no positive links between the set (x_1x_2) and the set (x_3x_4). QED



Links shown are +7.

Links not shown are -3.

Fig. 4.4. Cluster containing unrelated subsets.

The inclusion of unrelated subsets in the same cluster is considered an undesirable characteristic for a cluster to have. The reason why this is so involves the design of the procedure of Chapter V. It was decided that the procedure could be greatly simplified if one were to assume that each request for information from the system has only one purpose. A person who has several areas of interest on which he desires information is expected to make a separate request for each area. It follows that if each request has a single purpose, then the document clusters which are to answer these requests should not be divisible into unrelated subsets.

4.2 Subset Clusters

In an attempt to keep completely unrelated sets of documents from becoming part of the same cluster, a definition was devised based on the addition of subsets or the deletion of subsets of documents as opposed to the single changes allowed in the local maximum definition. This definition was accepted as the one most suitable for this project for a number of months. In this section we shall describe it, note its characteristics, and explain why it was finally discarded.

Definition 1: Subset Cluster

A subset cluster is defined to be any set of documents $X_\alpha = (x_{\alpha_1}, \ldots, x_{\alpha_r})$ for which both of the following conditions hold.

1. Every subset of documents X_β included within X_α is positively correlated to the remainder of X_α.

$$C[X_\beta(X_\alpha \cap \overline{X_\beta})] > 0 \qquad \text{for all } X_\beta \subset X_\alpha.$$

2. Every subset of documents X_ρ external to X_α is negatively correlated to X_α.

$$C(X_\rho X_\alpha) \leq 0 \qquad \text{for all } X_\rho \subset \overline{X_\alpha}.$$

It is worth noting that Condition 2 of the local maximum cluster definition is equivalent to Condition 2 above. If each document external to X_α is negatively correlated to X_α, then certainly all external subsets are negatively correlated to X_α. Conversely if each subset is negatively correlated to X_α, then, of course, single documents, being subsets, are also negatively correlated to X_α. It should also be pointed out that all subset clusters are local maximum clusters but not vice versa.

Next let us present an alternative definition of a subset cluster.

<u>Definition 2: Subset Cluster</u>

A subset cluster is defined to be any set of documents $X_\alpha = (x_{\alpha_1}, \ldots, x_{\alpha_r})$ for which both of the following conditions hold.

1. The internal correlation of X_α as defined in Sec. 4.1 is greater than the sum of the internal correlation of the disjoint subsets of X_α created by any arbitrary partitioning.

$$C(X_\alpha) > \sum_{i=1}^{r} C(D_i) \qquad \text{for all partitionings in which } (D_1 \cup \ldots \cup D_r) = X_\alpha \text{ and } D_i \cap D_j = \text{null set.}$$

2. The sum of the internal correlations of X_α and some subset X_ρ external to X_α is greater than or equal to the internal correlation of the set formed by adding X_ρ to X_α.

$$C(X_\alpha)+C(X_\rho)\geq C(X_\alpha \cup X_\rho) \qquad \text{for all } X_\rho \subset \overline{X}_\alpha.$$

<u>Theorem</u>. Definition 1 and Definition 2 for subset clusters are equivalent.

<u>Proof</u>. The equivalence of the second conditions of both definitions is fairly obvious. The equivalence of the first conditions requires some verification.

Let us assume that Cond. 1 of Def. 2 holds and partition the clusters into two subsets.

$$C(X_\alpha) > C(X_\beta)+C(X_\alpha \cap \overline{X_\beta})$$

But: $C(X_\alpha)=C(X_\beta)+C(X_\alpha \cap \overline{X_\beta})+C[(X_\beta)(X_\alpha \cap \overline{X_\beta})]$

$\therefore\ C[(X_\beta)(X_\alpha \cap \overline{X_\beta})] > 0$

This last result is Cond. 1 of Def. 1.

Now let us assume that Cond. 1 of Def. 1 holds and partition the cluster into the disjoint subsets $D_1,\ldots,D_r$. By Def. 1:

$$C[(D_i)(X_\alpha \cap \overline{D_i})] > 0 \qquad \text{for all } D_1,\ldots,D_r$$

But:

$$C(X_\alpha)= \sum_{i=1}^{r} C(D_i)+1/2 \sum_{i=1}^{r} C[(D_i)(X_\alpha \cap \overline{D_i})]$$

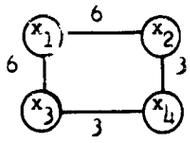
$$\therefore\ C(X_\alpha) > \sum_{i=1}^{r} C(D_i)$$

Thus if Cond. 1 of Def. 1 is true, Cond. 1 of Def. 2 is also. QED

Let us discuss now some of the characteristics of subset clusters. The comments and theorems on cluster size, overlap and coverage, which were made in Sec. 4.1 for local maximum clusters, hold for subset clusters also with the exception that one is no longer assured of having at least one cluster in any given document network.

<u>Theorem</u>. There exist document networks which contain no subset clusters.

<u>Proof</u>. Examination of each of the 2^4 possible subsets in the network of Fig. 4.5 reveals that none of them satisfy the two conditions necessary for subset clusters. QED



Links not shown are -5.

Fig. 4.5. Network containing no subset clusters.

<u>Structure</u>. Next we note that a hierarchal structure is no longer possible with subset clusters.

<u>Theorem</u>. No subset cluster X_β can be included within another subset cluster X_α.

<u>Proof</u>. Let us assume that X_α and X_β are subset clusters and that $X_\beta \subset X_\alpha$. Since X_α is a cluster and $X_\beta \subset X_\alpha$, then by Cond. 1 of the definition:

$$C[X_\beta(X_\alpha \cap \overline{X_\beta})] > 0$$

But since X_β is a cluster and $(X_\alpha \cap \overline{X_\beta}) \subset \overline{X_\beta}$ then by Cond. 2:

$$C[X_\beta(X_\alpha \cap \overline{X_\beta})] \leq 0$$

which contradicts the previous inequality QED

<u>Relatedness</u>. In the last section it pointed out that one of the difficulties with local maximum clusters lies in the fact that even completely uncorrelated sets of documents can occur in the same cluster. It was for this reason that the subset definition was devised. In subset clusters one is assured by definition that no subset of the cluster is negatively correlated to the remainder of the cluster.

Utility. The problems of coverage and hierarchy did not prove to be serious drawbacks to the subset definition of clusters. An extension to the definition was devised which allowed all documents to be in at least one cluster and provided for hierarchal relationships. This extension involved applying a bias to the links of the network. (See Sec. 4.4.) The reason the subset definition was finally abandoned was because no method could be found that would isolate subset clusters with a reasonable amount of effort.

Consider for a moment the problem of checking Condition 1 of the subset definition. One must determine whether there is a partitioning of a set of documents which results in two subsets that are negatively correlated to each other. The brute force method is to try every partitioning. This would involve 2^n tests for a set of n documents and would certainly be too much processing for an n of 20 or 30 even on a high speed digital computer. Several efforts were made to devise a more efficient method. Although they were not entirely successful, it might be well to briefly document a couple of them.

4.3 Finding Subset Clusters

In the first method for finding subset clusters which was investigated, an effort was made to determine if a partitioning of a set existed which would result in two negatively correlated subsets. Such a partitioning is called a 'split' of the set in the following discussion.

In the other approach emphasis was focused on the small, very highly correlated subsets called 'kernels' within the document set and an attempt was made to combine and expand these until a split appeared

4.31 Locating Splits

We wish to devise a method which will determine whether a set of documents can be split into two negatively correlated subsets and to locate where such splits are. Some of the theorems that were developed for this purpose will be stated below. In the interests of brevity the proofs will not be given. The symbols used in these theorems are defined as follows.

n - number of documents in S, the sets under consideration.

a - number of documents in a subset A of S.

b - number of documents in a subset B where $B=S\cap\bar{A}$. $(a+b=n, A\cup B=S)$

K - negative value assigned to links for which $N_{ij}=0$.

C_{min} - smallest value of the links for which $N_{ij}\neq 0$. It will be assumed in the following theorems that C_{min} is positive. (See Sec. 3.5.)

C_{max} - largest positive link in the network.

d - number of links in the set S which have the value K.

Theorem 1: Consider the partitioning of a set of documents into the subsets A and B.

Part A: Only those parritionings which satisfy the following inequality can possibly result in splits.

$$(a)(b)\leq\left(\frac{C_{min}+|K|}{C_{min}}\right)d$$

Part B: A necessary condition for a partitioning to result in a split is that the partitioning must be crossed by at least r negative links where:

$$r=\frac{(a)(b)(C_{min})}{C_{min}+|K|}$$

Part C: A sufficient condition for a partitioning to result in a split is that the partitioning be crossed by at least s negative links where:

$$s = \frac{(a)(b)(C_{max})}{C_{max} + |K|}$$

<u>Example of Theorem 1:</u>

$n = 20$

$K = -5$

$C_{min} = 4$

$d = 40$ (40 of the 190 links are negative)

By Part A of the theorem (a)(b) must be less than 90 to allow a split. Therefore partitionings with distributions a:b = 10:10, 9:11, 8:12, and 7:13 cannot possibly result in splits. This immediately eliminates about 90% of the possible partitionings as candidates for splitting the set. Unfortunately there are some 60,460 partitionings that still must be considered which is still out of the question.

However if the 40 negative links are all bunched on only 5 of the nodes (8 per node), then by Part B of the theorem only 61 partitionings can possibly cause splits and these can easily be checked.

If only 10% of the links are negative (19 instead of 40), then only partitionings with a:b = 1:19 and 2:18 can cause splits. There are 210 such partitionings and a check of these would also be possible.

However in the general case C_{min} may be small, d may be large, and the negative links may not be so fortuitously arranged so that the partitionings which must be examined may still remain very large.

Theorem 2 is concerned with the possibility of finding splits of the set S as it is being formed.

Theorem 2. Consider the possibility of a set of documents being split by the addition of another document. Three statements can be made.

1. If the new document is positively correlated to each item in the set, then no split can be created.

2. If a split is created, it must be crossed by at least one newly added negative link.

3. The sum of the newly added links crossing any split created must be negative.

The next two theorems will help to determine whether the set S is a subset cluster when it contains one or more documents that are positively correlated to all of the other documents in S.

Theorem 3. If a set of n documents has d or more documents that are positively linked to every other document in the set, then the set has no splits.

$$d = \frac{n\,|K|}{C_{min} + |K|}$$

Theorem 4. Assume that a set of documents has splits. Now remove all those documents that are positively correlated to every other document in the set. The reduced set must also have splits.

The sum of the links connecting documents in the subset A to documents in B is termed the cross correlation of the partitioning which created A and B. The following three theorems relate to this cross correlation.

Theorem 5. The cross correlations of all possible partitionings of a document set are equal if and only if every link has the value 0. $(n \geq 3)$

Theorem 6. The cross correlations of all possible partitionings of a document set of size a:b are equal if and only if every link has the same value.

Theorem 7. The average cross correlation of the partitionings of size a:b is $C(S)(a)(b)/\binom{n}{2}$ where $C(S)$ is the total internal correlation of the set.

4.32 Forming Kernels

Another method which was considered as a way for determining if a set was a subset cluster was to form highly correlated kernels within the set in question and thereby try to locate possible splits. The kernels might initially be those subsets wherein every document is positively correlated to every other document. These sets could then be combined in various ways to see if any splits appeared. The following two theorems relate to this approach.

The symbols used are as defined in the last section and as follows:

C_{avg} - average of the positive links of the set.

D_i - the ith disjoint kernel of the set S.

$D_1 \cup \cdots \cup D_t \subseteq S$

$D_i \cap D_j$ = null set for all i,j ($i \neq j$).

Theorem. If the sum of the internal correlations of a set of disjoint kernels is greater than or equal to the total internal correlation of the set, then there is at least one split in the set.

In other words, if: $\sum_{i=1}^{t} C(D_i) \geq C(S)$

then S has at least 1 split.

Theorem. A sufficient condition for having at least one

split in a set is that the set contain at least d negative links where:

$$d = \frac{\binom{n}{2}C_{avg} - \sum_{i=1}^{t} C(D_i)}{C_{avg} + |K|}$$

4.4 Biased Clusters

In this section an extension or modification to the cluster definitions is proposed. It was initially devised in order that subset clusters could have a hierarchal structure. It was found to be a useful modification to local maximum clusters also.

As a way of introducing the concept of a biased cluster, let us consider a large cluster (either local maximum or subset) of documents covering a rather broad field of interest. There will, of course, be users who want all of the documents in such a cluster, but what about the users whose interests are very specific and who want only a small portion of the cluster? As yet there has been no provision for such a narrowing of interest. Subset clusters and many local maximum clusters are not decomposable. We shall now present the theoretical basis of a method which will allow a cluster to be reduced to a more specific set or enlarged to a more general set.

Consider a set of documents, $W=(w_1,\ldots,w_r)$, which forms a cluster in the overall document network. The problem of retrieving a portion of this cluster is regarded as equivalent to the problem of finding a cluster in the sub-library consisting only of W.

In order to show how this might be done let us define a new sample space which has only 2^r points instead of the 2^n points of the original sample space. Each point in the new space represents a possible parti-

tioning of W. To distinguish between the probabilities of the two sample spaces, the probabilities of the old sample space will be given a subscript 'α' and the probabilities of the new sample space a subscript 'β'. Let the probabilities assigned to the points of this new sample space be initially equal to the marginal probabilities of the corresponding events over the old sample space.

$$p_\beta(w_1 \ldots w_r) = p_\alpha(w_1 \ldots w_r) = \sum_{\substack{\text{over all } x \\ \text{not in } W.}} p_\alpha(x_1 \ldots x_n)$$

The marginal probability, $p_\alpha(w_1^0 \ldots w_r^0)$, is the sum of the probabilities of all those elementary events in which none of the documents in W are in the subset of interest. Since these events are irrelevant when one is considering only the sub-library W, let us set $p_\beta(w_1^0 \ldots w_r^0)$ equal to 0. Such a step requires that the other $p_\beta(w_1 \ldots w_r)$'s all be increased by a normalizing factor k. The final values for the probabilities assigned to the new sample space can now be specified.

$$p_\beta(w_1^0 \ldots w_r^0) = 0$$

$$p_\beta(w_1 \ldots w_r) = kp_\alpha(w_1 \ldots w_r) \qquad \text{for all } p_\beta(w_1 \ldots w_r) \text{except } p_\beta(w_1^0 \ldots w_r^0)$$

$$k = 1/[1-p_\alpha(w_1^0 \ldots w_r^0)]$$

Now let us consider the effect of this change in the sample space on the correlation of any two documents in W.

$$C_\alpha(w_1^1 w_2^1) = \log \frac{p_\alpha(w_1^1 w_2^1)}{p_\alpha(w_1^1)p_\alpha(w_2^1)}$$

$$C_\beta(w_1^1 w_2^1) = \log \frac{p_\beta(w_1^1 w_2^1)}{p_\beta(w_1^1)p_\beta(w_2^1)}$$

$$= \log \frac{(k)p_\alpha(w_1^1 w_2^1)}{(k)p_\alpha(w_1^1)(k)p_\alpha(w_2^1)}$$

$$= \log \frac{p_\alpha(w_1^1 w_2^1)}{p_\alpha(w_1^1)p_\alpha(w_2^1)} - \log (k)$$

$$C_\beta(w_1^1 w_2^1) = C_\alpha(w_1^1 w_2^1) - \log (k)$$

Thus the correlations for the sub-library can be obtained by merely subtracting a constant or bias from the correlations for the full library.

An alternative way to describe this approach is through the frequency counts used in making the probability estimates. Instead of considering all the available partitionings of the document file, let us consider only those partitionings in which one or more of the documents in W occur in the subset of interest. Let us denote the counts based on this restricted set of partitionings by the letter M and use N for the original counts.

$N_i = M_i$ for all i in W.

$N_{ij} = M_{ij}$ for all i,j in W.

Now let us consider what happens to the approximation to C based on the probability estimates with the new frequency counts.

$$\tilde{C}_\beta(w_i^1 w_j^1) = \log \frac{M\, M_{ij}}{M_i M_j}$$

$$= \log \frac{M\, N_{ij}}{N_i N_j}$$

$$= \log \frac{N\, N_{ij}}{N_i N_j} - \log \frac{N}{M}$$

$$\tilde{C}_\beta(w_i^1 w_j^1) = \tilde{C}_\alpha(w_i^1 w_j^1) - \log (N/M)$$

Here again we note that we can in effect reduce the size of the library under consideration by merely subtracting a constant from each

correlation value.

In an analagous manner we can increase the size of the library and thereby obtain larger, more general clusters by adding some bias to each correlation in the network.

We now observe that of the three measures which meet the criteria outlined in Sec. 3.2 (3,4, and 8) only Measure 8 allows this type of narrowing an broadening of the request range. Measures 3 and 4 are insensitive to any change in the size of the library or partitioning file.

One final question arises concerning the biasing of the value K assigned to links for which $N_{ij}=0$. One could either let the bias affect all links equally or one could look upon K as a fixed value which is not changed by the bias. The latter approach was rather arbitrarily selected.

We are now ready to define what is meant by a biased cluster.

Definition: Biased Cluster

A biased local maximum cluster has the same definition as a regular local maximum cluster, but a non-zero bias has been applied to the document network in which the cluster is formed. The same is true of a biased subset cluster.

In summary, a simple, easy-to-use method has been suggested which will allow the size of clusters to be increased or decreased. Some arguments have been presented which show that the method has a sound theoretical basis.

4.5 Final Cluster Decision

The local maximum definition of clusters was reconsidered after no general method for finding subset clusters was found. It was pointed out in Sec. 4.1 that local maximum clusters were considered unacceptable

because totally unrelated subsets of documents could be part of the same cluster. The following theorem and lemmas show that this difficulty can be avoided by selecting an appropriate value for K.

During the remainder of this section it will be assumed that all of the links for which $N_{ij} \neq 0$ are positive (See Sec. 3.5). If this condition does not hold then the theorems and lemmas which follow can be restated in terms of links for which $N_{ij} = 0$ and links for which $N_{ij} \neq 0$ instead of positive and negative links.

> Theorem. Each document in a local maximum cluster of n documents is positively linked to over half of the remaining n-1 documents if $K \leq -C_{max}$.

Proof. By definition each document in a local maximum cluster is positively correlated to the remaining (n-1) documents in the cluster. Now if the positive links are smaller or equal in magnitude than the negative links, then it stands to reason that there must be more of the former to yield a positive sum.

> Lemma. Consider a local maximum cluster that is partitioned into 2 subsets, X_α and X_β, with X_β the larger if they differ in size. If $K \leq -C_{max}$, every document in X_α has at least one positive link to the other subset.

> Lemma. In a local maximum cluster with $K \leq -C_{max}$ there can be no subset that is totally uncorrelated (has no positive links) to the remainder of the cluster.

The choice of $K \leq -C_{max}$ does not insure that a local maximum cluster will be free of splits and thus be a subset cluster. Subsets can still be negatively correlated to the remainder of the cluster. But it does insure that the rather strong type of relatedness expressed by the above

two lemmas will exist for each partitioning of a local maximum cluster.

Another advantage to choosing $K \leq -C_{max}$ is that it provides the system with a very simple test of whether two documents can be in the same local maximum cluster.

Theorem. If $K \leq -C_{max}$ then two negatively linked documents can occur in a local maximum cluster together only if they are positively linked to at least one common document.

Proof. Consider a local maximum cluster of n documents. Assume that there are two negatively correlated documents, x_α and x_β, in the cluster. By the previous theorem x_α must be positively correlated to over half of the (n-1) other documents in the cluster. Since x_α is not positively correlated to x_β it must be positively correlated to more than half of the remaining (n-2) documents. This is true of x_β also. Thus they must be positively correlated to at least one common document.

Next let us consider what value should be assigned to K to insure that $K \leq -C_{max}$. In Sec. 3.5 it was shown that the largest value that the estimated correlation can possibly take is (log N) where N is the number of available partitionings of the document file. Thus if we make K equal to (-log N) we will be assured that $K \leq -C_{max}$.

So far some reasons have been given indicating that it might be expedient from a practical standpoint to make K equal to (-log N). Let us now consider whether this value for K is justifiable theoretically.

It was noted in Sec. 3.5 that if the frequency counts are based on a finite number (N) of partitionings, then none of the probability estimates can fall between 0 and 1/N. This results in those correlations which might have been in the range $-\infty$ to (2-log N) being estimated to be $-\infty$ (or perhaps some value greater than (2-log N)). It was suggested

that those correlation estimates that are $-\infty$ by the formula might be more appropriately adjusted to some finite negative value, K, since a correlation of $-\infty$ implies that there is absolutely no chance of the two documents ever occurring together.

Thus K can be considered an approximation to the correlations in the range $-\infty$ to $(2-\log N)$ and it would seem appropriate that it assume some value within that range. Consider also what value K should assume as N approaches ∞. It is suggested that K should approach $-\infty$ as N approaches ∞ since those document pairs for which N_{ij} still equals 0 in the limit do in fact never occur together and $C(x_i^1 x_j^1)$ should be $-\infty$.

There are two other consequences to making $K=-\log N$ that should be noted. It gives the correlation a symmetric range about 0 ($-\log N$ to $\log N$). It also forces the correlation of documents that have never occurred together to always be less than the correlation of documents that have co-occurred $[(-\log N) < (2-\log N)]$.

The local maximum definition is therefore selected for use in this project. Its definition is extended to include biased clusters and it is required that $K = -\log N$. Hereafter we will refer to a local maximum cluster as just a cluster.

CHAPTER V

SEARCH PROCEDURE

The last component of the theoretical model is the procedure which transforms a request for information into the set of documents that comprise the answer. The first step in describing the procedure will be to make a number of definitions. Then a list of features that a suitable procedure should have will be given. Finally the particular procedure developed for this project will be described and analyzed.

5.1 Definitions

Definition: Request

A request for information from the system is defined to consist of two subsets of documents. One subset, $Y=(y_1,\ldots,y_s)$, contains those papers known by the user to be pertinent to the current search. The other, $Z=(z_1,\ldots,z_t)$, contains those papers that are known to be not pertinent. The Y subset must be non-empty but the Z subset can be empty.

Definition: Answer

An answer to a request is defined to be a cluster of documents which includes the Y subset of the request and excludes the Z subset.

Definition: Clustering Procedure

Any algorithm which transforms a request into an answer will be termed a clustering procedure (sometimes hereafter just

called a procedure). We will consider for this project only clustering procedures which are iterative in nature and which on each iteration change the contents of a certain set of documents, $S=(s_1,\ldots,s_u)$. Upon termination of the procedure S is to be the answer set. For most of the procedures considered here only a single change is made to S on each iteration. The S generated by the ith iteration can be distinguished by a subscript (S_i).

Definition: Convergent Procedure

A convergent procedure is one that terminates after a finite number of iterations.

Definition: Inconsistent Request

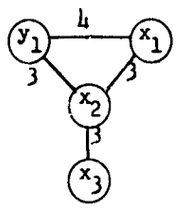
A request is said to be inconsistent if there is no answer cluster for any bias which satisfies the request.

Definition: Ambiguous Request

A request is said to be ambiguous if there is more than one answer cluster which satisfies the request. Note that one must consider all possible biases in determining ambiguity.

Requests with empty Z sets will generally be ambiguous. This is because larger and larger answer clusters can be formed by increasing the bias. For example, the request of Fig. 5.1 is ambiguous having the following four possible answers.

Answer	Bias
(y_1)	$-\infty \rightarrow -4$
(y_1x_1)	$-4 \rightarrow -3$
$(y_1x_1x_2)$	$-3 \rightarrow +7$
$(y_1x_1x_2x_3)$	$+7 \rightarrow +\infty$



Links not shown are -5

$Y=(y_1)$

$Z=(\ \ )$

Fig. 5.1. Ambiguous Request.

5.2 Attributes of a Good Clustering Procedure

In this section we shall list some characteristics which the clustering procedure should have. It will be assumed that the definition of a cluster of documents as given in Chapter 4 is suitable. If this is the case, then the basic objective of a clustering procedure would be to locate the appropriate cluster in an efficient way.

1. Request Satisfaction

If the request is unambiguous and consistent, then the procedure should produce the one cluster which satisfies the request.

2. Request Modification

If the request is ambiguous or inconsistent, then the procedure should be able to recognize this fact and should help the user to modify his request. This suggests that the procedure should allow close man-machine coupling so that information generated by the clustering process can be presented to the user for his examination and modifications to the request can be fed back into the system.

3. Convergence

The procedure should be convergent for every possible request and document network. Whether it is forming an answer cluster or determining

request ambiguity or inconsistency, it should never fall into a repetitive, non-terminating cycle.

4. Minimal Number of Iterations

The procedure should find the answer in as few iterations as possible. An excessively large number of deletions of previously added documents from the set being formed would be undesirable.

5.3 Description of Procedure

A description and flow chart of the procedure developed for this project will be presented in this section. An analysis of the procedure will be given in Sec. 5.5.

Fig. 5.2 is a block diagram showing the overall structure of the procedure. Before attempting to describe each block in Fig. 5.2 in detail let us make some general comments about the procedure.

There are three basic phases which the procedure can enter depending on the amount of bias required and the relationships of various documents and sets of documents.

Phase I: No Bias

The procedure starts in this phase, remains in it as long as no bias is required, and returns to it from Phase II if at some point the bias can be reduced to zero. The documents considered for addition to S in this phase are those (positive to S) which keep each y_i in Y positive to S (or at least increases its correlation to S) and keep each z_i in Z negative to S (or at least decreases its correlation to S). Of these candidates the one with the highest correlation to S is selected for addition to S. If at some point there are no more documents that are positive to S, then the procedure terminates. If there are documents

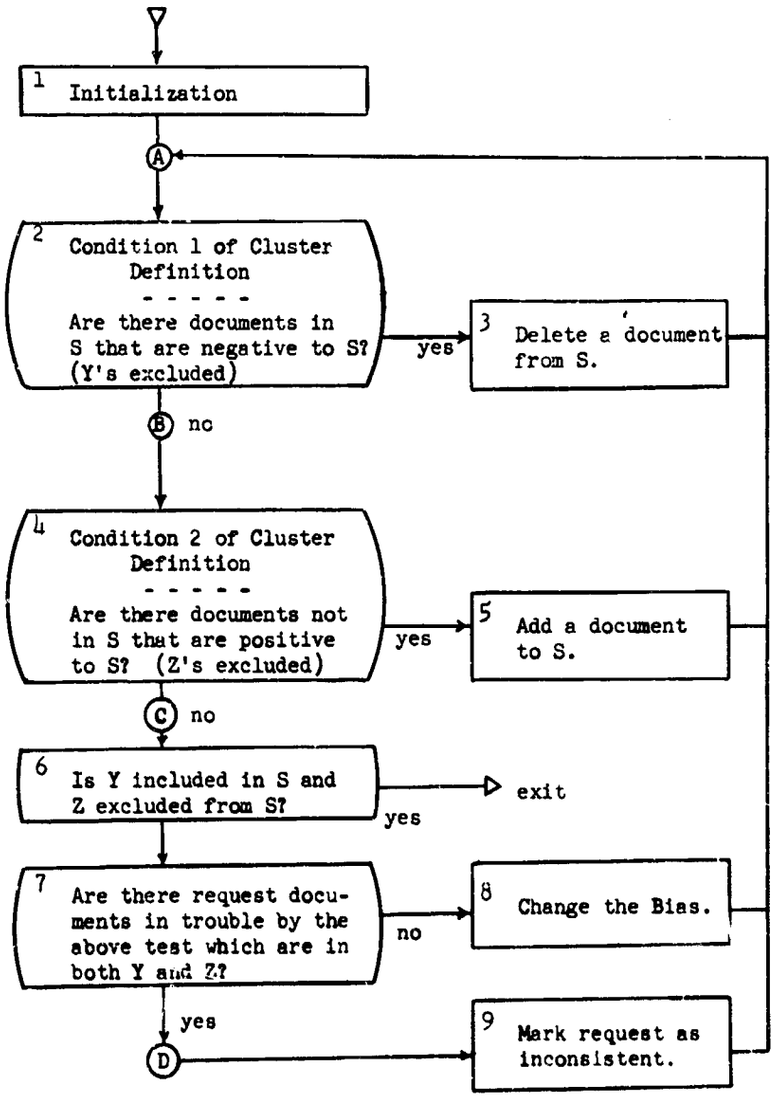


Fig. 5.2. Overall Flow Chart.

that are positive to S but none of them meet the above conditions with respect to Y and Z, then it is concluded that some bias will be needed and Phase II is entered.

Phase II: Bias

In Phase II the bias is either made positive enough to keep all the y_i's positive to S or made negative enough to keep all the z_i's negative to S. On each iteration those documents that are positive to S by the current bias are considered for addition to S. Of these candidates the document which requires the least bias when added to S is selected for addition to S. If at any time the bias becomes zero the procedure returns to Phase I.

When there are no more documents that are positive to S, the procedure either terminates or enters Phase III. Actually certain constraints are placed on the amount the bias can change on any one iteration. This means that all of the request documents may not be properly correlated to S (y_i's positive to S and z_i's negative to S) at the end of Phase II. If they are all properly correlated to S (i.e. the request is satisfied), the procedure terminates. If they are not yet properly correlated to S, the procedure enters Phase III.

Phase III: Monotonic Bias

The purpose of this phase is to either make positive to S certain y_i that are not currently positive to S or to make negative to S certain z_i that are currently negative to S. This is accomplished by allowing the bias to move in only one direction while suitable additions and/or deletions are made to S. One may not return to Phase I or II from Phase III. Phase III and the procedure terminate when the y_i's and z_i's are correctly linked to S.

The detailed flow charts for the general blocks of Fig. 5.2 will be greatly simplified if we first define a number of symbols.

Flow Chart Symbol Definitions

$\emptyset$: The null set.

$\cap$: Set intersection operator.

$\cup$: Set union operator.

$\bar{S}$: Set of all documents not in set S. (Complement)

$\subset$: Set inclusion: $A \subset B$ means set A is included in set B.

Y: The set of all documents specified as interesting by the user.

Z: The set of all documents specified as not interesting by the user.

S: The set which is being formed into the answer cluster by the procedure. ($Y \subset S$)

P: The set of all documents positively correlated to the set S by the current bias. A document in S is in P if it is positively correlated to the remainder of S.

Q: The set of documents included in P but not in S or Z. The document to be added to S will be chosen from this set. $Q = P \cap \bar{S} \cap \bar{Z}$

T: The set consisting of those documents in Q which will not require positive bias if added to S. Document t_i is in T if when it is added to S it will do one or both of the following operations for every document y_j in Y.

(1) Keep y_j positive to the new S. $C[y_j(S \cup t_i)] > 0$ (with 0 bias)

(2) Increase the correlation of y_j to S. $C(y_j t_i) > 0$ (with 0 bias)

V: The set consisting of those documents in Q which will not require a negative bias if added to S. Document v_i is in V if when it

is added to S it will do one or both of the following operations for every document z_j in Z.

(1) Keep z_j negative to the new S. $C[z_j(S \cup v_i)] \leq 0$ (with 0 bias)

(2) Decrease the correlation of z_j to S. $C(z_j v_i) \leq 0$ (with 0 bias)

X: The set of documents which are candidates for addition to S. If there are one or more documents in Q that require no bias if added to S, then X contains those documents. Otherwise it contains the documents that require a change in bias in only one direction.

W: The set of documents which are candidates for deletion from S. A document w_i is in W if it is negatively correlated to the remainder of S by the current bias and if it is not included in Y.

$$C[w_i(S \cap \bar{w}_i)] \leq 0 \qquad w_i \subset S \cap \bar{Y}$$

f: Number of positive links in the set S. (with no bias)

g_i: Number of positive links from document x_i to S. (with no bias)

d_i: Bias required for the set $(S \cup x_i)$. If $x_i \subset T \cap \bar{V}$ then d_i is just negative enough to keep each z_i negative to $(S \cup x_i)$. If $x_i \subset V \cap \bar{T}$ then d_i is just positive enough to keep each y_i positive to $(S \cup x_i)$. If $X = T \cap V$ then d_i is made 0.

BIAS: Current bias.

b_i: Allowable change in bias if x_i is added to S.

$$b_i = \text{minimum}\,[(d_i - \text{BIAS}), 1, 10/(f+g_i), C(x_i S)/(f+g_i)]$$

(C above is by current bias.)

R: The set of documents in X that would keep the bias at 0 or allow it

to be reduced to 0 if added to S.

$$|BIAS + b_i| = 0 \qquad \text{for all } x_i \subset R$$

We are now ready to present more detailed flow charts for the blocks of Fig. 5.2. Fig. 5.3 covers block 1, Fig. 5.4 covers blocks 2 and 3, Fig. 5.5 covers blocks 4 and 5, and Fig. 5.6 covers blocks 6-9. A brief comment is made to the right of each step in these detailed flow charts as an aid to understanding them. More precise statements of their functions are given in Sec. 5.5.

5.4 Earlier Procedures

For historical purposes and for comparison and analysis, let us briefly document some of the earlier procedures which were considered.

Procedure 1

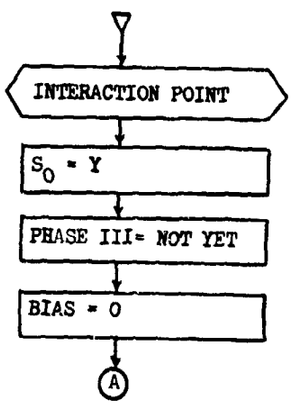
Briefly this procedure transforms a request into three subsets—

A: the set of documents related to the request.

B: the set of some of the documents not related to the request.

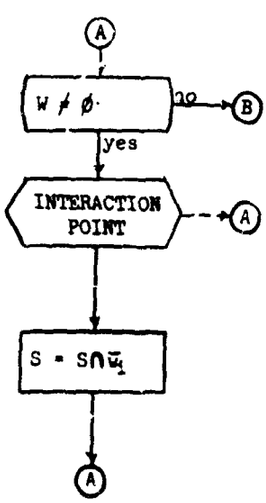
C: a 'limbo' set of documents positively correlated to both sets A and B.

Initially set A contains only those documents specified as interesting by the user, and set B contains those documents specified as non-interesting. On each iteration all documents positively (negatively) linked to A(B) and negatively (positively) linked to B(A) are added to A(B). Documents positively linked to both A and B are placed in limbo while those negatively linked to both are ignored. All changes to the sets A, B, and C are made concurrently at the end of each iteration.



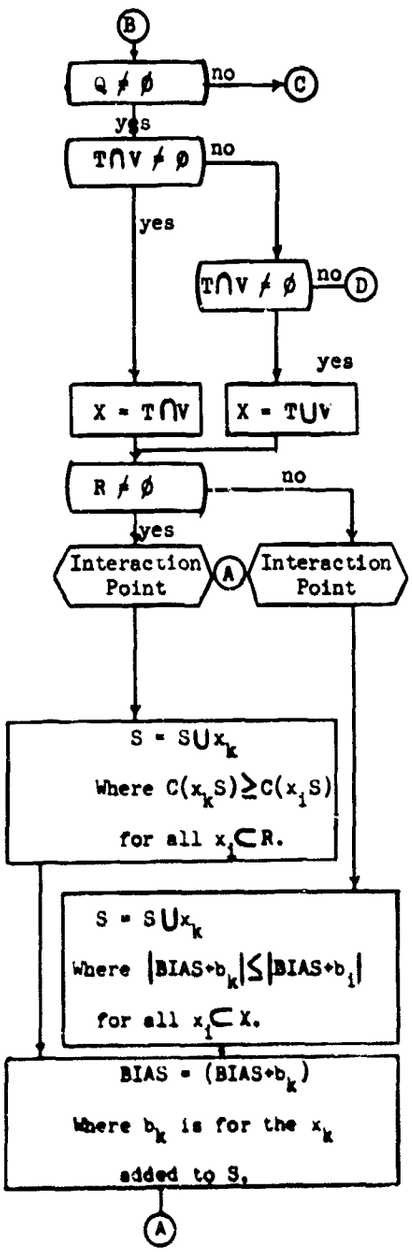
1. Allow user to specify initial Y and Z sets.
2. Put the interesting documents in S.
3. Indicate that the procedure is not yet in the third phase.
4. Start with an initial bias of 0.

Fig. 5.3. Initialization



5. Check if there are documents in S that are negative to the remainder of S.
6. Point at which information can flow between the user and the system. (e.g. status of clustering procedure, data on particular documents, modifications to the request.etc.)
7. Delete a document from S.

Fig. 5.4. Condition 1 and Deletions.



8. Check if there are any more documents positive to S.

9. Check if there are documents positive to S that keep (or try to keep) all the y's positive and all the z's negative.

10. Check if there are documents which require a change in bias in only one direction. Note that $T \cup V = (T \cap \bar{V}) \cup (V \cap \bar{T})$ at this point.

11. Load the set X with the candidates for addition to S.

12. Check if one or more documents in X can allow the bias to drop to zero.

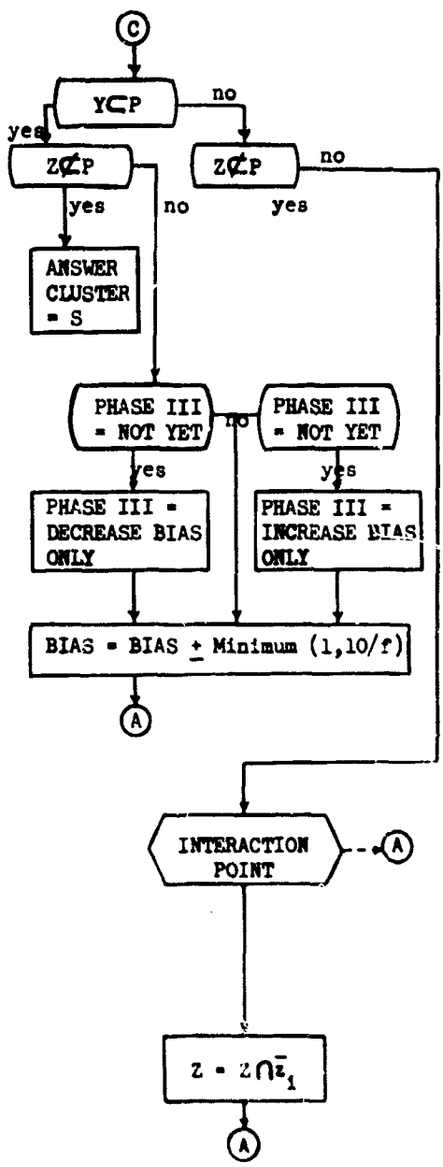
13. Point at which information can flow between the user and the system. (e.g. status of clustering procedure, data on particular documents, modifications to the request, etc.)

14. Add a document to S. The document x_k is the x_i in R for which $C(x_i S)$ is a maximum. (Based on current bias.)

15. Add a document to S. The document x_k is the x_i in X for which the magnitude of the allowable new bias, $|BIAS+b_i|$, is a minimum.

16. Change the bias if necessary. (Sign of b_k is modified by PHASE III to allow change in one direction only.

Fig. 5.5. Condition 2 and Additions.



Tests for Request Documents in Trouble

17. Check if all the documents in Y are positive to S.

18. Check if all the documents in Z are negative to S.

19. Termination of procedure. The answer cluster is S.

Phase III Bias Change

20. Check if this is the first time through Phase III.

21. Set PHASE III switch to allow bias to change in only one direction.

22. Make maximum change in bias. (The sign depends on the Phase III switch.)

Inconsistent Request

23. The request is considered inconsistent since the bias must go up and down simultaneously. The user is informed of this fact and allowed to ask questions and/or modify the request.

24. A document is chosen for deletion from Z if the user has not already modified the request.

Fig. 5.6. Phase III and other tests.

Procedure 2

This procedure is the same as Procedure 1 except that only one change is made to set A or set B at a time. Thus, the most positively correlated document is added and then the most negative document is deleted from each set.

Procedure 3

The basic difference between this procedure and Procedure 2 is that the criteria used to determine which document to add to set A or B is that it be most positively related to the original request instead of the current trial subset (S). Only those documents that are positively correlated to S are considered for addition. Within this set, selection is on the basis of correlation to the original request.

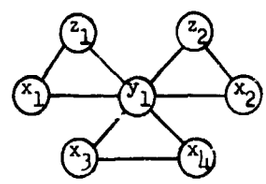
Procedure 4

This procedure attempts to combine the advantage of Procedures 1 and 2. All documents positively correlated to either sets A or B (but not both) should be added to them on the first iteration as in Procedure 1. Subsequently only single changes are made to the subsets as in Procedure 2.

Let us briefly note here why these earlier procedures were rejected. All of these procedures have a single subset B into which the documents considered not pertinent to the search are placed. This subset is treated just like the subset of pertinent documents and an attempt is made to form it into a cluster also.

The difficulty with such an approach can be seen by the example of Fig. 5.7. By the above procedures the non-pertinent set B is initialized with $Z=(x_1 x_2)$. Further additions to B are not possible because x_1

and x_2 are both negative to B. This is because the non-pertinent set is really not one cluster but two clusters. Since x_1 and x_2 are negative to B, one of them can be added to A. This will make x_3 and x_4 negative to A and divert the procedure from the desired cluster. Basically what has happened is that the usefulness of the documents in Z has been hindered by requiring that they form a single cluster.



Links shown are +5

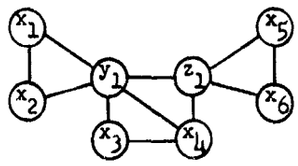
Links not shown are -6

$Y = (y_1)$

$Z = (z_1 z_2)$

Fig. 5.7. Example showing why non-pertinent documents should not all be grouped into one cluster.

This would lead one to suggest that perhaps a separate cluster should be formed around each document in Z. There are some reasons why this would not prove useful in addition to the fact that it would eat up an excessive amount of effort in the formation of non-pertinent clusters. Consider the example of Fig. 5.8. Let us assume that x_3 is added to A and x_5 to B on the first iteration. Now on the second iteration x_4 can be added to A because it is no longer positive to B. The cluster $(x_1 x_2 y_1)$ is again not found because the non-pertinent cluster formed around z_1 was $(z_1 x_5 x_6)$ instead of $(y_1 x_3 x_4 z_1)$. The point here is that the z_i's will be in a number of clusters and one does not know exactly which cluster to form around z_i in order to divert S in another direction.



Links shown are +5

Links not shown are -6

$Y=(y_1)$

$Z=(z_1)$

Desired cluster: $(y_1x_1x_2)$

Cluster to be excluded by z_1: $(y_1x_3x_4z_1)$

Fig. 5.8. Example of difficulty with forming clusters around non-pertinent documents.

5.5 Analysis of Procedure

Thus far the clustering procedure selected has been described and flow charted and a brief explanation of the purpose of each block has been given. Also certain earlier procedures have been briefly sketched. We shall now analyze the effectiveness of the selected procedure in terms of the objectives of Sec. 5.2.

5.51 Request Satisfaction

The procedure selected and most of the other procedures considered to date operate by making single changes to a set S which initially contains the Y set of the request. Documents not in S that are positively correlated to S are considered for addition to S and documents in S that are negative to S are considered for deletion from S. Let us first settle the question of whether it is possible in general for a procedure of this type to locate an answer cluster if one exists.

Theorem. It is always possible to transform a set S which initially contains only the Y set of the request into a (subset)

answer cluster if one exists by successively adding to S documents that are positively correlated to S.

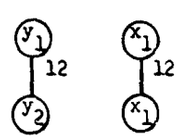
Proof. The proof of this theorem will be constructive.

(1) Initialize the set S with Y.

(2) If S coincides with the answer cluster A, the procedure can terminate.

(3) Otherwise, consider the set of documents $(A \cap \bar{S})$ yet to be added to S to form A. By the definition of a subset cluster in Sec. 4.2, $(A \cap \bar{S})$ must be positively correlated to S and thus there is at least one document in $(A \cap \bar{S})$ that is positively correlated to S. Add this document to S and go back to Step (2). QED

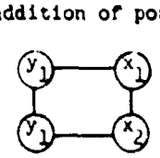
Note that this theorem is true only for subset clusters. We can show that it does not hold for local maximum clusters by the example of Fig. 5.9. The set $(y_1 y_2 x_1 x_2)$ forms a local maximum cluster, but it cannot be reached from the set $S_0 = (y_1 y_2)$ by the addition of documents positively correlated to S.



Links now shown are -5

Fig. 5.9. Local maximum cluster not accessible to procedure.

Even when $K \leq -C_{max}$ the theorem still does not hold for local maximum clusters. In the network of Fig. 5.10 the set $(y_1 y_2 x_1 x_2)$ again forms a local maximum cluster, but it cannot be reached from the set $S_0 = (y_1 y_2)$ by the addition of positively correlated documents.



Links shown are +4

Links not shown are -5

Fig. 5.10. Local maximum cluster not accessible to procedure.

Actually it may be a distinct advantage if procedures of the type being considered cannot reach certain local maximum clusters. It was noted in Sec. 4.5 that a procedure which produces subset clusters only would be preferred over one that results in local maximum clusters; but that such a procedure had not been found. The above theorem and comments show that procedures of the type selected can generate for a given request all of the subset clusters which satisfy a given request. In addition they may locate some (but not all) of the additional local maximum clusters which satisfy the request.

Let us now observe that we have so far only proved that a suitable clustering procedure of the type suggested may exist. The 'constructive proof' of the theorem does not indicate how to choose the correct document to add to S in Step (3) if several documents are positive to S. One could, of course, try all possibilities. Let us represent these possible additions by a tree where each branch out of a node represents the addition of a positively correlated document to S. In the example of Fig. 5.11 there are three documents positively correlated to y_1, two positively correlated to the set (y_1x_1), etc.

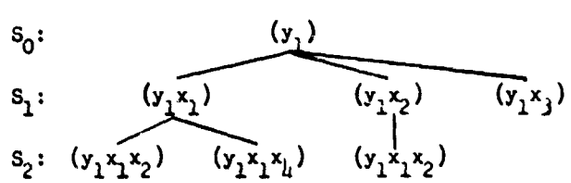
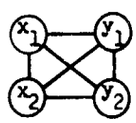


Fig. 5.11. Possible additions to S.

A procedure which traversed all of the branches of such a tree would be assured by the preceding theorem of finding an answer (subset) cluster if one existed. However, one can quickly convince himself that

such an exhaustive examination of all possible positively correlated additions is, in general, completely impractical because of the magnitude of the task. What is needed is some way of determining which of the positively correlated documents should be added to S on each iteration.

There will, of course, be cases where the answer cluster is obtained no matter which of the positively correlated documents is added to S on a given iteration. A simple example of a request and network for which this is the case is given in Fig. 5.12. On the first iteration one can add either x_1 or x_2 and still end up with the answer cluster $(y_1 y_2 x_1 x_2)$.



Links shown are +4

Fig. 5.12. Network where it does not matter which document is added to S first.

However, in the more general case the choice of which document to add to S on each iteration is a very critical aspect of the clustering procedure. The answer to a request may not even be found if the wrong document is added to S on one or more of the iterations. As an example, consider the network and request of Fig. 5.7. If the procedure were to add x_1 to S on the first iteration, then $(y_1 x_3 x_4)$, the only cluster which satisfies the request, would not be found.

Let us now describe the criteria used by the procedure of Sec. 5.3 to decide which document to add to S on each iteration and note how these criteria might help in obtaining an answer cluster if one exists.

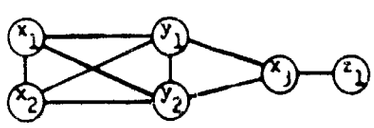
In Steps 9-11 of Fig. 5.5 preference is given to documents that are

positively linked to each y_i (or else leave the y_i positive to S) and negatively linked to each z_i (or else leave the z_i negative to S). The network of Fig. 5.7 serves as an example of how this preference might aid in obtaining the answer cluster. Documents x_3 and x_4 are considered for addition to S before x_1 and x_2 and the answer cluster $(y_1x_3x_4)$ is obtained.

Steps 12 and 15 of Fig. 5.5 are for the purpose of minimizing the bias on each iteration and will be discussed when we talk about request modification and ambiguity.

In Step 14 the document which is selected for addition to S is the one that has the highest positive correlation to S from among those documents that have met all of the earlier criteria.

The theorem at the beginning of this section shows that the only operation that a procedure needs to perform is the addition of positively correlated documents to S if the appropriate document to be added on each iteration can be determined. If, in fact, the procedure mistakenly adds on a given iteration a document which is not part of the answer, then it may still be possible to arrive at the answer if the procedure is allowed to also delete documents that have become negatively correlated to S (Steps 5-7 of Fig. 5.4). In the network of Fig. 5.13 the answer $S_4=(y_1y_2x_1x_2)$ is obtained even though $S_1=(y_1y_2x_3)$.



Links shown are +4

Links not shown are -5

Fig. 5.13. Network showing that the procedure must be allowed to delete as well as add.

Despite the above features which help in the choice of the document to be added on each iteration, there are still cases where the

procedure of Sec. 5.3 does not find an answer cluster even when one exists. Consider the request and network of Fig. 5.14. Documents x_1, x_2, and x_3 are linked to the documents in sets Y and Z by exactly the same values and are all candidates for addition to S on the first iteration. If the first document to be added is either x_1 or x_2, then the procedure finds the cluster $(x_1x_2y_1y_2)$ which is the only valid answer cluster for the request. If, however, x_3 is added to S first, then the procedure reaches a point where no bias can be chosen which will simultaneously keep y_1 and y_2 positive to S and x_1 negative to S and the request is judged inconsistent.

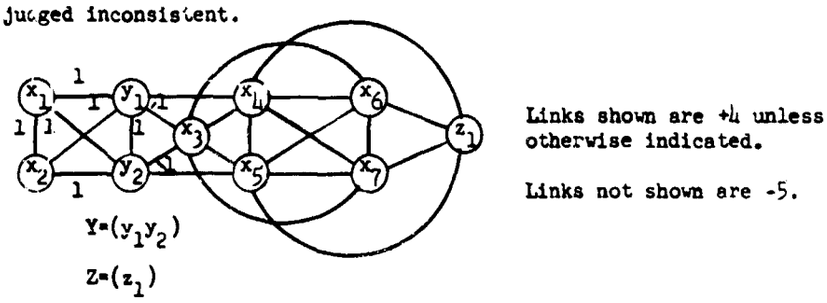


Fig. 5.14. Network illustrating the difficulties involved in knowing which document to add to S on a given iteration.

The alternatives open to the procedure for the network of Fig. 5.14 are shown in the decision tree of Fig. 5.15. It should be pointed out that all of the procedures discussed in this chapter decide which document to add to S on each iteration on the basis of the relatedness of the document being considered to the documents in the S, Y, and Z sets only. The inter-relatedness of the documents not in S, Y, and Z is not a factor in the selection. Indeed, from a practical standpoint, it cannot be used as a factor in the decision, since it would necessitate

considering the consequences of adding subsets of documents instead of single documents and for r documents under consideration there are as many as 2^r subsets to consider.

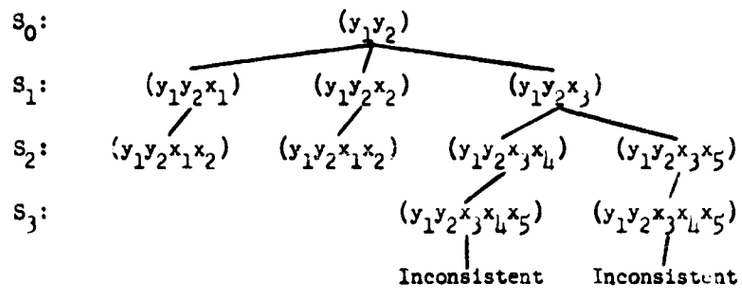


Fig. 5.15. Tree illustrating the possible additions to S for the network and request of Fig. 5.14.

If the documents to be added to S are chosen on the basis of their relatedness to the S, Y, and Z sets only, then there is no way of determining whether to add x_1, x_2, or x_3 to S_0 in Fig. 5.14. If one cannot tell beforehand whether to add x_1, x_2, or x_3, then perhaps a procedure should be devised that would at some later point back up and try another 'direction' if S becomes inconsistent with the request. In other words, if x_3 is added to S in Fig. 5.14, perhaps one could on the fourth iteration remove a subset containing x_3 from S and add x_1 and x_2. Such a step would require not only that the procedure be able to know which subset to remove but also that it remember all of the previous S sets so that it would not fall into a non-terminating cycle. This approach is also rejected as not being practical.

The philosophy adopted for this research project is that for those cases where the procedure has difficulty in locating an answer, that the user should be coupled into the procedure to guide the process in the right direction. This is the reason for the interaction points in the

procedure. The user can step in before the addition or deletion of any document and over-ride the decision of the procedure by changing the request, if he decides the cluster is moving into the wrong area. In the case of Fig. 5.14 the user could easily obtain the cluster $(y_1y_2x_1x_2)$ by specifying any member of the set $(x_3x_4x_5x_6x_7)$ to be uninteresting.

5.52 Request Modification

If the request as initially specified by the user is inconsistent or ambiguous, then some additional interplay may be needed between the system and the user so that it can be appropriately modified. Let us make some general comments about the suitability of the clustering procedure for interaction with a user and then deal specifically with the problem of what particular type of interaction is needed to resolve request inconsistency and ambiguity.

If a clustering procedure is to be used in close coupling with the user, then the process should be divisible into small units of effort. Each unit of effort should produce some useful piece of information that can be presented to the user and the user should be able to make changes to the request between these units of effort.

The natural unit of effort is, of course, the iteration. The information produced by the iteration is the document to be added to or deleted from S. The change in the request can be the response of the user to the document presented. An iterative clustering procedure, therefore, lends itself very well to close supervision by the user.

There are four interaction points shown for the procedure of Sec. 5.3. The initial specification of the request is made at Step 1. In Step 6, which immediately precedes the deletion of a document from S

(Step 7), the user is given a chance to examine the document to be deleted and to modify his request if he wishes to. In Step 13 the user is allowed to ask questions and change the request before the addition of a document to S. In Step 23 the request is judged inconsistent and the user is again allowed to obtain information from the system and modify the request. These four steps provide an interaction point before each change to S and on each iteration of the procedure. A description of the full range of questions that can be asked by the user at these interaction points will be given when the retrieval language is presented in Chapter VIII.

Let us now consider the problem of determining whether a request is inconsistent or ambiguous. One test for inconsistency has already been given. The last theorem or Sec. 4.5 states that in order for two negatively correlated documents to be in the same cluster they must be positively linked to at least one common document (if $K \leq -C_{max}$). Let us present three more theorems pertaining to whether two documents are assured of being in a cluster together or not.

> Theorem. Two documents x_1 and x_2 can be positively correlated to exactly the same documents and negatively correlated to the same documents and still not be in the same clusters.

Proof. Consider the example of Fig. 5.16. The documents x_1 and x_2 are both positively correlated to x_3 and x_4 and negatively correlated to x_5. However, $(x_1 x_3 x_4 x_5)$ forms a cluster which contains x_1 and excludes x_2. The link between x_1 and x_2 is dotted to show that they can be positively or negatively linked and the theorem would still be true. QED

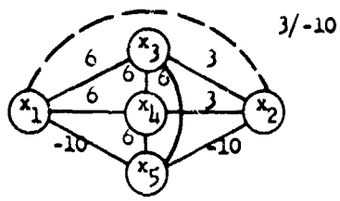


Fig. 5.16. Network with x_1 and x_2 not in the same cluster.

<u>Theorem</u>. A document x_1 can be positively correlated to every document that a document x_2 is negatively correlated to (and vice versa) and x_1 and x_2 can still be in a cluster together.

<u>Proof</u>. The networks in Fig. 5.17 offer a proof of this theorem. The documents x_1 and x_2 are in the same cluster $(x_1x_2x_3x_4)$ and yet the values of their links to x_3 and x_4 have the opposite signs. QED

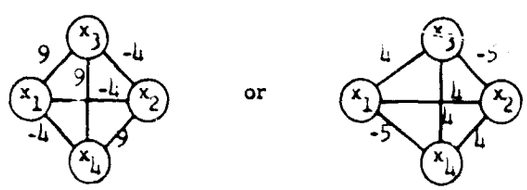


Fig. 5.17. Network with x_1 and x_2 in the same cluster.

If one adds the restriction that $K \leq -C_{max}$, then the above theorem is only true for positively correlated document pairs. The last theorem of Sec. 4.5 states that when $K \leq -C_{max}$ two negatively correlated documents can occur in a cluster together only if they are positively linked to one or more of the same documents.

<u>Theorem</u>. Two documents x_1 and x_2 are assured of always being in the same clusters together if $C(x_1^1 x_2^1)$ is greater than the absolute magnitude of the difference in the correlations of x_1 and x_2 to every possible subset of other documents.

<u>Proof</u>. To prove this theorem let us assume that x_1 and x_2 are not in the same cluster and then show a contradiction. Let us say that x_1 forms a cluster with the set of documents A which does not include x_2 as indicated in Fig. 5.18.

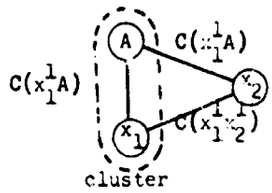


Fig. 5.18. Network for proof of theorem.

Since $x_1 \cup A$ is a cluster:

$$C(x_1^1A) > 0$$

$$C[(x_2^1)(A \cup x_1^1)] \leq 0$$

Rearranging and combining these inequalities--

$$C(x_2^1A) + C(x_1^1x_2^1) \leq 0$$

$$C(x_1^1x_2^1) \leq -C(x_2^1A)$$

$$C(x_1^1x_2^1) \leq C(x_1^1A) - C(x_2^1A)$$

$$C(x_1^1x_2^1) \leq |C(x_1^1A) - C(x_2^1A)|$$

This last inequality is in conflict with the part of the theorem which states that for any A:

$$C(x_1^1x_2^1) > |C(x_1^1A) - C(x_2^1A)|$$

QED

These three theorems give some indication of the difficulties involved in determining if two documents are in the same cluster on the basis of the links from those documents to the other documents of the network. The third theorem here and the last theorem of Sec. 4.5 would help in some cases to determine whether documents can co-occur in

clusters, but they have far from general applicability.

It was, therefore, concluded that there was no easy test which could be initially performed to determine if the request was inconsistent or ambiguous. The tests which were devised consisted of attempts to find one or more clusters which satisfied the request and required at least as much effort as the finding of an answer for a valid request. It was decided that the procedure should not concern itself with the problems of request ambiguity and consistency at first but should assume that the request is valid and start trying to find the answer cluster. If during this process it was decided that the request was inconsistent, then the user would be notified of this fact. And if the user was still worried about ambiguity after a cluster had been found, then he could perform some further searching to satisfy himself that he had retrieved what he was after.

It was further decided that the user should be given the option of being able to interact with the procedure on any or all of the iterations in order to monitor what was being retrieved and in order to modify the request if the situation demanded it. Thus a user who suspected his request to be ambiguous or inconsistent could carefully watch what documents were being added to S to make sure that he was obtaining what he wanted, while the user who had confidence in the validity of his request could let the procedure run to completion unattended.

The rule which was followed in the design of the procedure of Sec. 5.3 was, therefore, to allow the user to interact at any point he wished to (and especially in cases where an invalid request was suspected), but to never require that he respond before the clustering could continue. Thus in Steps 23 and 24 of Fig. 5.6 the request appears

to be inconsistent. The user is given the chance of changing his request if he wishes. If no change is made, then the procedure picks a document to be deleted from Z so that clustering can continue.

Also in the case of ambiguity the procedure is designed to find the most reasonable answer cluster it can for presentation and not to depend on the user to clear up the ambiguity. This is the purpose of Steps 12 and 15 in Fig. 5.5. If two clusters with different biases are both valid answers to the request, then the one with the smaller bias is considered a better selection. Therefore, an attempt is made to make the bias as small as possible on each iteration.

5.53 Convergence

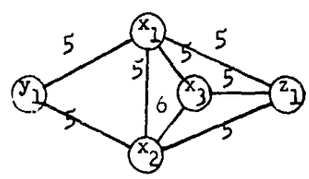
A major objective in the design of the clustering procedure is to insure that it will always terminate in a finite number of steps for every possible document network and every possible request. A procedure which occasionally drops into an infinite loop would, of course, be completely unacceptable. The possibility of an infinite loop comes about because of the fact that the procedure can delete as well as add documents to the set S. If on some iterations the set S has the same composition as it had on a previous iteration, and if the procedure does not remember all of the previous S sets, then a non-terminating cyclic behavior is possible.

In Phase I of the procedure convergence is assured by the following theorem.

> Theorem. A procedure is convergent if the only types of changes made to the set S being formed are the addition of documents positively correlated to S and the deletion of documents negatively correlated to S.

Proof. The internal correlation of S is increased by the addition of a document positive to S. It is also increased by the deletion of a document negative to S. Thus C(S) increases monotonically as these two types of changes are made to S. This means that C(S) is larger on a given iteration than for any earlier iteration. Therefore the composition of S must be different on each iteration. Since there are at most 2^n possible S sets (for a network of n documents), there are at most 2^n iterations of the procedure before it terminates. QED

If the bias of the network is changed as it is in Phase II, then the above theorem no longer insures convergence. For example, the following steps might possibly be taken by a hypothetical procedure in trying to obtain a cluster in the network of Fig. 5.19.



Links not shown are -6

Fig. 5.19. Network which may cause a procedure to cycle.

(1) $S_0=(y_1)$

(2) $S_1=(y_1x_1)$ $C(x_1S_0)=5$

(3) $S_2=(y_1x_1x_2)$ $C(x_2S_1)=10$

(4) Bias =-2 to keep z_1 negative

(5) $S_3=(y_1x_1x_2x_3)$ $C(x_3S_2)=1$

(6) Bias =-3 to keep z_1 negative

(7) $S_4=(y_1x_1x_2)$ $C(x_3S_4)=-1$

(8) Bias =-2 to just keep z_1 negative

At this point the procedure returns to Step (5) in a never ending loop.

In order to avoid such cycles Phase II of the procedure selected (Sec. 5.3) synchronizes each change in bias with the addition of a document to S. If the document being added increases the internal correlation of S by k bits, then a decrease in bias is allowed which decreases the internal correlation by up to k bits. Thus the total internal correlation of S is still increased on each iteration and convergence is again assured.

In the above example Phase II would combine (synchronize) Steps (3) and (4) and allow the bias to still be -2 bits. Steps (5) and (6) would also be combined but the bias would only be allowed to go to -2.2 bits ($b_3=C(x_3S)/5$). Step (7) would not be taken because x_3 would not be negative. [$C(x_3S)=0.6$].

Thus far we have talked about the effect of decreasing the bias on convergence. An increase in bias does not reduce the total internal correlation and would not necessarily have to be synchronized with additions to the set. For purposes of symmetry, however, bias increases are placed under the same restrictions that bias decreases are.

Finally, let us consider convergence in Phase III. Bias changes that are not synchronized with the addition of a document are now allowed, but the bias can change in only one direction. We have already shown that the clustering procedure is limited to a finite number of iterations for a given bias (by the above theorem). Phase III permits only a finite number of bias changes so the total number of iterations is finite and we are assured of convergence once more.

5.54 Minimum Number of Iterations

Those steps which are taken to improve the proper selection of the document to be added on each iteration should also help to decrease the

number of deletions necessary on later iterations. We have already discussed the problem of choosing the correct document on a given iteration.

PART THREE: EXPERIMENTAL SYSTEM

In the last three chapters the basic components of the theoretical model were presented. The next three chapters describe the experimental system which was developed so that the ideas and concepts of the model could be tested in a realistic environment.

The four aspects ot the experimental system that will be covered are:

Chapter VI: Computational Facilities and Data Base

Chapter VII: File Structure

Chapter VIII: Interaction Language

CHAPTER VI

COMPUTATIONAL FACILITIES AND DATA BASE

There are two projects at M.I.T. on which this research endeavor is highly dependent. Project MAC supplied the computational facilities for the experimental phase of the project. The Technical Information Project supplied the document collection and data base on which the experiments were performed. In addition these two projects provided considerable other technical and general assistance. Since the computational facilities and data base are essential components of the experimental system, they will now be described.

6.1 Computational Facilities

The experimental portion of this project was designed for the Project MAC time-sharing system[21]. In this section we shall describe the MAC system and note some of its features that are of particular significance to this project. A more complete description of the objectives and characteristics of the MAC system can be found in the references[12,21]

Fig. 6.1 is an abbreviated diagram of the equipment included in the MAC system. Some of the more significant parameters of this equipment are given in Fig. 6.2. All of the equipment shown in Fig. 6.1 is physically located at M.I.T.'s Technology Square with the exception of the time-sharing consoles. Over 100 of these consoles are located at various places on the M.I.T. campus and can be connected to the 7750

through the M.I.T. telephone exchange. There are also MAC consoles at more remote locations. Indeed any TWX or TELEX telegraph station has the capability of being connected into the MAC system. Each console has a dual purpose. It communicates to the 7750 what characters have been typed on its keyboard and it also types out messages originating in the 7094 that are routed to it through the 7750.

In a time-shared computer a number of consoles can be simultaneously connected into the system and can independently obtain the services of the central processor. A limit is normally placed on the number of consoles that can be actively connected at any one time. The purpose of this limit is to help insure that those who are connected will be promptly serviced. The current limit for the MAC system is 30, but it varies periodically as changes and improvements are made in the system.

One of the core storage banks (bank A) contains the time-sharing supervisory program. This program decides which of the users who currently want service has the highest priority. The program of the highest priority user is loaded into core (bank B) from the disc or drum and allowed to run for up to two or three seconds. Then the program is removed (swapped) and the new highest priority program is loaded and run.

The IBM 1302 disc is used for permanent or temporary storage of programs and data. The data file to be described in the next section is stored on this disc as well as programs which arrange and structure it and allow the user to communicate with it.

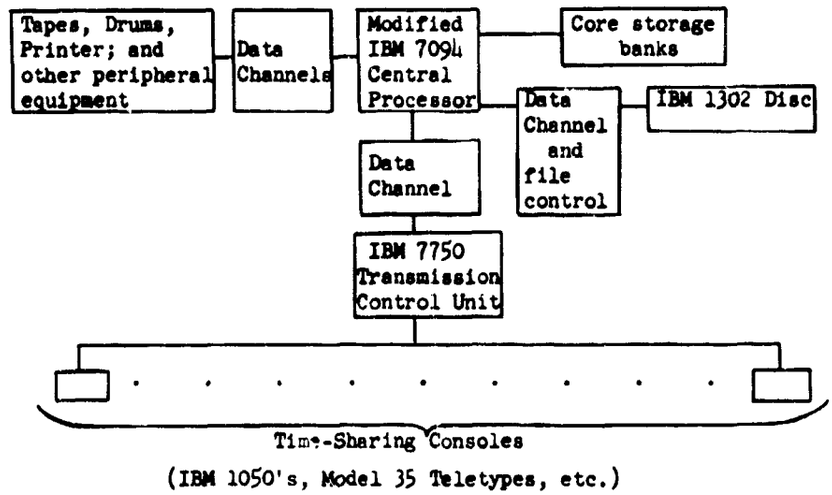


Fig. 6.1. Project MAC Equipment Configuration.

Basic word size	36 bits
Core storage operating cycle (to read or write 1 word)	2 microseconds
Size of core storage banks A and B	32,768 words each
1302 disc storage capacity (80,000 tracks of 432 words each)	34.56 million words
1302 Disc scan time	50-180 milliseconds to position on track; 50 milliseconds to read track.
Transmission rate to and from time-sharing consoles	about 100 bits/second.
Physical limit on number of consoles connected to 7750 (The actual limit is lower)	112

Fig. 6.2. Significant Parameters of MAC System.

6.2 Data Base

The basic data needed to implement the theoretical model of Part Two is a document collection and a file of partitionings of that collection. The document collection selected is described in the next section and the final section of the chapter contains a discussion of the type of partitioning data that will be used.

6.21 Document Collection

The Technical Information Project at M.I.T. is currently accumulating a file of information on articles found in the physics periodical literature.[29] This file covers about 26,000 articles from 25 different journals. Fig. 6.3 lists the names of the journals and the extent of the coverage in terms of volumes. The time period covered for each journal is 1 Jan. 1963 to the present. Note that all of the articles in the volumes listed are included.

One can gain some appreciation of the extent of the coverage of the file by noting that the 25 journals account for over 50% of the articles that are abstracted for Physics Abstracts.

The file is currently growing at the rate of 1500 articles a month. Periodically new journals are added to the file. Journals to be included are selected on the basis of a statistical analysis of their citations. This selection criteria is described more fully elsewhere .

The information extracted for each article is the journal identification, volume and page number, title, author(s), author location(s), and coded bibliographic citations. Fig. 6.4 is an example of the information available in a given article. Fig. 6.5 summarizes some of the parameters of the file.

Journal	Journal Code	Volume Range	Number of Articles
1. Annals of Physics	384	21-36	275
2. Applied Physics Letters	646	2-8	592
3. Canadian Journal of Physics	55	41-44	531
4. Helvetica Physica Acta	43	36-38	202
5. Indian Journal of Physics	164	37-39	165
6. Japanese Journal of Applied Physics	612	2-4	328
7. JETP Letters	821	1-2	65
8. Journal of Applied Physics	11	34-37	1643
9. Journal of Chemical Physics	12	38-44	3398
10. Journal of Mathematical Physics	227	6	193
11. Journal of the Physical Society of Japan	80	13-20	759
12. Nuovo Cimento	17	27-40	1385
13. Nuclear Physics	682	46-75	1529
14. Physica	21	29-31	359
15. Physical Review	1	129-142	3713
16. Physical Review (Series B)	199	133-140	1791
17. Physical Review Letters	41	10-16	1585
18. Physics Letters	49	3-20	2880
19. Physics of Fluids	799	6-8	607
20. Proceedings of the Physical Society (London)	3	81-87	738
21. Progress of Theoretical Physics (Kyoto)	29	29-34	392
22. Soviet Journal of Nuclear Physics	825	1	144
23. Soviet Physics - JETP	669	16-21	1465
24. Soviet Physics - Solid State	510	5-7	814
25. Soviet Physics - Technical Physics	790	6-10	898
		178	26,471

Fig. 6.3. Journals covered by the physics periodical file of the Technical Information Project (March 20, 1966).

Physical Review
Volume 136
Page: 0001
Spectral properties of a single-mode ruby laser. Evidence of homogeneous broadening of the zero-phonon lines in solids
Tang, C. L.
Statz, H.
Demars, G. A.
Wilson, D. T.
Waltham, Massachusetts
Raytheon Research Division

J001 V102 P1252	J001 V112 P1940	J001 V128 P1726
J001 V133 P1029	J011 V034 P1682	J011 V034 P2289
J011 V034 P2935	J018 V187 P0493	J018 V195 P0587
J041 V006 P0106	J046 V009 P0399	J646 V002 P0222

Search completed, 257 articles.
1.99 seconds, 129.1 articles/sec.

Fig. 6.4. Example of the information available on a given article. The last four lines are the coded citations (J=journal, V=volume, P=page).

Number of articles available on the disc	26,471
Time span covered	Jan. 1963 to present

Files key-punched but not currently on the disc:

(1) Physical Review, Vol. 77-128 (1950-1962)

(2) Journal of Chemical Physics, Vol. 28-37 (1958-1962)

Average number of articles per track	6.7
Average number of authors per article	2.02
Average number of citations per article	12.
Average number of words per title	8.

Fig. 6.5. Parameters of T.I.P. data file (March 20, 1966).

Initially the information is key-punched on IBM cards. After some preliminary editing and correction it is then loaded on the IBM 1302 disc of the Project MAC computer. On the disc it undergoes more editing and is transformed into the format selected for permanent storage (see Sec. 7.1).

The T.I.P. file has certain features which make it attractive for use by this research project. It is of sufficient size and interest to attract serious users. The articles covered contain a substantial number of citations which will be shown to be of particular use shortly. The generation of the data involves only clerical and mechanical operators (i.e. no human indexing or evaluation is required).

6.22 Partitions

Some of the advantages to having a retrieval system based on user feedback were discussed in Chapter II. A basic objective of this project was stated to be the investigation of the feasibility of such a system. In Chapter III a particular form that user feedback could take was described. Basically it consisted of each interaction of a user with the document collection resulting in a partitioning of the documents into a set of interesting documents and a set of uninteresting documents.

This type of interaction was described so that one could better understand the motivation behind the choice of the sample space, probabilities, and other aspects of the theoretical model. Actually the theoretical model as developed in Chapters III, IV, and V in no way requires that the partitionings on which the probability estimates are based be generated by user interactions. Any type of partitioning data

could be used, even data that has been arbitrarily contrived. Indeed, in the experimental system another type of partitioning was used because usage data is not readily available at the present time.

Let us consider whether a change in the type of partitioning data employed by the experimental system will impair its effectiveness in testing whether a system based on usage data is feasible. First it can be observed that much of this investigation has very little, if any, dependence on the particular type of data being utilized. For example, the objective of a procedure of Chapter V is to find a cluster of documents. Its ability to do this could be examined and tested as well on the set of arbitrarily selected partitionings of a hypothetical document collection as on a set of partitionings generated by the interaction of a real user population with a real library.

There are some reasons, however, why it is advisable to use a set of partitionings for the experimental system that is not artificial and which resembles usage data as closely as possible. For example, the utility of the interaction points in the procedure are best tested by real users. This, of course, requires a data base which produces results that a user would be interested in. Also the overall effectiveness of the system to produce useful results can be properly evaluated only in a realistic environment.

With this objective in mind let us now consider what types of partitionings are available for the document collection described in the last section. There were five types of partitionings that were evaluated for this project. They consist of dividing the set of documents into two subsets based on whether or not the documents--

(1) were written by a given author.

(2) contain a certain word in their titles.

(3) cite a given article.

(4) were cited by a given article.

(5) occur in a given subject category.

Thus by criterion (1) there are as many partitions as there are authors in the file, with each author dividing the document file into those papers he wrote and those he didn't write.

A detailed analysis of each of the above types of partitionings was conducted on one volume (vol. 128) of the Physical Review. Certain tests were also conducted on much larger parts of the document collection. Let us summarize the results of these tests and evaluate each of the five partitioning criteria.

(1) Author Partitions.

Difficulty was encountered in devising an algorithm that could determine if two author names referred to the same individual. A surprisingly large number of the authors were not consistent in the way they gave their names. Given names were sometimes supplied in full, sometimes represented by an initial, and sometimes left off altogether. The method which yielded the best results required an exact match of the surname and required that given names either match exactly or match on the first letter if one of the names was a single letter (i.e. an initial). We at first allowed a missing given name to be a match for anything, but this produced too many false matches. We, therefore, required that in order for a match to occur the number of given names had to coincide.

Another difficulty was that roughly half of the authors were the authors of only one paper. This produced a large number of partitionings with only one document in the subset of "interest", with the consequence

that there were many of the papers that did not co-occur with any other paper by this method.

A third drawback to this type of partitioning arises in those cases where an author changes his area of interest and publishes articles on unrelated subjects.

(2) Word Partitions.

If every title word is allowed to create a partition of the file, then practically every document will co-occur with every other document because of the common function words like "of", "the", etc. The alternative is to try to identify and exclude from use function words. However there is no clear distinction between function words and keywords. It is fairly clear that certain words should be eliminated if co-occurrences are to be meaningful. However there is a large grey area of words such as "effect", "wave", "theory", of "electronic" that in and of themselves create little meaningful linkage, but in combination with other words are very significant. The approach adopted for the tests was to eliminate all words that occurred in over 5-10% of the titles. This unfortunately eliminated the word "nuclear" while allowing words like "between" and "theory" to create partitions.

A second problem in using word partitions is that there are a number of words which differ from each other by only a suffix (i.e. superconductor, superconductors, superconducting, superconductive, superconductivity). A table was compiled of 40 of the more commonly occurring suffixes of the title words in the document file. All of the words which differed from each other by one of these suffixes were considered equivalent in creating partitionings.

An even more basic problem involves the use of synonomous words for the same concept. Some type of thesaurus would be necessary to link up articles with synonymous title words. It was decided that there are too many problems involved in the generation (or selection) and use of a thesaurus to warrant any effort in this direction in this research endeavor.

(3) Cite-same Partitions.

When two papers cite one or more of the same papers they are said to be bibliographically coupled. A number of studies have been conducted to analyze the characteristics of bibliographic coupling[28]. These studies indicate that bibliographic coupling constitutes a very meaningful and important type of relationship between papers, especially in those document collections which have a sizable amount of citation information. In the T.I.P. file of Sec. 6.21 there are an average of 12 citations per article and strict editorial policies make it easy to identify the articles that are cited.

(4) Cited-by same Partitions.

We note from Fig. 6.3 that the documents covered by the T.I.P. file have all been written in the last three years. Due to the time required to review and publish articles there is usually a period of at least six months between the time an article is published and the time citations to it begin to appear in the literature. And even after a span of two to three years over half of the articles in the Physical Review have still not been cited by subsequent articles in the Physical Review[27]. Thus this type of partitioning will have a very small yield for the current T.I.P. file in terms of the number of documents that will occur in one or more subsets of interest and in terms of the total number of

co-occurrences of articles that will be generated.

(5) Subject Category Partitions.

A subject index is published of the articles in the Physical Review. Each article is assigned to from one to four categories. These category groupings form another type of file partitioning. However, not all of the 25 journals have subject indexes and there is no general agreement on category headings among the indexes that do exist. Also the categories even within a single journal are constantly changing.

In the beginning we decided to use all five of the above types of partitionings for the experimental system with the hope that each would add meaningful links to the resulting document network. However, the results of the above tests led us to conclude that the use of criterion (3) only would result in an adequate set of partitionings, and would avoid some of the problems encountered in using the other criteria. The final experimental system is, therefore, based on partitionings of type (3) only.

CHAPTER VII

FILE STRUCTURE

Thus far we have described the computational facility on which the experimental system operates and the data it uses. Let us now turn our attention to the problem of how the data should be arranged and structured for storage on the disc or in core. The first section of this chapter describes the general approach adopted in this project for the storage of data. Then four basic types of files are suggested and various combinations of the basic types are proposed for the overall data storage system of the project. Certain arguments favoring the overall storage system that was selected are set forth. In the last section a brief discussion is presented of the type of data structure that would be appropriate for the data that has been loaded into the high speed core storage for processing.

7.1 Description and Arrangement of Data

A few rather general comments on the problem of data storage are in order before we launch into a description of the particular types of files considered for this project.

It will be useful in our discussion to hink of the data to be stored as forming a tree-like structure. For example, the information file generated by the Technical Information Project (Sec. 6.21) can be subdivided into journals. Each of the journals can be broken down into a number of volumes. Each volume in turn consists of some articles.

Within an article there are several information types--title, author(s), etc. Some of these information types may be further subdivided. For example, one can split the author information into the separate authors of the article. Fig. 7.1 portrays this tree structure.

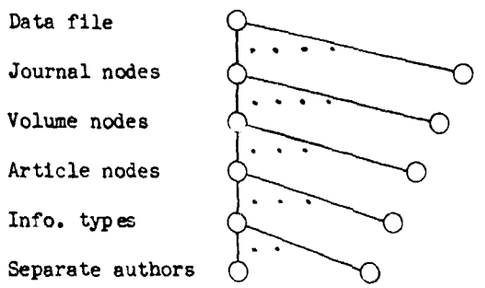


Fig. 7.1. Example of tree-like structure of data.

Each terminal node at the bottom of this tree represents a piece of data which must be stored, such as an author's name or a citation. Each parent node represents the grouping together of one or more pieces of logically related data. For example, a volume node groups together all the articles which are contained in that volume.

Let us first consider a couple of problems involved in storing the data represented by the terminal nodes. Much of this data is variable in length. For example, titles might vary from 20-200 characters. Two ways of handling variable size data suggest themselves. One might use a special code or flag to indicate the end of the piece of data or one might explicitly store the length somewhere in the file. The latter approach was selected since one would always have to perform a search to determine the end of the data if a flag were used.

In addition to knowing how long a piece of data is we must know its type or identification. For example, it is not possible, in general, to

determine whether a string of characters is a title or an author without being explicitly told this fact. If there were one and only one title, author, citation, etc. for each article, then the information type could be specified by the relative position or order of the pieces of data. However, for a given article there may be none or several citations and one cannot specify the information type implicitly by the order.

Thus, in addition to storing the actual data for each terminal node, one must give two additional facts--length and type. The storage of these two additional facts is useful for the parent nodes in the above tree as well as for the terminal nodes. The type of information for a given node serves to identify that node from all of its sister nodes which are under the same parent node. The length information delimits the scope of the node. For example, a volume node would have for its identification the volume number, and for its length either the number of articles in the volume or the amount of storage occupied by those articles. Thus one can summarize the storage requirements of a data file by the following two statements. An identification and length must be stored for every node in the related tree structure. In addition one must store a piece of literal data for each terminal node.

The last question to be discussed here relates to the actual physical order in which data is to be stored. Let us use the example of Fig. 7.2 to describe the arrangement selected. One can flatten the tree of Fig. 7.2 out into the linear array of nodes shown in Fig. 7.3 such that no two connecting lines cross, and such that each parent node is to the left of its subnodes.

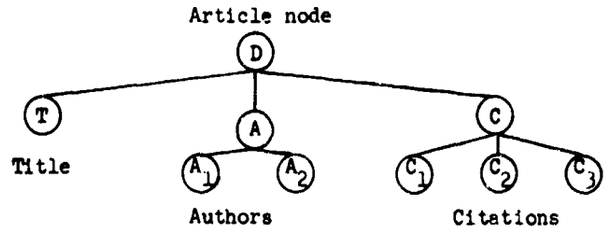


Fig. 7.2. Example used to show physical order given the data.

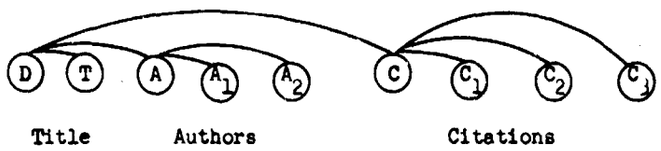


Fig. 7.3. Linear arrangement of data in Fig. 7.2.

This is the physical order in which the data is stored for this project. For the example of Fig. 7.3 the article identification and length are first (node D). This is followed by the code for title information, the title length, and the actual title (node T). Next is the code for author information and the length of the author data (node A). Then the information on a particular author is given (node A_1). This includes the author's identification (his position among the authors of the article), the length of his name, and his actual name. The description for the remaining nodes is similar.

It may be of interest to note that the above approach is analagous to polish prefix notation. Consider the algebraic equation $[A \cdot (B+C)]$. Its polish prefix form, $\cdot[A,+(B,C)]$, is obtained by flattening the tree of Fig. 7.4 such that no lines cross. If one equates terminal nodes to operands and parent nodes to operators, then our storage arrangement is the polish prefix form of the data.

Fig. 7.4. Polish prefix notation.

7.2 Types of Files

In this section four basic types of data files are described. An overall data storage system might consist of only one of the file types or it might include a combination of several types.

7.21 Raw Data File

The file of data generated by the Technical Information Project (Sec. 6.21) will be termed the raw data file. It currently has the 'polish prefix' structure described above. The precise substructure of a given article is shown in Fig. 7.5. The relative amoung of storage occupied by each of the types of information is given in the table of Fig. 7.6.

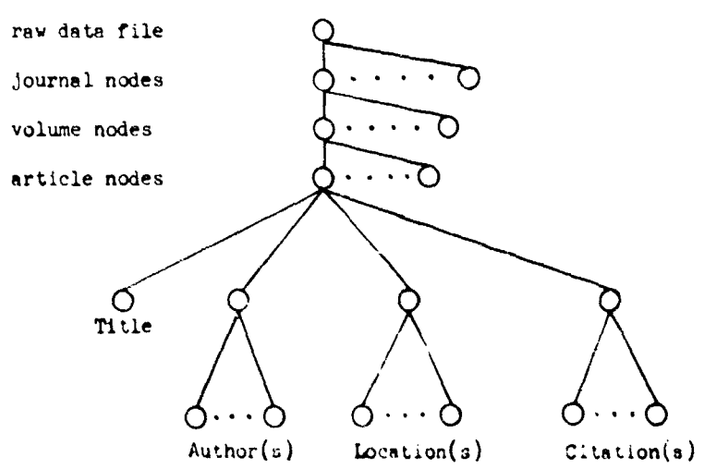


Fig. 7.5. Structure of raw data file.

article node (ident. and length)	5 %
title	21 %
authors	14 %
author locations	28 %
citations	32 %
	100 %

Fig. 7.6. Percent of storage occupied by each information type.

7.22 Inverted Files

An inverted file is a type of index to the raw data file. For example, one might create an inverted author file by extracting from each article the authors' names. These names could be alphabetized and the duplicates deleted. Such a file would have the structure shown in Fig. 7.7. In this figure nodes $D_1 \ldots D_k$ are the identifications of the articles written by Author A_1.

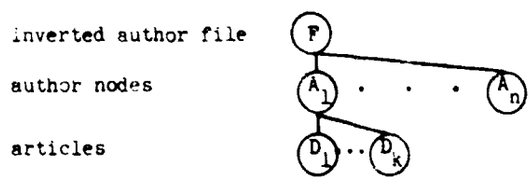


Fig. 7.7. Structure of inverted author file.

Inverted files have been created for title words, authors, locations, and citations. Because of a current lack of storage space, the inverted files cover only a part of the total raw data file. This partial coverage was found to be sufficient for experimental purposes, however.

On the basis of the experience gained with these partially completed inverted files, it is estimated that inverted files for the full raw data file will increase storage requirements by the percentages given in Fig. 7.8.

title word file	17.7%	of raw data file
author file	15.3%	" " " "
location file	15.0%	" " " "
citation file	47.5%	" " " "
Total	95.5%	" " " "

Fig. 7.8. Storage requirements for inverted files.

There are certain additional steps that can be taken which will probably reduce the additional storage required to only about 70% of the raw data file. Thus adding inverted files increases storage requirements by a factor of 1.5→2.0. It is suspected that the amount of storage needed for file inversion is a relatively standard factor for most types of information. Certainly the types of information found in the test file of this project (title, words, authors, locations, citations) varied markedly in their characteristics but still followed roughly this factor of two increase.

Fig. 7.9 shows that the relative amount of storage required for an inverted author file decreases as the size of the file increases. The leveling off shown leads one to believe that an order of magnitude increase in the test file would not significantly change the percent increase in storage required for an inverted author file. A similar leveling off was found for title words.

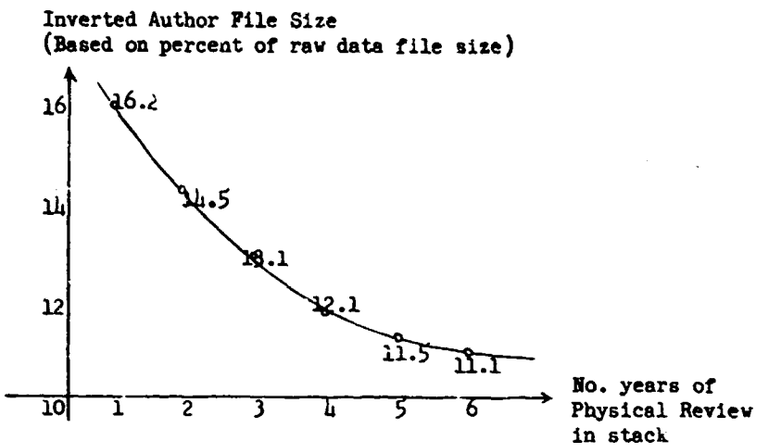


Fig. 7.9. Storage required for inverted author file.
(For articles in Physical Review 1959-64)

There is a good theoretical reason why the inverted files should require about the same amount of storage as the raw data itself. The reason is that the inverted files store the same information as the raw data file (except perhaps for the relative order of some of the data). Indeed one could reconstruct the raw data file from the inverted files by merely collecting together the title words, authors, etc. for each article. The one exception to the equivalence of the information found in the two types of files concerns order. One cannot determine from the inverted word file the order that the words originally had in the titles of the raw data file, but only which words belong to each title. Of course, some additional provision might be made so that inverted files contained order information as well as the article identifications. However the point here is that the two types of files should require about the same amount of storage.

7.23 Linkage Files

A linkage file contains a description of a document network of the type described in Chapter III. The basic information needed to describe such a network consists of document node identifications and link values.

The structure of a linkage file is shown in Fig. 7.10. For each document node in the network there is an entry in the filw which consists of the identification of the document along with the information on the links emanating from the node. The linkage information consists of the identifications of the other document nodes connected to the node in question along with the values of the connecting links. In such a file it is necessary to store only those links for which $N_{ij} \neq 0$ with the understanding that the value of all other links is K.

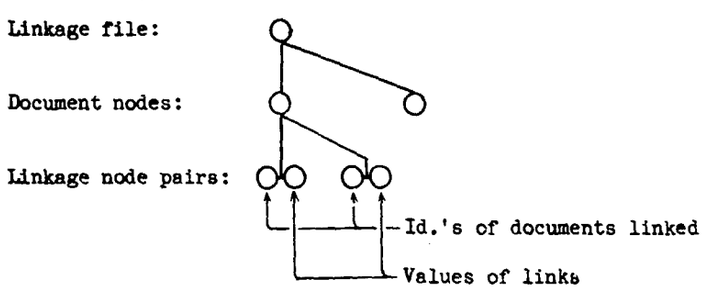


Fig. 7.10. Structure of Linkage File.

Note that the information on each link is specified in two places in a linkage file. For example, the value of $C(x_i^1 x_j^1)$ is stored in the entry for document x_i and also in the entry for x_j. This redundancy makes it so that once the entry on a given document is located, one immediately knows all of the documents to which it is linked as well as the values of the links.

In an attempt to gain some insight into the size and characteristics of linkage files, a test was conducted on one volume (Vol. 128) of the Physical Review. Linkage files were created based on each of the five types of partitions discussed in Sec. 6.22. The results of this test are summarized in Fig. 7.11.

Partitioning criterion on which links are based	File Size (Based on size of Phys. Rev. Vol. 128)	Percent of total possible links for which $N_{ij} \neq 0$
(1) Authors (estimated)	15% of raw data file	1/2%
(2) Title words (for words occurring less than 20 times)	58% " " " "	4%
(3) Cite-same	24% " " " "	1 1/2%
(4) Cited-by-same (Citations to v.128 from v.128-133)	5% " " " "	small
(5) Subject Category	175% " " " "	15%

Fig. 7.11. Table of linkage file sizes for vol. 128 of the Physical Review.

Fig. 7.11 indicates that partitioning criterion (3) generates a network in which about 1 1/2% of the links have values other than K (i.e. $N_{ij} \neq 0$). This is for a single volume of the Physical Review. It would seem reasonable that this percentage would be somewhat less for the total document file. We shall assume in the analysis of the next section that approximately 1% of the possible links in the network of the total file have non-K values. This means that each document in the T.I.P. file is linked to about (.01)(26,000)=260 other documents on the average.

7.24 Request - Answer File

The actual generation of this type of file was never seriously contemplated because of the immense amount of processing time and storage space that would be required. It is described here because it represents an extreme case to which we wish to make reference in the next section.

A request-answer file contains the answer cluster for each possible request. Its possible structure could be represented by Fig. 7.12. $D_1 \ldots D_k$ in this figure are the documents contained in the particular answer cluster in question.

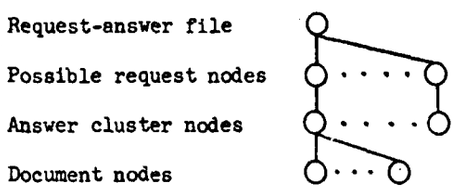


Fig. 7.12. Structure of request-answer file.

Retrieval from this type of file would consist of a simple table look-up for the request and then presentation of the associated answer cluster.

7.3 Storage Systems

The overall storage system selected for this project could consist of any combination of one or more of the types of files described in the preceding section. For purposes of discussion and comparison let us suggest four types of storage systems. The first three were implemented and tested to some extent. System (2) is the one that was finally selected for this project.

(1) Raw data file only.

(2) Raw data file and inverted files.

(3) Raw data file and linkage file.

(4) Raw data file and request-answer file.

The raw data file is included in each of the four storage systems so that information on specific articles can be presented to the user at any time he wants it. For instance, a user might want to know the title and author(s) of an article that is about to be added to the set S. This information would be obtained from the raw data file.

Each of the four suggested data storage systems could serve as base for the clustering procedure of Chapter V. There are some significant differences in the characteristics of the retrieval system that would result, however. Let us indicate some of the differences by discussing four important characteristics of the resulting retrieval systems.

7.31 Storage Space Required

Since the raw data file is basic to all four systems, we will express storage requirements in terms of the size of that file. It has already been noted that the inverted files require about as much storage as the raw data file. If we make the assumption that 1% of all possible links have non-K values as was suggested in Sec. 7.22, then the linkage file for the TIP document collection would be about six times as large as the raw data file. If we assume that every request for information consists of only two documents of interest and every answer cluster contains 20 documents, then a request-answer file would be about 35 times the size of the raw data file. Much more space would be required if larger requests were allowed. These figures are summarized in Fig. 7.13.

(1)	Raw data only	100% of raw data file
(2)	Raw data plus inverted	200% " " " "
(3)	Raw data plus linkage	700% " " " "
(4)	Raw data plus request-answer .	3500% " " " "

Fig. 7.13. Comparison of storage requirements for the four types of data systems.

7.32 Processing Time

Let us next determine the average amount of processing time that would be needed to transform a request into an answer cluster for each of the proposed storage systems. By processing time we mean the amount of time allocated by the central processor of the Project MAC system to running the clustering program. The time spent in swapping the program in and out of core storage is excluded. The rario of the real time that the MAC user must wait to the processing time varies with the number and type of users on the system and can range from one to forty or fifty.

The time required to access a piece of data on the 1302 disc is about 1/2 second. This includes both the time spent by the disc control supervisor and by the disc in locating and reading a track. Thus the request-answer system would require about a second in order to find an answer, since very little computational or manipulative work is required.

For a linkage file system at least 20 accesses to the disc would be required (for a cluster of 20 documents). This would involve about 10 seconds of processing time in addition to some computational time which was found to be small in comparison. We pick 15 seconds as the average amount of time required to find a 20-document cluster if linkage files are available.

The amount of processing time required to find a 20-document cluster with an inverted file storage structure has been found to 50-60 seconds. This includes 60 or so accesses to the disc and a fair amount of manipulation and computation.

If only the raw data file is available, then one must pass through the total data file two or three times looking for documents that are linked to the documents in sets Y, Z, and S. One complete pass through the raw data file takes 200-300 seconds. Thus the average processing time would be on the order of 600 seconds. Fig. 7.14 summarizes the processing time required for each of the four systems.

(1)	Raw data only	600 sec.
(2)	Raw data plus inverted	60 "
(3)	Raw data plus linkage	15 "
(4)	Raw data plus request-answer . . .	1 "

Fig. 7.14. Average processing time required to find a cluster of 20 documents for the four types of storage systems.

7.33 Updating and Editing

Besides the processing time involved in answering requests there is a certain amount of time required for updating and editing the file, since it is constantly changing. For purposes of comparison let us consider the problem of adding 335 articles (50 tracks or raw data) to an existing file of 20,000 articles (3000 tracks). The time required to load and structure the raw data file will not be considered since it is common to all four storage systems.

In order to update the inverted files one must extract the appropriate fields from the new raw data, sort them into the desired

sequences and merge the sorted data with the old inverted files. The current programs for doing this would take about 400 seconds for the 50 tracks of data. The time needed for each information type is as follows: words - 90 sec., authors - 50 sec., citations - 210 sec., locations - 50 sec. The time for each process is as follows: extraction - 25 sec., sorting - 150 sec., merging - 230 sec.

Consider the problem of updating a linkage file with the links based on whether or not two papers cite the same paper (partition type (3) in Sec. 6.22). Updating can be accomplished by the following steps. First, extract the citations from the 50 tracks of new articles. Sort these citations and compare them with the total raw data file to determine which articles are linked to each new article. During this comparison process generate a file of information on the new links. Sort this file and merge it into the old linkage file. The programs which were written to perform this updating process were only tested on small files of several hundred articles. Let us extrapolate the results and estimate how long it would take to update the linkage file for the case under consideration. Extracting and sorting the citations of the 335 new articles would take about 100 seconds. Matching the citations with the total raw data file would take about 1800 seconds and merging them into the old linkage file would require about 1200 seconds for a total of 4000 seconds.

The amount of time required to update a request-answer file would be more of a guess than an estimate. It would take at least 7000 seconds to rewrite the file and probably 10 to 100 times more to find all the clusters. These figures are tabulated in Fig. 7.15 for ease in comparison.

(1) Raw data only	0	sec.
(2) Raw data plus inverted	400	"
(3) Raw data plus linkage	4000	"
(4) Raw data plus request-answer	7000+	"

Fig. 7.15. Processing time required to update a file of 2000 articles with 335 new articles for each of the four storage systems.

7.34 Flexibility and Compatability

So far we have been mainly concerned with how much storage space and processing time is required for a system which finds answer clusters. Actually the process of finding clusters as proposed in this thesis is not considered to be the only retrieval tool which will be made available to the user. Rather clustering is looked upon as one possible component in a larger, more general retrieval system. It follows that the storage structure of the data should not be designed with just the clustering process in mind, but it should be chosen on the basis of its utility and adaptability to a large class of retrieval functions.

Even if the data file for the experimental system were to be used exclusively for clustering, it would still be useful to make the structure selected as general as possible. One reason why this is so stems from the fact that any experimental system is generally in a constant state of flux and any rigid or specialized data structure may soon be rendered obsolete.

Let us suggest that the following objective might yield a data storage structure which would provide an adequate base for a large number of different retrieval functions and at the same time strike a

suitable compromise between storage and time requirements.

"The amount of storage required should be minimized subject to the restriction that at no time should one have to serially search through the total file to obtain a given piece of information. By serial search we mean a sequential examination of every article in the file."

7.4 Selection of Storage System

From Sec. 7.31 and 7.32 it is evident that no data structure will at the same time minimize the processing time and storage space required. Some type of engineering compromise is needed. This compromise must be influenced by such factors as the characteristics of the computational facilities to be used and by the type of retrieval service that is to be offered. One must also consider the costs involved in updating the file and how often updating is to be performed. The decision is further complicated by the fact that the structure selected should be compatible with other retrieval functions and flexible to change.

A storage system consisting of the raw data only requires the least amount of storage space and the least effort to update. Its major drawback is in the time required to answer a request. Even now with the current file of about 26,000 articles the time required to find information is generally too great to allow for close man-machine coupling. And if the file size were to increase by an order of magnitude, a system based on this structure would certainly be too slow.

The linkage and request-answer files have excellent response times but require an excessively large amount of storage space and are very hard to update. In addition they are designed specifically for the

purpose of finding clusters and have little or no real value to other retrieval operations.

The second type of data storage system consisting of the raw data file and the inverted files was the one selected for this project. Its storage requirements were less than double that required for the raw data file alone. The processing time required to find a cluster was high, but not so high as to exclude close man-machine interaction, and it appears that an order of magnitude increase in the file size would not appreciably increase these time requirements. Updating of the system could be done on a daily or weekly basis without consuming an excessive amount of computational effort. The structure is also useful in a large number of other retrieval operations as will become more obvious in the next chapter.

7.5 High Speed Storage Structure

So far in this chapter we have discussed how the data should be structured for permanent storage on the disc. A related problem concerns the form the data should take once it has been selected for processing and is loaded into high speed core storage.

The approach that was used in the earlier versions of the experimental system was to convert the data to a "list" structure as it was loaded into core. This involves associating one or more address pointers with each piece of data. The pointers preserve the original sequence of the data without requiring that it occupy contiguous locations in memory. One of the major advantages of such a structure is the relative ease with which the data can be re-arranged and with which particular pieces of data can be added and deleted. Some of the

programming languages that have been developed to facilitate the creation and manipulation of list structures are COMIT, LISP, SLIP, and SNOBOL.[51,54]

It was later decided that the added flexibility obtained through the use of list structures was not, in general, needed for library-type data that remains relatively fixed. Indeed the processing time required to reformat the data into lists was considerable. Therefore the approach that was finally adopted was to leave the data in core in the same form that it was on the disc.

It is actually easier to perform some of the operations needed in the formation of a cluster on this disc structure than it is to do them on the equivalent list structure. Take, for example, the calculation of the N_{ij}'s. For the partitioning criterion selected this would involve the comparison of two tables of citations. The most efficient way that has been found to do this is to have the citation codes of each article in numeric order on the disc, and to make a single synchronous pass through the two tables tallying the number of matching entries. The time required to do this match if the data has a list structure would probably at least double. There are also certain other operations (e.g. binary or logarithmic searches) for which a list structure is not well suited.

For the final version of the experimental system a rather simple storage allocation system was adopted which kept track of the available free core storage. Through this system blocks of storage could be allocated, changed in size, or freed up for other uses. Reference to each block was through a numeric code so that the actual address of the block could change. This made it so that all the free storage could be kept in one contiguous block. Data from the disc was loaded into these

blocks of storage and processed there.

The S, Y, and Z document sets were also placed in blocks obtained from the storage allocator. It was later decided that this was a distinct disadvantage to the system because the sets were constantly changing and should have had the flexibility available from a list structure.

CHAPTER VIII

INTERACTION LANGUAGE

The description of the experimental system is now almost complete. The clustering procedure which is used in answering requests has been defined in Chapter V. The computational facilities and data base on which the system operates have been described in Chapter VI. In Chapter VII the way the data is structured was explained.

The one aspect of the experimental system that has not been covered concerns the interface between the user and the system. In this chapter we will describe the language which permits the user to communicate and interact with the system.

8.1 Background to Language

As a way of introducing the language we will present in this section some of the general design objectives that were selected for the language and an example of a typical interaction using the language.

8.11 Design Objectives of Language

The first retrieval language developed for this project was designed specifically for clustering and bore little resemblance to the language used by the Technical Information Project programs in performing the more conventional matching functions (author, citation, and keyword searches, bibliographic coupling, etc.). It was found to be inconvenient and confusing to have to shift from one program and one language to

another program and another language every time one wanted to shift from a clustering request to a T.I.P. request and vice versa. It was decided that the same general language should be used for both functions. This goal is related to the idea expressed in the last chapter that the clustering function should be considered a component of a larger retrieval system (Sec. 7.34). Not only should the data structure be designed for the larger, more general system, but the retrieval language should also. In the remainder of the chapter the clustering and matching functions will, therefore, be treated equally.

In addition to having adequate expressiveness for the current clustering and T.I.P. commands, it was considered desirable that the language be flexible enough so that it might be easily extended to other types of retrieval operations.

A second objective of the language is that it should be easy to learn, use, and remember. It was decided that if the vocabulary and syntax of the language resembled normal English it would be easiest to learn and remember.[11] However, it was found to be rather tedious after a while to have to type a complete English sentence for each request. An abbreviated version of the language was, therefore, developed for the experienced user which allowed much of the vocabulary to be abbreviated. The abbreviated version was such that one could make a smooth transition from the full English request to the abbreviated request as he became more familiar with the system. An example of a complete request and the equivalent abbreviated request follow.

"Print the authors and locations of all the articles cited by the article, Physical Review, volume 135, page 3."

"p art loc of art cited by 1 135 1."

A third goal of the language is that it be simple enough to process efficiently and quickly. Even a rather complex request in the language that was adopted takes much less than a second of central processor time to interpret.

8.12 Example of Language

In Fig. 8.1 is an example of an interaction that might occur between a user and the system. The lines that the user types are underlined. First he initiates the MARS (Machine Aided Retrieval System) program. We assume that the one fact the user knows is that he is interested in something about Langmuir probes. He could just as well have known an author or paper that interested him or perhaps a combination of these.

In the first command he asks for a list of those articles containing the word, "Langmuir", in their titles. Let us say that after examination of the list produced, the user decides that the papers by three of the authors are the most interesting. He now asks for all papers written by these three authors (that have not already been retrieved).

Next we assume that the user selects two of the papers as of particular interest and wishes to form a cluster around them. Further he decides that one of the papers is definitely not what he wants and he, therefore, specifies that it is not of interest. A close interaction sequence follows with the system presenting papers that are about to be added to or deleted from the set S and the user deciding which are of interest and which are not.

Finally a cluster is formed and the user stores it on the disc for future reference. He then analyzes its characteristics by making various lists of frequency counts.

RESUME MARS
W 1348.4

PRINT THE TITLES AND AUTHORS OF ARTICLES CONTAINING THE WORD,'LANGMUIR'.

17 ARTICLES IN SET 1.

PHYSICA
VOLUME: 30
PAGE: 182
STUDIES OF THE DYNAMIC PROPERTIES OF LANGMUIR PROBES I: MEASURING METHODS.
CARLSON R. W.
OKUDA T.
OSKAM H. J.

NUOVO CIMENTO
VOLUME: 29
PAGE: 487
EFFECT OF A R.F. SIGNAL ON THE CHARACTERISTIC OF A LANGMUIR PROBE=
BOSCHI A.
MAGISTRELLI F.

END.

PRINT THE TITLES AND AUTHORS OF ARTICLES BY R. W. CARLSON OR T. OKUDA OR H. J. OSKAM BUT NOT IN SET 1.

6 ARTICLES IN SET 2.

JOURNAL OF THE PHYSICAL SOCIETY OF JAPAN
VOLUME: 13
PAGE: 1212
DISTURBANCE PHENOMENA IN PROBE MEASUREMENT OF IONIZED GASES.
OKUDA T.
YAMAMOTO K.

END.

PRINT FOR DECISION THE TITLES AND AUTHORS OF ARTICLES RELATED TO PHYSICA, V. 30, P. 182, AND J. PHYSICAL SOCIETY OF JAPAN, V. 13, P. 1212, BUT NOT NUOVO CIMENTO, V. 29, P. 487.

TO BE ADDED:
PHYSICS LETTERS
VOLUME: 11
PAGE: 126
THE PLASMA RESONANCE PROBE IN A MAGNETIC FIELD.
CRAWFORD F. W.
HARP R. S.

IS THIS OF INTEREST: YES

TO BE ADDED:

END.

```
SAVE SET 3.

FILE  SET   3  CREATED.

END.

PRINT THE FREQUENCY OF AUTHORS IN SET 3.

23 AUTHORS IN SET 3.

4  OKUDA T.
3  CARLSON R. W.

END.
```

Fig. 8.1. Example of possible user interaction with data using retrieval language. (Lines typed by user are underlined.)

8.2 Description of Language

Two methods of describing the retrieval language have been selected. In the first the syntax of the language is described by means of a finite state (sequential) machine.[35] In the second the syntax and vocabulary are defined by means of Backus normal (ALGOL 60) notation.[37] The equivalence of these two descriptions is also shown.

8.21 Finite State Machine Description

There are a number of different methods that could be used to describe the retrieval language that was developed for this project. Perhaps the most appropriate way to describe the syntax of the language would be to present the same table that is actually used by the interpretive part of the retrieval system. Fig. 8.2 is the syntax table which has been extracted from a program listing. It is a tabular description of a finite state machine[35]. The first column contains the identifications of the various states. Column two pertains to one of the languages used to write the system (it is the name of a MACRO in FAP)

and is not pertinent to our discussion here. The third column contains the valid state transitions that can occur. For example, the entry (V,2) for S1 means that the machine will change from state S1 to S2 if the input signal is V (verb).

S1	STATE	((V,2)(X,1)(A,1))
S2	STATE	((V,2)(C,3)(N,4)(L,8)(E,10)(X,2)(A,2))
S3	STATE	((V,2)(X,3)(A,3))
S4	STATE	((N,4)(C,5)(P,6)(X,4)(A,4))
S5	STATE	((N,4)(X,5)(A,5))
S6	STATE	((N,7)(X,6)(A,6))
S7	STATE	((P,6)(L,8)(X,7)(A,7))
S8	STATE	((L,8)(C,9)(E,10)(X,8)(A,8))
S9	STATE	((P,6)(L,8)(X,9)(A,9))
S10	STATE	()

Fig. 8.2. Finite state machine description of syntax of retrieval language.

Fig. 8.3 is the state diagram for the machine of Fig. 8.2. We have left off the self loops on each state due to the X and A inputs to keep from cluttering up the diagram. Also not shown is the sink state which the machine enters when the input sequence being analyzed has an invalid syntax. For example, if the machine is in state S_2 and the input signal is a P, then the sink state is entered. The initial or starting state of the machine is S_1. The final or accepted state is S_{10}. Thus an input sequence is considered to have an acceptable syntax if it transforms the machine of Fig. 8.3 from S_1 to S_{10}.

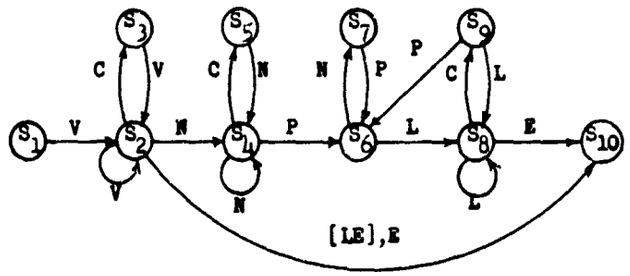


Fig. 8.3. Finite State Diagram for the Table of Fig. 8.2. (Transitions not shown go to an error or sink state.)

The input symbols of Fig. 8.2 and 8.3 represent classes of words. Fig. 8.4 gives the general titles and some examples of the classes. The interpretive procedure first classifies each word in the input statement into one of the classes and then checks the syntax by the Table of Fig. 8.2. In Fig. 8.5 we present a specific example of an acceptable and an unacceptable statement.

Input Symbol	Class Name	Specific Examples
V	Verbs	print, count
N	Nouns	article, title
P	Prepositions	by, of
A	Adjectives and Adverbs	first, last
C	Conjunction	and, or
X	Filler Words	the, a
L	Undefined (literal) words	Jones, laser
E	Terminator	.(carriage return)

Fig. 8.4. Classes of Input Symbols.

Statement:	Count the articles by John Jones.							
Word classes:	V	X	N	P	L	L	E	
States traversed:	S_1	S_2	S_2	S_4	S_6	S_8	S_8	S_{10}

Statement:	Print the titles of articles and.						
Word classes:	V	X	N	P	N	C	E
States traversed:	S_1	S_2	S_2	S_4	S_6	S_7	Sink State

Fig. 8.5. Example of statement with acceptable syntax and statement with unacceptable syntax.

Let us comment briefly on the purpose of each state in the diagram of Fig. 8.3. Preliminary to doing this it should be noted that there are generally three main parts to an acceptable statement (request):

(1) Verb (states [illegible] and S_3)

(2) Direct object (states S_4 and S_5)

(3) Modifying phrase (states S_6 S_9)

State S_1 is the starting state of the machine. State S_2 requires that each request begin with a verb describing what the system should do. The verb can be either simple (e.g. print) or compound (e.g. count and save). State S_3 excludes the possibility of a double conjunction between elements of a compound verb (e.g. print and or store). It also prevents the verb from ending in a conjunction.

State S_4 requires that the next part of a request be a list of one or more nouns signifying the type of information that is to be produced by the system. This can again be simple (e.g. title) or compound (e.g. title, authors, and locations). State S_5 has a purpose similar to S_3.

The last part of the request is the modifying phrase which contains the structure of the articles and other entities that are

specified by the user in making the request. States S_6 and S_7 allow the request to have a complex structure with several levels of prepositional phrases modifying other phrases. For example, one could find the co-authors of a given author by the request: "Find the authors of articles by John Jones."

States S_8 and S_9 allow the user to specify some logical combination of a number of specific fields. For example: "Print the articles by John Jones and Robert Smith but not Joseph Adams."

The E transition from S_2 to S_{10} is so that certain commands will be accepted that consist of a verb only. The LE transition between S_2 and S_{10} allows for an abbreviated mode of reference to certain data (e.g. Print set 3.). Adjectives and adverbs can occur anywhere in a request and can modify verbs, nouns, etc.

8.22 Backus Normal Description

Let us leave the finite state description of the syntax of the language now and provide a more conventional description. The statements of Fig. 8.6-8 constitute the Backus normal (ALGOL 60) description of the language. In this notation "::=" means "is defined to be", " | " means "or", and "⟨ ⟩" encloses the defined elements of the language[37].

Two additional explanations are necessary for the Backus normal description of Fig. 8.6-8. All elements (words) in the statements are separated by one or more word separators (blanks, commas or periods) except in the definitions for ⟨word⟩ and ⟨integer⟩ where the characters have no separation. Adjectives, adverbs, and filler words can occur at any point in a request, but this fact is omitted from the description to simplify its statement.

```
<request> ::= <compound verb> <compound object> <compound modifier>
              <terminator> | <abbreviated command>

<compound verb> ::= <verb> | <compound verb> <verb> |
                    <compound verb> <conjunction> <verb>

<compound object> ::= <noun> | <compound object> <noun> |
                      <compound object> <conjunction> <noun>

<compound modifier> ::= <modifying phrase> | <compound modifier>
                        <conjunction> <modifying phrase>

<modifying phrase> ::= <preposition> <compound literal> |
                       <preposition> <noun> <modifying phrase>

<compound literal> ::= <literal> | <compound literal> <conjunction>
                       <literal> | <compound literal> <literal>

<abbreviated command> ::= <compound verb> <terminator> |
                          <compound verb> <literal> <terminator>
```

Fig. 8.6. Backus normal statements describing syntax of language.

```
<vocabulary word> ::= <verb>|<conjunction>|<noun>|<preposition>|
                      <adjective>|<adverb>|<filler>|<terminator>

<verb> ::= <find verb>|<print verb>|<delete verb>|<save verb>|
           <read verb>|<other verb>

<find verb> ::=  count | find | fetch | f | get | g | keep

<print verb> ::=  list | print | p

<delete verb> ::=  delete

<save verb> ::=  dump | save | store

<read verb> ::=  read

<other verb> ::=  load | return | search | trace | unload | yes | no | skip

<conjunction> ::=  and | and not | but not | not | or

<noun> ::= <article noun>|<title noun>|<word noun>|<author noun>|
           <location noun>|<citation noun>

<article noun> ::=  art | article | articles | doc | document | documents |
                    id | ids | identification | identifications | paper |
                    papers

<word noun> ::=  keyword | keywords | word | words

<author noun> ::=  aut | author | authors

<location noun> ::=  loc | location | locations

<citation noun> ::=  biblio | bibliography | bibliographies | cit | citation |
                     citations | ref | reference | references

<preposition> ::= <article preposition>| <word preposition>|
                  <author preposition>|<location preposition>|
                  <citing preposition>|<cited by preposition>|
                  <set preposition>|<clustering preposition>

<article preposition> ::=  of | used by

<word preposition> ::=  contain | contains | containing | use | using

<author preposition> ::=  by

<location preposition> ::=  at

<citing preposition> ::=  cite | citing

<cited by preposition> ::=  cited by

<set preposition> ::=  in

<clustering preposition>::=  related to | related by authors to |
                             related by citations to

<filler> ::=  a | all | all of | an | any | any of | are | been | each | every |
              have | is | the | this | these | those | were | written

<adjective> ::=  first | last | most recent

<adverb> ::=  by frequency | for decision

<terminator> ::=  .↵  (↵ is a carriage return
```

Fig. 8.7. Backus normal statements describing vocabulary of language.

```
<literal> ::= <article literal>|<word literal>|<author literal>|
              <location literal>|<set literal>
<article literal> ::= <journal><volume><page>
<word literal> ::= <literal string>
<author literal> ::= <literal string>
<location literal> ::= <literal string>
<set literal> ::=  set <integer>
<journal> ::= <journal name>|<alphabetic code>|<numeric code>
<journal name> ::=  Phys. Rev.|Physical Review|...|Physics of Fluids
<alphabetic code> ::=  phyrev|phyreb|...|spjetp
<numeric code> ::= <integer>
<volume> ::= <word><integer>|<integer>
<page> ::= <word><integer>|<integer>
<literal string> ::= <word string>|<word string>
          (the first word string in this definition cannot include a
           vocabulary word.)
<word string> ::= <word>|<word string><word>
<word> ::= <character>|<character><character>|<character><character>
           <character>|...
<integer> ::= <digit>|<digit><digit>|<digit><digit><digit>|...
<character> ::= <letter>|<digit>|<special character>
<letter> ::=  a|b|...|z
<digit> ::=  0|1|...|9
<special character> ::=  -|/|=|*|:|;|...
<word separator> ::=  (blank)|,|.
```

Fig. 8.8. Backus normal description of literals.

8.23 Equivalence of Descriptions

The equivalence of the Backus normal definition of Sec. 8.22 to the finite state diagram of Sec. 8.21 can be shown by successively applying the four transformations of Fig. 8.9 to the statements of Fig. 8.6. Fig. 8.10 is a brief outline of the steps which would be taken in this process. One is referred to the literature for an explanation of the additional concepts (e.g. non-deterministic machines, equivalent states, etc.) introduced in this Figure.

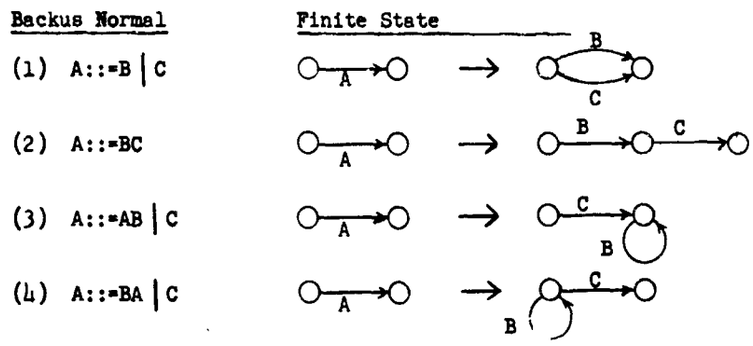


Fig. 8.9. Rules for transforming Backus normal statements to finite state diagram.

8.3 Interpretive Algorithm

In this section we will describe how the retrieval system interprets and processes the language of Sec. 8.2. The discussion will initially cover some general aspects of requests and of the words that they contain. Sections 8.32-8.34 will describe the various functions that requests can perform (the verb), the types of data that can be generated as output (the direct object), and the structure that specifies the actual request (the modifying phrase).

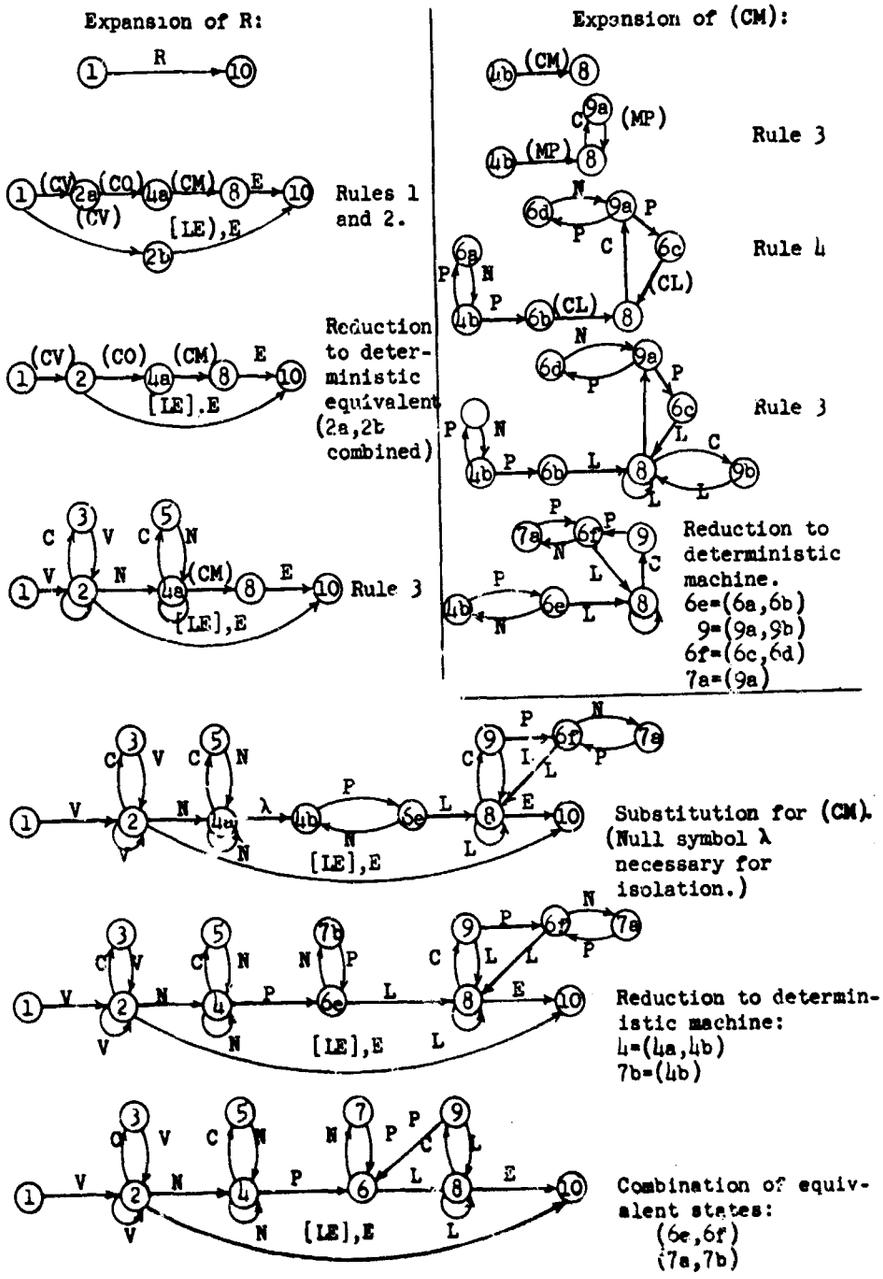


Fig. 8.10. Outline of steps proving equivalence of Backus-normal and finite state descriptions.

8.31 Vocabulary and Literals

A request consists of one or more lines of characters that the user types on his time-sharing console. The maximum length of a request is currently 400 characters. The end of a request is indicated by a period followed by a carriage return. The request character string is initially broken up into words. Words are defined to be character strings separated by blanks, commas, and/or periods. There are two types of words: those found in the vocabulary table and those not found in the table. All words not found in the table are called literals. Their function is to specify the particular authors, title words, citations, etc. that the user wishes to designate in defining his request. The vocabulary words are for indicating the function and structure of the request.

In some cases a user may want to use one of the words in the vocabulary table as a literal. For example, he may want to find all titles that contain the vocabulary word, "store". To do this he can explicitly specify the word as a literal by the use of the literal mark, " ' ". For the above example the user would say, "print the titles of all articles containing 'store' ."

Note that the retrieval system makes no distinction between lower and uppercase letters. The T.I.P. file does not contain information on whether a letter is lower or upper case either.

8.32 Available Functions

The verb part of each request specifies the particular operation or operations that are to be performed. For example, if the user wants the results of the search to be printed on his time-sharing console, he

would use the verb, "print". There are currently twenty-three verbs in the vocabulary and thirteen different functions that they specify. Let us describe five of the thirteen functions.

(1) Scratchpad Storage

One of the most useful features of the retrieval system is its scratchpad storage capability. Basically this involves the storage in core memory of various kinds of data for later reference. For example, one can create in scratchpad storage a file of all articles written by a given author by the command, "Find the articles by John Jones." After creating the set, the system tells the user its size and identification number (e.g. 4 articles in set 3). Later on the user could find out what articles cite articles by John Jones by the request, "Print the articles citing articles in set 3," or just "p art citing set 3."

Each data set in scratchpad storage is currently homogeneous with respect to the type of information it contains. In other words one could not create a set that consisted of both author and citation data.

Some of the verbs that create sets in scratchpad storage are: count, find, fetch, f, get, g, and keep. These words are completely equivalent so far as the system is concerned.

(2) Console Print-out

The verbs that will cause the data in question to be printed on the user's console are list, print, and p. A scratchpad set will also be automatically created (if the output is homogeneous and if it isn't already a set).

The first line of each print-out consists of the number of items that will follow. Thus the user is always aware of the ultimate size of the listing and can interrupt it if he wishes.

(3) Delete Data Sets

Sets or groups of sets can be erased from scratchpad storage by commands such as "Delete set 4", "Delete all sets."

(4) Save Data Sets

Any scratchpad data set can be placed on the disc for permanent storage by the verbs save, store, or dump. The form of the command would be: "Save set 2."

(5) Read Data Sets

Data sets that have been stored on the disc by the above command can be written back into scratchpad storage by commands of the type: "Read set 6."

The functions of some of the verbs can be modified by adverbs or adverbial phrases. Let us describe two such modifications that have been implemented.

(1) Frequency Lists

The print verb can be modified to list items in terms of their frequency of occurrence in the data from which they are extracted. For example, the command, "Print frequency of title words in Phys. Rev. Vol. 132." would produce a list of the number of times each word appears in the titles of articles in Phys. Rev. Vol. 132 (most frequent first and alphabetical within the same frequency).

(2) Decision Print-outs

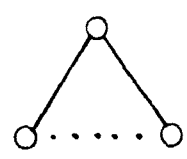
The print verb can also be modified so that there is a pause after each item is printed out to allow the user to decide upon and respond to the item. This would be the command used, for example, by a user who wished to be coupled into the clustering procedure. For the command,

"Print for decision the titles of articles related to Nuovo Cimento Vol. 30, page 1.", the procedure would pause after printing the title of each article about to be added to or deleted from the set S and allow the user to place the article in the Y or Z set if he wished.

8.33 Data Generated

The second part of the request is the direct object of the verb. It is a list of the types of information (nouns) that the user specifies he wants in the system's response to the request. Fig. 8.7 indicates six different types of nouns that can be used for this purpose (article, title, word, author, location, and citation nouns). The correspondence of these words to the various types of data found in the T.I.P. file is fairly obvious. Any combination of these types of data can be printed on the user's console, but only one type can be put in scratchpad storage for a given request. The form of the data as it is printed on the console is shown in Fig. 6.4. The data placed in scratchpad has the single level structure indicated by Fig. 8.11 (see Sec. 7.1).

Set Node:



Author Name Nodes:

Fig. 8.11. File structure of data in scratchpad storage.

8.34 Request Structure

The third and final component of the request is the phrase which modifies the direct object of the verb. It consists of a series of prepositional phrases which either modify the direct object itself or

else modify the noun object of one of the other prepositional phrases. Let us define the structure of this modifying phrase and describe how it is interpreted.

8.341 Determination of Literal Type

The object of each preposition can be a noun or a literal. In the case of a literal some indication must be given of its type, since there is no intrinsic difference between most of the types (e.g. a word literal might look exactly like an author literal). The first preposition to the left of a literal is currently used to determine the type. Fig. 8.12 lists the literal type which is assumed to follow each preposition. For example, any word not in the vocabulary that follows the preposition, "by", is assumed to be an author's name.

The one exception to this is the set literal which can be the object of any preposition. It is distinguished from other literals, not by the preceding preposition, but by the word, "set", at the beginning of the literal.

There is one additional way of indicating the literal type which has been partially implemented but is not described in Sec. 8.2. This involves the use of a noun between the preposition and the literal. An example of this would be the phrase, "with the word, phonon", which is acceptable and identical to the phrase, "using phonon". A change such as this would become essential if the number of data types increased substantially, since there would not be enough suitable prepositions.

Preposition Type	Type of Object
⟨article preposition⟩	⟨article noun⟩, ⟨citation noun⟩, ⟨article literal⟩
⟨word preposition⟩	⟨word noun⟩, ⟨word literal⟩
⟨author preposition⟩	⟨author noun⟩, ⟨author literal⟩
⟨location preposition⟩	⟨location noun⟩, ⟨location literal⟩
⟨citing preposition⟩	⟨article noun⟩, ⟨citation noun⟩, ⟨article literal⟩
⟨cited by preposition⟩	⟨article noun⟩, ⟨citation noun⟩, ⟨article literal⟩
⟨set preposition⟩	⟨set literal⟩
⟨clustering preposition⟩	⟨article noun⟩, ⟨citation noun⟩, ⟨article literal⟩

Fig. 8.12. Valid types of objects for each preposition class. (Set literals are valid objects for any preposition and are not listed.)

8.342 Form of Literals

After the general type of information that a literal contains is determined, one must next interpret what specifically is meant by each literal. To this end let us describe the conventions which govern the form that each type of literal can take.

Article literals generally consist of three parts: the journal, volume, and page. The journal can be specified by using the full title, the standard abbreviation of the title, or a special alphabetic or numeric code. The volume and page number can each consist of an integer or a word followed by an integer. Some examples of acceptable article literals are:

Physical Review, volume 128, page 1

Phys. Rev., vol. 128, p. 1

Phyrev v 128 p 1

1 128 1

The volume and page number have been made optional so that one can refer to all articles in a given journal or in a given volume by a single literal.

Each word literal should consist of a single word. If one wishes to search for a phrase of two or more words, he should use two or more literals (e.g. "print titles of articles using thin and film.").

A word literal represents (matches) not only the word in the file which is identical to it, but also all words to which it is the prefix. Thus the command, "Get the art using supercon." would get all articles with titles containing superconductor, superconductivity, etc.

If one does not want prefix matching, he can use a "*" to designate an explicit blank. The command, "p art using laser*.", would not produce those articles whose titles contain the word, "lasers".

Author literals are to be written with the surname last (e.g. John H. Jones). A literal that consists of a surname only will retrieve all authors with that surname. A literal containing one or more given names will match those author names in the file for which the surname matches exactly and for which every given name in the literal is the prefix of the corresponding given name in the file. Thus, "p art by Al Jones.", would print all articles by "Albert Jones," "Alden Jones", and "Allen S. Jones".

Location literals must be given in a request exactly as they are found in the data file if retrieval is to be accomplished.

Set literals consist of the word, "set", followed by the identification number of the desired set.

8.343 Action Initiated by Each Preposition

Each prepositional phrase in a request initiates a file search (table look-up) in an appropriate data file. If the object of the preposition is an author, location, word, or citation literal, then the file used is the corresponding inverted file. If the object of the phrase is an article literal then the raw data file is used.

The information obtained from an inverted file is, of course, always a list of article identifications. The type of information obtained from the raw data file is determined by the type of noun that is modified by the prepositional phrase in question. For example, in the command, "Print authors of Phys. Rev. 128 1.", the table look-up for the "of" preposition would be in the raw data file and would select the author information.

The set of articles (or other data) produced by each table look-up can in turn be the object of another preposition and another table look-up. Consider the request, "Print the titles of articles cited by articles by John Jones." The procedure first looks up the articles by John Jones. Then it finds the articles cited by the articles by John Jones. And finally it retrieves and prints the titles of the articles so obtained. Note that each of the three prepositions, of, (cited) by, and by initiated a particular type of file search.

There are two types of prepositions that do not cause a table look-up in a file. A clustering preposition performs more than just a table look-up. The procedure of Chapter V is executed, resulting in the set of articles of the appropriate cluster.

The set preposition does not initiate a file search but produces the input set as its output (a unitary transformation). Thus in the

request, "Print the title of articles in set 4.", the preposition, "in", merely passes on the articles in set 4 to the next preposition, "of", which looks up their titles.

8.344 Logical Operations

The results of the table look-ups (or clustering) for two or more prepositional phrases can be combined by the standard logical operations (and, or, not). Consider, for example, the request, "Print the articles by John Jones and by Robert Smith or by Charles White but not by David Allen." The logical operation performed can be represented by the equation $[((J.J. \cap R.S.) \cup C.W.) \cap \overline{D.A.}]$ where the initials J.J. stand for the set of papers by John Jones and $\overline{D.A.}$ is the set of papers not written by David White. It will be noted that the logical operations are performed from left to right through the request in the same sequence in which the user typed them in. It was thought that this might be a more useful convention for a system that is closely coupled to the user than to have a parenthesized system with a hierarchy of the types of operations to perform first (as in MAD, FORTRAN, etc.).

Any arbitrarily complex logical structure can be obtained by this kind of approach (without having to use parentheses) if one creates sets in scratchpad storage. For example the set of articles represented by the logical expression, $(J.J. \cap R.S.) \cup (C.W. \cap \overline{D.A.})$, could be created by the sequence of commands.

```
Find art by John Jones and by Robert Smith.
     3 articles in set 1.
Find art by Charles White but not by David Allen.
     1 article in set 2.
Print art in set 1 or in set 2.
```

There is one logical structure that is not allowed in the system since it makes little sense in retrieval applications. This is the negation of any of the operands of the "or" operation. Consider the command, "Print articles by John Jones or not by Robert Smith." If this means $(J.J. \cup \overline{R.S.})$, then the articles requested would include most of the file since Robert Smith would have authored at most 20-30 articles.

The conjunctive operation between each pair of prepositional phrases must be explicitly stated. One could not say, "Print art by John Jones, by Robert Smith, and by Charles White." However, one can omit the prepositions after the first one (e.g. "Print art by John Jones and Robert Smith.").

8.345 Selection of Predecessor

The next problem to be considered is the determination of what noun(s) each prepositional phrase modifies (its predecessor). Consider the request, "Find the articles citing articles by John Jones and cited by Physics of Fluids, v. 7, p. 1." The last phrase, "cited by..." can conceivably modify either of the two preceding "articles" words. However, the answer to the request is markedly different depending on the interpretation selected. The approach adopted here is to "attach" each prepositional phrase to the first noun to the left of the phrase that is a valid type for the preposition in question. In Fig. 8.13 the valid noun types that can be modified by each preposition are listed.

Note that each preposition that immediately follows a noun and not a conjunction, must modify that noun and cannot be attached to other nouns further to the left. If the noun is not valid for the preposition by Fig. 8.13, then the request is considered in error. The request,

"Find the articles by John Jones and the citations at Harvard University.", would not be valid because the preposition, "at", is not a valid modifier of "citations" and cannot be attached to the earlier "articles" word because it does not immediately follow a conjunction.

Modifiable Noun Types	Preposition Type
⟨noun⟩	⟨article preposition⟩
⟨article noun⟩,⟨citation noun⟩	⟨word preposition⟩
⟨article noun⟩,⟨citation noun⟩	⟨author preposition⟩
⟨article noun⟩,⟨citation noun⟩	⟨location preposition⟩
⟨article noun⟩,⟨citation noun⟩	⟨citing preposition⟩
⟨article noun⟩,⟨citation noun⟩	⟨cited by preposition⟩
⟨noun⟩	⟨set preposition⟩
⟨article noun⟩,⟨citation noun⟩	⟨clustering preposition⟩

Fig. 8.13. Types of nouns that each class of prepositions can modify.

8.346 Interpretation of Adjectives

Let us make two final comments concerning the interpretation of the language. Filler words are adjectives, adverbs and certain other words that initiate no action in the interpretor. They are effectively ignored. Their only use is to make the statement of the request more smooth and natural.

There are other adjectives and adverbs that do effect the interpretor, however. Some of them are listed in Fig. 8.7. A large number of adjectives and adverbs come to mind that would be very useful if implemented. However only enough of them were made part of the experimental system so the possibility of their use in the language could be tested.

PART FOUR: RESULTS AND CONCLUSIONS

Part Two introduced a theoretical model for a document retrieval system. The experimental system developed to test the model in a realistic environment was described in Part Three. In this part we present the experimental results obtained with the system and the conclusions about the model that can be drawn from them.

This final part is divided into two chapters.

Chapter IX: Experimental Results

Chapter X: Conclusions

CHAPTER IX

EXPERIMENTAL RESULTS

In the first section of this chapter some data on the general characteristics of clusters will be presented. Then some specific examples will be given illustrating the composition of clusters in terms of the frequency of occurrence of title words, authors, and citations of the included articles.

In the next two sections clusters will be compared with some existing sets of documents which have already been judged to be mutually pertinent. Three bibliographies found in review articles that are not part of the T.I.P. file and two subject categories compiled by indexers will be used for this purpose.

Finally, the results of two tests will be presented in which clusters were evaluated by representative users of the document file.

9.1 Cluster Parameters

Before attacking the problem of whether or not clusters contain sets of documents that are mutually interesting to users, it may be appropriate to first summarize some of the more general features of clusters. This section will, accordingly, present statistics on certain cluster parameters.

The data from which the statistics are drawn come from the tests of Sec.'s 9.3 to 9.5. They are, of course, a function of the particular requests presented to the system during the tests and of the composition

of the T.I.P. file at the time. It was thought, however, that this would serve as an introduction to the experimental results.

The first parameter that will be described is cluster size. Fig. 9.1 shows the distribution by size of some different clusters generated by the procedure. The largest cluster found so far contains 159 documents, while the smallest contains only one document.

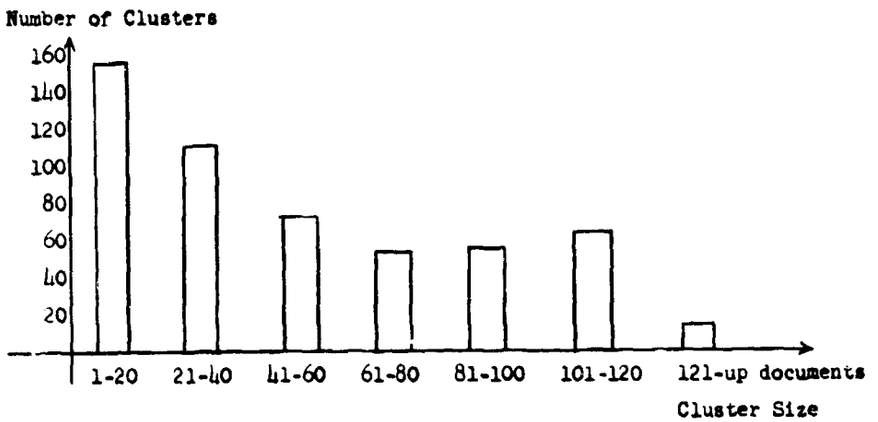


Fig. 9.1. Distribution of cluster size for 490 clusters.

One of the important features of the clustering procedure as described in Chapter V is its ability to adjust the size of the answer to fit the request. This is accomplished by applying a bias to the links of the document network (See Sec. 4.4). About 82% of the clusters examined utilized either a positive or negative bias with the other 18% having no (zero) bias.

In Fig. 9.2 the distribution of clusters for various ranges of bias is shown. Fig. 9.3 indicates that the average cluster size increases monotonically as the bias increases. This curve seems to follow the equation $y^2=80(x-12)$ where y is the cluster size and x is the bias. We will not attempt to explain why this is the case here.

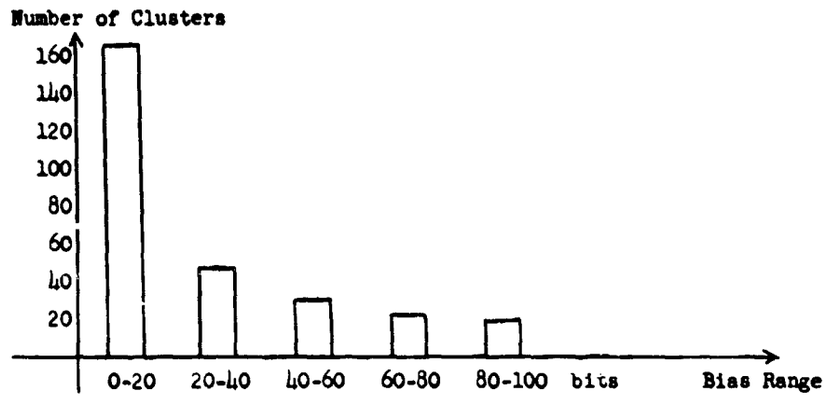


Fig. 9.2. Distribution of clusters by bias for 275 clusters.

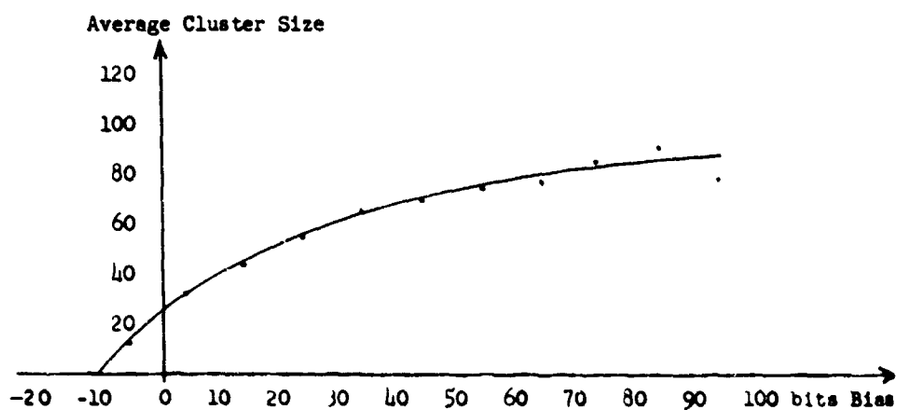


Fig. 9.3. Plot of average cluster size versus bias for 340 clusters.

Another characteristic of the procedure that can be studied is the way documents are deleted from the set (S) that is being formed. The formation of 37 clusters was observed. It was found that an average of three documents were deleted per cluster. This resulted in an average deletion of one document in every 15 iterations. It was also found that about 90% of the documents that were deleted from S were added to S

some later time during the clustering.

Let us next ask when during the clustering process deletions occur. Fig. 9.4 indicates that deletions are more likely to occur toward the end of the clustering process.

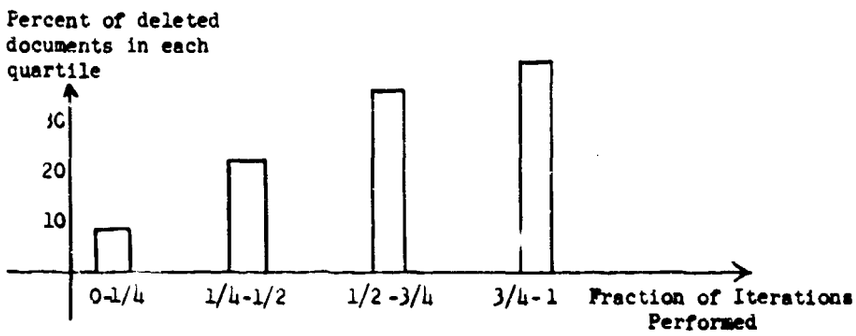


Fig. 9.4. Percent of deletions occurring in each quartile of the clustering process. (average for 75 clusters)

In the final portion of this section we will describe the way the procedure responds to requests that are inconsistent or ambiguous. A specific example, (Cluster A_1 of Sec. 9.33) is used for this purpose. The first test consisted of holding the pertinent (Y) set of the request constant and in successively placing every other member of the Cluster A in the non-pertinent (Z) set ($y=a_1$; $z=a_i$ $i=1,\ldots,n$). The results are shown in Fig. 9.5 and 9.6.

There are three basic types of responses that resulted. In seven cases the size of the Cluster was reduced. This was, in general, what happened when the document specified as not pertinent had a smaller bias to A than a_1 did. In eight other cases the procedure was found to select another cluster (B,D, or E) containing some documents that were

not part of the original cluster. In the remaining twelve cases the request was judged to be inconsistent. A careful examination of the network revealed that in each of the twelve cases there was at least one cluster which could have satisfied the request. The reasons why the procedure was not able to locate a valid answer cluster in these cases have already been discussed in Sec. 5.51.

Fig.'s 9.5 and 9.6 illustrate two types of request ambiguity. The first type is hierarchal in nature involving clusters that are subsets of larger clusters. Take, for example, the request, $Y=a$; $Z=a_{18}$. It can be satisfied not only by the cluster listed for it in Fig. 9.5, but also by the smaller clusters listed for a_7, a_{10}, and a_{20}. The second type of ambiguity is due to the fact that clusters overlap. Thus the clusters B, D, or E also satisfy the request $Y=a_1$;$Z=a_{18}$.

A second test was conducted in order to further study the extent of the second type of ambiguity. In this test a given document was specified as pertinent and a cluster was found. The document which had the highest correlation to the cluster found was then specified as non-pertinent and another search was conducted. If a second cluster was found then the document with the highest correlation to the new cluster was added to Z and the process was continued. At some point the request became inconsistent.

The results of this type of test on six articles is given in Fig. 9.7. Note that document a_1 of Fig. 9.7 would result in the test pattern of Example 4 since a_{21} is most highly correlated to A and the answer to the request ($Y=a_1$;$Z=a_{21}$) is inconsistent.

Articles in Cluster (A)	Bias of a_i to A	Rank by bias (largest first)	Answer to the Request: $Y=a_1$; $Z=a_i$
a_1	114.9 bits	20	Inconsistent
a_2	132.7	5	B
a_3	121.0	15	Inconsistent
a_4	130.3	8	Inconsistent
a_5	103.2	26	$A \cap a_5$
a_6	118.4	16	B
a_7	116.3	17	$A \cap (\overline{a_5 a_6 a_7 a_{10} a_{12} a_{15} a_{16} a_{18}})$
a_8	131.9	6	Inconsistent
a_9	123.2	13	Inconsistent
a_{10}	109.8	23	$A \cap (\overline{a_5 a_{10} a_{12} a_{15} a_{16} a_{18}})$
a_{11}	127.4	9	Inconsistent
a_{12}	104.6	25	$A \cap (\overline{a_5 a_{12} a_{16}})$
a_{13}	136.6	4	Inconsistent
a_{14}	126.1	11	Inconsistent
a_{15}	110.4	22	D
a_{16}	102.8	27	$A \cap (\overline{a_{16}})$
a_{17}	122.0	14	B
a_{18}	106.6	24	$A \cap (\overline{a_5 a_{12} a_{16} a_{18}})$
a_{19}	116.2	18	E
a_{20}	112.3	21	$A \cap (\overline{a_5 a_{10} a_{12} a_{15} a_{16} a_{18} a_{20}})$
a_{21}	146.4	2	E
a_{22}	124.1	12	Inconsistent
a_{23}	155.6	1	Inconsistent
a_{24}	141.8	3	Inconsistent
a_{25}	115.4	19	E
a_{26}	130.4	7	Inconsistent
a_{27}	127.0	10	E

$B=(a_1 a_3 a_{15} a_{18} a_{20})$ plus 12 other articles

$D=(a_1 a_2 a_4 a_5 a_{17} a_{20})$ plus 11 other articles

$E=(a_1 a_2 a_{20})$ plus 20 other articles

Fig. 9.5. Example of clusters which result when documents are specified as non-pertinent.

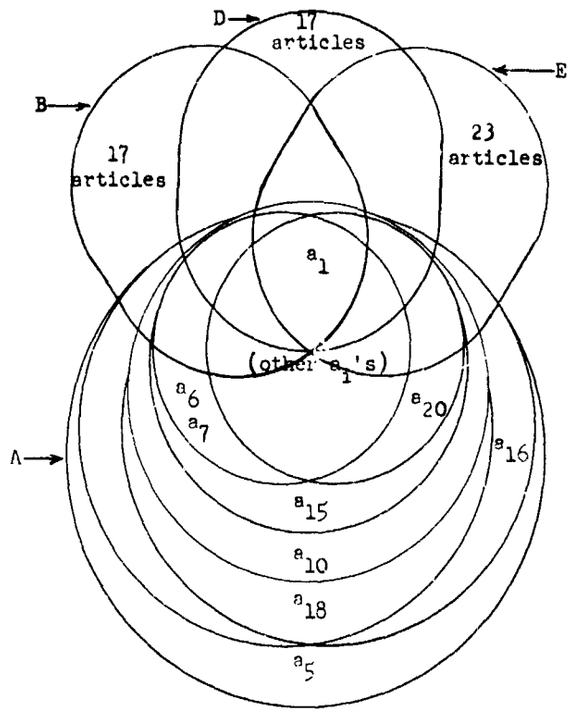


Fig. 9.6. Diagram of relationship of clusters of Fig. 9.5. (Each circle represents a cluster)

Example	Size of successive answer clusters
1	31, 22, 27, inconsistent
2	17, 125, 4, 2, inconsistent
3	22, 36, 23, 23, inconsistent
4	27, inconsistent
5	33, 27, inconsistent
6	39, 33, 14, inconsistent

Fig. 9.7. Test of request ambiguity.

9.2 Cluster Composition

In the last section statistics on some of the more general features of clusters such as size and bias were presented. In this section the composition of clusters will be described in terms of data available in the T.I.P. file. In particular, examples will be given of the composition of clusters in terms of the title words, authors, and citations of the included articles.

In Fig. 9.8 we list in order of frequency of occurrence the title words for six clusters. Note that the common "function" words (in, of, the, and, on, etc.) have been omitted from all of the lists except for Example A. Also the lists have been truncated to include only the words that occurred most often in the titles. The full titles of Example B are shown in Fig. 9.16.

In none of the cases studied did the title of every article in a cluster contain the same word. For Fig. 9.8 the word that comes closest to occurring in every title is "plasma" of Example D, which occurs in 18/22=82% of the titles. If one were to group together words of equivalent meaning, then "superconducting" and "superconductors" in Example A would be highest with 27/31=88 .

In Fig. 9.9 some similar examples are given for the authors of the articles in clusters. In Example A it was found that E. Schlomann is the author of two other papers in the T.I.P. file (in addition to the four listed), R. I. Joseph of one other, and W. Strauss of two others.

In Fig. 9.10 citation counts are given for the same three clusters that were used in Fig. 9.9. In Example A there is one citation which is found in all of the articles in the cluster. In Example B, 46/64=72% of the articles cite the same paper, while only 10/35=28% do in Example

Example A

Cluster A_7 of
Sec. 9.33.
31 articles
99 words

22	in
22	superconducting
19	of
13	ultrasonic
10	energy
10	gap
9	the
8	attenuation
5	and
5	superconductors
5	tin
4	by
4	determination
4	waves
3	(11 words)
2	(16 words)
1	(58 words)

Example B

Cluster A_1 of
Sec. 9.31.
12 articles
66 words

7	waves
5	spin
3	garnet
3	iron
3	magnetic
3	magneto-elastic
3	microwave
3	nonuniform
3	propagation
3	yttrium
2	crystal
.	
.	
.	

Example C

Cluster A_5 of
Sec. 9.33.
22 articles
75 words

12	quantum
11	oscillations
8	ultrasonic
6	attenuation
6	field
6	giant
6	metals
5	effect
4	magnetic
4	magnetoacoustic
3	absorption
3	sound
2	alphen
.	
.	
.	

Example D

Cluster A_8 of
Sec. 9.52.
22 articles
84 words

18	plasma
9	turbulent
8	waves
5	particles
4	electromagnetic
4	turbulence
3	charged
.	
.	
.	

Example E

Cluster A_{12} of
Sec. 9.51.
40 articles
154 words

20	plasma
17	probe
11	langmuir
9	probes
5	characteristics
5	field
5	magnetic
4	electrostatic
4	resonance
4	studies
3	double
.	
.	
.	

Example F

Cluster for article
8 of Fig. 9.11
22 articles
81 words

16	optical
7	generation
7	harmonic
6	nonlinear
5	theory
3	second
.	
.	
.	

Fig. 9.8. Title-word frequency counts for six clusters. (The number to the left of each word is the number of times it occurs in the titles of the cluster.)

Example A	Example B	Example C
Cluster A_1 of Sec. 9.31.	Cluster A_4 of Sec. 9.32.	Cluster A_5 of Sec. 9.52
12 articles	64 articles	35 articles
13 authors	75 authors	38 authors
4 Schlomann Ernst	7 Spector Harold N.	7 Kraichnan Robert H.
3 Joseph R. I.	4 Prohofsky E. W.	2 Deissler Robert G.
2 Damon R. W.	3 Gurevich V. L.	2 Eschenroeder Allan Q.
2 Strauss W.	3 Kroger Harry	1 (35 authors)
2 Van De Vaart H.	3 Pustovoit V. I.	
1 (8 authors)	2 (8 authors)	
	1 (62 authors)	

Fig. 9.9. Author frequency counts for three clusters.

Example A	Example B	Example C
Cluster A_1 of Sec. 9.31.	Cluster A_4 of Sec. 9.32.	Cluster A_5 of Sec. 9.52.
12 articles	64 articles	35 articles
35 citations	369 citations	195 citations
12 11-34-1298	46 41-7-237	10 802-5-497
7 41-8-357	31 11-33-2457	6 227-2-124
6 11-35-159	29 41-9-87	5 8-30-301
4 11-35-167	22 11-33-40	5 799-7-1030
3 1-105-390	19 11-34-1548	5 802-12-242
3 1-120-2004	19 41-9-296	5 802-13-369
3 11-35-1022	18 1-127-1084	5 802-16-33
2 1-125-1950	14 1-126-1974	4 (3 citations)
2 11-31-1647	14 41-8-4	3 (13 citations)
2 11-35-2382	10 41-4-505	2 (33 citations)
2 11-35-2382	9 1-134-1302	1 (139 citations)
2 11-36-875	9 28-8-161	
2 41-6-620	7 (4 citations)	
2 41-12-583	6 (7 citations)	
2 708-19-308	5 (12 citations)	
1 (21 citations)	4 (12 citations)	
	3 (18 citations)	
	2 (49 citations)	
	1 (262 citations)	

Fig. 9.10. Citation frequency counts for three clusters.

C. Example C is an illustration of an area where all of the articles do not cite one central paper and yet through the use of a large positive bias they can be pulled together into a cluster.

The papers listed in Fig. 9.10 are identified by three numbers: The journal code (see Fig. 6.3), volume, and page number. Thus 1-136-441 is the paper beginning on page 441 in volume 136 of the Physical Review.

9.3 Comparison to Bibliographies

The next test will be to compare the bibliographies found in certain papers with clusters formed by the procedure. Consider, for example, a paper with 20 citations. It would be of interest to know if a cluster can be formed which includes most, if not all, of the 20 citations.

For this purpose three articles were selected from the special October 1965 issue of the IEEE Proceedings on ultrasonics. It was decided that these articles which are not part of the T.I.P. file would insure some degree of independence between the data base and evaluation criteria. The IEEE Proceedings represented a journal which is closely related to the T.I.P. physics file and yet is not actually part of the file. Since the T.I.P. file covers only the last three years, a recent issue of the IEEE Proceedings was needed if a suitable fraction of the bibliographies of the evaluating papers were to be found in the T.I.P. file.

Of the twenty-seven articles in the October IEEE Proceedings, only ten cite ten or more articles in the T.I.P. file. Fig. 9.11 tabulates these ten papers. For the three articles to be used in evaluating the clustering procedure we selected the two papers with the highest percent

of their bibliographies in the T.I.P. file (1 and 2) and the paper with the most references to the T.I.P. file (7).

	Articles in Proc. IEEE Vol. 53	Total Citations	Citations to T.I.P. file	Percent of Bibliography in T.I.P. file
1.	pp. 1495-1507	22	10	46 %
2.	pp. 1452-1464	38	16	42
3.	pp. 1517-1533	58	22	38
4.	pp. 1438-1451	86	32	37
5.	pp. 1508-1517	47	17	36
6.	pp. 1320-1336	33	11	33
7.	pp. 1586-1603	128	36	28
8.	pp. 1604-1623	67	18	27
9.	pp. 1387-1399	56	13	23
10.	pp. 1547-1573	101	15	15

Fig. 9.11. Articles in the October 1965 Issue of the IEEE Proceedings that have 10 or more references to the T.I.P. file.

9.31 Bibliography 1 (IEEE Proc., v. 53, p. 1495)

From Fig. 9.11 we note that the article beginning on page 1495 has 22 citations, 10 of which are to articles in the T.I.P. file. Fig. 9.12 lists the 10 articles as set B and also lists some other sets of papers that will be found useful in the discussion that follows. The ith document in set B will be referred to as b_i, etc.

The answer clusters obtained by the procedure for 18 different requests are tabulated in Fig. 9.13. The symbol $A[Y(b_i)Z(b_j)]$ stands for the answer cluster with b_i specified as interesting and b_j specified as not interesting (i.e. $Y=b_i$), $Z=(b_j)$).

B	E	H
1-136-442	11-36-3453	1-129-991
11-35-159	646-5-176	1-130-439
11-35-167		1-134-172
11-35-1022		1-134-407
11-36-108	**F**	1-136-1657
11-36-1243	11-36-2426	1-137-182
11-36-1267	11-36-3599	11-34-1629
11-36-1579	41-12-325	11-34-2639
41-12-583	646-6-18	11-36-2387
646-5-33		11-36-3102
		41-11-69
	G	41-11-69
D	1-130-647	41-14-254
11-36-1245	11-35-836	49-4-129
11-36-3402	11-35-993	310-7-1892
	11-36-661	146-2-38
	11-36-1845	669-16-410
		669-18-235
		790-8-594

Fig. 9.12. The sets of articles included in the clusters for Bibliography 1.

Answers to Selected Requests:

$A[Y(b_i)]=A_1$ for $i=2\ldots5,7,8,10$

$A[Y(b_1)]=A_4$

$A[Y(b_6)]=A_2$

$A[Y(b_9)]=A_3$

$A[Y(b_9),A(h_4)]=A_1$

$A[Y(b_9),Z(h_{14})]=A_1$

$A[Y(b_ib_9)]=A_1$

$A[Y(b_1b_2)]=A_1\cup F$ plus 5 members of H and 50 other articles

$A[Y(b_2\ldots b_{10})]=A_2$

$A[Y(b_1\ldots b_{10})]=A_2\cup A_3$

Definitions of Clusters:

$A_1=(b_2\ldots b_5,b_7,b_8,b_{10})\cup D\cup E$

$A_2=A_1\cup(b_6)\cup F$

$A_3=(b_9)\cup E\cup H$

$A_4=(b_1)\cup G$

Fig. 9.13. List of the answer clusters formed for Bibliography 1.

In Fig. 9.14 the probable answers for requests consisting of other combinations of b's are suggested. All of the requests listed in this figure have not been actually tested, but experience with the clustering procedure and the results of Fig. 9.13 make it appear reasonably safe to assume that the conclusions are correct.

$A[Y(b_i b_j)]=A_1$ for $i,j=2\ldots5,7\ldots10$ $(i \neq j)$

$A[Y(b_6 b_i)]=A_2$ for $i=2\ldots10$

$A[Y(b_1 b_i)]=$ (large set of 70-100 articles) for $i=2\ldots10$

$A[Y(b_9)Z(h_i)]=A_1$ for $i=1\ldots18$

$A[Y(\text{Any combination of } b_2\ldots b_5, b_7\ldots b_{10})]=A_1$

$A[Y(b_6 \text{ plus any combination of } b_2\ldots b_{10})]=A_2$

$A[Y(b_1 \text{ plus any combination of other b's})=$(large set of 70-100 articles)

Fig. 9.14. Generalizations suggested by the results of Fig. 9.13.

A diagram showing the amount of overlap of the various answer clusters is shown in Fig. 9.15.

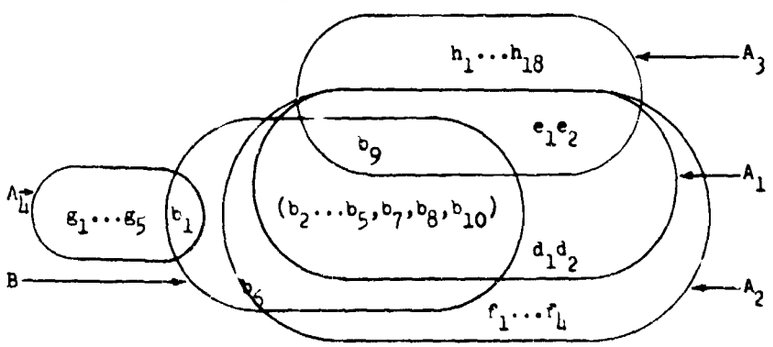


Fig. 9.15. Sketch showing the relationship of the answer clusters of Bibliography 1.

Some comments will now be made concerning the results given in Fig.'s 9.12 - 9.15. When the request consists of a single member of the bibliography, the same answer results in 7 out of 10 cases. This cluster, A_1, contains 8 of the 10 articles in the bibliography (b_1 and b_6 are omitted).

The article b_9 is included in A_1 but does not result in A_1 when used as a request. It results in an almost completely different set of documents (A_3) which contains only one member of the bibliography. The request $Y(b_9)$ is, therefore, ambiguous with either A_1 or A_3 being a valid answer. To resolve the ambiguity various documents from the set H were placed in the non-pertinent set Z. This shifted the answer from A_3 to A_1. It was found that the ambiguity could also be resolved by placing an additional document in the Y set. Thus a request of $Y(b_2b_9)$ also resulted in the answer A_1.

The cluster A_2 exemplifies another type of ambiguity. The set A_1 is a subset of the set A_2 and thus the requests $Y(b_i)$ where i=2...5,7, 8,10, could be satisfied by either A_1 or A_2. The request $Y(b_6)$ can only be satisfied by A_2, however, since b_6 is not included in A_1. Thus the article b_6 is slightly "beyond" the cluster A_1 and if used in the Y set of the request results in more general cluster A_2 of 17 documents instead of the cluster A_1 of 12 documents. Note that both requests of the form $Y(b_ib_6)$ with i=2...10 and the larger request $Y(b_2...b_{10})$ result in the cluster A_2.

The only article from Bibliography 1 which is not included in A_2 is b_1. The request $Y(b_1)$ results in the cluster A_4 which is disjoint from any of the clusters discussed so far. When requests of the form $Y(b_1b_i)$ i=2...10 are used, very large clusters result including most

of the documents listed in Fig. 9.12 and many more. A check of the paper from which Bibliography 1 was taken reveals that b_1 is cited only as a source for the values of some constants. It is suggested that this may be the reason it does not fit into the closely-related cluster A_2 which includes the other nine papers.

One final observation will be made. There are four articles in A_1, and nine in A_2 that are not part of the original bibliography. The question of whether these papers constitute valid additions to the bibliography will be discussed in Chapter X. Let us at this point, however, present the titles of the papers in A_1 (Fig. 9.16) as an illustration of the type of additional articles included in the clusters.

9.32 Bibliography 2 (IEEE Proc., v. 53, p. 1452)

In Fig.'s 9.17 - 9.20 we present the same data for Bibliography 2 that were given for Bibliography 1. Here again a large majority of the documents (11 of 16) in the bibliography lead to the same cluster (A_1) when specified as interesting in the request.

From Fig. 9.20 we observe that clusters $A_1,\ldots,A_4$ form a hierarchal series of increasingly larger sets with each new set including the previous set. The set A_4 contains 14 of 16 members of the bibliography and 50 other documents. The set A_1 is the only set in the series that has 0 bias. The series can, of course, be extended to sets which are larger than A_4 or to subsets of A_1 by additional changes in the bias.

There are two members of the bibliography (b_6 and b_{13}) that do not fit into the pattern set by the other 14 members. The article b_6 has no positive connection to any other paper (i.e. none of the papers it

Print the titles of the articles related to J Appl Phys v. 35 p. 159.

12 documents in set 1.

Journal of Applied Physics, Volume 35, page 159.
Generation of spin waves in nonuniform magnetic fields I. Conversion of electromagnetic power into spin-wave power and vice versa.

Page 167
Generation of spin waves in nonuniform magnetic fields II. Calculation of coupling strength

Page 1022
Magneto-elastic waves in yttrium iron garnet

Volume 36, page 118
Magneto-elastic waves in yttrium iron garnet

*Page 1245
Electronically variable delay of microwave pulses in single-crystal YIG rods

Page 1267
Microwave magneto-elastic resonances in a nonuniform magnetic field

Page 1579
Demagnetizing field in nonellipsoidal bodies

* Page 3402
Anisotropic spin-wave propagation in ferrites

*Page 3453
Propagation of magnetostatic spin waves at microwave frequencies in a normally-magnetized disc

Physical Review Letters, Volume 12, page 583
Dispersion of long-wavelength spin waves from pulse-echo experiments 1

Applied Physics Letters, Volume 5, page 33
Propagation, dispersion, and attenuation of backward-traveling magneto-elastic waves in YIG

*Page 176
Hall effects in single-crystal spheres of Yttrium iron garnet (YIG)

End. 9.6 sec. used.

Fig. 9.16. Titles of articles in the A_1 cluster. (The four * articles were not part of the original bibliography.)

B	D	D (Con't.)	E	H (Con't.)
1-134-1302	1-129-1009	49-4-194	41-14-706	1-135-51
1-135-1761	1-130-910	49-13-285	310-6-2233	1-135-1562
1-136-772	1-131-1087	49-17-14		1-137-801
1-136-1731	1-131-2512	80-19-674	**F**	1-137-1305
1-138-1721	1-132-522	80-20-1131	669-17-1432	1-138-534
11-35-125	1-132-679	80-30-1424		1-138-1559
11-36-528	1-134-507	80-20-1647	**G**	1-139-539
41-11-246	1-135-1388	80-20-1946	1-135-869	1-140-2110
41-12-47	1-137-311	80-20-2160	41-12-241	1-142-126
41-12-555	1-138-1250	310-5-1818	49-19-268	3-82-401
41-13-434	1-139-1949	310-7-688	310-6-2473	3-86-709
41-14-372	3-81-130	384-32-100	646-7-45	11-36-22
646-4-82	11-35-137	612-3-448	646-7-82	11-36-3281
646-4-190	11-35-1483	612-3-698		12-39-1493
646-4-212	11-36-3728	669-16-383	**H**	21-30-1717
146-6-81	21-31-1700	669-16-1612	1-130-919	21-30-1817
	29-30-149	669-19-242	1-131-95	41-11-14
	29-31-957	669-19-1407	1-131-1469	41-11-146
	41-13-308	669-12-1113	1-133-133	80-20-363
	43-37-545	821-2-149	1-133-1493	669-21-1034
	49-4-45		1-134-728	821-2-141
			1-134-1313	
			1-134-1429	

Fig. 9.17. The sets of articles included in the clusters for Bibliography 2.

Answers to Selected Requests:

$A[Y(b_i)]=A_1 \quad i=1,2,3,5,7,8,9,11,12,14,16$

$A[Y(b_{10})]=A_2$

$A[Y(b_4)]=A_3$

$A[Y(b_{15})]=A_4$

$A[Y(b_6)]=(b_6)$

$A[Y(b_{13})]=A_5$

$A[Y(b_2b_4)]=A_3$

$A[Y(b_{15}b_{16})]=A_4$

$A[Y(b_4b_{15})]=A_4$

$A[Y(b_4b_{13})]=A_4 \cup b_{13} \cup$ (29 others)

$A[Y(b_1 \ldots b_5b_7 \ldots b_{12}b_{14} \ldots b_{16})]=A_4$

$A[Y(b_{14})Z(d_{22})]=A_5$

$A[Y(b_{14})Z(b_3)=A_5 \cap (\overline{b_9b_{11}b_{18}b_{19}b_{22}b_3})$

$A[Y(b_{14})Z(b_3b_{13})]=(b_8b_9b_{11}b_{14}) \cup (d_2d_6d_{20}d_{22}d_{24}d_{25}d_{41})$

Definitions of Clusters:

$B_1=(b_1b_2b_3b_5b_7b_8b_9b_{11}b_{12}b_{14}b_{16})$

$B_2=B_1 \cup b_{10}$

$B_3=B_2 \cup b_4$

$B_4=B_3 \cup b_{15}$

$A_1=B_1 \cup D$

$A_2=B_2 \cup D \cup E$

$A_3=B_3 \cup D \cup E \cup F$

$A_4=B_4 \cup D \cup E \cup F \cup G$

$A_5=(b_3b_{13}b_{14}) \cup H$

Fig. 9.18. List of the answer clusters formed for Bibliography 2.

$A[Y(b_i b_j)]=A_1$	for $b_i, b_j \subset B_1$
$A[Y(b_{10} b_i)]=A_2$	for $b_i \subset B_1$
$A[Y(b_4 b_i)]=A_3$	for $b_i \subset B_2$
$A[Y(b_{15} b_i)]=A_4$	for $b_i \subset B_3$
$A[Y(b_6 b_i)]=$ Inconsistent	(b_6 is not linked to any other paper.)
$A[Y(b_{13} b_i)]=A_4 \cup b_{13}$ (29 others)	for $b_i \subset B_3$
$A[Y(X_1)]=A_1$	for $X_1 \subset B_1$
$A[Y(b_{10} X_1)]=A_2$	for $X_1 \subset B_1$
$A[Y(b_4 X_2)]=A_3$	for $X_2 \subset B_2$
$A[Y(b_{15} X_3)]=A_4$	for X_3 B_3

Fig. 9.19. Generalizations suggested by the results of Fig. 9.19.

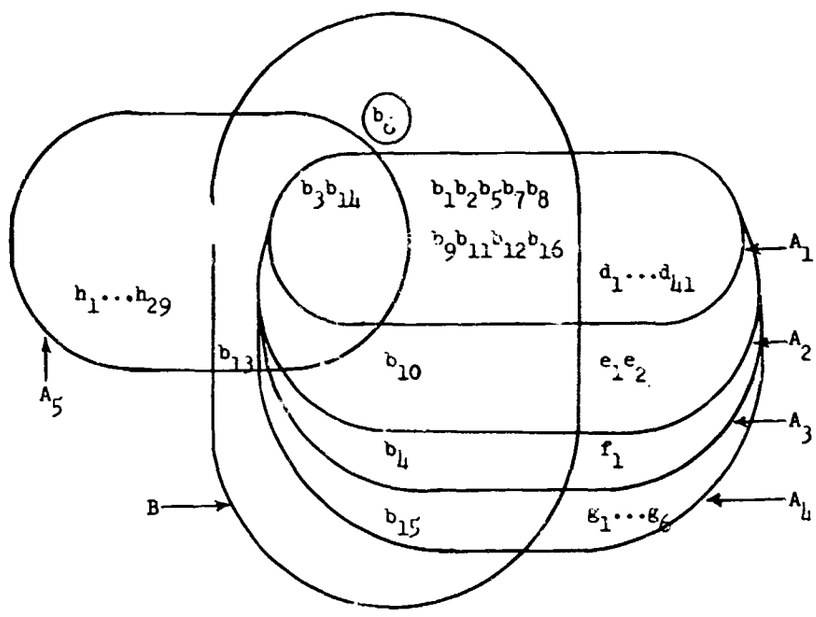


Fig. 9.20. Relationship of answer clusters of Bibliography 2.

cites are cited by other papers) and is thus isolated from the rest of the file. Article b_{13} can be included in a cluster with the rest of the papers if the bias is made large enough. The cluster $A[Y(b_4 b_{13})]$ contains, for example, all of the bibliography except b_6.

There is one significant characteristic that the five papers not included in A_1 have. They all have relatively few citations. Articles b_6 and b_{13} have only two citations each. Articles b_{10} and b_{15} have only three. Article b_4 has seven. In contrast the bibliography articles in A_1 all have seven or more citations except b_7 and b_{14} which have five each. It is suggested that perhaps the reason b_6 and b_{13} are not included in the cluster A_1 is that they have insufficient references to position them properly in the network.

9.33 Bibliography 3 (IEEE Proc., v. 53, p. 1586)

In Fig.'s 9.21 to 9.24 the data for bibliography 3 is presented. The paper from which this bibliography is taken has four sections (I,II,III,IV) with section III haveing four subsections (III A, B, C, D). The particular section (and subsection) in which each bibliographic item is first cited is noted in Fig. 9.21. These section numbers are also noted over the symbols for the documents in Fig. 9.23. Some of the documents in Fig. 9.23 are inclosed in parenthesis. This is to indicate that the document has already appeared elsewhere in the diagram.

From Fig. 9.23 we note that a hierarchal series of clusters (A_1 to A_4) similar to the one in Fig. 9.20 is formed by 13 of the documents of Sec. III. A similar but separate series (A_6 to A_8) is formed by the documents of Sec. IV. There also appears to be a separation of the

B

1-129-12	IIIA
1-129-18	IIIC
1-129-652	IIIA
1-131-111	IIIA
1-131-653	IIIA
1-131-1497	IV
1-131-2420	IIID
1-132-1062	IV
1-132-1073	IV
1-132-2039	IV
1-133-1487	IV
1-135-740	IIIA
1-135-1161	IV
1-136-1096	IIID
1-137-211	IIIC
1-137-889	IIIC
1-137-1400	IIIC
1-138-487	IIIC
21-29-357	IV
41-11-316	IIID
41-12-104	IIIC
41-12-166	IIIC
41-12-360	IIIE
41-13-162	IIIC
49-7-112	IIID
49-8-155	IIIA
49-8-160	IV
49-12-297	IIIC
49-13-287	IIIC
49-14-13	IIIA
49-14-73	IIIC
49-17-184	IIIC
646-6-111	IV
669-17-50	IIIA
669-18-403	IIIC
669-20-552	IIIA

D

1-130-929
1-132-522
1-132-535
1-135-181
1-137-883
1-140-1355

E

1-129-1990
1-131-2512
1-133-1589
1-134-507
1-136-1170
1-137-1717
1-138-88
1-138-1453
1-139-1849
41-12-357
310-7-383
669-17-628

F

669-18-1125
669-19-159

G

1-138-1191
669-16-154
669-18-419

H

1-133-84
1-136-22
41-11-552

J

49-5-233
49-7-133
80-20-1424

K

1-131-73
1-132-621
1-134-1
1-135-19
1-136-306
1-136-203
1-136-893
1-136-1471
1-138-1661
1-139-746
1-140-1902
1-141-452
1-143-229
41-15-862
669-16-945
669-18-834
669-21-704

R

669-18-1260

M

1-129-1088
1-130-92
1-130-565
1-131-617
1-131-1995
1-131-2078
1-132-1512
1-133-443
1-133-1546
1-135-1698
1-137-1172
1-137-1706
1-139-823
1-139-1459
1-140-2051
1-140-2065
1-141-452
1-141-553
1-143-406

M (Con't.)

80-18-1569
669-16-1481
669-17-87
669-18-51
669-18-896
669-20-267
669-20-560
669-20-583
669-21-75

N

1-131-2433
1-131-2463
1-132-1991
1-136-998
1-137-431
41-12-558
80-20-1136

P

1-133-1104
1-139-1876
1-143-452
49-13-282

Q

1-129-2055
1-132-1885
1-140-187
1-140-1429
1-141-592
49-7-7
49-12-297
80-20-1374
310-6-2565
669-16-818
669-16-1459
669-18-908

9.21. The sets of articles included in the clusters for Bibliography 3.

Answers to Selected Requests:

$A[Y(b_i)]=A_1$ $i=1,2,20,23,36$

$A[Y(b_{14})]=A_2$

$A[Y(b_{35})]=A_3$

$A[Y(b_5)]=A_4$

$A[Y(b_i)]=A_5$ $i=15\ldots17,22,24,28,29,32$

$A[Y(b_i)]=A_6$ $i=8\ldots11,13,27$

$A[Y(b_6)]=A_7$

$A[Y(b_i)]=A_8$ $i=18,19$

$A[Y(b_i)]=A_9$ $i=4,34$

$A[Y(b_7)]=A_{10}$

$A[Y(b_{30})]=A_{11}$

$A[Y(b_4)]=A_3 \cup (b_{15}b_{17}b_{21}h_3j_1)$

$A[Y(b_i)]=$ Misc. large sets of documents (88-159 articles) $i=3,12,25,26,31,33$

$A[Y(b_{18}b_{21})]=A_5$

$A[Y(b_2b_{22}b_{24}b_{35})]=[A_1 \cup A_5 \cup (b_7b_{35}f_2)] \cap (\overline{b_{29}})$

$A[Y(b_5b_{29})]=$(cluster of 108)

$A[Y(b_{16}b_{18}b_{29}b_{35})]=A_{12}$

Definitions of Clusters:

$A_1=(b_1b_2b_4b_{23}b_{34}b_{36}b_{16}b_{18}b_{20}) \cup D \cup E$

$A_2=A_1 \cup (b_7b_{14}) \cup F$

$A_3=A_2 \cup (b_{35}) \cup G$

$A_4=A_3 \cup (b_5) \cup H$

$A_5=(b_{15}b_{16}b_{17}b_{18}b_{20}b_{21}b_{22}b_{24}b_{28}b_{29}b_{32}) \cup D \cup J \cup (g_1h_1)$

$A_6=(b_8b_9b_{10}b_{11}b_{13}b_{27}) \cup K \cup (h_1h_2e_5e_8)$

$A_7=A_6 \cup (b_6) \cup L$

$A_8=A_7 \cup (b_{18}b_{19})$

$A_9=(b_4b_5b_{14}b_{34}b_{36}) \cup M$

$A_{10}=A_9 \cup (b_7) \cup N \cup (e_7)$

$A_{11}=(b_1b_5b_7b_{30}) \cup P \cup (d_6e_1e_6e_8h_1h_2m_{15}m_{17}q_6)$

$A_{12}=A_3 \cup A_5 \cup (m_{12}q_7)$

Fig. 9.22. List of answer clusters formed for Bibliography 3.

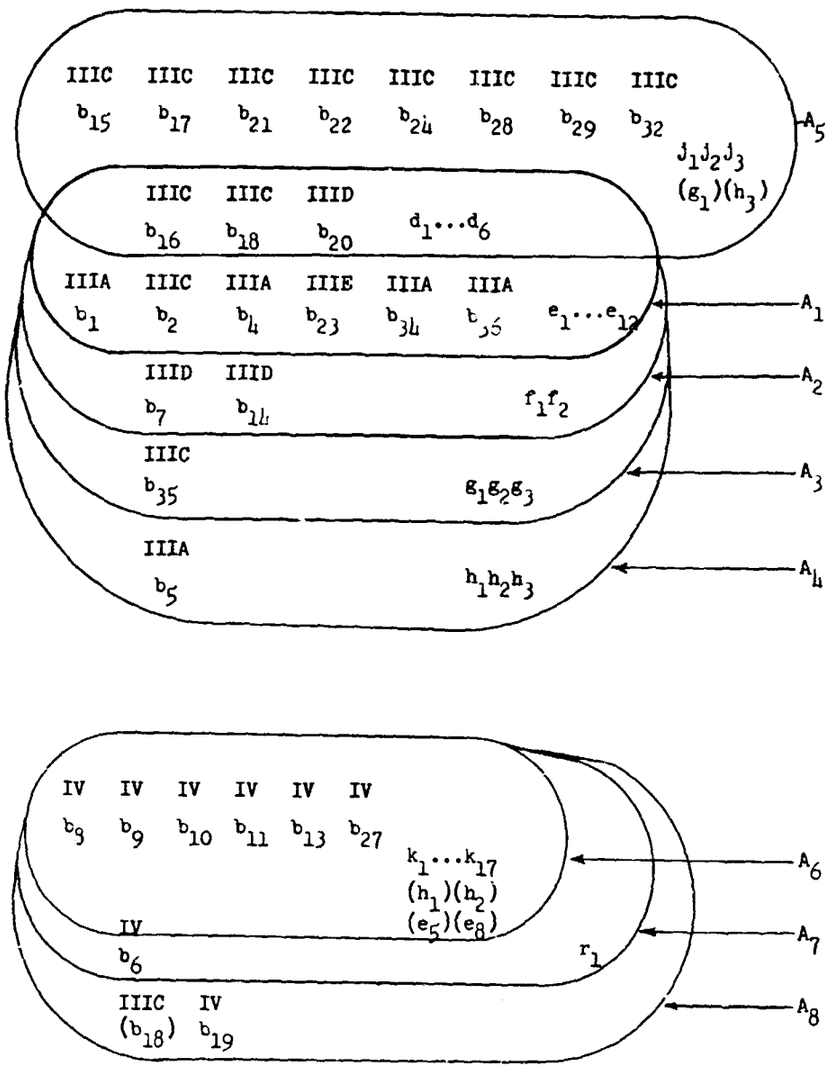


Fig. 9.23. Relationship of answer clusters of Bibliography 3.

documents by subsection within Sec. III. Note that 10 of the 13 documents cited in subsetion IIIC are included in cluster A_5.

The structure of the clusters in this example was found to be considerably more complex than in the previous two examples and no attempt is made to predict the results of requests that have not been explicitly tested. One can gain some appreciation of the complexity of the interrelationships between the clusters by an examination of clusters A_9 to A_{11}.

As with Bibliographies 1 and 2 there are a few of the documents that are not included in the clusters of Fig. 9.23. Nine articles are cited by Sec. IV. All of these except b_{33} are included in the cluster A_8. Thirteen articles are cited by Sec. IIIC. All of them but b_2,b_{31}, and b_{23} are in A_5 and all but b_{31} are in A_{12}. The cluster A_{12} is more general in that it includes not only articles cited by Sec. IIIC but also those cited by Sec.'s IIIA, D and E. Of the 27 articles cited by Sec. III, 20 are included in A_{12}. The seven missing articles are b_3,b_5, b_{12},b_{25},b_{26},b_{30}, and b_{31}.

The article b_3 was examined in detail in an attempt to discover why it was not included in A_{12}. It was found to have six references. Of the six, one was keypunched incorrectly. Two of them are to articles in a Russian journal (Soviet Physics - JETP), whereas the other references to these articles in the T.I.P. file are to the journal in which the English translation is found. A fourth reference is to a paper written by the same author and not cited by anyone else, and a fifth is to a bulletin, which was evidently not sufficient to cause it to be included in A_{12}. It was found that if the references had been correctly keypunched and had been to the correct English translations, b_3 would

have been included in A_{11} and probably A_{12}.

There is one other feature of the article from which Bibliography 3 was taken. In the final paragraph the author made this comment.

> "I wish to thank ...A. R. Mackintosh for calling B. I. Miller's work to my attention."

The article by B. I. Miller was checked to see if it would have been included in any of the clusters if it had been part of the T.I.P. file. It was found to have only one reference but this reference was sufficient to cause it to be included in A_{11}. Thus this procedure could have performed the same reference service that A. R. Mackintosh did.

9.4 Comparison to Categories

In the last section we compared clusters to the bibliographies compiled by the authors of three articles. Another source of sets of articles that have been judged to be related would be the subject index found in one of the journals or in Physics Abstracts. For this purpose one category was selected from the subject index of Physical Review and one category was selected from Physics Abstracts.

9.41 Physical Review Category

Most of the categories in the Physical Review Subject Index are very broad. The sets formed by clusters, on the other hand, are in general much smaller and much more specific. Of course, larger clusters could be formed by including a large number of articles in the Y set of the request, but they would require a large amount of effort to process and compare. For this reason a category with relatively few entries was

selected. Its title changed periodically over the three year period, but it was identified as the one which was referred to when one looked up the word, "luminescence" in the word list which was supplied with the subject index. The various titles used for the category are as follows:

1963		Luminescence	(18 articles)
1964	46.4	Luminescence and Fluorescence	(6 articles)
1965	42.3	Optical Emission and Absorption	(17 articles)
1966	44.3	Optical Emission and Absorption	(2 articles)

The same format used for presenting the data in Sec. 9.3 is used here in Fig. 9.24-26.

It will be seen from Fig. 9.26 that most of the papers separate into the three major areas represented by A_{25}, A_9, and A_{26}. A statistical analysis of the composition of each of these three clusters is given in Fig. 9.27. It is found that the only words that appear more than once in the titles of two or more of the clusters are optical, absorption, radiation, and crystals. The correspondence of these words to the title of the original category (optical absorption and emission) is of interest.

A similar analysis of the author lists showed that N. Bloembergen was the only author that appeared more than once in two or more of the lists. The citation lists were also found to have very little overlap. The greatest overlap occurred between A_9 and A_{26}. For example, the 1st, 3rd, 5th, 7th entries in the list for A_9 were found in the list for A_{26} with a count of 2.

It is thus concluded that the articles in the clusters A_{25}, A_9, and A_{26} do have different characteristics. Whether the distinction

B

1-129-169
1-129-593
1-129-2422
1-130-502
1-130-639
1-130-945
1-130-2257
1-131-127
1-131-501
1-131-508
1-131-1114
1-131-1456
1-131-1543
1-131-2036
1-132-224
1-132-1023
1-132-1482
1-132-2501
1-133-1163
1-136-141
1-136-271
1-136-508
1-136-541
1-136-1091
1-137-508
1-137-536
1-137-1117
1-137-1651
1-137-1787
1-138-63
1-138-180
1-138-806
1-138-1741
1-139-321
1-139-544
1-139-1239
1-139-1616
1-140-155
1-140-263
1-140-601
1-140-1867
1-143-372
1-143-574

D

1-134-1166
1-137-801
1-138-1
1-138-960
3-82-393
3-85-565
3-86-709
41-12-504
41-13-334
41-13-657
41-13-720
49-10-52
49-11-294
646-6-25

E

1-139-10
1-140-1051
1-141-287
1-141-306
41-14-68
199-138-753
199-139-202

F

1-129-125
1-132-2023
1-137-1515
1-138-1472
1-138-1477
1-139-1262
1-139-1991
1-140-352
41-14-64
49-19-89

G

1-139-588
1-140-576

H

1-129-1980
1-132-2450

J

1-131-1912
1-132-1029
1-135-950
1-135-1622
1-137-1087
1-138-1287
1-139-314
11-34-1682
11-35-1183
12-38-1544
12-38-1607
12-38-2289
12-39-3118
12-42-1999
49-18-219
49-19-98
80-18-1448
80-19-1096

K

1-133-1029
1-136-481
12-42-3404

R

1-133-163
1-133-1717
1-134-299
1-134-423
1-135-1676
1-137-583
1-137-1016
1-138-276
1-139-1687
1-139-1965
1-140-880
80-19-2260
669-21-204

M

80-19-924

N

1-140-957
49-5-186
612-4-264

P

1-139-970

Fig. 9.24. The sets of articles included in the clusters for Category 1.

Answers to Requests:

$A[Y(b_i)]=A_1$ $i=29,42$

$A[Y(b_i)]=A_2$ $i=26,43$

$A[Y(b_{34})]=A_3$

$A[Y(b_i)]=A_4$ $i=33,37,38$

$A[Y(b_{28})]=A_5$

$A[Y(b_{30})]=A_6$

$A[Y(b_i)]=A_7$ $i=8,19$

$A[Y(b_{16})]=A_8$

$A[Y(b_{14})]=A_9$

$A[Y(b_{39})]=A_{10}$

$A[Y(b_2)]=A_{11}$

$A[Y(b_{17})]=A_{12}$

$A[Y(b_i)]=A_{13}$ $i=5,12,27$

$A[Y(b_{21})]=A_{14}$

$A[Y(b_{31})]=A_{15}$

$A[Y(b_{40})]=A_{16}$

$A[Y(b_1)]=A_{17}$

$A[Y(b_i)]=A_{18}$ $i=7,22,24$

$A[Y(b_i)]=A_{19}$ $i=10,11$

$A[Y(b_i)]=A_{20}$ $i=13,18,20$

$A[Y(b_{25})]=A_{21}$

$A[Y(b_{35})]=A_{22}$

$A[Y(b_i)]=A_{23}$ $i=4,6$

$A[Y(b_{15})]=A_{24}$

$A[Y(b_i)]=(b_i)$ $i=3,9,41$

$A[Y(b_i)]=$(large clusters) $i=23,32,36$

$A[Y(b_1b_2b_{12})]=$(107 articles)

$A[Y(b_{28}b_{34})]=A_3 \cup A_5=A_{25}$

$A[Y(b_{28}b_{30}b_{34})]=$(104 articles)

$A[Y(b_{35}b_{42})]=$(large)

$A[Y(b_8b_{17})]=$(large)

$A[Y(b_2b_{39})]=$(large)

$A[Y(b_{29}b_{40})]=$(large)

$A[Y(b_{27}b_{31}b_{40})]=(A_{15} \cup A_{17} \cup b_6) \cap (\overline{r_2r_4r_8b_1b_7})$

$A[Y(b_{18}b_{24}b_{27})]=A_{15} \cup A_{17} \cup A_{18} \cup A_{20} \cup (b_6g_1p_1f_6)=A_{26}$

Definitions of Clusters:

$A_1=(b_{29}b_{33}b_{42}) \cup D$

$A_2=A_1 \cup (b_{26}b_{43})$

$A_3=A_2 \cup (b_{34}b_{38})$

$A_4=(b_{29}b_{33}b_{37}b_{38}) \cup E$

$A_5=A_4 \cup (b_{28})$

$A_6=(b_{30}d_1)$

$A_7=(b_8b_{19}) \cup F \cup G$

$A_8=A_7 \cup (b_{16}h_2)$

$A_9=A_8 \cup (b_{14}h_1)$

$A_{10}=(b_{39}g_2)$

$A_{11}=(b_2g_1g_2)$

$A_{12}=(b_{17}f_1g_1)$

$A_{13}=(b_5b_{12}b_{27}) \cup (r_3r_9r_{10}r_{12}) \cup J$

$A_{14}=A_{13} \cup (b_{21}) \cup K$

$A_{15}=A_{14} \cup (b_{31}r_5)$

$A_{16}=(b_1b_7b_{27}b_{40}) \cup (r_1 \cdots r_8r_9r_{11})$

$A_{17}=(b_1b_7b_{27}b_{40}) \cup R$

$A_{18}=(b_7b_{22}b_{24}m_1r_2r_8)$

$A_{19}=(b_{10}b_{11}m_1)$

$A_{20}=(b_{13}b_{18}b_{20}j_{18}m_1) \cup N$

$A_{21}=(b_{25}k_2)$

$A_{22}=(b_{25}b_{35}p_1)$

$A_{23}=(b_4b_6)$

$A_{24}=(b_{15}r_7r_{11})$

$A_{25}=A_3 \cup A_5$

$A_{26}=A_{15} \cup A_{17} \cup A_{18} \cup A_{20}$ $(b_6g_1p_1f_6)$

Fig. 9.25. Answers to selected requests for Category 1.

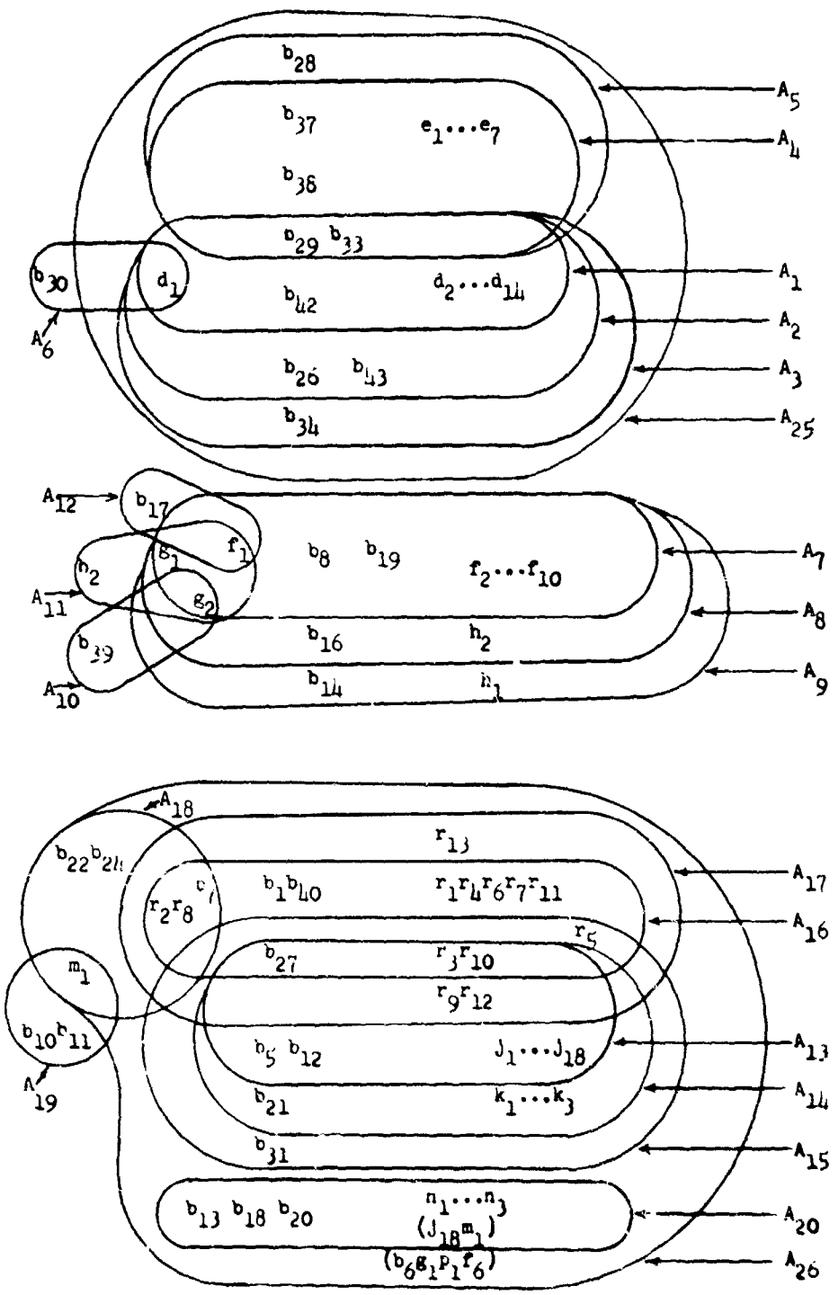


Fig. 9.26. Relationship of answer clusters for Category 1.

CLUSTER A_{25} (30 articles)		CLUSTER A_9 (18 articles)		CLUSTER A_{26} (55 articles)	
109 words:		84 words:		214 words:	
13	raman	7	SiC	12	ruby
9	stimulated	6	Exciton	11	optical
6	laser	5	Complexes	9	lines
6	radiation	4	Absorption	8	KCL
6	scattering	4	Luminescence	8	spectra
5	theory	3	CdS	7	crystals
4	fluctuations	3	Effects	6	absorption
4	intensity	3	Emission	6	thermoluminescence
3	effects	3	Nitrogen	5	excited
3	emission	3	Optical	5	F
3	liquids	3	Radiation	5	MgO
3	media	3	Recombination	4	center
3	optical	2	Cadmium	4	Cr_3^+
3	order	⋮		4	irradiated
3	waves			4	R
2	anti			4	relaxation
⋮				3	alkali
				⋮	
37 authors:		25 authors:		85 authors:	
5	Shen Y. R.	6	Choycke W. J.	6	Sturge M. D.
4	Bloembergen N.	6	Hamilton D. R.	5	McCumber D. E.
2	Armstrong J. A.	2	Patrick Lyle	3	Bloembergen N.
2	London R.	2	Dean P. J.	3	Schawlow A. L.
2	Smith Archibald W.	2	Reynolds D. C.	3	Yen W. M.
2	Tang C. L.	1	Anders W. A.	2	Arten J. O.
1	Anderson H. G.	⋮		⋮	
⋮					
292 citations:		248 citations:		846 citations:	
12	1-127-1918	13	41-4-361	22	80-13-880
10	1-130-2529	11	1-128-2135	15	1-122-381
10	1-131-2766	11	41-1-450	15	12-36-2757
10	1-133-37	10	1-127-1868	14	11-34-1682
10	41-9-455	8	1-131-127	13	1-122-1469
10	41-11-160	7	1-116-473	10	1-130-639
10	49-7-186	6	1-133-1163	10	12-20-1752
9	646-3-181	5	1-120-1664	9	80-13-899
8	41-11-419	5	1-127-1878	8	1-57-428
8	41-12-504	5	1-132-2023	8	30-31-956
7	1-134-1429	4	(5 citations)	7	(5 citations)
7	646-3-137	3	(7 citations)	6	(12 citations)
6	41-12-290	2	(42 citations)	5	(8 citations)
5	(5 citations)	1	(184 citations)	4	(18 citations)
4	(11 citations)	⋮		3	(33 citations)
3	(17 citations)			2	(121 citations)
2	(34 citations)			1	(741 citations)
1	(212 citations)				

Fig. 9.27. Comparison of the three clusters formed for Category 1.

between the clusters is of practical significance to a user would, of course, require further experimental justification.

As an additional comparison the results of this section were compared with the articles found in the category in Physics Abstracts with the title, "luminescence." This category contained 22 of the articles listed in Fig. 9.24. (14 in set B and 8 others.) All of these 22 articles were included in A_9 or A_{26}. This would tend to indicate that the Physics Abstracts indexers considered the articles of A_{25} to be in a different area than A_9 and A_{26} also.

9.42 Physics Abstracts Category

Since a property (luminescence) was chosen for the last section, it was decided that a category covering a substance might be appropriate for this test. We again sought a category with relatively few entries so that it would be easier to compare it with the related clusters. The category with the heading, "Erbium", was selected. The articles classified in this category from January 1963 to the present are listed in set B of Fig. 9.28. Fig.'s 9.29 and 9.30 present the related clusters.

9.5 User Experience

In the last two sections we compared the results of the clustering procedure to the three bibliographies and two categories. In this section we will present the response of the system to some actual requests for information. The response to both a relatively simple request and to a more comples request are studied.

B

1-131-1043
1-131-1586
1-132-1609
1-137-138
1-137-1109
11-35-1047
11-36-1001
11-36-1127
11-36-1249
12-38-2190
12-39-1285
12-39-1629
12-39-2128
12-40-2751
12-40-3606
12-41-1225
12-41-3363
12-42-873
12-43-847
29-29-477
49-8-5
49-11-100
49-13-112
49-15-301
49-16-265
49-17-95
80-20-808
80-20-1332
199-137-790
310-6-2225

D

1-129-2072
1-130-1337
1-130-1825
1-131-932
1-131-1039
1-138-216
1-139-1606
1-140-1896
3-81-846
3-84-63
3-84-693
11-36-906
11-36-1078
11-36-3628
12-39-1449
29-31-1
49-6-19

F

1-131-158
1-134-1620
1-137-1139
1-138-241
3-85-955
11-36-1209
49-17-96

F

1-132-542
1-133-219
1-134-94
11-35-800
12-43-2087

G

1-129-1601
1-130-1100
1-133-1571
1-134-320
1-134-1492
1-136-175
1-136-231
1-136-271
1-136-711
1-136-717
1-136-726
1-137-627
1-137-1449
1-140-1968
1-141-352
1-141-461
3-81-663
12-39-1422
12-39-1455
12-39-3503
12-42-377
12-42-981
12-42-1423
21-29-948
21-31-845
21-31-1325
49-10-16
49-10-496
310-7-1150

H

1-139-241
3-82-874
12-38-2750
12-42-4000
12-43-1680
80-18-1636

J

1-130-2325
1-132-280
1-133-881
1-136-1433
1-140-2005
1-142-115
12-41-565
12-41-617
41-11-196

K

1-141-4
43-36-505
1-137-1886
1-139-2008
3-84-297
12-38-976
12-38-2171
12-39-3251
12-40-796
12-40-3428
12-42-162
12-42-993
12-42-3797
12-43-2124
41-11-253

M

1-130-945
1-130-1370
1-133-34
1-133-494
1-134-172
1-134-1504
1-137-1749
1-138-1682
1-141-259

M (Con't.)

12-39-1024
12-39-1154
12-40-743
12-41-892
12-42-743
164-39-342
310-7-1450

N

1-138-1544
12-38-1476
12-38-2190
12-39-2134
12-41-1305
12-41-3227
12-43-1702

P

1-133-1364
49-19-463

Q

12-41-1970

R

11-36-2422
80-20-997

S

1-133-1364

T

21-29-974
49-20-496

U

669-17-1118
669-18-1022

V

1-135-97

W

1-140-1188
1-141-251

X

11-36-984
12-41-892

Fig. 9.28. The sets of articles included in the clusters for Category 2.

Answers to Requests:

$A[Y(b_i)]=A_1$ $i=1,6,11,20$

$A[Y(b_{27})]=A_2$

$A[Y(b_7)]=A_3$

$A[Y(b_{17})]=A_4$

$A[Y(b_{14})]=A_5$

$A[Y(b_{30})]=A_6$

$A[Y(b_{19})]=A_7$

$A[Y(b_{15})]=A_8$

$A[Y(b_{18})]=A_9$

$A[Y(b_{16})]=A_{10}$

$A[Y(b_{23})]=A_{11}$

$A[Y(b_i)]=A_{12}$ $i=22,24$

$A[Y(b_9)]=A_{13}$

$A[Y(b_9)]=A_{14}$

$A[Y(b_{12})]=A_{15}$

$A[Y(b_{28})]=A_{16}$

$A[Y(b_{29})]=A_{17}$

$A[Y(b_{26})]=A_{18}$

$A[Y(b_{25})]=(b_{25})$

$A[Y(b_{10})]=A_{19}$

$A[Y(b_{13})]=A_{20}$

$A[Y(b_5)]=A_{21}$

$A[Y(b_2)]=A_{22}$

$A[Y(b_i)]=A_{23}$ $i=3,21$

$A[Y(b_4)]=A_{24}$

Definitions of Clusters:

$A_1=(b_1b_6b_{11}b_{20})\cup D$

$A_2=A_1\cup(b_{27})\cup E$

$A_3=A_2\cup(b_7)\cup F$

$A_4=(b_3b_4b_{17})\cup G\cup(d_4e_4)$

$A_5=A_4\cup(b_{14})\cup H$

$A_6=A_5\cup(b_2b_{20}b_{30}d_5d_7f_3)\cup J$

$A_7=A_6\cup(b_{13}b_{19}f_4k_1k_2)$

$A_8=A_7\cup(b_{15}k_3\cdots k_{15})$

$A_9=A_8\cup(b_{18})\cup M$

$A_{10}=A_9\cup(b_{16})\cup N$

$A_{11}=A_{10}\cup(b_{21}b_{23}f_5)\cup P$

$A_{12}=(b_{22}b_{24}d_4f_4)$

$A_{13}=(b_8)\cup Q$

$A_{14}=(b_9)\cup R$

$A_{15}=(b_{12}n_2)\cup S$

$A_{16}=(b_{28}g_{26}m_{15})\cup T$

$A_{17}=(b_{29})$

$A_{18}=(b_{26})\cup V$

$A_{19}=(b_{10}b_{14}b_{17}g_{10}g_{19}g_{23}g_{26}h_2j_2j_4j_7$
$k_4k_7k_{10}k_{13}m_{11}n_1n_6n_7)$

$A_{20}=(b_{13}b_{17}b_{19}g_3g_4g_{14}g_{17}g_{18}g_{19}g_{21}g_{22}g_{26}$
$h_2h_3h_4j_7k_3k_4k_5k_6k_{11}k_{14}m_{12}n_4)$

$A_{21}=(b_5b_{16}g_8j_6k_{14})\cup W$

$A_{22}=(b_2b_{17}b_{20}d_5d_7e_4f_3g_2\cdots g_6g_{12}\cdots g_{15}$
$g_{17}g_{18}g_{21}g_{23}g_{25}g_{27}h_2h_3h_4j_1\cdots j_6j_{11})$

$A_{23}=(b_3b_{14}b_{18}b_{21}b_{30}f_5g_5g_{15}g_{18}g_{27}g_{29}$
$h_1j_8j_9k_9m_2x_1x_2)$

$A_{24}=(A_{23}\cup b_4)\cap(\overline{b_{14}})$

Fig. 9.29. Answers to selected requests for Category 2.

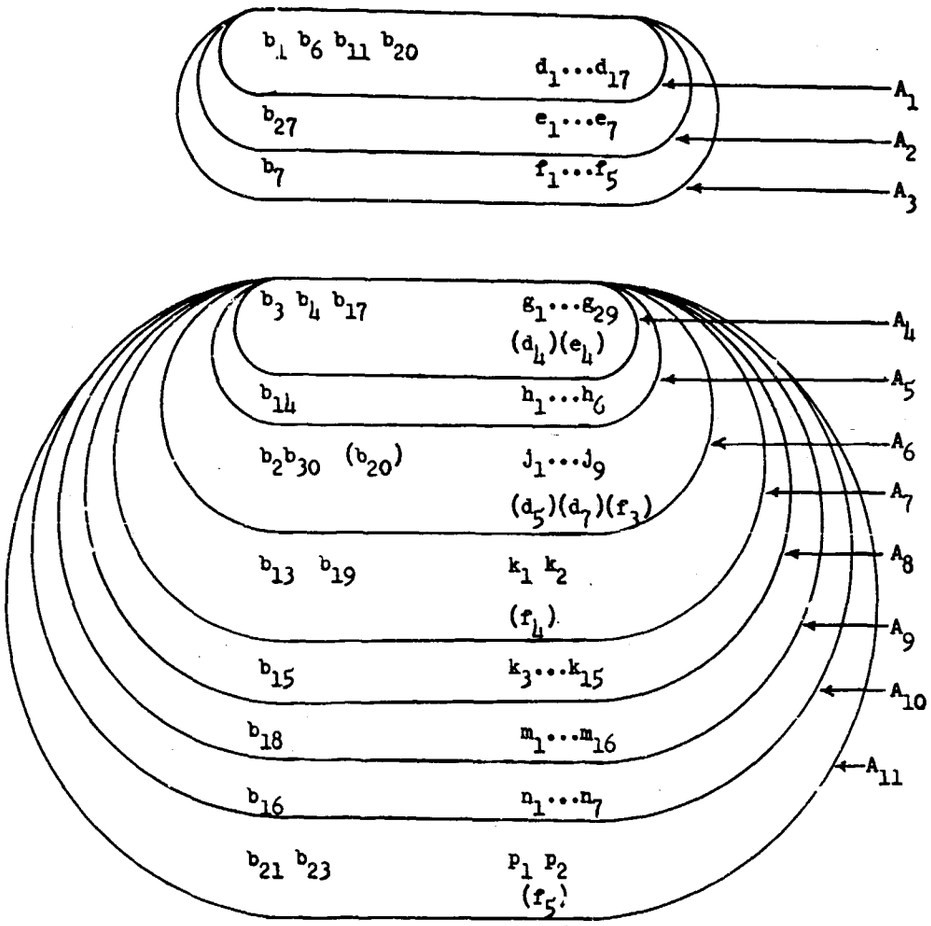


Fig. 9.30. Relationship of answer clusters for Category 2.

9.51 Simple Request

This test was performed in cooperation with a research physicist from Lincoln Laboratory. His initial request consisted of the following relatively brief specification:

words: turbulence
subsonic
hypersonic } perhaps
wake

authors: Lees
Hromas

articles: none

No articles were found which were written by the two authors (actually there were three papers by a Lees but in a completely different area). There were 70 articles that had either "turbulence or "turbulent in their titles (set T of Fig. 9.31). There were 27 which contained one or more of the words "wake, "subsonic", or "hypersonic". (Set W of Fig. 9.31.)

At this point a number of the articles in Set T were used as requests to the clustering procedure. The cluster structure shown in Fig. 9.32 and 9.33 resulted. The physicist was asked to evaluate the pertinence of each of the articles presented. He gave three types of responses: pertinent (y), non-pertinent (n), and questionable pertinence (m). The responses are indicated in Fig. 9.31 and also in Fig. 9.32 by the superscripts. It will be noted that nine of the twelve articles specified as pertinent are in the A_3 cluster.

The physicist was asked if there was any detectable difference between the article in the A_3 and A_7 clusters which were disjoint by the procedure. Of the 16 articles in A_7, 15 were from Russian journals, while 27 of the 35 articles in A_3 were from American journals. It was

T		T (Con't.)		W	D	
11-36-2075	y	799-6-1016	m	1-134-581	11-36-3609	y
11-36-2201	n	799-6-1048	m	1-135-1761	17-32-298	n
21-31-141	n	799-6-1250	n	1-138-934	669-18-698	n
29-30-17	y	799-6-1260	n	3-82-669	669-18-1014	n
41-14-813	n	799-6-1693	m	11-36-34	669-19-499	n
41-14-892	n	799-7-190	n	41-10-127	669-19-1165	n
41-15-381	n	799-7-335	m	41-13-437	669-20-135	n
49-9-144	n	799-7-562	m	41-12-592	790-10-605	n
49-12-201	y	799-7-629	m	41-13-742	799-6-1603	n
49-13-297	m	799-7-816	m	41-15-346		
49-18-224	n	799-7-1030	m	49-19-459		
80-19-1430	n	799-7-1048	m	80-18-288		
384-32-292	n	799-7-1156	y	80-18-1515		
646-7-285	y	799-7-1160	m	646-4-28		
669-16-295	n	799-7-1163	m	646-7-187		
669-16-1578	n	799-7-1169	m	799-6-946		
669-17-403	m	799-7-1178	m	799-6-1388		
669-17-1449	n	799-7-1191	n	799-7-197		
669-18-847	n	799-7-1403	n	799-7-667		
669-18-1251	n	799-7-1723	y	799-7-1147		
669-18-1268	m	799-7-1735	y	799-7-1198		
669-19-349	m	799-7-1920	n	799-8-44		
669-20-445	n	799-8-391	n	799-8-211		
669-20-1519	n	799-8-492	n	799-8-956		
669-21-744	y	799-8-575	m	799-8-1428		
669-21-774	m	799-8-598	y	799-8-1456		
669-21-1161	n	799-8-1063	m	799-8-1792		
790-6-882	n	799-8-1509	n			
790-6-1017	m	799-8-1647	n			
790-7-344	n	799-8-1659	n			
790-8-54	n	799-8-1775	m			
790-9-1057	n	799-8-1792	y			
790-9-1429	n	799-8-2219	y			
790-10-191	n	799-8-2225	n			
790-10-1041	n	821-2-332	n			

Fig. 9.31. Sets of articles included in the clusters for Physicist 1. (y=pertinent, n=non-pertinent, m=questionable pertinence)

$A[Y(t_i)]=(t_{46}t_{47}t_{49}t_{50}t_{55}t_{60}t_{62}t_{64}t_{65}t_{68}d_9)$
$=A_1 \quad i=46,47,49,50,55,60,62,64,65,68$

$A[Y(d_9)]=A_1$

$A[Y(t_{36})]=A_1 \cup (t_{36})$

$A[Y(t_{52})]=A_1 \cup (t_{36}t_{52})$

$A[Y(t_{48})]=A_1 \cup (t_{36}t_{48})$

$A[Y(t_{61})]=A_1 \cup (t_{36}t_{48}t_{52}t_{61})$

$A[Y(t_{51})]=A_1 \cup (t_{36}t_{48}t_{52}t_{61}t_{51})$
$=A_2$

$A[Y(t_i)]=(t_{19}t_{23}t_{24}t_{25}t_{26}t_{27}d_3d_4d_5d_6d_7d_8)$
$=A_6 \cup \; i=19,24,25,26,27$

$A[Y(d_i)]=A_6 \cup \; i=3,4,5$

$A[Y(t_{32})]=A_6 \cup (t_{32})$

$A[Y(t_{17})]=A_6 \cup (t_{32}t_{22}t_{17})$

$A[Y(t_8)]=A_6 \cup (t_{32}t_{22}t_{17}t_8)$

$A[Y(t_{16})]=A_6 \cup (t_{32}t_{22}t_{17}t_{16})$

$A[Y(t_i)]=(t_{37}t_{41}t_{55}t_{62}t_{66}) \quad i=37,66$

$A[Y(t_{13})]=(t_{12}t_{13}t_{38})$

$A[Y(t_{31})]=(t_{31}t_{34}t_{65})$

$A[Y(t_{33})]=(t_{33}t_{38}t_{65})$

$A[Y(t_i)]=(t_{38}t_{43}t_{58}) \quad i=38,43,58$

$A[Y(t_4)]=(t_4t_{68})$

$A[Y(t_{18})]=(t_{18}t_{62})$

$A[Y(t_{28})]=(t_{16}t_{28})$

$A[Y(t_{54})]=(t_{54}w_{17})$

$A[Y(d_2)]=(d_2t_{67})$

$A[Y(t_{70})]=(t_{25}t_{70})$

$A[Y(x)]=(d_1t_{67}) \quad x=d_1,t_{67}$

$A[Y(t_i)]=(t_2t_{69}) \quad i=2,69$

$A[Y(t_i)]=(t_3t_{12}) \quad i=3,12$

$A[Y(t_i)]=(t_5t_{20}) \quad i=5,20$

$A[Y(t_i)]=(t_9t_{23}) \quad i=9,23$

$A[Y(t_i)]=(t_{21}t_{22}) \quad i=21,22$

$A[Y(t_i)]=(t_{34}t_{39}) \quad i=34,39$

$A[Y(t_i)]=(t_{53}t_{57}) \quad i=53,56$

$A[Y(t_i)]=(t_{14}t_{56}) \quad i=14,56$

$A[Y(t_i)]=(t_i) \quad i=1,6,7,10,11,15,29,30,35,40,41,42,44,45,59,63$

Fig. 9.32. Answers to selected requests for Physicist 1.

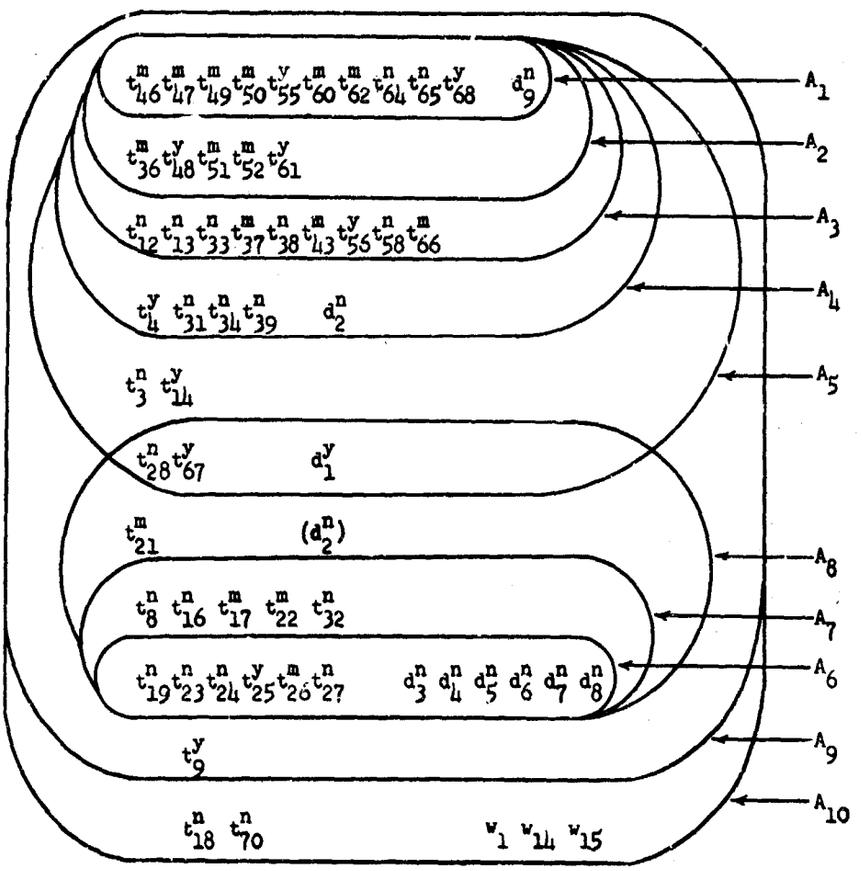


Fig. 9.33. Relationship of answer clusters for Physicist 1.
(y=pertinent, n=non-pertinent, m=questionable pertinence)

initially thought that the cause of the separation of the two clusters was probably due to the fact that the Russians generally cited Russians while the Americans cited Americans. After examining the two sets, the physicist expressed the opinion, however, that A_7 appeared to be more concerned with the upper atmosphere and ionosphere.

Also supporting the contention that there is a valid and useful distinction between A_3 and A_7 is the fact that nine of the eleven articles judged to be pertinent were from the A_3 cluster.

Because of the incompletely inverted files and the delays caused thereby, the actual searches were performed by the author of this thesis and later discussed with the physicist. It was interesting to note that at one point in the discussion, he stated that he could have more correctly shaped the final cluster by being able to specify as non-pertinent some articles on turbulence in helium that appeared in one of the clusters.

We note in passing that the physicist who aided in this test is the author of article t_{67}.

9.52 Expand Extensive Bibliography

In this section an example is given of how the clustering procedure might be used to supplement or extend an already sizable collection of papers on a given subject.

A bibliography of 112 articles on Langmuir probes was supplied to the author by another research physicist at Lincoln Laboratory. Of the 112 articles, 89 are to journals, 54 are to the 25 journals covered by the T.I.P. file, and 21 are actually in the T.I.P. file. The identifications of the 21 articles in the T.I.P. file are given in Fig. 9.14.

Fig. 9.35 shows the distribution of the articles in the file with time. Fig. 9.36 lists the words occurring in five or more of the 112 titles. In this list words such as "of, "the", "theory", etc., have been omitted. Also words have been grouped by stem. Thus, the words, "ion", "ions", "ionized", etc., are all grouped under the word, "ion".

Set B	B (Con't.)	B (Con't.)	B (Con't.)
3-82-243	11-36-1866	49-11-126	799-6-1492
11-34-1165	11-36-2363	80-18-260	799-4-1433
11-34-3209	21-30-182	80-18-1908	799-7-1843
11-35-1130	21-30-193	690-8-720	799-8-56
11-36-337	21-30-325	799-6-1479	799-8-73
11-36-675			

Fig. 9.34. 21 Articles in Langmuir Probe that are in T.I.P. file.

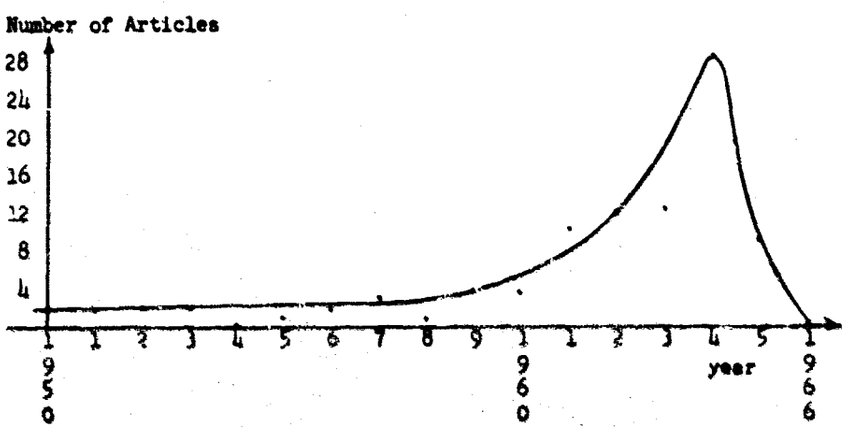


Fig. 9.35. Publication year distribution of initial Langmuir Probe bibliography.

Words	Number of articles
probe	87
plasma	40
Langmuir	35
ion	18
gas	15
discharge	13
electron	12
collection	10
density	8
low	7
pressure	6
spherical	6
electrostatic	6
probe and plasma	32
probe and Langmuir	35
probe and ion	16
probe and gas	7
probe and discharge	6

Fig. 9.36. Title word distribution for the 112 titles of the initial Langmuir probe bibliography.

As an additional part of this test it was decided that five other types of search strategies would also be used and their results would be compared to the results of clustering. The five search strategies selected will now be described.

TITLE WORD SEARCH

One possible search strategy would be to retrieve all those articles which have some word or logical combination of words in their titles. The choice of the word or words to be used was made on the basis of the frequency of occurrence of the words in the bibliography (Fig. 9.36) and in the T.I.P. file and with the advice of the physicist. Several test runs were made with various word combinations. A simple request for all articles with the word,"probe", in their titles was selected. This retrieved 58 articles including 20 members of the original bibliography.

AUTHOR SEARCH

There are 114 different authors of the 112 articles in the bibliography. A search of the T.I.P. file for articles by these 114 authors yielded 120 articles (21 from the original bibliography and 99 other papers). This search was not exhaustive but involved looking for authors only in those journals where it was thought they might publish.

CITATION SEARCH

The third type of search consisted of finding all of the articles that cite one or more of the 112 articles in the bibliography. A search of the T.I.P. file using this criteria yielded 78 articles.

BIBLIOGRAPHIC COUPLING SEARCH

When two papers cite one or more of the same papers they are said to be bibliographically coupled (Sec. 6.22). There are 270 articles that are bibliographically coupled to one or more of the 21 articles in set B of Fig. 9.34.

The coupling strength between two papers is defined to be the number of identical citations that they have. The coupling strength between one paper and a set of papers is defined to be the number of citations in the single paper which are also found in one or more of the papers in the set. In Fig. 9.37 we show the distribution of the 270 articles by their coupling strength to the set B.

JOINTLY CITED SEARCH

Bibliographic coupling occurs between two papers if they cite one or more of the same papers. Another type of coupling occurs if two papers are cited by one or more of the same papers. There are 605 papers which occur in one or more bibliographies with articles of set B. Of the 605, 101 are in the T.I.P. file.

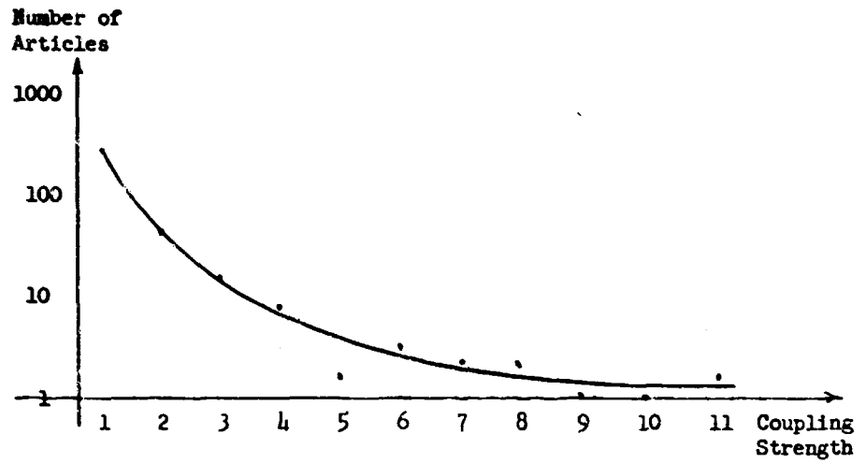


Fig. 9.37. Distribution of articles with various bibliographic coupling strengths.

CLUSTERING

The user specified the article b_{17} as the article of greatest interest in the bibliography. The articles b_6, b_8, b_{16}, and b_{19} were ranked next in terms of interest. The clusters which resulted when these and various other articles were used as requests to the system are shown in Fig.'s 9.38 - 9.40.

D	E (Con't.)	G	J
11-34-1897	41-11-310	3-83-473	11-35-1365
55-41-132	41-15-286	11-35-130	790-10-1102
80-19-1915	646-4-186	55-41-391	799-6-1762
612-2-719		55-41-1405	799-7-1834
799-7-2329	F	790-7-921	
799-8-748	3-81-682		K
	11-36-342	H	80-18-426
E	11-36-2361	799-7-110	80-18-1056
3-83-971	11-36-3526	799-8-920	80-20-845
11-36-3135	612-3-18	799-8-2097	612-2-58
11-36-3142	790-7-708		
11-37-180			M
			11-37-377

Fig. 9.38. The sets of articles included in the clusters for Langmuir Probe Bibliography (Physics 2).

Answers to Requests:

$A[Y(b_i)]=A_1 \quad i=14,16,17$

$A[Y(b_i)]=A_2 \quad i=1,7$

$A[Y(b_i)]=A_3 \quad i=8,9,11$

$A[Y(b_3)]=A_4$

$A[Y(b_i)]=A_5 \quad i=4,6,20,21$

$A[Y(b_{19})]=A_6$

$A[Y(b_5)]=A_7$

$A[Y(b_2)]=A_8$

$A[Y(b_{10})]=$(cluster of 82 articles)

$A[Y(b_{12})]=A_9$

$A[Y(b_{15})]=A_{10}$

$A[Y(b_i)]=(b_i) \quad i=13,18$

$A[Y(b_6b_8b_{16}b_{17}b_{19})]=A_{11}$

$A[Y(b_1b_3b_4b_6b_7b_8b_9b_{11}b_{14}b_{16}b_{17}b_{19}b_{20}b_{21})]=A_{12}$

$A[Y(d_i)]=A_1 \quad i=1,\ldots,6$

$A[Y(e_i)]=A_2 \quad i=1,3,\ldots,6$

$A[Y(e_2)]=A_8$

Definitions of Clusters:

$A_1=(b_8b_{14}b_{16}b_{17})\cup D$

$A_2=(b_1b_7b_8b_{14})\cup E$

$A_3=(b_3b_8b_9b_{11}b_{19})\cup(d_1d_4d_5)\cup F$

$A_4=(b_3b_8b_9)\cup(f_1f_2f_4)\cup G$

$A_5=(b_4b_6b_8b_{16}b_{20}b_{21})\cup(d_2g_1g_4)$

$A_6=(b_{16}b_{17}b_{19}b_{20}b_{21})\cup H$

$A_7=(b_5f_5)\cup K$

$A_8=(b_2b_{19}d_5e_1e_2e_4g_2)\cup J$

$A_9=(b_{12}b_{14}e_1e_2m_1)$

$A_{10}=(b_{15}f_5j_2)$

$A_{11}=A_5\cup(b_{17}b_{19}f_1)$

$A_{12}=A_1\cup A_2\cup A_3\cup A_4\cup A_5\cup(b_{19}j_1j_2)$

$A_{13}=A_{12}\cup(b_2j_3j_4)$

Fig. 9.39. Answers to selected requests for Langmuir Probe Bibliography (Physicist 2).

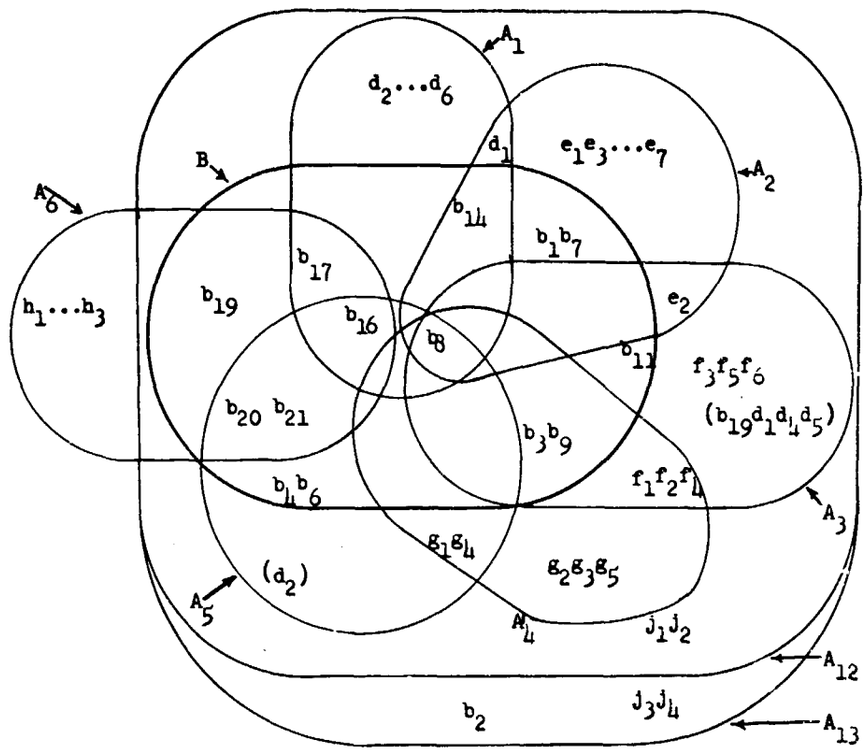


Fig. 9.40. Relationship of Clusters for Langmuir Probe Bibliography (Physicist 2).

COMPARISON

The six preceding search strategies produced a total of about 500 different articles. It was decided that this constituted too large a file to ask the user to evaluate. The file was, therefore, reduced to the 104 articles which appeared to have the greatest chance of being of interest to the user. These included the 83 articles which were retrieved by two or more of the six search strategies, the 15 additional articles which were bibliographically coupled to the set B with a value of three or more and another six articles which contained the word, "probe", in

their titles in the sense of a measuring device. In seven other articles the word, "probe", was found in the title but it was used as a synonym for investigation (e.g. "three-field model as a probe of higher group symmetries").

The 104 articles presented for evaluation are listed in Fig. 9.41. The first column (A) is the identification. The next column (B) contains an indication (1) of those articles which are members of set B. The next six columns (C-H) note which articles were retrieved by each of the six search strategies:

C - Column contains a one if the paper has the word, "probe", in its title.

D - Number of authors of the paper that are also authors of 112 papers in the Bibliography.

E - Number of the 112 papers in the Bibliography that are cited by the paper.

F - Bibliographic coupling strength of the paper to the set B.

G - Number of papers which cite the paper and also cite one or more of the 112 papers in the Bibliography.

H - Symbol of the paper in the clusters of Fig. 9.38 to 9.40.

(Note that the counts in Columns D and F do not include the authors or citations which match only because the article itself is in the set B.)

The last column (J) contains the evaluation code. Each document was assigned to one of the following five categories:

1 - Of personal interest to user.

2 - Of general interest.

3 - Perhaps of general interest.
(e.g. a probe may have been used as a tool in the experiment.)

A	B	C	D	E	F	G	H	J
1-129-1181	-	-	-	-	7	2	-	3
1-132-1435	-	-	1	-	3	1	-	3
1-132-1445	-	-	1	-	4	2	-	3
1-132-2363	-	-	1	1	-	-	-	3
1-132-2554	-	-	-	-	3	-	-	3
1-134-1215	-	-	-	-	6	-	-	3
1-137-346	-	-	-	-	4	-	-	3
1-138-1015	-	-	1	-	1	-	-	3
1-140-748	-	-	-	-	3	-	-	3
1-140-778	-	-	-	-	4	-	-	3
1-141-146	-	-	-	-	4	-	-	5
3-81-682	-	-	-	-	6	-	f_1	3
3-82-243	1	1	-	3	2	3	b_1	1
3-83-473	-	-	-	1	2	1	g_1	5
3-83-971	-	-	-	-	2	1	e_1	2
3-84-133	-	-	-	-	4	-	-	3
11-34-1665	1	1	-	-	1	1	b_2	1
11-34-1897	-	1	1	4	5	4	d_2	1
11-34-2613	-	1	1	-	-	-	-	2
11-34-3209	1	1	-	1	3	1	b_3	1
11-35-130	-	1	-	1	8	1	g_2	1
11-35-1130	1	1	-	1	1	1	b_4	1
11-35-1365	-	-	1	-	1	-	j_1	3
11-36-337	1	-	-	-	-	-	b_5	3
11-36-342	-	1	-	-	6	-	f_2	5
11-36-435	-	1	1	-	1	-	-	2
11-36-675	1	1	1	-	2	-	b_6	1
11-36-1659	-	1	-	-	-	-	-	5
11-36-1866	1	1	2	-	2	-	b_7	1
11-36-2361	-	1	2	-	2	-	f_3	1
11-36-2363	1	1	-	-	8	-	b_8	1
11-36-2672	-	1	-	-	1	-	-	1
11-36-3135	-	-	1	-	9	-	e_3	2
11-36-3142	-	1	1	1	4	1	e_3	2
11-36-3526	-	-	-	-	7	-	f_4	3
11-36-3740	-	-	1	-	1	-	-	3
11-37-180	-	1	1	-	2	-	e_4	1
11-37-215	-	1	1	4	4	-	-	2
11-37-377	-	-	2	1	2	-	m_1	3
11-37-[illegible]	-	-	1	-	2	-	-	3
17-27-674	-	-	1	-	1	-	-	4
21-29-93	-	-	1	-	1	-	-	3
21-29-1165	-	1	1	-	-	-	-	1
21-29-1313	-	1	1	-	-	-	-	1
21-30-182	1	1	3	2	11	3	b_9	1
21-30-193	1	1	3	1	1	2	b_{10}	1
21-30-375	1	1	3	4	10	1	b_{11}	1
21-30-2021	-	-	1	-	3	-	-	3
21-31-1632	-	-	1	-	1	-	-	4
41-11-310	-	-	1	-	1	2	e_5	2
41-13-83	-	1	-	-	-	-	-	5
41-15-286	-	-	-	-	2	-	e_6	3
41-15-1018	-	-	1	-	1	-	-	3
49-4-135	-	-	1	-	1	-	-	5
49-5-244	-	-	1	-	-	2	-	5
49-11-126	1	1	2	-	3	1	b_{12}	2
49-19-118	-	1	-	1	-	-	-	2
49-20-7	-	1	-	-	-	-	-	5
49-20-269	-	1	-	-	1	-	-	1
55-41-132	-	-	-	-	3	-	d_2	1
55-41-391	-	1	-	-	1	-	g_3	3
55-41-1405	-	-	-	-	3	-	g_4	3
55-41-1980	-	1	-	-	-	-	-	2
80-18-260	1	1	3	1	-	-	b_{13}	1
80-18-426	-	-	1	-	2	1	k_1	5
80-18-558	-	-	-	-	3	-	-	5
80-18-1056	-	-	1	-	2	-	k_2	5
80-19-566	-	-	1	-	1	-	-	5
80-19-1908	1	1	2	2	3	-	b_{14}	1
80-19-1915	-	1	1	2	2	-	d_3	2
80-19-2313	-	-	1	1	1	-	-	3
80-20-845	-	-	1	-	1	-	k_3	5
164-37-241	-	1	-	-	-	-	-	5
612-2-58	-	-	1	-	1	-	k_4	5
612-2-719	-	1	-	1	6	1	d_4	2
612-3-18	-	-	-	1	8	1	f_5	5
612-3-24	-	1	-	-	-	-	-	4
612-3-789	-	1	2	-	-	-	-	2
646-4-186	-	1	1	1	3	1	e_7	2
646-7-324	-	1	1	-	1	-	-	1
669-16-887	-	-	-	-	3	-	-	3
790-6-947	-	-	-	1	1	1	-	3
790-6-990	-	1	-	-	-	-	-	4
790-7-580	-	-	-	1	2	1	-	5
790-7-788	-	1	-	-	3	-	f_6	1
790-7-921	-	-	-	1	1	1	g_5	5
790-8-319	-	1	-	-	-	1	-	5
790-8-720	1	1	-	-	-	1	b_{15}	1
790-9-961	-	-	-	1	1	-	-	3
790-10-1102	-	-	-	-	3	-	j_2	3
799-6-1479	1	1	2	4	4	6	b_{16}	1
799-6-1492	1	1	1	3	3	7	b_{17}	1
799-6-1762	-	-	-	-	2	2	j_3	2
799-7-110	-	-	-	4	2	1	h_1	2
799-7-1329	-	-	-	2	7	-	d_5	5
799-7-1433	1	1	-	-	-	-	b_{18}	1
799-7-1517	-	-	1	1	1	-	-	3
799-7-1834	-	-	1	-	1	-	j_4	5
799-7-1843	1	1	-	4	11	-	b_{19}	1
799-8-56	1	1	1	4	5	1	b_{20}	1
799-8-73	1	1	1	3	4	-	b_{21}	1
799-8-748	-	1	1	1	1	-	d_6	1
799-8-920	-	-	1	1	1	1	h_2	3
799-8-2097	-	-	1	2	2	-	h_3	3

Fig. 9.41. Langmuir Probe papers evaluated by physicist. (Explanations of columns are given in text.)

4 - Degree of interest cannot be determined by examination of the author(s).

5 - Not of interest.

In Fig. 9.42 the results of each of the six search strategies are tabulated for comparison. The results for bibliographic coupling are separated into two entries depending on the coupling strength.

An examination of Fig. 9.42 indicates that the search strategies using the author, citation, and cited-by-same criteria yield comparatively large sets of documents containing relatively few of the articles judged to be of specific pertinence by the user (evaluation category 1).

Bibliographic coupling with the coupling strength greater than or equal to one yields such a large set of articles (270) that it would be more appropriate to compare it with a larger cluster such as the 85-article cluster which contained 26 of the category-1 documents. Let us therefore compare cluster A_{13} with the set of articles with coupling strength greater than or equal to two. It will be seen that A_{13} is less than half as large and yet contains three more of the category-1 documents.

It will be observed that the clustering procedure uses the same data used in bibliographic coupling but in a different way. Consider, for example, the 27 articles in A_{13} which are not part of the original bibliography. Seven have a coupling strength to B of only 1 and six have a coupling strength of 2. Whereas an articles like 1-129-1181 with a coupling strength of 7 is not included in A_{13}.

Search Strategy	Number of articles retrieved	Number of articles in each evaluation category				
		1	2	3	4	5
Title word	58	30	11	1	2	6
Author	120	18	10	15	2	8
Citation	78	16	7	8	0	5
Bibliographic coupling (strength _ 2	88	19	10	19	0	9
Bibliographic coupling (strength _ 1)	270	26	12	29	2	15
Cited-by-same articles	101	13	8	4	0	7
Clustering (A_{13})	43	22	8	7	0	6
Total	abt. 500	31	16	32	4	21

Fig. 9.42. Comparison of results of seven search strategies.

Let us now turn our attention to the title word search. Fig. 9.42 incidates that this search strategy retrieved four more of the category-1 documents than were retrieved by the search strategies based on citations (i.e. bibliographic coupling and the 85-document cluster). This result provides an example of a case where title words provide a better basis for retrieval than do citations. Previous experience would indicate that such is not generally the case.

To determine why the clustering procedure was less effective in this case the five category-1 documents which did not appear in any of the clusters generated were examined. It was found that three of them (b_{13}, b_{15}, and 21-29-1165) contain only a single citation and the other two (b_{18} and 21-29-1313) contain only two citations. We are thus led to the same conclusion arrived at earlier that the clustering system,

in general, has trouble properly placing documents with three or fewer citations.

The remedy for this difficulty would be to use some additional types of partitioning data. In the example at hand, all 31 of the category-1 documents could be retrieved in the same cluster if the system used not only the partitions generated by citations but also those generated by certain keywords like "probe".

One other observation may be worth noting. The article, b_5, was part of the original bibliography but was not included in any clusters with other members of the bibliography. A check of its bibliography showed that it had nine citations, which experience indicated should be enough to place it in the correct cluster. The author of this thesis decided, therefore, to ask the physicist if b_5 was in a different area from the other 20 members of the bibliography. Before this was asked, however, the evaluation of the 104 articles of Fig. 9.41 was made. A check of this evaluation revealed that 19 of the 21 members of the original bibliography were placed in evaluation category 1 while b_{12} was placed in category 3.

9.6 Summary of Results

For purposes of comparison and emphasis let us summarize some of the significant features of the last three sections. In Fig. 9.43 two measures of the success of the clustering procedure are tabulated. Column four indicates how many of the pertinent articles were retrieved by the clustering system in each test. Column five indicates what fraction of the articles retrieved were pertinent. The particular cluster selected for each test is specified in parenthesis in column three.

Name of Test	Number of papers specified as pertinent	Size of Related Cluster	Percent of pertinent papers in cluster	Percent of cluster specified as pertinent
Bibliography 1 (Sec. 9.31)	10	17(A_2)	9/10=90%	9/17=53%
Bibliography 2 (Sec. 9.32)	16	64(A_4)	14/16=88	14/64=22
Bibliography 3(III) (Sec. 9.33)	27	48(A_{12})	20/27=74	20/48=42
Bibliography 3(IV) (Sec. 9.33)	9	31(A_8)	8/9=89	8/31=26
Bibliography 3(IIIC) (Sec. 9.33)	13	22(A_5)	10/13=77	10/22=46
Category 1 (Sec. 9.41)	43	105 ($A_9 \cup A_{25} \cup A_{26}$)	28/43=65	28/105=27
Category 2 (Sec. 9.42)	30	133 ($A_1 \cup A_{11}$)	19/30=64	19/133=14
User 1 (Sec. 9.51)	12(y)	59(A_{10})	9/12=75	9/59=15
User 2 (Sec. 9.52)	31(1)	43(A_{13})	22/31=71	22/43=51

Fig. 9.43. Summary of the experimental results of Sections 9.3-5.

One additional statistic may be of interest. This relates to whether the documents that are pertinent to a search are added to the cluster early or late in the process. For this purpose 50 clusters from Sec. 9.33 and 9.41 were analyzed and the number of articles of specified pertinence added in each quarter of the process was noted. These figures were averaged for the 50 clusters. The results are shown in Fig. 9.44. It will be seen that on the average almost half (45 %) of the pertinent articles which are included in the final cluster are added during the first quarter of the process.

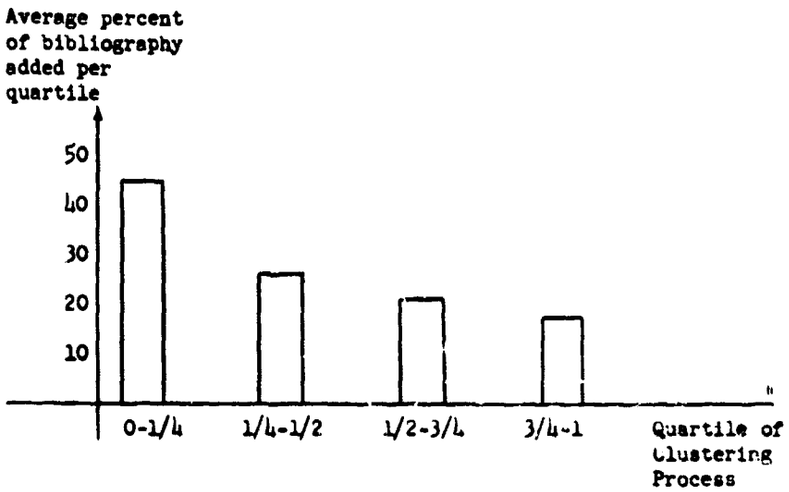


Fig. 9.44. Graph showing average percent of bibliography (or category) articles added during each quartile of the clustering process.

CHAPTER X

CONCLUSIONS

In this chapter we shall make some initial comments concerning the adequacy of the various components of the experimental system. Then certain conclusions about the clustering procedure will be given. Next the effectiveness of the overall model and system in retrieveing useful sets of documents will be evaluated. In the final section some possible avenues for further research will be suggested.

10.11 MAC Time-Sharing System

After five years' experience with batch processing computers, the author of this thesis found the MAC time-sharing system a refreshing change with some significant advantages. Let us briefly comment on the use of the MAC system in three areas: in debugging programs, in testing and evaluating systems, and in operational retrieval functions.

DEBUGGING

It is estimated that the use of the MAC system cut by a factor of somewhere between two and ten the amount of time required to debug the experimental program. This, of course, is due to the fact that turn-around time for a run with time-sharing is of the order of a few minutes, whereas with batch processing it is usually several hours or days.

The availability of more sophisticated debugging routines would have reduced debugging time even further. Some features that would

have been of special help are multiple break points, conditional break points, an interpretive mode, more convenient patching, automatic updating of the English text, etc.

One problem in using time-sharing for debugging is that it is almost too easy to make changes to a program and re-run it. This results in one making a change before its consequences have been fully considered. Part of the answer to this problem lies in self discipline on the part of the programmer. It will also help when a computer becomes available on a 24-hour basis so one is not tempted to try to rush through a change before a maintenance or test session.

Two minor improvements to the consoles would help. A less noisy console would allow the user to more effectively contemplate a problem at the same time the computer is printing out some results on the console. Also a neon light showing when the console is being serviced by the central processor would be of considerable value.

SYSTEM TESTING

After one has obtained a program that is debugged and performs according to specification, it often becomes apparent that the original specifications for the program need changing. This may result in some modifications to the program, or if the change is extensive, it may require rewriting the whole program. The same advantages and problems that time-sharing has in debugging are also in evidence in this cycle of program specification and respecification.

OPERATIONAL RETRIEVAL

Let us now consider what would happen if one were to decide to use the MAC system or one like it as an operational information retrieval system serving a community of real users.

If all of IBM 1302 disc were used for data, a file 30 times the size of the current T.I.P. file could be stored. This would allow one to increase the time span covered by the periodical literature from 3 to perhaps 10-15 years and also add some non-periodical literature. All of the files could also be completely inverted. There would probably still be room left for coverage of another discipline about the size of physics. If magnetic tapes were used, coverage could be increased even further by loading the disc with different data on different days of the week.

Let us assume that the current limit of 30 users on line at once is maintained. The response time for simple requests for information would probably be acceptable to most users. This would be 1 second of computer time and 1-30 seconds of real time. The response time to more complex requests would probably be found objectionable to some users. Retrieval of a cluster, for example, might take 40-50 seconds of computer time and 5-10 minutes of real time.

The response time to complex requests could be improved by a factor of 5-10 if the supervisory system were modified to allow some type of direct access to the disc. The current supervisory program is designed for the storage of files that are constantly changing. This places a penalty factor of 5-10 of the accessing of files that never change, such as those found in a library.

One of the biggest difficulties with using the MAC system as an information retrieval service is that it has no provision for the transmission, display and reproduction of analog information. Such a capability would probably be needed, for example, if the system were to supply the abstracts or total text of articles.

Thus, with the current system a person with a console in his office might be able to identify which articles are of interest, but he would still have to go to the library to get them. (He could perhaps have his own microfilm system, but this would be very expensive.)

10.12 T.I.P. Document Collection

The first tests of the clustering procedure were performed using a single volume of the Physical Review. As the data base was increased, some marked changes in the characteristics of the procedure were noted. One of the major causes of these changes was the fact that the partitioning sets for the single volume are all quite small, whereas the partitions for the total T.I.P. file have a wide range of sizes.

The question arises as to whether an increase of perhaps one or two orders of magnitude in the current document file might further change the way the procedure operates. In an attempt to answer this question, let us first note that such an increase would necessarily involve coverage of some additional branches of science such as chemistry, mathematics and/or electrical engineering. This would be true since a sizeable fraction of the significant physics periodical literature that is being published is already being added to the T.I.P. file. This implies that the size of the clusters generated by the procedure would not significantly change even if the size of the collection were greatly increased.

Also the use of an inverted data storage system would keep the access time to any one piece of information relatively constant even when the size of the file were measurably increased. It is, therefore, concluded that the system would operate in essentially the same way it

currently does even if the document file were scaled up in size by several orders of magnitude.

10.13 Partitions

The experimental results as summarized in Fig. 9.43 are evidence of the fact that partitions based on citation information constitute a useful data base for the measure of relatedness and the clustering procedure. There were, of course, a few documents which were not included in the cluster to which it appeared they should belong. In almost all of these cases it was found that the documents had three or fewer citations which was evidently an insufficient number to properly place them in their appropriate cluster.

From this, one might conclude that the clustering system as presently programmed may not be an effective retrieval tool for a file in which a large fraction of the documents have three or fewer citations. Actually what may be needed in such a file is a modification in the type or types of partitioning information utilized so that partitions are also generated by users, title words, authors or some other parameter(s). A case where other types of partitionings would have helped even in the citation-rich T.I.P. file was described in Sec. 9.52.

10.14 Storage Structure

One general conclusion that was reached in this project is that in a dynamic system an attempt should be made to give the data a general structure instead of a structure tailored to one specific requirement. This will allow a flexible approach to new uses of the data. An inverted file structure coupled with the raw data file was suggested as a

possible general filing system.

It is argued in Sec. 7.22 that an inverted file should occupy about the same amount of storage as is occupied by the file which is being inverted. This claim was verified for the data in the T.I.P. file.

10.15 Retrieval Language

The fact that both the syntax and vocabulary of the retrieval language is table-driven(i.e. they are specified by tables) was considered to be a significant advantage. As modifications in the structure of the request and in the words used to describe the request suggested themselves, they were easily incorporated into the system by a minor modification in the appropriate table.

Currently no one besides the author of this thesis has had sufficient experience with the retrieval language to evaluate it. Let me, therefore, make some admittedly biased observations.

First, the language was found to be easy to remember even after a lapse of several months in which it was not used. The language was also found to have considerable room for future growth. Indeed a large number of additional verbs and adjectives that would be useful in retrieval suggested themselves. The ability to make a request for information as complex or as simple as needed was also found helpful. Actually only a maximum of about three or four levels of structure has been utilized so far.

10.2 Evaluation of Procedure

In this section we shall discuss whether the procedure as described in Chapter V has the general characteristics which it needs for operation as a retrieval tool. An evaluation of the actual utility of the current procedure and experimental system in satisfying user requests will be discussed in the next section.

CONVERGENCE

Considerable difficulty was encountered with the earlier clustering procedures because they occasionally entered into a non-terminating cycle. The steps taken to prevent such cycles have been described in Sec. 5.53. The experience gained over the past several months supports the contention that the current procedure will always converge in a finite number of iterations to an answer cluster or to a comment that the request is inconsistent.

GENERAL-SPECIFIC

From Fig. 9.3 one can conclude that the use of a bias in the correlation network does, indeed, allow one to increase or decrease the size of the answer cluster. That the value to be given the bias can be automatically determined by the composition of the request has been experimentally verified by the results of Sec.'s 9.3-5.

AMBIGUITY RESOLUTION

In Chapter IX examples are given showing how some of the possible answer clusters that satisfy a given request can be eliminated by specifying additional documents to be of interest or not of interest (additions to the Y and Z sets). It is clear that one can arrive at a point at which only one cluster satisfies the request by the appropriate additions to the Y and Z sets. From Fig. 9.7 one might conclude that

on the average at least two members of Z are required to make a request unambiguous. Of course, even if the request is ambiguous, the desired answer cluster may still be found. For example, in Sec. 9.31 seven out of the ten requests with $Y=(b_1)$ resulted in A_1 and yet all seven are ambiguous.

INCONSISTENCY RECOGNITION

From the results of Fig. 9.5 we conclude that not only does the procedure mark as inconsistent those requests for which there is no answer cluster, but it also decides that some of the requests are inconsistent, for which a valid answer cluster exists. This difficulty is not considered serious, however, since the user can be coupled into the system and can guide the procedure in the right direction and reshape the request if an inconsistent situation is reached.

10.3 Evaluation of System

In the last section several conclusions were stated concerning the characteristics of the clustering procedure. In this section we will discuss the more general problem of the effectiveness of the overall system as a retrieval tool.

From Fig. 9.43 we note that the percent of pertinent documents retrieved by clustering ranges from 64 to 90 %. This compares favorably with a published retrieval efficiency of about 50% for other automatic retrieval systems.

Almost all of the pertinent documents which were not retrieved were found to have three or fewer citations. This would give one the hope that with an expanded data base for the partitions the 64-90 % retrieval efficiency could be improved even more.

We next note from Fig. 9.43 that from 47 to 86% of the retrieval documents are not part of the set of documents of known pertinence. Let us assume for a moment that all of these documents are irrelevant. Many users would still find this acceptable since a quick examination of the titles could be used to select the articles of interest from the larger set.

Now let us consider whether or not some of the additional articles might really be found to be of interest by a user who has selected the cluster in which they are found.

First, we observe that for the tests of Sec. 9.3 some of the articles in the clusters were published after the October IEEE Proceedings came out and thus had no chance of being part of the bibliographies even if they were pertinent. This is the case, for example, with the following documents of Fig. 9.21: d_6, e_9, k_{11}, k_{12}, k_{13}, k_{17}, m_{12},..., m_{18}, m_{27}, p_3, q_3, q_4, and q_5.

Also the authors of the three bibliographies used probably did not intend to exhaustively cover the area. They may have only selected what they considered to be the best reference(s) available for each specific concept or topic.

These arguments do not hold for the articles added by the clustering procedure to the categories of Sec. 9.4. The categories are supposedly exhaustive and should include all but the most recent articles. In defense of the additional articles in the clusters let us give two examples. The first title below is included in the Physical Review category on "Luminescence" while the second is not.

1-133-1163
Optical properties of cubic SiC, luminescence of nitrogen-exciton complexes, and interband absorption.

1-133-2023
Optical properties of 15R SiC, luminescence of nitrogen-exciton complexes, and interband absorption.

As a second example, consider cluster A_4 of Sec. 9.42. This cluster contains three articles that are classified in the category, "Erbium", in Physics Abstracts. Of the 31 other articles in the cluster three contain the word, "erbium", in their title and seven more contain the word, "erbium", in the abstract or text. All of the remaining articles have at least one of the other 14 rare earth elements mentioned in the title. The following is an example of an article contained in the cluster A_3 but not included in the erbium category.

1-126-726
Energy levels and crystal-field calculations of Er_3^+ in yttrium aluminum garnet.

For the tests with users described in Sec. 9.5 the percentage of the cluster that is pertinent would be 27/59=46% for User 1 and 27/43=86% for User 2 if all of the articles of questionable (or general) pertinence were counted. The user might even find some of those articles judged non-pertinent to be of interest if he were allowed to examine the actual article instead of just the title.

The foregoing arguments and data suggest that a user might, on the average, find at least half of the documents in a cluster of interest.

It is perhaps significant that the percentage of pertinent documents retrieved is lower in the tests for the two categories than for the other tests. The other tests involved bibliographies compiled by experts (authors and users) while the categories were generated by

indexers.

One might also note that the tests of Sec. 9.3 have higher percentages of pertinent documents retrieved on the whole than do the tests of Sec. 9.5. This could be explained by the fact that the users of Sec. 9.5 based their decisions on the titles, authors, and citations of the articles, while the authors of Sec. 9.3 had undoubtedly read the articles they cited. The conclusion to be reached here is that the clustering procedure tends to do best in those tests where it was compared to sets generated by the careful consideration of experts.

In conclusion, the experience of this thesis indicates that clustering may be a useful tool to research workers who desire information covering either a very specific or a very broad area of interest. It is our opinion that further development and research is both warranted and essential.

10.4 Suggestions for Further Research

The suggestions to be presented here have been divided into three general categories:

(1) Data base and data structure

(2) Clustering procedure and interaction language

(3) Theoretical problem

10.41 Data Base and Structure

OTHER DATA BASES

It has already been suggested (Sec. 10.13) that the clustering system should be tested on other types of partition data. Some of the

other types of partitions that might be tried are listed in Sec. 6.22.

It is also suggested that tests be made of the simultaneous use of several types of partitioning data. In this connection one might consider the use of a weighting factor for the partitions which might, for example, give a larger weight to partitions generated by citations than to those generated by title words.

Of particular interest would be a system which utilized the type of usage data described in Chapters II and III.

CHANGING FILE

There are a number of questions relating the fact that a document collection is continually changing. What should happen when documents are added to or deleted from the file? Can the user be automatically notified of new documents of interest? In this connection one might want the user to permanently store those clusters found to be of interest. Then as nwe documents come into the file they can be compared against the clusters. The user would then be notified of those articles which were valid members of his clusters.

CODING

There is also need for additional work on the problem of data coding and compression. For example, one might be able to reduce storage requirements considerably by storing codes for all (or certain) author names in the raw data file. This may be true of the other types of data also.

10.42 Procedure and Language

There are a number of directions in which the clustering procedure and interaction language might be extended. One objective might be to

make a wider class of statements acceptable and understandable to the system. This might involve increasing the vocabulary and/or allowing other syntactic forms.

PARSING BY CONTEXT

As a specific suggestion we note that the current system determines the function of (parses) a word by a simple table look-up. A word cannot have a dual function depending on its context. Thus if one wants to use "p" as an abbreviation for print (p. the titles of set 1), this would currently exclude its use say as an abbreviation for paper or as the initial in an author's name ("get articles by 'P. A. Jones'" would however be acceptable). It should be possible, however, to distinguish between these different uses, if one utilizes the context.

GRAPHIC DISPLAY

A more radical extension of the language would be through the use of some type of graphical device. For example, it might prove useful to display part of the document network on an oscilloscope and to allow the user to specify the interesting and non-interesting documents by means of a light pen.

In addition to increasing the flexibility of the language, one might also want to allow the specification of some other functions. Let us suggest some additional functions that the clustering procedure might appropriately perform.

CLUSTER SIZE

A user might want to limit the size of the answer cluster to some specified range at the outset. (e.g. "Get between 3 and 7 articles related to Phys. Rev. v. 136 p. 1899.") This could be accomplished by

increasing or decreasing the bias enough so that the size of the answer cluster fell within the specified range.

DATA BASE

It would also be of value to a user if he could specify the type of partitioning data to be used by the clustering procedure. Thus the command, "Get the articles related by authors and users to Phys. Rev. Letters v. 11 p. 6", would use the partitions generated by both authors and usage data to create the answer cluster. This control could be extended to select for the data base certain classes of partitions within a broad type. For example, a request of the type, "Get the articles related by M.I.T. faculty users to Phys. Letters v. 7 p. 14", would allow the user to single out for use that type of partitioning which he thought would yield the best results.

CLUSTERS OF AUTHORS,ETC.

There is no real reason why clusters must be limited to sets of documents. It may be useful to generalize the system to allow clusters to be formed of other types of entities such as authors, locations, words, etc. It might be very helpful, for example, to be able to determine the cluster of scientists that are working in a given field or area.

10.43 Theoretical Problems

ANSWER CLUSTER DEFINITION

Some modification to the definition of an answer cluster may be of value. For example, should a change be made to the requirement that all the documents specified as interesting be in the cluster?

NOISE

There will, of course, be cases where certain documents are

mistakenly included together in a set of interest. This may arise, for example, from an incorrect judgement on the part of a user or perhaps by a clerical slip. The effect of this type of noise on the system should be investigated. Also suitable steps should be taken to maintain the integrity of the data base through editing processes.

SELF-SUSTAINING RUTS

Consider an information retrieval system which is based on the data generated by its users. This might be one based on usage data or on citations. Is it possible in such a system for a self-reinforcing feedback loop to be created which cannot be altered? For example, if users are supplied documents on the basis of past use, this may create new partitions which only serve to reinforce the results of the old partitions.

EVALUATION MEASURE

The measure described in Chapter III was not suggested for use in rating the merit or value of documents. Its function was to group together documents that were mutually pertinent. If a suitable way could be devised for measuring the worth of documents, this would be of considerable aid to users. Perhaps this would take the form of some type of concensus of opinion of the previous users of the documents.

TRAILS VS. SETS

In the article already cited by V. Bush the model suggested for information retrieval was a trail leading from one pertinent document to the next. The model used in this research endeavor is the partitioning of the file into two subsets. Actually both models have useful features. In some cases there is a definite pattern or trail which should be followed in consulting the documents related to a given

subject. In other cases the order in which the documents should be examined is apparent from their publication data. In still other cases there is no particular order in which the documents need be consulted. Thus it would seem that one might want to include both the ideas of sets of documents and trails of documents in a more general information retrieval model.

PREDICTIVE USAGE

As additional information becomes available on the types of questions that are asked by users and the sets of documents that seem to satisfy them, it may be possible to design a system involving some form of prediction of what a user really wants when he asks a given question. This might even be extended to involve trends in document usage, so that future document use is extrapolated on the basis of past use.

APPENDIX A

MEASURES OF RELATEDNESS

Some of the measures which have been proposed for use in information retrieval are tabulated below. Measures (1) to (6) were originally suggested in terms of frequency counts. Measures (7) and (8) were first proposed in terms of probabilities. For purposes of comparison we have attempted to express each measure in the table both in terms of probabilities and frequency counts. In the case of measure (5) this was not possible.

The definitions for the symbols used in the table and the conversion formulae for going from probabilities to frequency counts and back again are found in Sec. 3.1. It was necessary to add superscripts to the frequency counts in the table to distinguish between some additional counts which appear in these measures. Thus $N_{i\bar{j}}^{01}$ is the number of partitions in which the subset of interest contains document j but not i.

Name	Range	C - Probabilities	C - Frequency Counts
1. Comparison Function (Martin)	$0 \to 1/2$	$C = \dfrac{p(x_i^1 x_j^1)}{p(x_i^1)+p(x_j^1)}$	$\tilde{C} = \dfrac{N_{ij}^{11}}{N_i^1+N_j^1}$
2. Association Measure[15,44] (Doyle-1962)	$0 \to 1$	$S = \dfrac{p(x_i^1 x_j^1)}{p(x_i^1)+p(x_j^1)-p(x_i^1 x_j^1)}$	$\tilde{S} = \dfrac{N_{ij}^{11}}{N_i^1+N_j^1-N_{ij}^{11}}$
3. Modified Coefficient of Colligation[33,55] (Maron-1960)	$-1 \to 1$	$Q = \dfrac{p(x_i^1 x_j^1)-p(x_i^1)p(x_j^1)}{p(x_i^1 x_j^1)p(x_i^0 x_j^0)+p(x_i^1 x_j^0)p(x_i^0 x_j^1)}$	$\tilde{Q} = \dfrac{N_{ij}^{11}N_{ij}^{00}-N_{ij}^{10}N_{ij}^{01}}{N_{ij}^{11}N_{ij}^{00}+N_{ij}^{10}N_{ij}^{01}}$
4. Pearson Correlation Coefficient[6,7] (Borko-1962)	$-1 \to 1$	$r = \dfrac{p(x_i^1 x_j^1)-p(x_i^1)p(x_j^1)}{\sqrt{p(x_i^1)p(x_i^1)p(x_j^0)p(x_j^0)}}$	$\tilde{r} = \dfrac{N_{ij}^{11}N_{ij}^{00}-N_{ij}^{10}N_{ij}^{01}}{\sqrt{N_i^1 N_i^0 N_j^1 N_j^0}}$
5. Chi Square Formula with Yates Correction[43] (Stiles-1961)	$0 \to \infty$	- - -	$\tilde{X}^2 = N\dfrac{\left(\left\lvert N_{ij}^{11}N_{ij}^{00}-N_{ij}^{10}N_{ij}^{01}\right\rvert - \frac{N}{2}\right)^2}{N_i^1\ N_i^0\ N_j^1\ N_j^0}$
6. Cosine Function[42] (Salton-1963)	$0 \to 1$	$R = \dfrac{p(x_i^1 x_j^1)}{\sqrt{p(x_i^1)p(x_j^1)}}$	$\tilde{R} = \dfrac{N_{ij}^{11}}{\sqrt{N_i^1 N_j^1}}$
7. Average Information-Theoretic Correlation Coefficient[49,50] (Watanabi-1960)	$0 \to 1$	$C = \sum\limits_{a,b=0,1} p(x_i^a x_j^b)\log\dfrac{p(x_i^a x_j^b)}{p(x_i^a)p(x_j^b)}$	$\tilde{C} = \sum\limits_{a,b=0,1} \dfrac{N_{ij}^{ab}}{N}\log\dfrac{N\ N_{ij}^{ab}}{N_i^a N_j^b}$
8. Information-Theoretic Correlation Coefficient[19] (Fano-1958)	$-\infty \to \infty$	$C = \log\dfrac{p(x_i^1 x_j^1)}{p(x_i^1)p(x_j^1)}$	$\tilde{C} = \log\dfrac{N\ N_{ij}^{11}}{N_i^1 N_j^1}$

BIBLIOGRAPHY

1. Baker, Frank B., "Information Retrieval Based Upon Latent Class Analysis," Journal of the ACM, (Oct. 1962), vol. 9, no. 4, pp. 512-521.

2. Bar-Hillel, Y., "A Logician's Reaction to Information Search System," American Documentation, (Apr. 1957), vol. 8, no. 2, pp. 103-114.

3. Bar-Hillel, Y., "Some Theoretical Aspects of the Mechanization of Literature Searching," Technical Report No. 3, PB 161-547, Office of Technical Services, Washington, D. C., April 1960.

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

5. Bonner, R. E., "On Some Clustering Techniques," IBM Journal of Res. and Dev., (Jan. 1964), vol. 8, no. 1, pp. 22-31.

6. Borko, H., "The Construction of an Empirically Based Mathematically Derived Classification System," Proceedings of the 1962 Spring Joint Computer Conference, pp. 279-289.

7. Borko, Harold and Bernick, Myrna, "Automatic Document Classification," Journal of the ACM, (April 1963), vol. 10, no. 2, pp. 151-162.

8. Borko, Harold and Bernick, Myrna, "Automatic Document Classificationn- Part II. Additional Experiments," Journal of the ACM, (April 1964), vol. 11, no. 2, pp. 138-151.

9. Bourne, C. P., "Bibliography on the Mechanization of Information Retrieval," Stanford Research Institute, Menlo Park, Calif., 1958.

10. Bush, Vannevar, "As We May Think," The Atlantic Monthly, (July 1945), vol. 176, no. 1, pp. 101-108.

11. Cheatham, T. E., Jr., and Warshall, S., "Translation of Retrieval Requests Couched in a 'Semiformal' English-Like Language," Comm. of the ACM, (Jan. 1962), vol. 5, no. 1, pp. 34-39.

12. Crisman, P. A., editor, "The Compatible Time-Sharing System-- A Programmer's Guide, Second Edition," The MIT Computation Center, MIT Press, Cambridge, Mass., 1965.

13. Dale, A. G., and N., "Some Clumping Experiments for Associative Document Retrieval," American Documentation, (Jan. 1965), vol. 16, no. 1, pp. 5-9.

14. Doyle, Lauren B., "Semantic Road Maps for Literature Searchers," Journal of the ACM, (Oct. 1961), vol. 8, no. 4, pp. 553-578.

15. Doyle, Lauren B., "Indexing and Abstracting by Association," American Documentation, (Oct. 1962), vol. 13, no. 4, pp. 378-390.

16. Doyle, Lauren B., "Is Automatic Classification a Reasonable Application of Statistical Analysis of Text?" Journal of the ACM, (Oct. 1965), vol. 12, no. 4, pp. 473-489.

17. Fairthorne, R. A., Towards Information Retrieval, Butterworths, London, 1961.

18. Fano, R. M., "Information Theory and the Retrieval of Recorded Information," Documentation in Action, pp. 238-244, Reinhold, N. Y., 1956.

19. Fano, R. M., "Example of Storage and Retrieval Procedure," Research Laboratory of Electronics Memorandum, MIT, Cambridge, Mass., 9 March 1959.

20. Fano, R. M., Transmission of Information, MIT Press, Cambridge, Mass., 1961.

21. Fano, R. M., "The MAC System: the Computer Utility Approach," IEEE Spectrum, (Jan. 1965), vol. 2, no. 1, pp. 56-64.

22. Feller, William, Probability Theory and its Applications, John Wiley and Sons, New York, 1950.

23. Henderson, Madeline Berry, "Organizations Active in Machine Indexing Research," Machine Indexing: Progress and Problems, Am. Univ., Feb. 13-17, 1961, pp. 23-39.

24. Kelley, John L., General Topology, D. Van Nostrand Co., Princeton, N. J., 1955.

25. Kessler, M. M., "An Experimental Communication Center for Scientific and Technical Information," Lincoln Laboratory, Report 4G-0002, Lexington, Mass., 31 March 1960.

26. Kessler, M. M., "Background to Scientific Communication," IRE Transactions, (Apr. 1960), vol. EWS-3, no. 1, pp. 3-6.

27. Kessler, M. M., and Heart, F. E., "Concerning the Probability that a Given Paper Will be Cited," MIT Technical Information Project Report Number 6 (Nov. 5, 1962).

28. Kessler, M. M., "Bibliographic Coupling Between Scientific Papers," American Documentation, (Jan. 1963), vol. 14, no. 1, pp. 10-25.

29. Kessler, M. M., "The MIT Technical Information Project," Physics Today, (March 1965), vol. 18, no. 3, pp. 28-36.

30. Kessler, M. M., TIP User's Manual, First Edition, Mimeographed at MIT, Dec. 1, 1965.

31. Luhn, H. P., "Automatic Creation of Literature Abstracts," IBM Journal of Res. & Dev., (Apr. 1958), vol. 2, no. 2, pp. 159-165.

32. Maron, M. E., and Kuhns, J. L., "On Relevance, Probabilistic Indexing and Information Retrieval," Journal of the ACM, (July 1960), vol. 7, no. 3, pp. 216-244.

33. Maron, M. E., "Automatic Indexing: An Experiment Inquiry," Journal of the ACM, (July 1961), vol. 8, no. 3, pp. 404-417.

34. Mooers, Calvin N., "The Next Twenty Years in Information Retrieval," American Documentation, (July 1960), vol. 11, no. 3, pp. 231-232.

35. Moore, E. F., Sequential Machines, Addison-Wesley, Reading, Mass., 1964.

36. National Science Foundation, "Current Research Development in Scientific Documentation No. 11," Office of Technical Services, Dept. of Commerce, Washington, D. C., Nov. 1962.

37. Naur, P., et al., "Report on the Algorithmic Language ALGOL 60," Comm. of the ACM, (May 1960), vol. 3, no. 5, pp. 299-314.

38. Needham, R. M., and Parker-Rhodes, A. F., "The Theory of Clumps," Report M. L. 126, Cambridge Language Research Unit, Cambridge, England, Feb. 1960.

39. Needham, R. M., "Theory of Clumps II," Report M. L. 139, Cambridge Language Research Unit, Cambridge, England, March 1961.

40. O'Connor, John, "Mechanized Indexing Methods and Their Testing," Journal of the ACM, (Oct. 1964), vol. 11, no. 4, pp. 437-449.

41. Parker-Rhodes, A. F., "Contributions to the Theory of Clumps: the Usefulness and Feasibility of the Theory," Report M. L. 138, Cambridge Language Research Unit, Cambridge, England, March 1961.

42. Salton, Gerald, "Associative Document Retrieval Techniques Using Bibliographic Information," Journal of the ACM, (Oct. 1963), vol. 10, no. 4, pp. 440-457.

43. Stiles, H. Edmund, "The Association Factor in Information Retrieval," Journal of the ACM, (April 1961), vol. 8, no. 2, pp. 271-279.

44. Tanimoto, T. T., "An Elementary Mathematical Theory of Classification and Prediction," IBM internal report, New York, N. Y., Nov. 1958.

45. Taube, Mortimer, "A Note on the Pseudo-Mathematics of Relevance," American Documentation, (April 1965), vol. 16, no. 2, pp. 69-72.

46. Taube, M., and Wooster, H., Information Storage and Retrieval Theory, Systems, and Devices, N. Y., 1958, Columbia Univ. Press.

47. Taube, M.; Gull, C. D.; and Wachtel, I. W., "Unit Terms in Coordinate Indexing," American Documentation, (Oct. 1952), vol. 3, no. 4, pp. 213-218.

48. Vickery, B. C., On Retrieval System Theory, Butterworths, London, 1961.

49. Watanabi, Satosi, "Information Theoretical Analysis of Multi-Variate Correlation," IBM Journal of Res. & Dev., (Jan. 1960), vol. 4, no. 1, pp. 66-82.

50. Watanabi, Satosi, "A Note on the Formation of Concept and of Association by Information--Theoretical Correlation Analysis," Information and Control, (Sept. 1961), vol. 4, no. 3, pp. 291-296.

51. Weizenbaum, J., "Symmetric List Processor," Comm. of the ACM, (Sept. 1963), vol. 6, no. 9, pp. 524-544.

52. Winters, William K., "A Modified Method of Latent Class Analysis For File Organization in Information Retrieval," Journal of the ACM, (July 1965), vol. 12, no. 3, pp. 356-363.

53. Yngve, Victor H., "COMIT as an IR Language," Comm. of the ACM, (Jan. 1962), vol. 5, no. 1, pp. 19-28.

54. Yngve, V., "COMIT," Comm. of the ACM, (March 1963), vol. 6, no. 3, pp. 83-84.

55. Yule, G. U., "On Measuring Association Between Attributes," Journal of the Royal Statistical Society, (May 1912), vol. 75, Part 6, pp. 579-642.

BIOGRAPHY OF AUTHOR

Evan Leon Ivie was born May 15, 1931, in American Fork, Utah. He graduated from Washington-Lee High School in Arlington, Virginia in 1949 as valedictorian. He then entered Brigham Young University In 1951-53 he served as a missionary for his Church in Canada. He returned to Brigham Young University receiving in 1956 with high honors the degrees of Bachelor of Science in physics and Bachelor of Engineering Science in electrical engineering.

In June 1957 he completed the requirements for the Master of Science degree at Stanford University. While at Stanford he held the Consolidated Electrodynamics fellowship. For the three following years he served as Lieutenant in the United States Air Force assigned to the Defense Intelligence Agency.

Mr. Ivie then began studies at the Massachusetts Institute of Technology. He was awarded a National Science Foundation Fellowship for three years. At M.I.T. he was a teaching assistant for various courses in computer technology and a research assistant with Project MAC and the Technical Information Project. He is co-author of the article, "The M.I.T. Technical Information Project", (Parameters of Information Science, 1964).

He is a member of Phi Kappa Phi, Sigma Xi, Sigma Pi Sigma, the Association for Computing Machinery and the Institute of Electronic and Electrical Engineers. He has held summer positions at Convair-Pomona, Convair-Astronautics, and Lincoln Laboratory. In March 19[illegible] he married the former Betty Jo Beck. They are the parents of three sons and three daughters.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Massachusetts Institute of Technology
Project MAC

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
None

3. REPORT TITLE
Search Procedures Based on Measures of Relatedness Between Documents

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Doctoral Thesis, Electrical Engineering

5. AUTHOR(S) (Last name, first name, initial)
Ivie, Evan L.

6. REPORT DATE
June 1966

7a. TOTAL NO. OF PAGES
240

7b. NO. OF REFS
55

8a. CONTRACT OR GRANT NO.
Office of Naval Research, Nonr-4102(01)

b. PROJECT NO.
Nr-048-129

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
MAC-TR-29 (THESIS)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Advanced Research Projects Agency
3D-200 Pentagon
Washington, D. C. 20301

13. ABSTRACT
A new type of information retrieval system is suggested which utilizes data of the type generated by users of the system instead of data generated by indexers. The theoretical model on which the system is based consists of three basic elements. The first element is a measure of the relatedness between document-pairs. It is derived from information theory. The second element is a definition of what constitutes a set (cluster) of inter-related documents. This definition is based on the measure of relatedness. The last element is a procedure which transforms a request for information into a cluster of answer documents. An experimental system was developed to test the model in a realistic environment. It was programmed for the Project MAC time-sharing system and utilized the physics data file of the Technical Information Project. Citations were used as the data base for the measure of relatedness. A file structure and retrieval language were designed which allowed close man-machine coupling. Retrieval efficiency compared to known sets was 60 - 90 percent, and ways of improving this further are suggested.

14. KEY WORDS

Computers	Machine-aided cognition	Real-time computer systems
Document searching	Multiple-access computers	Time-sharing
Information retrieval	On-line computer systems	Time-shared computer systems

DD Form 1473 (M.I.T.)

UNCLASSIFIED
Security Classification