

STATISTICAL INFORMATION RETRIEVAL

By

CHARLES DOUGLAS PARSONS

Bachelor of Science

New Mexico State University

Las Cruces, New Mexico

1957

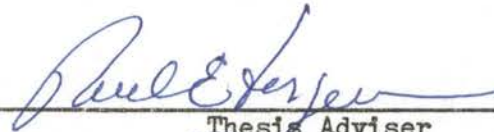
Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
July, 1966

OKLAHOMA
STATE UNIVERSITY
LIBRARY

JAN 27 1967

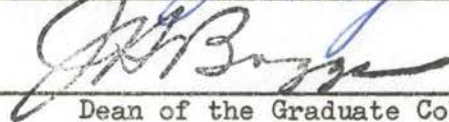
STATISTICAL INFORMATION RETRIEVAL

Thesis Approved:



Thesis Adviser





Dean of the Graduate College

627249

PREFACE

This thesis discusses the three major aspects of information retrieval -- indexing, storage, and retrieval. The history and some of the basic techniques are reviewed with special emphasis on those areas which are applicable to computer processing.

Concluding this thesis is a description of a production information retrieval system developed for the Research and Development Department of Phillips Petroleum Company. Included are samples of inquiries and the resulting output.

I wish to acknowledge and express my appreciation to all the individuals who assisted in the preparation of this report. I am particularly indebted to Mr. H. P. Luhn (posthumously) for his advice and counsel. Also, I wish to thank Mrs. Molly Wolfe and Mr. L. H. Dilliman for their assistance in testing and programming.

TABLE OF CONTENTS

| Chapter | Page |
|---|------|
| I. REVIEW OF INFORMATION RETRIEVAL | 1 |
| History | 2 |
| II. INDEXING | 4 |
| Manual Indexing | 4 |
| Automatic Indexing | 7 |
| Review | 8 |
| Associative Analysis | 9 |
| Factor Analysis | 13 |
| Latent Class Analysis | 19 |
| Association Factor | 24 |
| Summary of Indexing | 28 |
| III. STORAGE | 29 |
| File Organization | 29 |
| Folded-Word Sum Algorithm | 31 |
| Predestined Storage | 33 |
| IV. RETRIEVAL | 39 |
| Coordinate Concept | 39 |
| Retrieval Search Strategy | 41 |
| Retrieval Logic | 42 |
| Summary of Retrieval | 45 |
| V. THE SIR SYSTEM | 46 |
| System Design | 46 |
| Storage Processing | 48 |
| Retrieval | 52 |
| VI. TESTS, CONCLUSIONS, AND RECOMMENDATIONS | 59 |
| Tests | 59 |
| Conclusions | 61 |
| Indexing | 61 |
| Storage | 62 |
| Retrieval | 62 |

| Chapter | Page |
|---------------------------|------|
| VI. (Continued) | |
| Recommendations | 62 |
| Indexing | 63 |
| Storage | 63 |
| Retrieval | 64 |
| BIBLIOGRAPHY | 65 |

LIST OF TABLES

| Table | Page |
|--|------|
| I. Group Attributes | 15 |
| II. Response Pattern | 20 |
| III. Latent Structure | 22 |
| IV. Estimated Latent Structure | 23 |
| V. Term Frequencies | 26 |
| VI. Association Factor Example | 27 |
| VII. Vector-Location Examples | 36 |
| VIII. Test Results | 60 |

LIST OF FIGURES

| Figure | Page |
|--|------|
| 1. Classification Matrix | 10 |
| 2. Example Data Matrix | 12 |
| 3. Example Classification | 13 |
| 4. Associative Matrix | 14 |
| 5. Example Matrix | 15 |
| 6. Example Matrix | 16 |
| 7. Geometrical Representation of Eigenvalues | 17 |
| 8. Magnetic Tape Format | 30 |
| 9. Modulated Storage | 37 |
| 10. Coordinate Concepts | 40 |
| 11. Pertinency Levels | 42 |
| 12. Keyword Matrix | 43 |
| 13. SIR System Flow Diagram | 47 |
| 14. Storage Data Form | 49 |
| 15. Retrieval Procedures | 53 |
| 16. Retrieval Data Form | 54 |
| 17. Sample Output from Retrieval Program | 58 |

CHAPTER I

REVIEW OF INFORMATION RETRIEVAL

The technological -- and non-technical - information has been increasing at an ever-increasing rate. The "Information Explosion" problem has been recognized for many years but only in the last seven or eight years have steps been taken to relieve the scientist, engineer, and researcher from the awesome task of keeping up with all the information impinging on his work area. The thousands of technical journals, papers, and letters deluge the technical and professional people to such an extent that he is only able to scan a small percentage of the material for the pertinent facts relating to his work. This lack of pertinent information leads to losses for his employer in the form of reduced creative effort, duplication of effort and voids in the technical communications so necessary in a progressive and successful company.

This thesis reviews the history of statistical information retrieval and discusses the various aspects in detail. The thesis also describes a statistical information retrieval system on which the author was principally responsible for the design and specification. In order to relate the detailed material, reviews are presented in the appropriate chapters discussing the material.

History

With the advent of the electronic computer, such men as J. W. Mauchly (1) in 1949, (co-inventor of ENIAC and UNIVAC, the first electronic digital computers), envisioned the use of computers for documentation and library science activities. But it was several years before the computer sciences developed high-speed storage capacities to make information storage and retrieval a practical and economical undertaking. Several more years went by before H. P. Luhn (2) and M. Taube (3) gained recognition for their ideas on automatic information storage and retrieval.

One of the first mentions of a statistical approach to information retrieval was a paper by H. P. Luhn presented to the American Chemical Society in April, 1957 (4). Mr. Luhn talks of automatic encoding of documents using frequency as a criteria for selecting keywords or notions to represent a document. But, more important to this thesis, Mr. Luhn discussed the statistical aspects of communicating ideas or concepts. He reasoned, "In order to communicate an idea, we break it down into a series of little ideas, i.e., more elementary ideas (words) for which previous and common experience might have led to an agreement of meaning." The natural extension of this principle enable the expression of more complex concepts as combinations of the elemental ideas (or keywords or descriptors). Now, the statistical assumption on which Mr. Luhn bases his thesis is that similar ideas will have, as common denominators, the same elemental ideas. Or as Mr. Luhn states "Thus the statistical probability of combinations of similar ideas being similarly inter-

preted must be very high." This is Mr. Luhn's case for keyword indexing of literary material.

The case for retrieving an idea or notion based on this same principle will therefore be valid. An inquiry for information can also be represented by selected elemental ideas or notions thus putting the documented and inquiry information on the same basis.

Now, the retrieval of information pertinent to a particular inquiry can be accomplished by matching the elemental ideas (keywords) of the inquiry with those of the documented information. When the documented information obtains a required level of matches with inquiry ideas (keywords), the pertinency of the document to the inquiry is assured. The number of matches required depends on the type of documents, depth of indexing and the detail of the inquiry.

Taube and his associates are credited with the first tests of statistical association for information indexing. Much of Taube's early work parallels that of Mr. Luhn.

The above principles evolved into an information indexing technique called "coordinate concepts." The ideas (keywords) are thought of as the coordinate values of a concept (document) in a n -dimensional space, where " n " would be the number of ideas (keywords) used to represent the concept (document). A detailed example of the coordinate concept technique is given in Chapter 4.

CHAPTER II

INDEXING

The over-all problem of information storage and retrieval can be divided into three areas -- indexing, storage, and retrieval. The next three chapters are devoted to developments in these areas and, where applicable, the author's contributions while working in information processing.

Indexing of information is probably as old as written history, beginning with the classification of documents in ancient libraries and data collected by early astronomers. The two major purposes for indexing or classification are (1) ease of storage and retrieval and (2) identification. This chapter is mainly concerned with storage and retrieval indexing, but benefits are derived from taxonomy techniques (Tanimoto) for categorizing unknown information into known groupings.

Manual Indexing

The subject headings in the local library are probably the best known. Indexing can be the classification of information into fixed concepts. Subject headings for example, or concepts can be extracted directly from the information and used as the index. The type of indexing depends on the material to be indexed and the depth of

indexing required or desired. Subject headings are generally used as broad categories of interest, whereas indexes, based on concepts extracted from the material being indexed, are usually deep and relate to detailed information of a complex nature.

Manual indexing for computer information retrieval (IR) systems is usually based on an extensive vocabulary developed from the type of information being indexed. For example, the vocabulary or thesaurus sponsored by the Joint Engineers Council has between five and six thousand keyword descriptors, which are sufficient for indexing most engineering material not including speciality vocabularies, such as chemical compounds. Use of this type of thesaurus requires the indexer to read the document to be indexed extracting the key concepts and then, correlate these with the thesaurus, choosing a synonym or generic key concept if the extracted concept is not in the thesaurus. But, using this fixed thesaurus induces limitations in vocabulary expansion and causes additional effort in indexing the documents. The author has experienced some advantages in allowing the indexer to use any keyword concepts extracted from the documents but providing for some vocabulary control. Advantages are faster indexing and an open-ended vocabulary which changes with the introduction of new concepts and meanings. This principle of indexing gives the system a statistical nature, i.e., introduces the probability that the documents pertaining to a particular request will be indexed under some of the keywords used to describe the request.

To further specify the indexing of particularly complex information, several methods are used which relate the index term to the document subject. The first of these is an extension of indexing

called a "role indicator." This coded indicator accompanies the keyword and shows the relationship to the main subject. In the Engineers Joint Council system, the keyword indexing of a primary topic of a document is followed by an "8" and a keyword indicating the input or raw material used in a documented process would have a "1" to show this relationship to the topic.

A second method called "link" codes are used to group the keywords in a document discussing several topics into their related categories.

Although the above methods reduce the misconceptions, all IR systems have inherent weaknesses in the manner in which they handle synonyms, semantics, and syntax.

Synonyms - Two or more words having the same or nearly the same meaning.

Semantics - Changes in the meanings of words with the passage of time.

Syntax - A change in the order of word phrases which also changes the meaning.

Some IR systems attempt to reduce the effect of these weaknesses by introducing more complex indexing. But, the more complex indexing will not completely eliminate these weaknesses and will also introduce inefficiencies which cause most systems of this complexity to be impractical.

This author chooses to cope with the above weaknesses by exploiting the information available from the inquirer. For instance, the inquirer will furnish synonyms, if any, for all keywords in an inquiry and the computer system will search the synonyms, setting

them equivalent to the original keyword.

In the case of semantic changes, the type of vocabulary used will determine the extent of this problem. With a fixed vocabulary, it is necessary to periodically up-date the entire file with a new vocabulary incorporating these semantic changes. However, using an open-ended vocabulary allows the semantic changes to be introduced as new information is processed, thus up-dating on a continual basis.

In most IR systems, syntax is a problem left to the indexer and inquirer. This is not considered to be a significant problem despite the considerable attention given in the literature. When a syntactical error occurs, it is usually quite apparent and easily corrected.

The following steps summarize the indexing technique which this author has used with considerable success.

1. Extract keywords for indexing from the context of the document.
2. Assign role indicators qualifying the keywords to the context.
3. Exercise vocabulary control as an option to the indexer.
4. Place the responsibility for synonyms and syntax on the inquiry or retrieval portion of the system.

Although this technique of indexing is not unique, the author's review of IR literature did not disclose any systems using the above open-ended vocabulary with corresponding retrieval technique.

Automatic Indexing

Computer mechanization has successfully conquered the storage and

retrieval of information and the degree of this success continues to increase as better and more sophisticated equipment and techniques become available. But, the mechanization of indexing (hereafter referred to as automatic indexing) seems to defy solutions which are both effective and economically feasible. For automatic indexing to be effective, methods have to be developed to mechanically scan the printed text and extract or assign indexes which will sufficiently described the contents of the text for future retrieval.

Review

Luhn (5), Maron (6), Doyle (7) and Tanimoto (8) have attempted automatic indexing using various techniques, but with limited success. These techniques range from methods using the original text to methods which modify or classify manual indexing. One of the earlier methods used was simply frequency counts. This involves counting the number of occurrences of significant words (non-significant words are eliminated by using selected lists) from a text. Those having high counts and considered representative are used to index the text. This technique has obvious shortcomings which are inherent in our natural language. The most common of these being the possibility of having a major concept tied to one word which may only occur once.

Many of the techniques used in automatic indexing experiments are based on "factor analysis" and, as such, can be referred to as classification. Factor analysis has been used to classify information in a variety of ways -- modifying inputs and multi-interpretation of outputs. Experiments in this area have also been referred to as facet analysis and latent class analysis, but basically the work is as

follows.

The technique of Factor Analysis is statistical in nature and is not designed to handle or extract exact data. Items which are classified under numerous categories have the most applicability to this technique, such as an information retrieval system where a document is classified under a fixed vocabulary or a personnel file where an employee may be classified under numerous headings. Factor Analysis provides a means of reducing the number of categories or headings necessary to describe an item. Essentially, it groups the dependent categories into a new derived category and provides a quantitative means of describing an item in terms of this new set of categories. This same procedure can be performed on several levels and could be used to organize storage rather than reduce it. Should large scale random storage of information ever become a reality, this technique combined with Barycentric Coordinates¹ could be developed into a very useful tool.

Associative Analysis

Before Factor Analysis, the associative relationship of items to be classified and the desired categories are established. This section describes this relationship and shows a simple comparison analysis for automatically assigning items to pre-assigned categories.

Most information or data processing problems involve some form of classification. Either an item is to be found that is classified

¹Barycentric coordinates are the coordinating values assigned to represent a desired location (item, concept) in relation to a number of fixed locations (items, concepts).

under certain categories or it is desired to store an item under certain categories. This item could be data, relative information, etc.

Assuming some form of classification, a matrix can be generated of items (I_i) versus categories (C_j) in Figure 1.

| | C_1 | C_2 | C_3 | ... | C_j |
|-------|----------|----------|------------|-----|----------|
| I_1 | X_{11} | X_{12} | X_{13} | ... | X_{1j} |
| I_2 | X_{21} | | . | | |
| I_3 | X_{31} | . | (X_{3j}) | | |
| I_4 | X_{41} | | | | |
| . | | | | | |
| . | | | | | |
| I_i | X_{i1} | | | | |

i = number of items

j = number of categories

Figure 1. Classification Matrix

This classification matrix could be called item-category, document-keyword or named for any other data which is classified.

One of the first applications can be found in the simple comparison of one item against some pre-assigned groups of items. For example, say it is desired to classify personnel in pre-assigned classes - 25-30, 30-40, or 40 plus years of age and male or female categories. Now, the personnel file is to be grouped by different age brackets and sex. Each new employee would have a profile record in terms of the above categories, and this profile would be compared with the group profile to determine which group the new employee would be assigned to:

C_1 = 25 - 30 years of age.

C_2 = 30 - 40 years of age.

C_3 = 40 plus years of age.

C_4 = male.

C_5 = female.

I_1 = group of all females 25 - 30 years of age.

I_2 = group of all males 25 - 30 years of age.

I_3 = group of all females 30 - 40 years of age.

etc.

This grouping could be expressed in a matrix similar to matrix (1).

| | 25-30 | 30-40 | 40+ | Male | Female |
|---------------|-------|-------|-----|------|--------|
| Female, 25-30 | 1 | 0 | 0 | 0 | 1 |
| Male, 25-30 | 1 | 0 | 0 | 1 | 0 |
| Female, 30-40 | 0 | 1 | 0 | 0 | 1 |
| Male, 30-40 | 0 | 1 | 0 | 1 | 0 |
| Female, 40+ | 0 | 0 | 1 | 0 | 1 |
| Male, 40+ | 0 | 0 | 1 | 1 | 0 |

Figure 2. Example Data Matrix

Then, given a vector profile of a new employee -- for example, a woman who is 35 years old,

(0 1 0 0 1)

(1)

The transpose of this profile vector can now be multiplied by the matrix in Figures 2.

| | 25-30 | 30-40 | 40+ | Male | Female | New Employee | Product Vector |
|---------------|-------|-------|-----|------|--------|---|--|
| Female, 25-30 | 1 | 0 | 0 | 0 | 1 | $\begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{Bmatrix}$ | $= \begin{Bmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 1 \\ 0 \end{Bmatrix}$ |
| Male, 25-30 | 1 | 0 | 0 | 1 | 0 | | |
| Female, 30-40 | 0 | 1 | 0 | 0 | 1 | | |
| Male, 30-40 | 0 | 1 | 0 | 1 | 0 | | |
| Female, 40+ | 0 | 0 | 1 | 0 | 1 | | |
| Male, 40+ | 0 | 0 | 1 | 1 | 0 | | |

Figure 3. Example Classification

The largest element of the product vector in (Figure 3.) gives the group to which the new employee will be assigned (this is group for female, 30-40 in the example). Although it may not be practical in the above example, weights could be assigned to the profile vector in order to establish some desired priority.

Factor Analysis

The technique known as Factor Analysis is very useful in grouping data or eliminating the dependent categories of data. Assuming an example such as the matrix in Figure 1. (having numerous items and numerous classifications), it may be required to eliminate the dependent categories, i.e., categories which are highly correlated with other

categories. By application of this technique, any set of categories can be grouped into a new category labeled as the "center of gravity" of the original set of categories. Any of the original sets can be represented by a single point which is given the relative weights of the categories from the original set.

The first operation in factor analysis is to determine the category associative matrix. This is accomplished by multiplying matrix (Figure 1.) by its transpose to get a matrix which is known as the variance-co-variance matrix. This multiplication insures the requirements for an eigenvalue analysis, i.e., the determinant of the associative matrix is equal to zero. The elements (A_{ij}) in the associative matrix indicate the relative correlation between the i th and the j th categories in matrix (Figure 1.).

$$\left\{ \begin{array}{cccccc} A_{11} & A_{12} & A_{13} & \dots & A_{1j} \\ A_{21} & A_{22} & & & \\ A_{31} & & & & \\ \vdots & & & & \\ \vdots & & (A_{ij}) & & \\ A_{j1} & & & & \end{array} \right\}$$

$$i = 1, \dots, j$$

$$j = 1, \dots, j$$

Figure 4. Associative Matrix

On the matrix in Figure 4, the second operation is to proceed with an eigenvalue analysis by subtracting a λ from the diagonal elements of the associative matrix, and the resulting determinant times $(-1)^j$ is expanded to a poly-nominal of the order j . This is called the "characteristic polynominal" of the associative matrix and is solved for the j roots of λ . These roots are known as "Characteristic values" or eigenvalues of the associative matrix.

For an example of eigenvalue analysis, the following conditions are assumed.

3 group classifications

4 possible attributes

TABLE I
GROUP ATTRIBUTES

| Group | Attributes | | | |
|-------|------------|---|---|---|
| | A | B | C | D |
| 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 1 | 1 | 0 | 1 |

The responses of the groups to the attributes are shown in Table 1.

There responses form the classification matrix in Figure 5.

$$\begin{Bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{Bmatrix}$$

Figure 5. Example Matrix

This matrix is multiplied by its transpose to give the associative matrix in Figure 6.

$$\begin{Bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 3 \end{Bmatrix}$$

Figure 6. Example Matrix

For the eigenvalues, a λ is subtracted from the diagonal elements, and the resulting determinant is expanded to give Equation 2.

$$\lambda^3 - 7\lambda^2 + 13\lambda - 7 = 0 \quad (2)$$

the roots, of which, are:

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 3 + \sqrt{2} \\ \lambda_3 &= 3 - \sqrt{2} \end{aligned} \quad (3)$$

The characteristic value in (3) are called eigenvalues.

Now, it is necessary to lend some interpretation to these eigenvectors. First, the associative matrix is symmetrical since it is the product of a matrix times its transpose. Therefore, the geometrical representation of the space described by the associative matrix is the surface of a n-dimensional ellipsoid. For diagrammatic reasons, Figure 7 is shown as the second-order surface or a three-dimensional ellipsoid as an example.

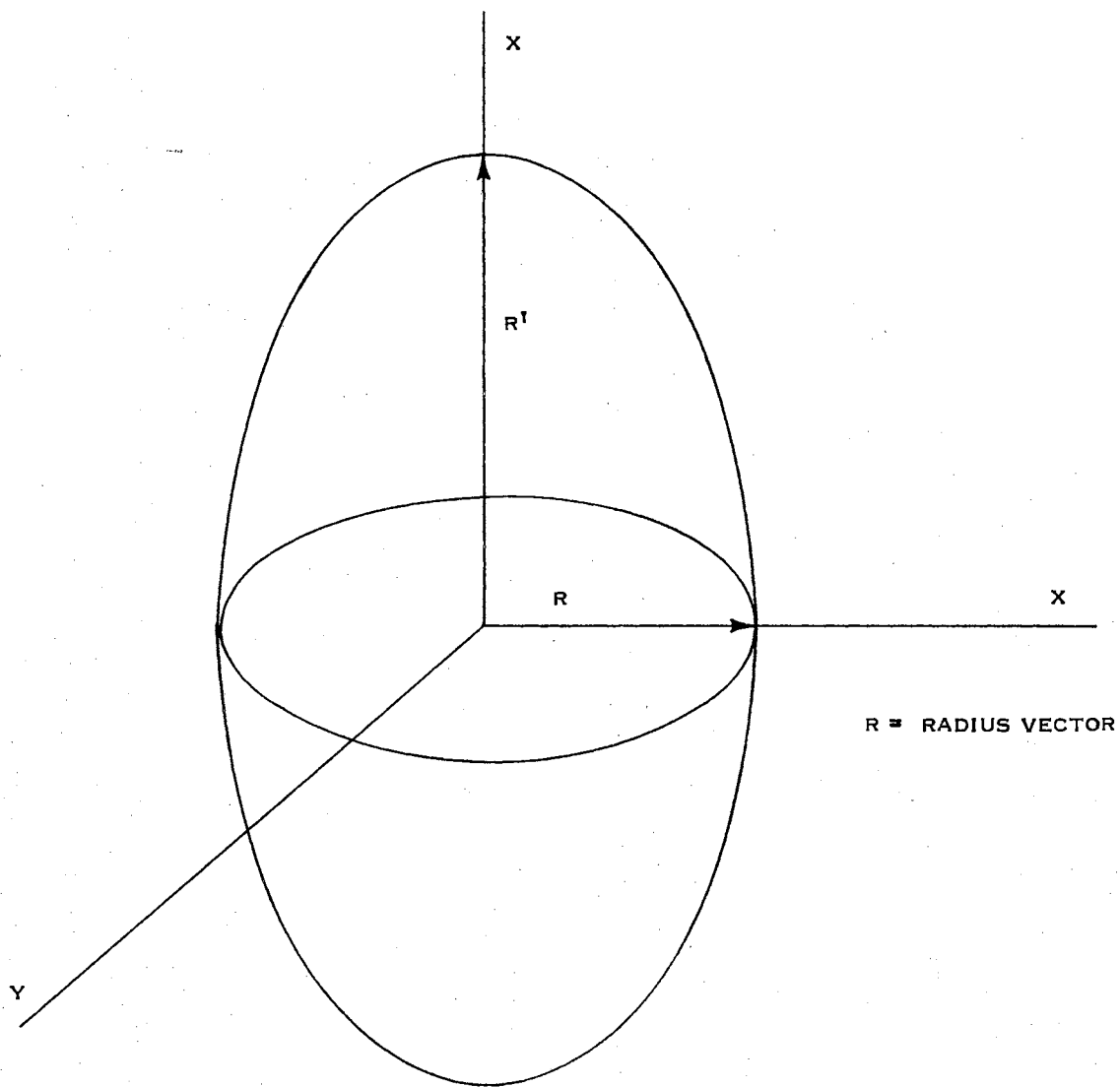


Figure 7. Geometrical Representation of Eigenvalues

Any vector $X = (X_1, X_2, X_3)$ in three dimensions has the significance of the "radius vector" r which connects an arbitrary point on the ellipsoid surface with the surface origin. When the point on the surface is in the position where a tangent to the surface is perpendicular to the radius vector r , it indicates a principal axis at that point.

It can be shown (9) that the eigenvalues λ_j are each the reciprocal of the square of the distance r and, therefore, r is expressed as the reciprocal of the square root of λ ,

$$r = \frac{1}{\sqrt{\lambda}} \quad (4)$$

where r represents a principal axis. Hence, a large eigenvalue means that in the direction of a certain principal axis, the ellipsoid surface is nearer the center of origin and vice versa. Therefore, all eigenvalues of a symmetrical matrix of real numbers are real and not complex.

For the example in Figure 5, the eigenvalues are interpreted as the radius vectors of a second-order surface.

$$\begin{aligned} r_1 &= 1 \\ r_2 &= \frac{1}{\sqrt{3 + \sqrt{2}}} \\ r_3 &= \frac{1}{\sqrt{3 - \sqrt{2}}} \end{aligned}$$

The radius vector r_2 shows to be the principal axis nearest the origin and Group 2 of Table I has the highest correlation with all responses.

The factor analysis technique seeks to derive from word association in representative documents, and automatically generated classification for use in actual indexing. Other techniques using this inter-relationship among words are referenced under the "Theory of Clumps," "Facet Analysis" and others, most of which are modifications of factor analysis.

The following are two of the most promising techniques in automatic indexing, although varying degrees of processing and preliminary term assignment are required.

Latent Class Analysis

The latent class analysis approach of Baker (10) to problems of automatic classification is theoretical rather than experimental in nature. Baker claims that the latent class model developed for the determination of sub-classes among individuals responding to a "Yes/No" questionnaire would have application to the categorization and search of information. The model would be based on a response pattern which indicates the presence or absence of clue words or phrases in processed documents and the analysis yields an ordering ratio which would serve as a weighting factor.

This technique requires key words be assigned to the documents and these be processed against a number of clue words which have been selected to represent the sub-classes. Baker admits, "The determination of appropriate key words to identify each document is a problem of major dimensions for which an adequate solution does not presently exist." Others commenting on the latent class technique

indicate that they feel the effectiveness of an information retrieval system is due more to the appropriateness of the key words than the subsequent processing.

After the key words are selected, it is possible to relate the response pattern of individual documents to the key words. Assuming there are K key words and individual documents that will respond positively or negatively, a K - valued vector of +'s or 0's will be generated.

TABLE II
RESPONSE PATTERN

| | |
|-------|---|
| K_1 | + |
| K_2 | 0 |
| K_3 | + |
| . | . |
| . | . |
| . | . |
| K_k | 0 |

With a given set of key words, it is possible to have 2^k response patterns.

Let the probability of an individual document responding to a key word be \prod_i , where $i = 1, 2, \dots, I$ keywords. The latent class model farther assumes the document population can be divided into m , mutually exclusive, subpopulations (classes) where α denotes a subpopulation.

Next, let V^α be the probability of an individual document being randomly drawn from the α^{th} sub-population, $\alpha = 1, 2, \dots, M$. Also, let X_i^α be the probability of a document from the α^{th} subpopulation responding positively to the i^{th} keyword, where $X_i^{-\alpha} = 1 - X_i^\alpha$ denotes a negative response. The probability that a document drawn from the α^{th} class will match both keyword I and J positively is given by $X_i^\alpha X_j^\alpha$.

Therefore, the probability of obtaining a given response pattern to the keywords is the sum of the products of a probability of belonging to a latent class V^α and the probability of responding positively, X_i^α .

The classes are separated with computer automated instruction and those not directly stated in this topic. The response key words chosen as representative of the class are:

where $i = 1, 2, \dots, I$.

The latent class analysis is fundamentally the problem of solving the above series of accounting equations for the calculated values of V 's and X 's. Matrix methods for solution of the accounting equations is available by Doyle (7). With a m -dimensional matrix, the computational load on the computer is not too severe when the upper limit of two $m \times m$ matrices does not exceed the computer storage. The matrix method gives a solution for the roots of the m^{th} degree determinantal equation.

TABLE III
LATENT STRUCTURE

| Latent Class | Probability of Class | Probability of Response | | | |
|--------------|----------------------|-------------------------|---------|---------|---------|
| | | 1 | 2 | 3 | 4 |
| 1 | v^1 | x_1^1 | x_2^1 | x_3^1 | x_4^1 |
| 2 | v^2 | x_1^2 | x_2^2 | x_3^2 | x_4^2 |

Table II shows the tabulation of estimated probabilities, V for probability of class and X for probability of response. An example of documents which is to be divided into two classes is assumed. The classes are documents dealing with computer automated instruction and those not directly related to this topic. The response key words chosen as representative of the class are:

1. Computer
2. Automated
3. Teaching
4. Devices.

Mr. Baker warns that "The determination of appropriate key words to identify each document is a problem of major dimensions for which an

adequate solution does not presently exist."

Before determining the document responses, the estimated probabilities of the key words in relation to each class.

TABLE IV
ESTIMATED LATENT STRUCTURE

| Latent Class | Probability of Class | Probability of Response | | | |
|--------------|----------------------|-------------------------|----|----|----|
| | | 1 | 2 | 3 | 4 |
| 1 | .6 | .9 | .7 | .7 | .6 |
| 2 | .4 | .3 | .1 | .2 | .1 |

The values in Table III are the probability estimates of the key words responding should a document belong to the latent class. For example, the word "computer" has a high probability ($X_1^1 = .9$) of occurring if the document is pertinent to Class 1 (category of documents dealing with computer automated instruction).

Given a response of,

$$\begin{array}{rcl}
 K_1 & & 0 \\
 K_2 & & + \\
 K_3 & & + \\
 K_4 & & 0
 \end{array}$$

the resulting accounting equation in (4) becomes,

$$\bar{\Pi} = p_1 + p_2 = NV^1 (1-x_1^1) x_2^1 x_3^1 (1-x_4^1) + NW^2 (1-x_1^2) x_2^2 x_3^2 (1-x_4^2). \quad (6)$$

The example latent structure gives the following value.

$$1000 (.6) (1-.9) (.7) (.7) (1-.6) + 1000 (.4) (1-.3) (.1) (.2) (1-.1)$$

or

$$11.76 \quad + \quad 5.04 = 16.80$$

The probability of this response in Class 1 is

$$P_1 = \frac{P_1}{P_1 + P_2} \quad (7)$$

$$P_1 = \frac{11.76}{11.76 + 5.04} = .700$$

It is necessary to choose a threshold value for the probabilities in order to determine in which class a calculated probability is to be assigned.

Association Factor

Stiles (11), whose work in association factors relates to the use of statistical associations between the terms manually assigned to documents, has also considered the same technique for automatic index-

ing and classification.

The general procedure requires that manually indexed files be processed against an expanded list of request terms and followed with a modified chi-square test to grade the relevancy of the retrieved documents. This test requires the co-occurrence frequencies, between each term used in indexing the documents and each request term be determined and used in the following formula (7) for determining the relative frequencies of the co-occurring terms.

$$\text{Log}_{10} \left[\frac{(|f N - AB| - N/2)^2 N}{AB (N - A) (N - B)} \right] = \text{Association Factor} \quad (8)$$

Where A is the number of documents indexed by one term; B is the number of documents indexed by a second term; f is the number of documents indexed by the combination of both terms; and N is the total number of documents in the collection. This formula is a form of the chi-square formula using the marginal values of the 2 x 2 contingency table and the Yates' correction for small samples.

Patterns of term co-occurrence can then be developed in the sense of term-profiles which show, for each term, the more significant of its associational values in pairing with other terms in the collection.

For example, the number of times various terms have occurred with the term "Friction" to index a document were counted and the most frequent were:

TABLE V
TERM FREQUENCIES

| Term | Frequency |
|----------|-----------|
| Theory | 7 |
| Film | 6 |
| Crystal | 5 |
| Metal | 5 |
| Thin | 5 |
| Transfer | 4 |
| Clutch | 3 |
| Wear | 2 |

Although, the word "Theory" occurred the most number of times, it is obvious that "Theory" is no more related to "Friction" than to any other word about which there might be a theory.

To determine the relative frequency, the above formulation is applied to each of the term paired with "Friction" and the list becomes as follows:

TABLE VI
ASSOCIATION FACTOR EXAMPLE

| Term | f | A | B | Association Factor |
|-------------|---|----|----|--------------------|
| Wear | 2 | 4 | 25 | 3.35 |
| Thin | 5 | 49 | 25 | 3.31 |
| Lubrication | 2 | 9 | 25 | 3.00 |
| Belt | 1 | 2 | 25 | 2.70 |
| Theory | 7 | 2 | 25 | Neg. |

For example, substituting the conditions of the word "Wear" into Equation 7 and assuming $N = 10^5$, the calculation is as follows:

$$\text{Log}_{10} \left[\frac{\left(\left| 2 \times 10^5 - 4 \times 25 \right| - \frac{10^5}{2} \right)^2}{4 \times 25 (10^5 - 4) (10^5 - 25)} \right]^N$$

$$\text{Log}_{10} (4009) = 3.35$$

Now, the word "Theory" has dropped much lower and the word "Wear" has risen to the top even though it occurred only twice in association with "Friction." Anyone interested in friction would probably be interested in the documents indexed by "Wear" and those indexed under "Lubrication."

The Association Factor analysis by computers has one serious drawback, the relatively small matrix size which can be accommodated in the present computers. The IBM 7090 can work with a 1200 x 1200 association matrix but even this sorely limits the vocabulary and the document file. New matrix techniques and larger computer storage may provide new applications for the association factor technique.

Summary of Indexing

The techniques of automatic indexing presented in this chapter are not, in themselves, solutions to the problem of mechanized indexing. The author is convinced that this problem will be solved, possibly using an expanded version of one of the above techniques. From the present state-of-the-art, the solution will require either a breakthrough in computer equipment with extremely large and faster memories or a new technique which will effectively extract and compile more accurate indexes.

The author's IR system (Chapter V) uses the generally accepted form of manual indexing and allowing the use of a flexible vocabulary, a concept first advanced by Luhn (4).

CHAPTER III

STORAGE

In computer information retrieval, the storage of information is the major factor in developing an operational system. The speed with which a retrieval system can access the stored information is usually the determining factor in the success or failure of the system. This chapter discusses the techniques and procedures which affect computerized storage.

File Organization

Assuming, the document to be stored is now represented by a reference number and a list of keywords (terms), there are only two basic methods of organizing a file.

1. Look-up file - a typical file in which the keywords are stored under the reference numbers.
2. Inverted file - a file in which all the reference numbers of documents containing a particular term are stored under that term.

Advantages of the look-up file are mainly due to the cohesiveness of a logical set, while those of the inverted file are mainly due to the search strategies of term-oriented retrieval techniques.

It is the opinion of this writer that the additional advantages

in computer processing offered by the inverted file outweigh the look-up file. Some of these advantages will be discussed in the Statistical Information Retrieval (SIR) system description (Chapter V).

Storage in a computer system is usually limited to magnetic tape (or drum) and random-access core memory. The core-type storage is very limited; therefore, most IR systems require magnetic tape storage (a technique for IR storage and retrieval on random-access disc storage is discussed in the section titled "Predestined Storage"). Storage on tape is of the sequential-type and the file organization must reflect the search-strategy used. An example of the tape arrangement for an inverted file is given in Figure 1.

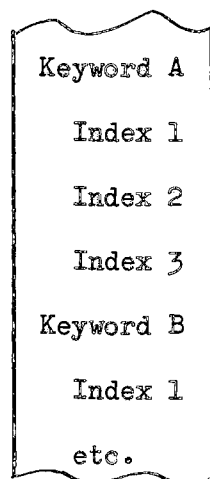


Figure 8. Magnetic Tape Format
for an Inverted
File

The greatest saving in computer run times are found in the file organization on tape and search-strategy used in the system. The best

configuration with an inverted file is to arrange both the keywords on tape and the keywords in the inquiry in the same order. This arrangement allows only the first inquiry keyword to be used until a comparison is made or it exceeds the arrangement order, then the second inquiry keyword is used for the next comparison without having to backspace the tape. This technique allows a complete search in one pass.

With tape file, another great savings in computer time is accomplished by compressing the stored information, as much as practical, and still obtain the desired results. Methods vary, but most involve some type of coding. The code can be the standard BCD² representation for alpha-numeric symbols or more complex telegraphic codes. Another technique developed for the IR system used in this thesis uses an algorithm which automatically creates a "folded-word sum" code.

Folded-Word Sum Algorithm

The development of the folded-word sum algorithm began with the realization that an IR system did not need to maintain the index terms on file in a legible form. The only necessary condition is the same code must be generated each time the identical keyword term is introduced in the algorithm. The code generated is sufficient for locating the same term in storage.

Words and phrases encountered in an IR system can vary and, in technical literature, may exceed 30 or 40 alphabetical characters.

²BCD - Binary-coded decimal is a standard code representing alpha-numeric characters and used throughout the computer industry (for example T is 63 octal or 110011 binary).

Words of this length would require 6 or 7 machine word³ expressed in BCD code. Not only would words of this length require excessive processing time, but would occupy considerably more space on magnetic tape. This problem is solved with a programmed algorithm which simply divides the word into six BCD-character lengths, and folds the lengths into an accumulative sum of 36-bits or one machine word. This sum is termed a "folded-word sum."

The folded-word sum technique is totally a concept of the author and no reference to this type of compression and storage of information has been found to date.

A detailed example of this algorithm is given in Chapter V.

Storage and retrieval are closely related or interrelated. In order to minimize search time during the retrieval of stored information, it is logical to attempt to store information in a known or prescribed location. When using storage facilities which provide for random access⁴, it is desirable to go directly to a pertinent concept or area of interest, rather than search through the entire file, starting at the first. This technique of direct access can be extended to any element of a file, as is practical and economical.

³Machine word is a term used to describe the basic internal word-length acceptable to the computer. The IBM 7094 word-length is 36 binary bits or 6 BCD digits.

⁴Random access refers to a storage ability which allows the system to directly access any stored information provided the known address or location symbol.

The new equipment, with the advent of disc memory⁵, provide large, random-access storage. The following section is a mathematical procedure and algorithm for the "predestined storage" of information.

Predestined Storage

In this section, the author expands the concept of multidimensional addressing by Hellerman (12) and converts the technique for use in predestined storage of information. The vector mathematics are from Hellerman (12) and the coded term and storage modulation are the author's concepts for predicting storage locations from the generated term codes.

Assuming, for the present, an element of information is given in coded form for storage in the random-access memory of a computer. The objective of this procedure is to store the coded term in a location which is prescribed by the term itself and, then, re-locate the storage position and make it available to the system.

Coded Term 020102

To locate, the coded term is partitioned into n separate integers.

$n = 3$ (02,01,02)

Now, consider these values as the indexes of an n -dimensional ($n = 3$) array denoted by A , an element of which is denoted by A_1 .

⁵Disc memory is a bank of mechanically rotated plates on which electromagnetic spots are used for storing numerically coded data. Read "heads" riding on each disc allows direct access to the stored data.

Here \underline{i} is a vector of indexes whose n elements are $i_0, i_1, \dots, i_{(n-1)}$.

Therefore, from the coded term:

$$i_0 = 02 \quad i_1 = 01 \quad i_2 = 02$$

and the vector,

$$\underline{i} = (02, 01, 02)$$

For a n -dimensional array A , a vector \underline{v} may be defined, each of whose elements is the largest number plus one which the corresponding element of \underline{i} can assume. Thus, the j th element of \underline{v} gives the number of units in the j th coordinate of the array. For example, the elements of \underline{i} are considered the largest, thus:

$$\underline{v} = (03, 02, 03)$$

In order to project the vector of array A into the linear array of addressed-memory, a vector \underline{s} is assumed which contains the elements of A . To insure uniqueness, a convention is adopted in which \underline{s} consists of a particular ordering of the elements of A wherein successive values of i_{n-1} correspond to successive values of \underline{a} , an index over \underline{s} . In general, successive values of i_k correspond to increments in \underline{a} of

$$\prod_{j=K+1}^{n-1} v_j \quad \text{for } K < n-1. \quad (8)$$

The dimension of \underline{s} is given by the product sum of \underline{v} in Equation (9).

$$(s)_d = \prod_{j=0}^{n-1} v_j \quad (9)$$

$$(s)_d = (3) (2) (3) = 18 \text{ elements}$$

The problem is: Given an array A, obtain for any element specified by its index vector \underline{i} , the single index value \underline{a} , such that

$$S_a = A^{\underline{i}}$$

where \underline{a} can be considered a storage address.

With the above parameters, it can be shown from similar work by Hellerman (12) that \underline{a} can be formulated (10) as:

$$a = \left[\sum_{j=0}^{n-2} i_j \left(\prod_{k=j}^{n-2} v_{k+1} \right) + i_{n-1} \right] \quad (10)$$

where n = dimension of storage vector

j = elements of the vector \underline{v} corresponding to elements of \underline{i} .

k = index of \underline{i} to correspond with increments in \underline{a} .

Evaluating the equation (10) for $n = 3$, which is sufficient for information storage systems, gives the location \underline{a} in terms of vector \underline{i} and \underline{v} :

$$a = i_0 v_1 v_2 + i_1 v_2 + i_2 \quad (11)$$

To find an address \underline{a} from a vector \underline{i} requires $(n-1)$ multiplications and $(n-1)$ additions. This does not count the evaluation of

$$\prod_{k=j}^{n-2} v_{k+1},$$

since it must be evaluated only once for any one dimension. From the example, where:

$$i = (02, 02, 02)$$

$$v = (02, 02, 03)$$

$$n = 3$$

Equation (10) gives:

$$\begin{aligned} a &= (02) (02) (03) + (01) (03) + (02) \\ &= 12 + 3 + 2 = 17 \end{aligned}$$

This is the maximum as limited by the example, and Table VII contains the values of a for the coordinates less than the maximum

$$i = (2, 1, 2) .$$

TABLE VII

VECTOR - LOCATION EXAMPLES

| i | a | i | a | i | a |
|-----|---|-----|----|-----|----|
| 000 | 0 | 100 | 6 | 200 | 12 |
| 001 | 1 | 101 | 7 | 201 | 13 |
| 002 | 2 | 102 | 8 | 202 | 14 |
| 010 | 3 | 110 | 9 | 210 | 15 |
| 011 | 4 | 111 | 10 | 211 | 16 |
| 012 | 5 | 112 | 11 | 212 | 17 |

Using the "folded-word sum" algorithm with the above storage technique would give access to a position containing the "indirect address" of the required information. But, in practice, this technique has a problem -- sums generated by the folded-word algorithm are not in sequence, thus giving a large array with many unused positions. To more efficiently employ this technique on the computer, a modulation procedure is required which will compress the storage positions in the array. For example, the array ($n = 2$) of maximum vector ($v = 5, 5$) would store a number ($i = 5, 3$) as shown in Figure 9:

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | Z | | |
| 4 | | | Y | | |
| 5 | | | X | | |

X is position
of (5,3)

Figure 9. Modulated Array

To modulate this number by (5) would cause it to be stored in (4, 3) as shown by Y in Figure 9. This modulation procedure would be indexed, one count for each modulation, until an occupied position was

found and the index stored for the last unoccupied position. The procedure would limit the unused positions in an array to a minimum and allow a limited memory to handle very large arrays. Retrieval procedures would also be needed to compare the position and its index.

The concept of predestined storage has not been completely evaluated in the field of Information Retrieval, but the author feels this concept offers some significant advantages for IR system on random-access equipment.

CHAPTER V

RETRIEVAL

The retrieval of information can be divided into two separate areas, the retrieval search strategy and the retrieval logic. The following discussion of these areas is centered around the "coordinate concept" technique of information retrieval.

Coordinate Concept

Most computer retrieval systems used the keyword concept approach to information retrieval. Some refer to this approach as "coordinate concepts" and others as "conceptual convergence," but all involve keywords either extracted from the documents or selected from a prepared list (vocabulary).

In a coordinate concept system, the document index can be considered to be stored in an n-dimensional space with the coordinate of its position being the "keywords" or "concepts." For example, a document titled "Physical Properties of Polyethylene" may have the following keywords:

polyethylene

tensile strength

melt index.

Looking at 3-dimensional space, its positional coordinates can be diagrammed as shown in Figure 10.

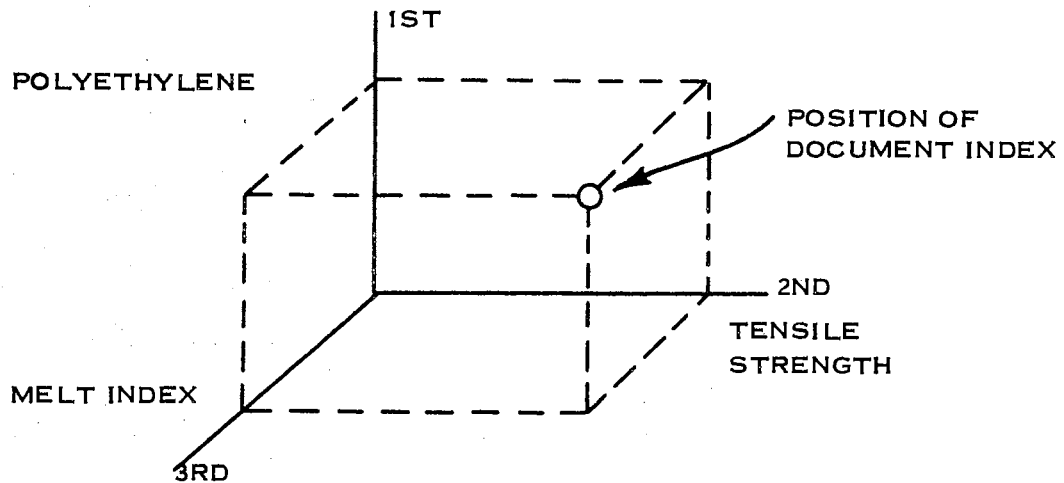


Figure 10. Coordinate Concepts

Thus, the coordinates give the conceptual content of different documents uniquely different storage positions.

Figure 10 is an analogy for the discussion of coordinate concepts. The actual first-step retrieval procedure in computer systems is

matching keyword concepts of an inquiry with the keyword concepts of stored information.

Retrieval Search Strategy

The initial search for matching keyword concepts is the most varied and dependent procedures of a retrieval system. The search depends on file organization, type of indexing, retrieval technique and the equipment to be used. For the purpose of this thesis, the discussion will be limited to search strategies used in computer systems.

In a computer system, the search strategy is all important, since the time required to match inquiries is directly effected by the amount of stored information which must be processed. To have an efficient computer retrieval, the search time must be kept to a minimum. Most current computer IR systems use magnetic tape, and, since tape slows the speed of computers, the tape operations are the key to an efficient system. For example, in file organization, the inverted file offers a distinct advantage in sequential tape storage because the file keywords can be stored in a particular order and the inquiry keywords in the same order, thus allowing the search to terminate after the last keyword is found or exceeded. This procedure requires that the tape only be processed one time. In another example, the information stored may only be indexed with a fixed vocabulary of limited terms and it may be more efficient to store the information on tape in order of use-frequency; i.e., the most used keywords occurring first on the tape. In random-access storage, the search strategy takes on a new set of conditions as discussed in Chapter III.

Retrieval Logic

Information retrieval logic is applicable to Boolean symbolism. For example, since the document reference number is stored under each of the keywords used to describe its content, the document will be available for retrieval at multiple-levels of "pertinency."

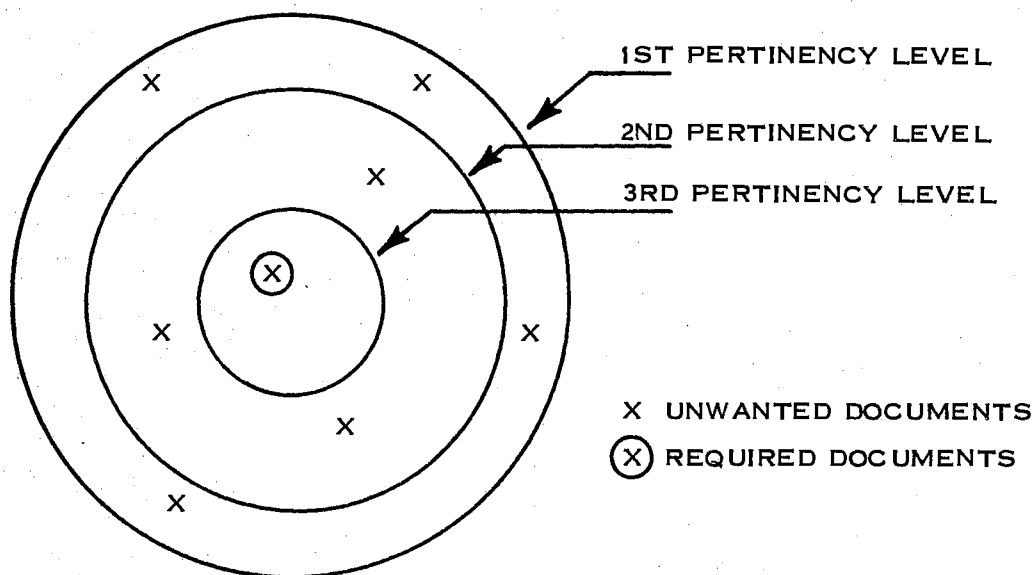


Figure 11. Pertinency Levels

Using the earlier example on polyethylene, all documents within the first pertinency level would have "polyethylene" as a keyword. The

second pertinency contains only those documents having both "polyethylene" and "tensile strength." The third pertinency level contains the document or documents having all three keywords. Figure 11 shows the most pertinent document as inclusive in the inner pertinency level, but this does not reduce the pertinency of the other documents. They would be ranked and shown at their proper pertinency level.

The basic logic used in any retrieval system is the "and" logic. For example, the inquirer might want to retrieve all documents containing the concepts, "X and Y." This logic is usually accomplished by the simple matching or comparing of the keywords in the document indexes with those requested by the inquirer. Therefore, the first retrieval operation is to compare the request keyword concepts with the keywords in the file, and when a match is obtained, the document indexes under the keyword are stored as vectors in the computer memory for further processing. After a set of vectors (termed the "keyword matrix") has been generated by matching keywords, the matrix forms the basis for the remaining logic.

Search Keywords

| | KW ₁ | KW ₂ | KW ₃ |
|---------------------|-----------------|-----------------|-----------------|
| Document Indexes | I ₁ | I ₂ | I ₁ |
| | I ₃ | I ₃ | I ₂ |
| | I ₅ | | I ₃ |

Figure 12. Keyword Matrix

Figure 12 is a representation of the Keyword Matrix as it may appear in the computer memory. Assuming only three search keywords and five document indexes, the matrix indicates as a result of the "and" logic that documents I_1 , I_3 and I_5 contain the keyword (KW_1). The matching keywords for the other documents are shown. A simple search of this relatively small matrix (internal searching in a large-scale computer is extremely fast and a matrix of 20,000+ positions can be processed in a fraction of one minute) will give the number of keywords matches for each document:

| <u>Index</u> | <u>Matches</u> |
|--------------|----------------|
| I_1 | 2 |
| I_2 | 2 |
| I_3 | 3 |
| I_4 | 0 |
| I_5 | 1 |

The third index has the most matches and would be ranked as the most pertinent of the documents. The index I_4 is not shown in Figure 10 and is not considered pertinent to the inquiry.

If more sophisticated logic is required, it can be accomplished by manipulating the keyword matrix. For example, if the inquiry specified that KW_2 is a synonym of KW_1 , the system puts the indexes in vector for KW_2 under KW_1 and considers the keywords as one. The synonym and the "or" logic are the same in all respects.

Another possible logic which is accomplished in the keyword matrix is termed "except" logic. The inquiry is conditioned to retrieve

documents when they are indexed by X, but only if they do not contain Y. If the keyword KW_2 in Figure 10 is tagged with an "except" code, this causes the example to retrieve documents containing KW_1 and KW_3 except those containing KW_2 . The system processes the indexes under KW_2 against all the other indexes and eliminates any that match. With these conditions, only I_1 and I_5 remain eligible for retrieval.

The logic termed "necessary" is the counter-logic of the "except." In the example, if KW_2 is tagged as "necessary," the system checks the other indexes and only those which also occur under KW_2 . This logic is very important in the detailed files where it allows more complex inquiries to be processed. The "necessary" conditions severely limits the file and causes the remaining conditions to be more selective.

Summary of Retrieval

The logic of retrieval is well established and the techniques shown in the preceding sections are used in several IR systems (13). Important to the system efficiency, are the procedures that the designer chooses to accomplish the desired logic. The procedures must take advantage of the computer equipment, the type of indexing and the resulting output.

The procedures discussed in this chapter are those of the author and were designed for use on an IBM 7094 computer system.

CHAPTER V

THE SIR SYSTEM

In this chapter, an existing information retrieval system is described and sample results are shown. The Statistical Information Retrieval (SIR) system was developed by the author for the Research and Development Department of Phillips Petroleum Company and has proved to be a successful production system the past three years. The system concept and programming specifications were initiated in October, 1961, with the first functioning program running pilot tests in March, 1962. A series of modifications were made, including multiple-search capabilities and an edit routine.

System Design

The SIR program is designed as a general-purpose information retrieval system capable of storing and retrieving any information index which can be expressed in key concepts or keywords. The system is mainly for document retrieval, but it can also be used on keyword indexes derived from engineering drawings or other classifiable items.

The flow diagram in Figure 13 shows the general flow of the information in the computer system. Two separate, but interrelated paths are formed -- storage processing and retrieval processing.

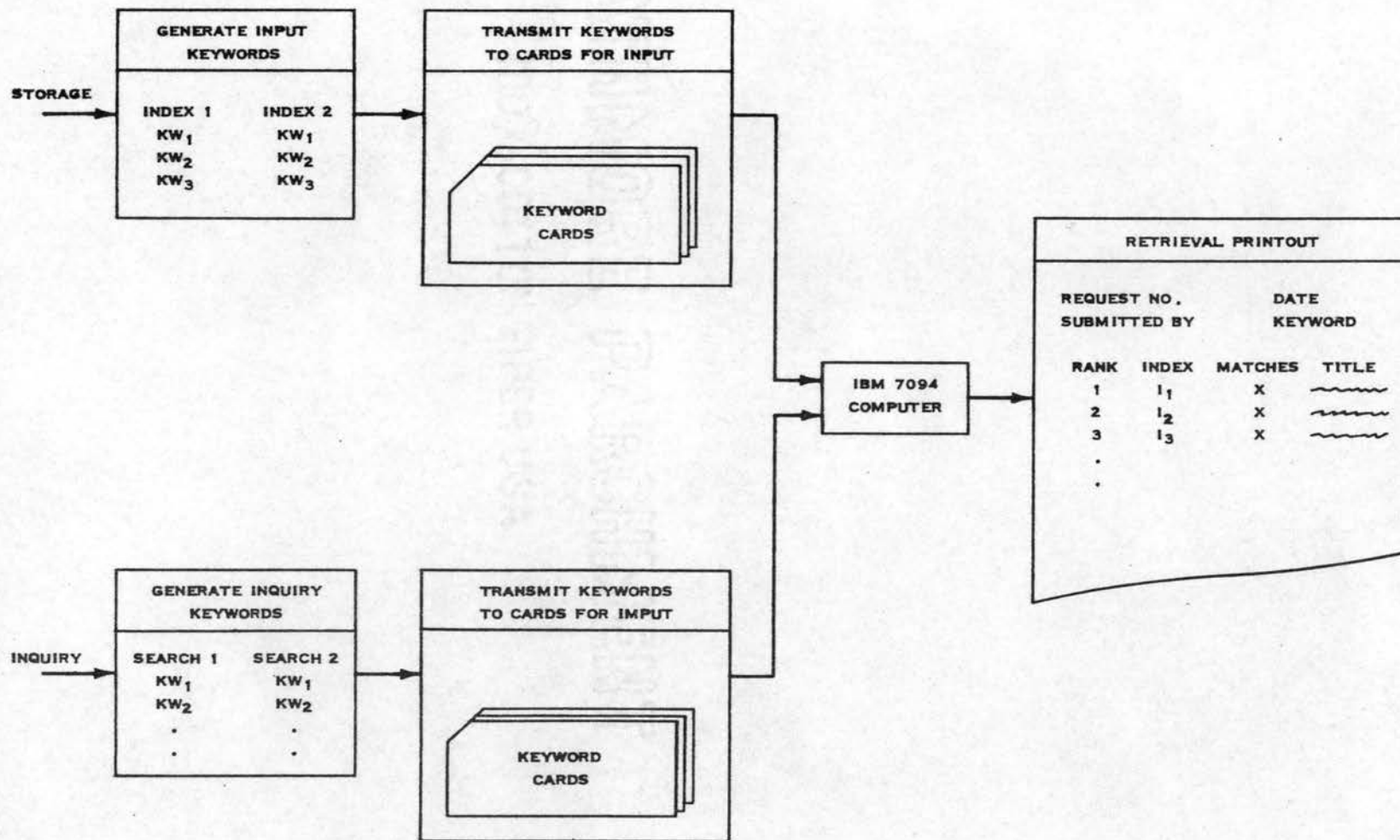


Figure 13. SIR System Flow Diagram

Storage Processing

The keyword indexes of information to be filed are compiled on data-forms (Figure 14) by the indexer. Using the simple example of a document titled "Physical Properties of Polyethylene," the title is printed in space provided after the asterisk in Column 19. The keywords "polyethylene," "tensile strength" and "melt index" follow, always starting in column 19 of the punched card. The role indicator "H" assigned "polyethylene" marks this keyword as the major topic and the indicator "G" on the other keywords qualifies them as physical properties.

At the top of the page, the index number, date and division are assigned. The index or reference number must be punched in the first 18 columns of each card, relating it to the item being stored.

After the storage data-forms are keypunched, the cards are batched in 5000 cards or less storage runs. These are transmitted to magnetic tape and processed with the storage routine. The routine performs the following functions:

1. Before the keywords are processed, the title and index number are written directly on an intermediate title tape. These are accumulated for several batches and then sorted onto an up-dated title tape in order of the index numbers. An option on the retrieval routine uses this tape to print the titles for documents retrieved.
2. Each keyword (which is now represented in the computer in BCD code) is processed through the folded-word sum

SUBMITTED BY: C. PARSONS

DATE: 6-10-66

| | | | FILE | REPORT | FOR S | YEAR | DIV | BLANKS | THE FIRST 18 SPACES ON EACH CARD | | | | | | | |
|--------------|----------------------------|-----------|-----------|--------|-------|------|-----|--------|-------------------------------------|--|--|--|--|--|--|--|
| INDEX: | | P 123R65R | | | | | | | | | | | | | | |
| TITLE: | 1 | 18 | PROG. NO. | | | | | | | | | | | | | |
| * PHYSICAL | PROPERTIES OF POLYETHYLENE | | | | | | | | | | | | | | | |
| * | | | | | | | | | | | | | | | | |
| * | | | | | | | | | | | | | | | | |
| POLYETHYLENE | H | | | | | | | | | | | | | | | |
| TENSILE | STRENGTH G | | | | | | | | | | | | | | | |
| MELT | INDEX G | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

19

76

80

* TO BE PUNCHED IN 19-COLUMN OF EACH TITLE CARD.

FIGURE 14. STORAGE DATA SHEET

algorithm. The keyword is broken into six-letter increments and the logical sum of the increments is used to represent the code for that keyword. For example:

BUTANE DEHYDROGENATION

First, all unnecessary symbols such as spaces between words, commas, periods etc. are eliminated.

BUTANEDEHYDROGENATION

Next, the term is chopped into six-letter increments and stacked.

BUTANE

DEHYDR

OGENAT

ION

The BCD code for the word appears as follows:

| | | | | | | |
|-------|----|----|----|----|----|---------------------|
| 22 | 64 | 63 | 21 | 45 | 25 | |
| 24 | 25 | 30 | 70 | 24 | 51 | |
| 46 | 27 | 25 | 45 | 21 | 63 | |
| 31 | 46 | 45 | | | | |
| <hr/> | | | | | | |
| 35 | 66 | 75 | 56 | 02 | 51 | Folded-word sum. |

The BCD increments are logically summed into the above octal code. The computer actually has this sum as a 36-bit binary machine word.

The folded-word sum is unique for all words 6 characters

or less in length. For longer words, the sum is near-unique. The probability of two different words generating the same sum is approximately 1 in over 34 billion (since the word sum is a near-random generation of a binary number of 2^{36} , the actual probability approaches 1 in 2^{36} or 68 billion). Due to the relatively low probability with seven-character words, the author chooses to reduce the probability ratio to 1 in 34 billion. To date, the system has not encountered any two words having the same sum.

3. After all the folded-word sums are generated, the sums and their related index numbers are sorted into numerical order by the keyword sums. Since the magnetic tape file is also filed in numeric order, the keyword sums to be stored can be processed one at a time and, consequently, do not require backspacing.
4. To actually place the information on tape, the first storage keyword (the smallest keyword sum) is compared with the keyword sums from the tape file. This procedure continues until the keyword sum to be stored matches or is exceeded by the keyword sums from the tape. If there is a match, the index number of the keyword to be stored is written on the tape in the matching keyword sum. If the keyword sum to be stored is exceeded without a match, the keyword sum is inserted and its index number is placed under the sum. This procedure creates the inverted file on tape.

5. After the new up-dated keyword and title tapes have been generated, the magnetic tapes are placed in the tape library. The current tape and the two preceding tapes are kept as back-up for this file. The cards are also placed in permanent storage. These precautions are taken because of the effort and expense required to generate the large volume of information stored in this type of file.

Retrieval

An inquiry for information can originate in numerous places, but the SIR system requires all inquiries to be processed through specialized personnel. Figure 15 shows the general path an inquiry might take, starting with the original request and ending with his answer.

An example inquiry may be stated, "What are the effects of antioxidant level, specifically AO 2246, on the physical properties of SBR 1500 type polymers?" Personnel then choose the keywords which best express this inquiry and put the words on the retrieval data form in Figure 16. This data form has as the first card the control data for the retrieval. Shown on this header card are:

1. The individual or group making the request.
2. The person who prepared the inquiry.
3. The code that indicates the particular file to be searched.
4. The limit of the search (the year back to which the search is required).

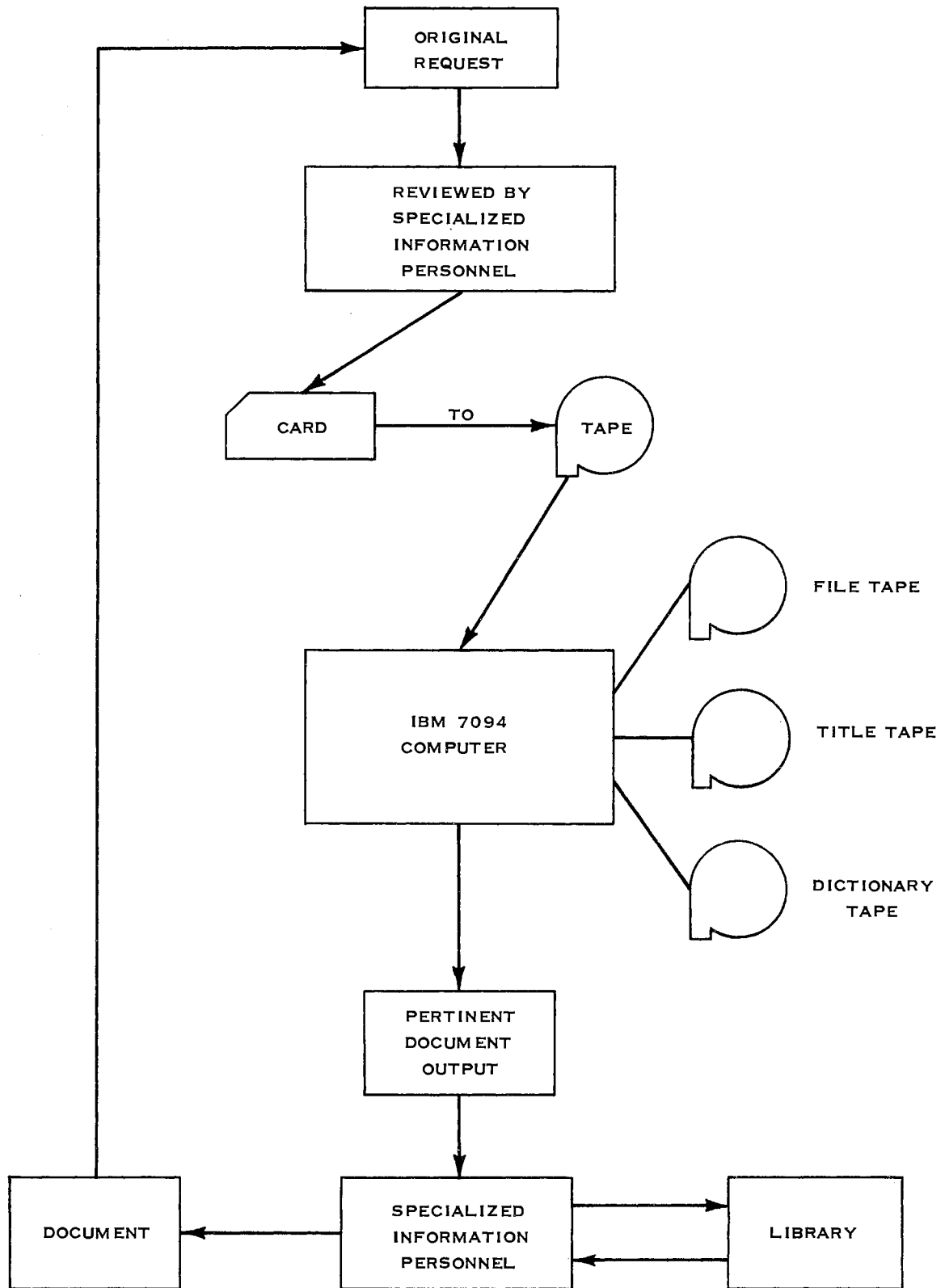


Figure 15. Retrieval Procedures

| HEADER CARD: | | | | | | | | |
|--------------|---------------------------|---------------------|--------------|----------------------|--------------------|----------|------------------|-----------|
| REQUEST BY | SUBMITTED BY | FILE TO BE SEARCHED | BACK TO YEAR | TOTAL WORDS SEARCHED | EFFECTIVE KEYWORDS | SYNONYMS | KEYWORDS MINIMUM | TITLE OP. |
| RES 2 | MAW | P 55 | 16 | 11 | 5 | 3 | 1 | |
| 19 | 24 | 29 | 33 | 35 | 38 | 41 | 44 | 47 |
| COLUMN 1 | KEYWORDS | | | | | | | PROG. NO. |
| | AØ 2246 K | | | | | | | |
| = | CYANØX SS K | | | | | | | |
| | AØ 2246 D | | | | | | | |
| = | CYANØX SS D | | | | | | | |
| | AØ 2246 LEVEL F | | | | | | | |
| = | AØ 2246 CØNCENTRATIØN F | | | | | | | |
| = | CYANØX SS CØNCENTRATIØN F | | | | | | | |
| | ANTIØXIDANT LEVEL F | | | | | | | |
| | ANTIØXIDANT H | | | | | | | |
| | MØDULUS G | | | | | | | |
| | TENSILE G | | | | | | | |
| = | TENSILE STRENGTH G | | | | | | | |
| | ELØNGATIØN G | | | | | | | |
| | SBR 1500 I | | | | | | | |
| | BUTADIENE CØPØLYMER K | | | | | | | |
| | STYRENE CØPØLYMER K | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

1 19 76 80

COLUMN 1: = FOR SYNONYM
 + FOR NECESSARY

FIGURE 16. RETRIEVAL DATA SHEET

5. The total number of keywords to be searched.
6. The number of effective keywords remaining without synonyms.
7. The number of synonyms to be searched.
8. The minimum number of keyword matches required for retrieval.
9. The indicator for the title option on those document retrieved.

Under the header card, the remaining cards are punched with the keywords chosen to describe the inquiry. The first column of each of these cards is reserved to tag the words with the appropriate logic. A blank in this column allows the "and" logic, but the equal sign indicate that the word is a synonym of the keyword preceding. The possible logical indicators used in this column are:

- synonym or "or" logic
- + necessary
- except

The cards generated from this retrieval data-form must be kept in sequence, particularly for synonym keywords.

The retrieval program has the function of passing the file on magnetic tape against the inquiry keywords and print out a list of document numbers the contents of which are pertinent to the inquiry. The processing step for the retrieval program are the following:

1. After the above input has been read into the computer, the program interprets the data on the header card,

checking the number of keywords and under which optional mode the program is to be run.

2. The first column of each keyword is checked and the appropriate logic assigned the keyword for later processing.
3. The keywords are now processed through the algorithm to generate the folded-word sum. This algorithm is the same as the one used in the storage routine and the word-sum generated here is the same as was generated in the storage routine, if the word exists in the file.
4. Next, the keyword sums are arranged in decending numerical order.
5. The keyword sums are passed against the tape, the smallest sum first, seeking a matching keyword sum. When the file word sum exceeds the inquiry sum, the program switches to the next word and continues the search from that point. If the program finds a match, the vector of document indexes following the keyword is transferred to the core storage in the computer.
6. After the last inquiry sum has been processed (the program considers the tape search complete when the last word-sum has been matched or the sums on file have exceeded the inquiry sum), the keyword matrix which was generated by the matching keywords in processed. First, the input of inquiry keywords are checked for logical tags and the appropriate manipulations are performed to accomplish the logic. The

remaining vectors are searched to determine the number of keyword matches for each document index and these matches are used to rank the documents in order of the pertinency to the inquiry.

7. The program now tests the indicator on the control card to determine if the title option of the retrieval is desired, if so, the index numbers of the documents retrieved are processed against the title tape and the first fifty or less document numbers have the titles printed out beside them.
8. An example of the printed output is shown in Figure 17.

The output listing of pertinent documents is returned to the personnel submitting the inquiry. These documents are pulled from the library and copies sent to the person or group who originated the inquiry.

REQUESTED BY - RES2
KEYWORDS SEARCHED- 16

SUBMITTED BY - MAW
SYNONYMS - 5
KEYWORDS MINIMUM- 3

FILE SEARCH - P
EFFECTIVE KW - 11

| RANK | INDEX | # KEYWORDS | |
|------|------------|------------|--|
| 1 | PO2840 61R | 7 | STUDY OF EFFECTS OF ANTIOXIDANT LEVEL |
| 2 | PO2612 60R | 5 | EVALUATION OF AO2246 ANTIOXIDANT |
| 3 | PO2949 61R | 4 | OIL - BLACK MASTERBATCH FOR TIRE TEST |
| 4 | PO2524 60R | 4 | EFFECT OF DRYING METHOD ON PHYSICAL PROPERTIES |
| 5 | PO2414 59R | 3 | ETC.* |
| 6 | PO2502 59R | 3 | |
| 7 | PO2512 59R | 3 | |
| 8 | PO2834 61R | 3 | |
| 9 | PO2530 60R | 3 | |

* REPORTS WOULD BE LISTED UNTIL PERTINENCY FELL BELOW THE KEYWORD MINIMUM.

Figure 17. Output Format

CHAPTER VI

TESTS, CONCLUSIONS, AND RECOMMENDATIONS

An information retrieval system can be evaluated by use, but for obvious reasons the test should be made on relatively small samples which can be manually checked. Since the results, and thereby the conclusions, are only as valid as the tests, the results from the following tests have limited validity.

Tests

A pilot test of 180 documents was devised to evaluate both the indexing technique and the computer storage and retrieval system. The documents were prepared using two different techniques with and without role indicators. Sample inquiries were submitted on the material indexed and the output compared with the documents which were manually retrieved from the original documents.

Using the pilot test results, recall and relevancy calculations were made. These calculations are made using the defined relationships developed at Western Reserve University's Documentation Center.

$$\text{Recall} = \frac{\text{No. of documents retrieved}}{\text{No. of documents on file}}$$

(12)

$$\text{Relevancy} = \frac{\text{No. of documents pertinent}}{\text{No. of documents retrieve}}$$

The data shown in Table VIII represents the test averages.

TABLE VIII
TEST RESULTS

| | <u>1st Runs</u> | <u>Including Reruns</u> | <u>With Role Indicators</u> |
|-----------|-----------------|-------------------------|-----------------------------|
| Recall | 88% | 98% | 98% |
| Relevancy | 64% | 72% | 84% |

Actually, the recall in Table VIII cannot be directly compared with the recall of other IR systems. The recall of the Statistical Information Retrieval system depends on two factors, the accuracy of indexing and the specified minimum number of matches required for retrieval. All systems are susceptible to indexing errors, but the "minimum number of matches" criterion may cause some related documents to be dropped from the ranking.

Another factor, which probably had some effect on both recall and relevancy, was the research data used in the tests. This data required greater indexing depth than other types of data (the number of keyword descriptors in these tests was approximately 65 keywords per document) thus enabling retrieval of topical information from the paragraph level.

The relevancy calculations from Formula 12 are also not strictly comparable with other systems. All the documents, which meet the "minimum number of matches" specification, are ranked and shown as output. However, only the first six documents are used, thus pre-setting the boundary conditions in the relevancy calculations.

Results from tests including role indicators in Table VIII show some increase in relevancy, but most of this increase can be attributed to the greater detail possible using indicators.

Conclusions

It can be concluded that the SIR system described in this thesis is a success. This conclusion is based on three years of practical experience with an operational (Phillips Petroleum Company) system. However, it is not implied that the described system is optimal. Considerable additional work and developments are required in all areas of information retrieval. Some of this work is described in the following conclusions and recommendations.

Indexing

Manual indexing, to date, is the only technique accurate for indexing research documents in depth. Experiments in automatic indexing (Chapter II) are not sufficient for this type of documentation. As improved techniques are developed, automatic indexing will replace manual techniques, penetrating first the simpler indexing areas and gradually including detailed indexing (note that developments in automatic indexing will probably parallel similar developments in automatic translation techniques).

The use of role indicators in manual indexing allows the greater specification and, thus, more accurate indexing. For indexing in less depth, the role indicator may become an unnecessary complication.

Indexing with a flexible vocabulary of keywords extracted for the document reduces indexing requirements and provides for accurate

retrieval on a statistical basis. The retrieval test data indicates a favorable comparison with similar systems using a fixed vocabulary.

Storage

The SIR system's extremely fast processing time, which is dependent on efficient storage, indicates the importance of file organization and data compression to information retrieval systems. The author estimates that the folded-word sum concept and file arrangement saved the SIR system 6 times the processing required for a similar straightforward system. It is only through these efficiencies in storage that it becomes economical to process information retrieval systems on large-scale computers.

Retrieval

The statistical nature of information retrieval has been verified by the SIR system. From the results of three years of useful production, it can be concluded that the keywords used to describe an inquiry and keywords used to index a pertinent document have a high probability of being the same keywords.

Recommendations

This thesis, and the information retrieval system described, are addressed to the problem of the "Information Explosion" outlined in Chapter I. The author, realizing the immense scope of the information problem, offers the SIR system as only a partial solution. To cope with the complete problem, systems and equipment must be developed to improve all levels on information processing and retrieval.

Indexing

Automatic indexing offers the only hope for processing the voluminous amounts of information generated by this civilization. Fortunately, this fact is known and considerable effort is in progress in this area. The author's recommendations in this area are:

1. Develop equipment and techniques which will automatically index written material.
2. Establish procedures which will initiate indexing of documents at their source, i.e., the author-publisher automatically extract index terms as the document is printed.
3. Encourage standardization of indexing in each interest area.
4. Establish service organizations which index written material in each interest area and provide companies (institutions and individuals) the opportunity for purchasing the reference services, thus eliminating unnecessary duplication in indexing.

Storage

Large, random-access memories are the key to most information storage problems, but, regardless of how large computer memories become, it will be more economical to use efficient file organization and data compression techniques. The author's recommendations for immediate development are:

1. Develop larger, faster and more economical random-access

computer memories.

2. Continue to use filing and compression techniques which effectively utilize random-access memories.
3. Pursue the concept of "Predestined Storage" which the author feels has considerable promise in the information retrieval field.

Retrieval

Due to their close interrelation, the retrieval aspect will be benefitted by any advances made in either indexing or storage. However, the new developments in indexing and storage will usually increase the complexity of retrieval. For example, the statistical nature of automatic indexing techniques will require additional interpretation and, therefore, more complex retrieval systems. The author's recommendation in this area is to keep the retrieval process as simple as possible, as it is the only portion of the system which must be repeated with each inquiry. Because of this iterative nature, the author recommends placing the burden of statistical interpretation on the indexing portion of the information retrieval system.

BIBLIOGRAPHY

- (1) Mauchly, J. W. No-Slip Library Machine, Science News Letter No. 56 (1949).
- (2) Luhn, H. P. A New Method of Recording and Searching Information, American Documentation Vol. 4 (1953).
- (3) Taube, M. Studies of Coordinate Indexing, American Documentation Vol. 6 (1955).
- (4) Luhn, H. P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information, IBM Journal (October, 1957).
- (5) Luhn, H. P. Auto-Encoding of Documents for Information Retrieval Systems, in M. Boaz [ed]. "Modern Trends in Documentation" (1959) 45-58.
- (6) Maron, M. E., J. L. Kuhns On Relevance, Probabilities Indexing and Information Retrieval, J. Assoc. for Computing Machinery Vol. 7 (1960) 216-44.
- (7) Doyle, L. B. Semantic Road Maps for Literature Searchers, J. Assoc. for Computing Machinery Vol. 8 (1961) 5533-78.
- (8) Tanimoto, T. T. The General Problem of Classification and Indexing, in "Machine Indexing," American Univ. (1962) 233-35.
- (9) Horn, F. E. Elementary Matrix Algebra, Third Printing, MacMillian Company, New York (1960) 209-211.
- (10) Baker, F. B. Information Retrieval Based on Latent Class Analysis, J. Assoc. for Computing Machinery Vol. 9 (1962) 512-521.
- (11) Stiles, H. E. The Association Factor in Information Retrieval, J. Assoc. for Computing Machinery Vol. 8 (1961) 271-279.
- (12) Hellerman, H. Addressing Multidimensional Arrays, Communications of the ACM Vol. 5 No. 4 (April, 1962) 205-207.
- (13) Biet, F., G. Picot, B. Reibell, F. Levery An Experiment in Automatic Selection of Documentation, Brochure by Compagnie de Saint-Gobain and IBM France (translated in memo Bro-86-61 by Phillips Petroleum Company, Technical Information Section, October, 1961).

VITA

Charles Douglas Parsons
Candidate for the Degree of
Masters of Science

Thesis: STATISTICAL INFORMATION RETRIEVAL

Major Field: Industrial Engineering

Biographical:

Personal Data: Born in Gainesville, Texas, December 17, 1933, the son of Thomas E. and Mabel Florine Parsons.

Education: Attended grade school in Los Angeles, California, Dallas, Texas and Carlsbad, New Mexico; graduated from Carlsbad High School in 1952; received the Bachelor of Science degree from New Mexico State University, with a major in Chemical Engineering, in August, 1957; completed requirements for the Masters of Science degree in July, 1966.

Professional Experience: Worked as a cooperative student with United States Department of Army, White Sands Proving Ground, New Mexico, from July, 1952 until August, 1957; employed at Phillips Petroleum Company, Research and Development Department in September, 1957; transferred to the Computing Department in September, 1959; present position as Operations Research Analyst; member of the Association for Computing Machinery and the Bartlesville Computing Association.