INFORMATION TO USERS

This dissertation was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document photographed is “Missing Page(s)”. If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in “sectioning” the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from “photographs” if essential to the understanding of the dissertation. Silver prints of “photographs” may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

University Microfilms
300 North Zeeb Road
Ann Arbor, Michigan 48106
A Xerox Education Company
CAMERON, James Slagle, 1932-

AUTOMATIC DOCUMENT PSEUDOCLASSIFICATION AND RETRIEVAL BY WORD FREQUENCY TECHNIQUES.

The Ohio State University, Ph.D., 1972
Information Services, information storage and retrieval systems

University Microfilms, A XEROX Company, Ann Arbor, Michigan

THIS DISSERTATION HAS BEEN MICROFIILMED EXACTLY AS RECEIVED
AUTOMATIC DOCUMENT PSEUDOCCLASSIFICATION AND RETRIEVAL

BY WORD FREQUENCY TECHNIQUES

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By

James Slagle Cameron, B.Sc., M.Sc.

The Ohio State University
1972

Approved by

[Signature]
Adviser
Department of Computer and
Information Science
PLEASE NOTE:

Some pages may have
indistinct print.
Filmed as received.

University Microfilms, A Xerox Education Company
ACKNOWLEDGMENTS

Very few people who receive the Doctor of Philosophy Degree can claim to have done it entirely by themselves. The completion of this dissertation, and the research upon which it is based, was due to several individuals whose advice and guidance I appreciate and acknowledge. My advisor, Dr. Anthony E. Petrarca, and the other members of my committee, Dr. Ronald L. Ernst and Dr. Gerald J. Lazorick, make many helpful suggestions during the course of the research and the writing of the dissertation. Dr. Arthur L. White, who assisted in finding other faculty members in the College of Education who would serve as information users also receives my thanks, as do those faculty members themselves.

Three organizations also made major contributions which should be recognized. The United States Air Force through the Air Force Institute of Technology made it possible for me to return to school. The Instruction and Research Computer Center of The Ohio State University and the National Science Foundation provided the technical support required in the research.

Last, but certainly not least, I would like to acknowledge the patience and understanding of my wife, Marianne, and my children, Leslie, Drake, Cathy, and Laura. Without their support, this would not have been possible.
VITA

October 22, 1932 . . . Born Springfield, Ohio

1953 . . . . . . . . B.Sc., The Ohio State University, Columbus, Ohio

1954-1955 . . . . U.S.A.F. Student, Department of Meteorology, Pennsylvania State University, State College, Pennsylvania


1960-1961 . . . . U.S.A.F. Chairman, Department of Physics, U.S. Naval Academy Preparatory School, Bainbridge, Maryland


1963-1964 . . . . U.S.A.F. Student, Department of Statistics, Leland Stanford Junior University, Stanford, California

1964 . . . . . . . . M.Sc., Leland Stanford Junior University, Stanford, California


1969-1972 . . . . U.S.A.F. Student, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio
FIELDS OF STUDY

Major Field: Information Science

Studies in Information Storage and Retrieval
Dr. Anthony E. Petrarca

Studies in Information Systems
Dr. Gerald J. Lazorick

Studies in Man Machine Interface
Dr. Ronald L. Ernst
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>ii</td>
</tr>
<tr>
<td>Vita</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td><strong>Chapter</strong></td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. HISTORICAL REVIEW</td>
<td>5</td>
</tr>
<tr>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>Word Occurrence Methods</td>
<td></td>
</tr>
<tr>
<td>Topological Methods</td>
<td></td>
</tr>
<tr>
<td>Retrieval Methods</td>
<td></td>
</tr>
<tr>
<td>Research Direction</td>
<td></td>
</tr>
<tr>
<td>III. EXPERIMENTAL DESIGN</td>
<td>20</td>
</tr>
<tr>
<td>Basic Approach</td>
<td></td>
</tr>
<tr>
<td>Underlying Assumptions</td>
<td></td>
</tr>
<tr>
<td>Data Base and User Selection</td>
<td></td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td></td>
</tr>
<tr>
<td>Nature of Parameters Used</td>
<td></td>
</tr>
<tr>
<td>Parameter Definitions</td>
<td></td>
</tr>
<tr>
<td>Evaluation Criteria</td>
<td></td>
</tr>
<tr>
<td>IV. IMPLEMENTATION OF THE TECHNIQUE</td>
<td>48</td>
</tr>
<tr>
<td>Corpus Generation</td>
<td></td>
</tr>
<tr>
<td>Relevance Criteria</td>
<td></td>
</tr>
<tr>
<td>Vocabulary Profile</td>
<td></td>
</tr>
<tr>
<td>Decomposition of the Abstract File</td>
<td></td>
</tr>
<tr>
<td>Final Parameter Selection</td>
<td></td>
</tr>
<tr>
<td>Pseudoclassification of the Abstract File</td>
<td></td>
</tr>
<tr>
<td>V. RESULTS</td>
<td>66</td>
</tr>
<tr>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>Counting Parameters</td>
<td></td>
</tr>
<tr>
<td>Vocabulary Correspondence Parameters</td>
<td></td>
</tr>
<tr>
<td>Redundancy Parameters</td>
<td></td>
</tr>
<tr>
<td>Chi-Square Parameters</td>
<td></td>
</tr>
<tr>
<td>Parameter Selection Summary</td>
<td></td>
</tr>
<tr>
<td>Alternate Form of Pseudoclassification Equation</td>
<td></td>
</tr>
<tr>
<td>Retrieval Results</td>
<td></td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User Search Profiles</td>
<td>50</td>
</tr>
<tr>
<td>2.</td>
<td>Stoplist Used for Pseudoclassification Operations</td>
<td>53</td>
</tr>
<tr>
<td>3.</td>
<td>Summary Data on Vocabulary Profiles</td>
<td>55</td>
</tr>
<tr>
<td>4.</td>
<td>Excerpts from Gl Vocabulary Profile</td>
<td>57</td>
</tr>
<tr>
<td>5.</td>
<td>Frequency of Parameter Selection in Initialization Process Using 46 Parameters and 12 Corpora</td>
<td>59</td>
</tr>
<tr>
<td>6.</td>
<td>Frequency of Parameter Selection in Discriminant Analysis (46 Parameters)</td>
<td>60</td>
</tr>
<tr>
<td>7.</td>
<td>Frequency of Parameter Selection in Discriminant Analysis (27 Parameters)</td>
<td>62</td>
</tr>
<tr>
<td>8.</td>
<td>Initialization Corpora Pseudoclassified by 27 and 46 Parameters (Comparative Results)</td>
<td>63</td>
</tr>
<tr>
<td>9.</td>
<td>Final Results--Master Abstracts File Pseudoclassified by 27 Parameters</td>
<td>65</td>
</tr>
<tr>
<td>10.</td>
<td>Order of Parameter Selection for Discrimination of Each Initialization Corpus (46 Parameters)</td>
<td>68</td>
</tr>
<tr>
<td>11.</td>
<td>Order of Parameter Selection for Discrimination of Each Initialization Corpus (27 Parameters)</td>
<td>69</td>
</tr>
<tr>
<td>12.</td>
<td>Parameter Versus Rank as Discriminator (Derived from Table 11)</td>
<td>70</td>
</tr>
<tr>
<td>13.</td>
<td>Comparison of Number of Relevant Abstracts Retrieved by The QUERY System and by Pseudoclassification</td>
<td>76</td>
</tr>
<tr>
<td>14.</td>
<td>Comparison of Precisions of Retrieval Obtained by the QUERY System and by the Pseudoclassification System</td>
<td>77</td>
</tr>
</tbody>
</table>
CHAPTER I. INTRODUCTION

In order to establish which documents of a particular collection might contain information of interest to an individual, he must have some idea of the contents of the individual documents. Since it is usually not feasible for the individual to read all the documents in the collection in their original form, he must rely on some type of condensed document description (1) to assist him. These document descriptions usually involve some type of human or automatic content analysis of the original document, involving operations such as classification, abstracting, extracting, and the like. Indexing and classification are more essential for the retrieval process, since they both embody categorization of documents into different classes and subclasses, the names (or representations) of which can be organized in some fashion to facilitate retrieval of documents broadly coinciding with (or matching) the user's categories of interest. On the other hand, abstracts and extracts, which are short natural language summaries of the original documents, help to reduce the set of items retrieved on the basis of indexing or classification to a subset which more narrowly coincides with the user's interests.

Traditionally the content analysis (i.e., condensed document description) and retrieval operations have been performed manually. In recent years, however, considerable effort has been expended in attempting to develop effective methods of automating these operations (33). This effort has been expended in three distinct areas of activity. One area of activity has been largely concerned with the development of techniques for automatic content analysis to aid in the production of
printed indexes or abstracts for use in manual systems of retrieval. The second area of activity has been concerned with the development of retrieval systems which automate the operations involved in matching a manually produced query to indexes or abstracts available in computer-readable form. The third area of activity is concerned with automatic indexing and retrieval of documents in an interactive manner, the interaction in this case referring to the use of evaluated results from one document retrieval operation to automatically modify an index or query (or both) to improve the next retrieval operation. The research described in this thesis falls into the last category. Chapter II discusses pertinent earlier work done in each of these areas to provide the necessary perspective for this work.

Since classification and indexing are implicit operations in all of this earlier work, a few words should be mentioned about these operations. Classification is essentially the result of an attempt to organize knowledge systematically. It involves partitioning knowledge into a fixed number of classes, each of which in turn is partitioned into subclasses, and sub-subclasses, etc. Ideally, every element of knowledge would occupy a point in this hierarchical structure, and its relationship to all other elements of knowledge would be known. The formal description of the classes, subclasses and sub-subclasses, usually in the form of an alphanumeric code, is the classification schedule, and the assignment of a document to one or more of these classification codes is called classification. Probably one of the best know classification schemes is the Dewey Decimal System (19) used in many libraries. A constraint imposed by many classification schemes that use the classification codes
for file location purposes is that the classification process is mutually exclusive, that is, a document may be assigned to only one class at a specific point in the hierarchy. This restriction, of course, applies to libraries, since a book may occupy only one shelf location.

Indexing, although similar to classification, differs in concept. Indexing is the assignment of a document to one or more subject categories to provide a condensed description of the document for retrieval purposes. However, the number of categories is not necessarily fixed, nor are there necessarily a formal set of rules for the assignment. If the indexing operation involves a formal schedule and hierarchical structure, then it is identical to classification, otherwise it is not the same.

The problems inherent in indexing and classification ought to be, but often are not immediately apparent to the information user. He is dependent upon another individual's assessment of a document's contents. Further, his interests may not coincide with either the order imposed by the classification schedule used or the method used in assigning index terms. Thus, a set of tools designed to assist him to gain information may become barriers to keep him from acquiring it.

If it were possible for the user to train, or educate, the indexer to consider his specific interests and create a new category for documents relevant to him, the problems mentioned above would be alleviated. However, this is an unrealistic approach, since it would be impossible to classify every document for every potential user according to his own classification scheme. The research in this thesis is oriented towards developing a suitable method whereby the documents in a collection would be automatically pseudoclassified for each user or
group of users. That is, each document in the collection would be classified as relevant or irrelevant for each user.

Just as a classifier must be taught the relevance criteria needed to be able to identify properly those documents in which a user is interested, an automated system must operate in similar fashion. The approach used as a basis for this research is to provide a set of known relevant documents adding others to this set whenever the user's interests shift. Then, the classifier or automated system must determine how the relevant documents resemble each other and how they differ from non-relevant documents. Thus, a set of similarity measures are required, as well as a set of rules for using those measures for the pseudoclassification system. Finally, a method of document retrieval must be provided so that the user can benefit from the system. The class of users who could be helped by this approach are those who are able to identify some minimal set of relevant documents that can be processed by the system. The measures selected and the technique used to determine which of them seemed to be the best for pseudoclassification are described in Chapter III. The specific procedures followed in the research, and the results, are described in Chapters IV and V. Conclusions and suggestions for future research are contained in Chapter VI.
CHAPTER II. HISTORICAL REVIEW

Introduction

As mentioned in Chapter I, efforts in developing effective methods of automatic content analysis and document retrieval have thus far been concentrated in three areas. One has dealt with the development of techniques for automatically analyzing a document's contents to aid in the production of printed indexes and abstracts for use in manual retrieval, the second has involved development of techniques whereby the retrieval and indexing operations interact so as to improve the quality of both, and the third has been concerned with the development of retrieval systems which automate the operations involved in matching a manually produced query to indexes or abstracts available in computer-readable format.

The most widely used techniques for automated content analysis are based on the identification and selection of significant words in a text and using them as index terms. The assumption is that a large number of index terms will adequately identify contents of a document for a majority of potential users. These techniques are described below under the heading Word Occurrence Methods.

A different approach to identifying document content is to represent the document as a point in n-dimensional space, where each dimension of the space represents an idea or concept. This approach, which can be categorized as topological, is similar to classification in that there are only a finite number of dimensions involved, and a document can be assigned coordinates only within the space. The purpose of this approach is to allow the modification of coordinates as users of the
system interact with the points (documents) during the retrieval process. The topological approach, then, represents those techniques in which indexing (or classification) interact with retrieval to enhance both processes. These are discussed under the heading Topological Methods.

Some aspects of the activities involved in automating the operations for matching a manually produced query to indexes or abstracts available in computer-readable form are discussed under the heading Retrieval Operations. The merits and deficiencies of the above approaches, which provided the basis for this research, are discussed in the section on Research Direction.

Word Occurrence Methods

The simplest word-occurrence technique for indexing is the KWIC (key-word in context) index (22). Although not a new idea, the technique gained wide usage, largely as an out-growth of work by Luhn (25), when computers made simple text manipulation on a large scale feasible.

Basically, every significant word in a text (usually the title of a document) is used as an index entry. Determination of the significance of words is handled in one of two ways. Either a stoplist is used to delete certain common words from the indexing process, while all remaining words are assumed to be significant (see for example Chemical Titles, a computer produced publication of the American Chemical Society), or each word in the text is manually evaluated for significance and flagged before the index is generated (see for example Chemical-Biological Activities (CBAC), another computer produced publication of the American Chemical Society). Since the index entries are generated from the text, the usual problems associated with the use of uncontrolled vocabularies
exist. In other words, the user, in searching the index must know not only the terms which he most often uses to describe topics of interest to him, but all synonyms, near synonyms, related terms, broader terms and narrower terms if he wants to ensure retrieving most of the documents on a particular topic.

A more serious problem, however, stems from the fact that the adequacy of the technique is entirely dependent on the assumption that the text from which the index terms were taken is adequately descriptive of the document. In the case of titles, this means that cute or misleading titles would result in the effective loss of the document for the information seeker. This specific problem is, however, becoming less critical, according to Tocatlian (39), since there seems to be a trend towards more descriptive titles, caused in part by the increase in KWIC-type title indexing.

In several early studies on automatic content analysis (4, 21, 25, 26, 41), it was observed that a frequency count of significant words in a document could isolate the special vocabulary used to convey information in any realm of discourse. As a result, most of the word occurrence techniques for indexing or classifying documents are based upon this observation. The technique most often used is to select index terms from a document by comparing the frequencies of occurrence of certain words or phrases in the document to their expected frequencies of occurrence. This approach, based on work by Luhn (25, 26) and Zipf (31), has been extensively tested by Baxendale (4) and Damereau (17). Measures and criteria for determining the acceptance (or rejection) of words or phrases to be used as index terms have been suggested by
Edmondson and Wyllys (19), Damereau (17), and Doyle (20). Implementation of the technique requires first that a corpus of documents pertaining to a subject be decomposed into its component words and/or phrases, or a subset selected by methods similar to those used to identify significant words for KWIC indexes. If the corpus selected is large enough, the relative frequencies of the words or phrases derived may be considered to be an accurate estimate of the relative frequencies for the universe of documents pertaining to the subject.

The indexing operations performed on documents known to belong to that universe require that each document be similarly decomposed. Then, if the relative frequency of a word or phrase in the document is significantly greater than its relative frequency in the corpus, the word or phrase is selected as an index entry. Carrol (13, 14) has recently tested this technique by comparing the results with those obtained manually by a group of professional indexers using a given set of documents. His conclusions were that, automatic indexing in this manner is more consistent than, and equivalent in quality to, indexing performed manually.

As with the KWIC index approach, the user must be sufficiently familiar with all the vocabulary used by others to describe his area of interest if he wishes to find all pertinent references. Again, the style of the author of a document may influence the outcome of the results because the use of too many synonyms for a particular concept may give rise to a situation where none of them has occurred frequently enough to be selected as an index entry. A final point, of theoretical significance, has to do with the size of the corpus used as a sample of
the universe. Damereau and Carroll defined their universe to be the specific collections they were using, but this definition, which obviously was a pragmatic one, would be inadequate in terms of the general applicability of the technique.

Word occurrence frequency techniques have also been studied for application in classification. Borko (8, 9), Borko and Bernick (10, 11), Maron (27) and Williams (40) have attempted to classify documents on the basis of discriminant or factor analysis using as inputs the actual and expected frequencies of a pre-selected set of words. The general approach followed by all the researchers was to take a set of documents and from them manually select a group of words which the researchers felt were most representative of the literature in a given field (e.g., forty-eight were selected by Williams, ninety by Borko) and the relationship of the words to the subfields were analyzed on the basis of word occurrence or cooccurrence frequencies in each subfield. Classification criteria were then established by factor or discriminant analysis, and the original sets of documents were classified by probabilistic methods (i.e., a document was assigned to that class for which the probability of class membership was greatest). Williams' results appear to be best, ranging from forty-three to eighty per cent correct document classification, while the other experiments gave about fifty per cent correct classifications. A possible explanation for the disparity may be that Williams used only four classes, while Borko and the other experimenters used twelve. Although the results seem to be poor, they are statistically better than random assignment would provide. Thus, this approach to classification seems to show some promise.
However, some possible limitations of the approach are as follows. First, it is unlikely that any field of knowledge can be completely described by a small number of words. Although such a restriction may be necessary because of the extensive use of word-association techniques (see below), the decision involved in having to decide which of a limited number of keywords to use for classification purposes places a heavy burden on the designers of such a system. In addition, the likelihood of being able to partition and repartition the field on the basis of the same set of words would seem to be remote. (Repartitioning, using a different set of words, might give rise to questions concerning the applicability of a word to a subfield when it is not applicable to its generic field).

Another difficulty, which is characteristic of most classification systems which assign each document to a single mutually exclusive class, arises when there are possibilities of multiple assignment (or no assignment in those cases where none of the pre-defined classes are adequate). The rule for classification used in the above experiments was to assign the document to that class for which the probability of class membership was greatest. However, unless there is some indication of the second or third highest probabilities, and their magnitudes relative to the highest probability, marginal classification errors will probably not be significantly reduced, nor will the magnitude of these errors be recognized. Alternatively, if the system allows multiple classification assignments to a document, the problem then would be to determine the minimum probability of class membership which warrants a classification assignment to a document. If this minimum is too low,
the system could assign every document to every class; if too high, the
document might not belong to any class or the approach would be effective­
ly the same as that used in the studies described.

The final technique to be discussed, term similarity calculations,
are not used as ends in themselves, but are used in conjunction with
other techniques partially to resolve difficulties caused by synonomy
or near synonomy. Consider two documents, one containing the phrase
information storage and retrieval, the other containing the phrase data
storage and retrieval. Although data and information do not have exactly
the same meaning, the context of the two words indicates some degree of
overlap, and there should be some method to determine this. Stiles (37)
has developed a technique for resolving this problem. The technique
requires the generation of large matrices (whose elements may be variances
and covariances, frequencies of specific terms in documents, etc.) and
multiplying them together or raising them to powers greater than one.
The elements in the resulting matrices will indicate the degree of similar­
ity or overlap (i.e., the indirect association) between the different
elements of the original matrix. For example, a matrix could be
generated in which the rows represent different terms. By taking this
matrix, and its transpose, one can derive a term-term matrix which
indicates the degree to which terms are related. Thus, in the example
above, the terms storage and retrieval would be strongly related, while
information and data would be less strongly related. In other cases,
some terms will be so weakly related that they can be considered as
completely unrelated.
The advantage of this approach is that related terms can be identified for use in augmenting indexes, for relating concepts (if concepts rather than terms are used in the original matrix), or for creating dictionaries or thesauri. The disadvantage is that matrix manipulations require large amounts of computer core storage and may require a significant amount of time for processing (12).

Topological Methods

The topological approach, which represents a document as a point in an n-dimensional space, has not been operationally implemented except as research projects in a few locations. The general approach taken thus far is to analyze a document, either manually or automatically, for content (i.e., determining what ideas or concepts are being discussed in the document). Then a vector is formed by assigning weights (relative importance numbers) to each concept. Since the elements of the vector are given in terms of coordinates, each coordinate representing one of a set of predetermined concepts, a vector representation is generated by using the weights as the coefficients of the vector.

The vector representation is, however, only the first step of the process. Following this representation, all points in the n-space are examined for relative similarity by computing a correlation function (7, 37) and forming centroids for those points which are within some predetermined range of correlation values. The centroids represent an average of the concepts for the documents represented by the points around the centroid. By retrieving documents whose points are clustered about a given centroid, and evaluating their individual relevances to a given user's needs, the points may be moved closer to, or farther from, a
centroid, effectively reindexing the document. The purpose of this reindexing is to allow a body of users continuously to reindex documents until the document is correctly indexed, correctness of indexing implying the set of index entries which most satisfies a majority of information users.

The experimental system which best illustrates this approach is the SMART system, first implemented at Harvard University and later at Cornell University by Salton (33, 34). In the version implemented at Cornell, the concept weights are automatically assigned to each document. The procedure is as follows. Documents are input into the computer and decomposed into individual words. These words, which may be modified by stem truncation techniques so that most variant forms will be considered as one single word, are then compared against a table identifying related words, concepts, and relative weights. Thus, a single word may cause the assignment of weights to several concept positions (coordinates). Related concepts, identified by techniques such as those proposed by Stiles (discussed in the previous section), may also be assigned weights. When all words are processed, the vector is normalized (i.e., its magnitude is reduced to a standard unit), and the resulting vector is correlated with all other vectors in the system. New centroids are generated if necessary.

As user feedback occurs after a query, or group of queries, those weights which caused erroneous retrieval (as judged by the user) are reduced algorithmically while weights which caused correct retrieval are increased. When the points are moved, correlation factors are again computed and the new centroids may be derived.
Although this approach appears to work well in an experimental system such as the SMART system, several questions remain unanswered. For example, the SMART system (at Cornell) actually handles approximately 4,000 documents which have been extensively analyzed, both manually and automatically (34). Would such a system work equally well for a larger data base of 50,000 or more documents which have not been carefully analyzed? Also, since many of the calculations involve large matrix operations, even though technological advances such as megabyte core storage, direct access storage devices, and increased speed are available with the newest computers, there is some doubt that the physical operations required could be accomplished in a reasonable time to improve the quality of information retrieval at a given point in time. Finally, with many different users having diverse points of view, how long would it take for the system to stabilize (i.e., to have documents occupy their proper point in space), particularly if the data base is updated with some frequency? At this time, it appears that there is little likelihood that systems of this type will be implemented on a large scale in the near future.

Retrieval Operations

This section describes some details concerning the effort expended in recent years toward development of retrieval systems which automate the operations required to determine which documents or abstracts match the user's request. The principles involved are similar whether they are performed manually or by automated retrieval systems. In either case, the determination of which documents in a collection may be relevant to a particular user's needs, is not normally a trivial task.
The user must first determine which index terms or classification codes best describe his area of interest. This means that, in the case of a controlled vocabulary system, he must use a thesaurus or classification schedule. In a free vocabulary system, he must determine not only the terms he might use to define his interests, but also the terms which an indexer or an author, in the case of KWIC title indexes, might have used to describe the same topics. The retrieval operation then amounts to a matching operation (i.e., use of a match function) between the query terms and the index terms.

In automated retrieval systems, a match function, in its simplest form, is a Boolean expression in which all variables in the expression have a value of one or zero (corresponding to true or false, present or absent, etc.). The expression, when evaluated, also has a value of either zero or one. The operators in such a function are AND, OR, and NOT, and have the following properties. The expression A AND B means that variables A and B must both have a value of one in order for the expression to have a value of one. If either or both are zero, the expression has the value zero. The expression A OR B means that either A or B, or both, must have a value of one for the expression to be evaluated as one. Thus, the value of an OR expression is one unless the variables are all zero. The expression NOT A will have a value of one if A is zero, zero if A is one.

For the purpose of retrieving documents, the user may decide that index records containing index terms X, Y, and Z are most likely to refer to documents in which he is interested. Then his expression will be F = X AND Y AND Z. In searching the file of index records, the following
procedure is used. Prior to the examination of each record, variables X, Y and Z are set to zero. The record is examined to see if the desired index terms are present. If an index term, say X, is present, then the associated variable, X, is reset to one. At the end of the examination of the record, the expression is evaluated. If F is one (i.e., all the index terms were present), the document associated with the record is retrieved. Then the process is repeated for the next record until all records in the index file are examined.

Although the above example used AND logic, OR logic could have been used if the user wanted documents which were indexed under either K, L or M. More complex Boolean expressions, utilizing multilevel logic, are possible in some systems, but the process is essentially the same.

Although the above process can be rapidly performed by automatic means, the limitations are the same as for manual retrieval from printed indexes. The user must still assume that the indexer's point of view coincides with his own. Specifically, he must assume that the indexer will assign the same index terms to a document that the user would. The user must also assume that his interests can be, and are, easily described in terms of the index. When the user's interests are very broad and cross many boundaries (as reflected in the index), it may not be possible to retrieve exactly what he wishes to retrieve. Apart from the difficulty of expressing one's interests in a Boolean expression, the user may have implicit substitute values for index terms which he is unable to state. Thus, a user may want all documents indexed by term A, but also be willing to accept documents indexed by B, C, D, but not E. However, sometimes he will accept a document which is indexed as B, C, and E. Briefly, there is more subjectivity in document retrieval and
relevance judgments than a Boolean expression permits. The topological
systems attempt to resolve this problem by moving points, but this
approach is not sufficiently widespread to help the average user. Thus,
while automating the retrieval function speeds up the process and allows
more documents to be retrieved and examined in a given time, the
problem of determining relevance criteria in advance of document retrieval
(i.e., in terms of his match function) is still left to the user.

Research Direction

As mentioned in Chapter I, the index is an essential form of
condensed document representation to aid in retrieval of information by
the traditional manual approach. The previous sections of this chapter
reviewed some of the accomplishments that have been made in automating
both the indexing operations and the matching operations involved in
attempting to identify those documents containing information of interest
to a user. However, all the existing approaches to retrieving information,
both manual and automatic, present problems to the user concerning how
best to define his interests for querying the system. As indicated in
one of the previous sections (Word Occurrence Methods) these problems
result mostly from the vocabulary used in the indexes. Thus, if a
technique could be developed to help the user find the vocabulary used
to describe his interests in the documents themselves (or alternatively
the indexes) this problem would be alleviated and would make the
collections more useful to the user. This research was directed toward
development of such a technique. Since search of a document collection
for an individual user is essentially a partitioning process in which
there are only two mutually exclusive classes (i.e., relevant and non-
relevant documents), a process called pseudoclassification by Jackson (23), the technique that was developed is described throughout this thesis as a pseudoclassification technique to partition document collections for individual users.

As envisioned, the technique would reduce or eliminate many of the communication problems that now exist between indexers, indexes, and users concerning the information content of a document. It is based on the premise that, since word occurrence techniques have been shown to work somewhat effectively to describe contents of documents, the same techniques should work equally as well to identify topics of interest to a user. The only thing required by the user would be an initial corpus of relevant and non-relevant documents, each segment of which would then be subjected to a word occurrence frequency analysis to identify relevant and non-relevant topics. The results of this analysis would be compared against a similar analysis of each document in the collection, using appropriate similarity measures to identify which documents in the collection would be of interest to the user. The results of the first few iterations, if evaluated and fed back into the system, could be used to refine the identification of a user's interests just as feedback is used in the topological systems described above to refine descriptions of the contents of items in a collection.

In addition to use for pseudoclassification purposes, the technique could also be used for certain types of classification, particularly that type which involves partitioning of broader classes of knowledge (e.g., education) into narrower subdivisions (e.g., math education, vocational and technical education, etc.).
The rest of this thesis is devoted to a description of the design and implementation of the technique, followed by a discussion of results, conclusions, and suggestions for future research.
CHAPTER III. EXPERIMENTAL DESIGN

Basic Approach

As stated in the preceding Chapters, the purpose of the research was to develop a method of pseudoclassification and retrieval of documents using word frequency of occurrence techniques as a basis for identifying those documents coinciding with a user's categories of interest. In order to accomplish this task, certain operations had to be performed.

First, it was necessary to determine the vocabulary used in documents relevant to the user's interests. This vocabulary, which is referred to as a user vocabulary profile, is discussed in detail in Chapter IV. Suffice it to say, at this point, that these profiles or comparison lists were generated by acquiring a corpus of user judged relevant abstracts which were decomposed, as a group, into word and token counts. (A word is a unique string of alphabetic characters, preceded and followed by a non-alphabetic character. A token is the occurrence of a word). It was also necessary to set up some similarity measures to be tested for their power to discriminate between relevant and non-relevant documents. Although the parameters actually chosen for testing are discussed in a later section, some general consideration of the types of measures chosen are introduced here.

Since the basic techniques involve comparison of the words in an abstract to the words on a comparison list, four possible measures or parameters become immediately available. These are:

1) the number of words in the abstract \( W \);
2) the number of tokens in the abstract \( T \);
3) the number of words common to both the abstract and the comparison list (M); and
4) the number of tokens in the abstracts represented by the M words (C).

Although these parameters would not be good discriminators, per se, certain combinations might be. For example, comparison of the words in an abstract against a particular vocabulary profile produced an M/W ratio of .75, this might indicate a high degree of correspondence, and thus be an indication of potential relevance. Similarly, a C/T ratio and other parameters of this type might also be appropriate.

Another group of word frequency variables chosen for consideration was the chi-square ($\chi^2$) group. If a particular comparison list contains those words whose tokens comprise some per cent (p) of a corpus of relevant documents, then a $\chi^2$ variable can be generated. Assume that if an abstract is relevant, $M = pW$. Then

$$X = \frac{(M-pW)^2}{pW} + \frac{(W-M-(1-p)W)^2}{(1-p)W}$$

will have a limiting distribution as a chi-square variable. Further, the probability of getting a computed value less than or equal to X is the probability that the text of the abstract came from the same universe as the words on the comparison list. This, then, an indirect measure of the abstract's relevance to a given user. Similarly, one may further assume that not only will the percentage (p) of the words on the comparison list appear in the abstract, but they will occur with the same ratio as in the corpus of relevant documents considered as a whole.
In general there appeared to be no shortage of possible parameters, and the problem would be to find some combination which would produce the correct pseudoclassification of the abstracts, a problem for which the use of discriminant analysis techniques seemed most appropriate.

The rest of this chapter is concerned with various aspects of the experimental design, including a consideration of the underlying assumptions involved; a discussion of criteria and constraints concerning database and user selection; a general description of what the discriminant analysis approach involves; the types of parameters tested; definitions of the specific parameters used; and the criteria used for evaluation of the results.

**Underlying Assumptions**

Based on the observation that vocabulary in any given field of knowledge tends to stabilize (26, 41), an important assumption in this research was that the same relations would hold for a subset of documents judged to be relevant to a user's interests. Another assumption was that word frequency techniques alone, devoid of syntactic and contextual relationships, would provide sufficient discriminating power to partition adequately a document collection for a user (20). Intuitively, it was also felt that there was no necessity to generate and manipulate similarity matrices of the type proposed by Stiles (37) and others (28, 34, 35). This was predicated on the assumption that those words which are used extensively in the literature of interest to a particular user will be a part of the vocabulary of his field, while words which are not extensively used are probably not critical in defining his area of interest.

Alternatively, one might suggest that a document of interest containing
low frequency words will contain many of the high frequency words as well. The generation of similarity matrices, particularly term-term matrices, would then only provide the information that some words co-occur frequently and some co-occur infrequently.

Pragmatic considerations and other constraints on the experimental procedures require mentioning of several other assumptions that have bearing on the results and conclusions to be drawn from this research. The first was that abstracts could be used for the pseudoclassification process. Abstracts had to be used in this research because no large files of full-text documents were immediately available in computer-readable form, while several abstract files were. (This is discussed in more detail in the next section).

The next assumption, which was made strictly for pragmatic reasons to try to simplify the processing operations involved, was that it would not be necessary to eliminate common words from the document word lists and the user comparison lists. Since common words tend to occur with relatively constant frequency in both lists, any biases introduced into the measuring parameters used for the pseudoclassification process would be constant and their effects would tend to cancel out. Needless to say, the benefits which would accrue from not having to introduce common-word deletion routines, into the processing programs would be significant. This approach had to be modified, however, when preliminary results showed that the biases were not constant and were larger than anticipated, probably because the experiment was performed on abstracts rather than full-text documents. It might be interesting to see if the original assumption would hold for whole-text documents.
A final assumption was that stem truncation and word-form modification techniques would not be required. The reasoning was that if the various forms of a word occurred with any frequency in the literature a user identified as relevant, they would appear as part of his vocabulary profile.

The validity of these assumptions, of course, can only be substantiated, qualified, or refuted by the experimental results and conclusions, the subject matter of the remaining chapters of this thesis.

Data Base and User Selection

Ideally, the approach to be utilized in this research would best be performed on full text computer-readable documents, and the preliminary research was actually performed on nineteen full text documents, some of which were available from other research activities and some of which were key-boarded directly. It was apparent, however, that this collection was too small and restricted in subject material to use in the final research, and a much larger document file would have to be obtained. Since no large full text document files in machine readable form were immediately available, a choice had to be made among several condensed text computer readable files that were available. These were of two basic types, document title files and abstract files. The title files were discarded from consideration since it was felt that the technique which was to be tested would require more text than would be available in a title. The available abstract files appeared to have a sufficient amount of text in the majority of abstracts, so the only problem was to decide which file to use.
The two files immediately available were Research in Education (RIE) abstracts compiled by the Educational Resources Information Center (ERIC), and the Chemical-Biological Activities (CBAC) Abstracts compiled by Chemical Abstracts Service (CAS). The CBAC files were eliminated from consideration after a brief examination because it was felt that they would present certain problems for initial testing of the techniques to be used. Specifically it was felt that problems would arise from the highly technical language and a very structured notation used in the abstracts. Although the technique to be used in pseudoclassifying documents should work with this type of data, it was felt that the initial research should be performed on a data base containing a more general vocabulary. The RIE files presently contain more than 40,000 abstracts covering all aspects of the field of Education. The inputs to RIE come from approximately twenty different ERIC Centers, each of which specializes in one of the major subject areas covered by RIE. The fact that there are two ERIC Centers on The Ohio State University campus (i.e., Scientific, Mathematical, and Environmental Education Center, Vocational and Technical Education Center) provided further motivation for selection of the RIE files, since personnel associated with either or both of these centers could provide assistance in case there were any peculiarities in the files which might present problems.

The decision to use ERIC abstracts meant that the class of potential users was also selected (i.e., the class of individuals who are interested in documents concerning education). Several meetings with faculty members in the College of Education resulted in seven faculty members of the Department of Mathematics and Science Education agreeing to act as
information users. In addition, one doctoral candidate in Chemical Education, from the Department of Chemistry, also agreed to participate.

**Discriminant Analysis**

Frequently the set of elements forming a sample can be partitioned into two or more mutually exclusive groups. When this occurs, it may be of interest to determine which characteristics of the elements were the most important in the partitioning process. When there is only one characteristic (or parameter) and several classes, the problem may be collapsed into a univariate analysis of variance problem. When, however, there are many parameters, which may be highly correlated, and many classes to which the elements could be assigned, the problem may become somewhat more complex, since not only must parameter variances be considered, but also covariances.

An approach which can be used, under the assumption that the sampling distributions of the parameters are normally distributed, is to generate a linear combination of the parameters, and treat the resulting sum as a single parameter. (The assumption that the sampling distributions of the individual parameters be normally distributed is essential in the development of the theory, but may be violated in practice providing that the actual distributions are not pathological in nature. However, when this assumption is violated, the results must be carefully reviewed). The linear combination which best performs the discriminating function is that combination whose ratio of the sum of squares between classes for any sample to the total sum of squares is a maximum. The reason for this is that the total variance, or sum of squares, of the sample can be expressed as the sum of two variances, the variance of the elements within
a class and the variance of the elements between classes. By keeping the total variance fixed and increasing the variance between classes, a corresponding reduction of variance within a class is obtained. Intuitively, this means that the elements within a class are more similar to each other than to elements in other classes.

The generation of a linear combination of parameters and the associated variances also relies heavily on two elementary statistical properties of variance. The first is that the sum of the variances of a set of parameters is the variance of the sum of the parameters. The second is that multiplying a parameter by a constant produces a variance equal to the product of the parameter's variance and the square of the constant. Thus, multiplying each parameter by a constant and summing the results produces a new parameter whose variance is easily computed. By considering this new parameter's total variance and within class variance, a function, \( f \), which maximizes between class variance may be obtained.

The technique may be expanded if the above function is derived on the basis of increasing total between class variances, so that within class variances are minimized. This, however, lumps all within class variances together. If one wished to expand the concept of additivity of variances, it can be shown that the total within class variances is the sum of the individual within class variances. Then, by introducing an additional constraint into the determination of the function, \( f \), such that the within class variance of a specific class is minimized after a parameter is chosen, only the coefficients would be different from the original function. If this procedure were carried out for each class, then there would be one discriminating function for each class.
In practice, the parameters describing each element in the sample could be used to generate a set of conditional discriminant functions, and the individual elements could be assigned to that class for which the value of the derived function was least. For example, if there were five parameters \((p_1, p_2, \ldots, p_5)\) available for partitioning a sample into two classes, and only the first three parameters were found to be significant discriminators, the derived functions would be:

\[
f_1 = c_1 p_1 + c_2 p_2 + c_3 p_3 + k_1
\]

\[
f_2 = d_1 p_1 + d_2 p_2 + d_3 p_3 + k_2
\]

Then the rule for assignment would be to assign the element to the first class if \(f_2 < f_1\), otherwise assign the element to the second class. There would be no assumptions made that \(p_1, p_2, p_3\) were the only, or the best, discriminators, only that the data available concerning these parameters and classes indicated that as a group they could discriminate better than any other combination of the five parameters. New parameters might produce a different combination.

A further refinement of this procedure would be to assign to the individual elements a conditional probability of class inclusion for each class so that a cost function for misclassification or correct classification could be generated. A more detailed explanation and derivation of the theory involved may be found in Rao (31) and Anderson (1).

For this research, an existing program to perform the mechanics of the discriminant analysis was used. The Stepwise Discriminant Analysis Program (BMD07M) of the Biomedical Computer Program Package (6) was
selected because the program was known to be well tested and simple
to run. Also, the approach used by the program was desirable within
the context of the research. The technique implemented within the
program is to examine, individually, each parameter not in the discrim­
inant function and include in the faction that parameter whose
inclusion would make the greatest contribution to the function. Then,
the parameters in the function whose contributions had become comparative­
ly small on inclusion of multiply-correlated parameters would be deleted.
Thus, it would be possible to trace through the printed results of a
program to determine which parameters were most significant at any
point in the process.

This program was written as one of a series of statistical programs
by the staff of the Health Sciences Computing Facility, Department of
Preventive Medicine and Public Health, College of Medicine, University
of California, Los Angeles. For a description of the control cards,
data preparation, output options, etc., see BMD: Biomedical Computer
Programs, the program description manual produced by the Health Sciences
Computing Facility, UCLA (6).

Variables are entered into the discriminating function one at a time,
and are selected for inclusion by the statistical counterpart of the
following two equivalent criteria:

a) The variable which, when partialed on the previously
entered variables, has the highest multiple correlation
with the groups.

b) The variable which gives the greatest decrease in the ratio
of within to total generalized variances.
The statistical counterpart to the above criteria is the F test. In this program, if the F ratio of a variable which has been included in the class of discriminators becomes too low, the variable is deleted from the class of discriminators.

The following description of the computational procedure is taken from the BMD manual.

Let
\[ p \] = number of variables.
\[ g \] = number of groups used for the analysis.
\[ t \] = total number of groups.
\[ n_m \] = number of cases in group m.
\[ n \] = total number of cases.
\[ x_{mki} \] = value of variable i for case k of group m.

Assume for simplicity that the first g of the t groups are used for the analysis. The computational steps are:

**Step 1.** The data are read and the following are formed:

**Means**
\[
\bar{x}_i = \frac{1}{n} \sum_{m=1}^{g} \sum_{k=1}^{n_m} x_{mki} \quad i = 1, 2, \ldots, p
\]

**Group means**
\[
\bar{x}_{mi} = \frac{1}{n_m} \sum_{k=1}^{n_m} x_{mki} \quad i = 1, 2, \ldots, p
\]

**Group standard deviations**
\[
s_{mi} = \frac{1}{n_m-1} \left( \sum_{k=1}^{n_m} (x_{mki} - \bar{x}_{mi})^2 \right)^{1/2} \quad i = 1, 2, \ldots, p
\]
\[ m = 1, 2, \ldots, t \]
Within and total cross-product matrices

\[
W = w_{ij}; \quad w_{ij} = \sum_{m=1}^{g} \sum_{k=1}^{n} (x_{mk_i} - \bar{x}_{m_i})(x_{mk_j} - \bar{x}_{m_j})
\]

\[
T = t_{ij}; \quad t_{ij} = \sum_{m=1}^{g} \sum_{k=1}^{n} (k_{mk_i} - \bar{x}_{i})(x_{mk_j} - \bar{x}_{j})
\]

\[i = 1, 2, \ldots, p\]

\[j = 1, 2, \ldots, p\]

Within groups covariance matrix

\[
V = v_{ij}; \quad v_{ij} = \frac{1}{n-g} w_{ij}
\]

\[i = 1, 2, \ldots, p\]

\[j = 1, 2, \ldots, p\]

Within groups correlation matrix

\[
R = r_{ij}; \quad r_{ij} = \frac{w_{ij}}{(w_{ii} w_{jj})^{1/2}}
\]

\[i = 1, 2, \ldots, p\]

\[j = 1, 2, \ldots, p\]

**Step 2.** At each step of the procedure the variables are divided into two disjoint sets; those included in the discriminant functions and those not included. Assume for simplicity that the first \(r\) are included.

Let \(W\) and \(T\) be matrices

\[
W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}
\]

where \(W_{11}\) and \(T_{11}\) are \(r \times r\).

Let \(A\) be a matrix

\[
A = \begin{bmatrix} W_{11}^{-1} & W_{11}^{-1} W_{12} \\ W_{21} W_{11}^{-1} & W_{22} - W_{21} W_{11}^{-1} W_{12} \end{bmatrix} = \{a_{ij}\}
\]
The following statistics are computed:

a) Coefficients and constant terms of the classification functions

\[
\begin{align*}
\mathbf{c}_{k1} &= (n-g) \sum_{j=1}^{r} x_{kj} a_{ij} & i = 1, 2, \ldots, r \\
\mathbf{c}_{ko} &= \frac{-1}{2} \sum_{i=1}^{r} \mathbf{c}_{ki} \bar{x}_{ki} & k = 1, 2, \ldots, g
\end{align*}
\]

b) The square of the Mahalanobis distance between each pair of groups

\[
D_{m,i} = \sum_{i=1}^{r} (\mathbf{c}_{mi} - \mathbf{c}_{ki}) (\bar{x}_{mi} - \bar{x}_{ki})
\]

with \( m,i = 1, \ldots, g \)

c) The F values for testing differences between each pair of groups

\[
F_{m,i} = \frac{(n-g-r+1) n_{m,i}}{r(n-g)(n+m_{i})} D_{m,i}
\]

with \( r \) and \( n-g-r+1 \) degrees of freedom.

d) F values for each variable

(1) If variable \( j \) has been entered

\[
F_{j} = \frac{a_{ij} - b_{ij}}{b_{jj}} \frac{n-r-g+1}{g-1}
\]

with degrees of freedom \( g-1 \) and \( n-r-g+1 \).

(2) If variable \( j \) has not been entered

\[
F_{j} = \frac{b_{jj} - a_{jj}}{a_{jj}} \frac{n-r-g}{g-1}
\]

with degrees of freedom \( g-1 \) and \( n-g-r \)
Under the usual normality assumptions these are the likelihood ratio tests of the equality over all g groups of the conditional distribution of variable j given the (remaining) entered variables.

e) U statistic to test equality of group means

\[ U = \frac{\text{Det}(W_{11})}{\text{Det}(T_{11})} \]

with degrees of freedom \((r, g-1, n-g)\)

f) Approximate F statistic to test equality of group means

\[ F = \frac{1-U^{1/s}}{1/s} \frac{ms+1-rq/2}{rq} \]

where \(s = \frac{(r^2 - q^2 - 4)}{r^2 + q^2 - 5}\), if \(r^2 + q^2 = 5\)

\[ s = 1, \text{ if } r^2 + q^2 = 5 \]

\[ m = n - \frac{r+q+3}{2} \]

\[ q = g-1 \]

its degrees of freedom are \(rq\) and \(ms+1-rq/2\). If either \(r\) or \(q\) is 1 or 2, the approximation is exact.

g) Tolerance values

\[ w_{ii} = a_{ii}/t_{ii} \quad i = r + 1, \ldots, p \]

A variable passes the tolerance test if and only if \(w_{ii}\) and \(t_{ii}\) equal or exceed either a user specified value or a default value.

Step 3. To move from one step to the next, one variable is added or removed from the discriminating set according to one of the following rules:
a) If there are one or more variables which are entered and have an $F$ value less than "$F$ to remove", the one with the smallest $F$ will be deleted.

b) If no variable satisfies a), then from among those variables which have not been included, which pass the tolerance test, the variable selected has greatest "$F$ to enter."

**Step 4.** When no more variables can be added to, or removed from, the discriminating set, the following are computed for

$$\mathbf{a} = 1, 2, \ldots, \mathbf{t}; \quad \mathbf{m} = 1, 2, \ldots, \mathbf{g}; \quad \mathbf{k} = 1, 2, \ldots, \mathbf{n}_2:

a) Value of the $\mathbf{m}$th classification-function evaluated at case $\mathbf{k}$ of group

$$s_{\mathbf{mk}} = c_0 + \sum_{\mathbf{j}=1}^{\mathbf{r}} c_{\mathbf{mj}} x_{\mathbf{mj}}$$

b) Posterior probability of case $\mathbf{k}$ in group $\mathbf{\lambda}$ having come from group $\mathbf{m}$

$$p_{\mathbf{\lambda mk}} = \frac{\text{Exp}(s_{\mathbf{\lambda mk}})}{\sum_{\mathbf{\lambda}=1}^{\mathbf{g}} \text{Exp}(s_{\mathbf{\lambda 1}})}$$

c) Square of Mahalanobis distance of case $\mathbf{k}$ in group $\mathbf{m}$ from group

$$D_{\mathbf{\lambda mk}}^2 = \sum_{\mathbf{i}=1}^{\mathbf{r}} \sum_{\mathbf{j}=1}^{\mathbf{r}} (x_{\mathbf{mki}} - \overline{x}_{\mathbf{\lambda i}}) a_{\mathbf{ij}} (x_{\mathbf{mkj}} - \overline{x}_{\mathbf{\lambda j}})$$

This may be used as a chi-square variable with $\mathbf{r}$ degrees of freedom for classification purposes.

**Nature of Parameters Used**

During the course of the research, forty-six parameters were developed and tested for their discriminating capabilities. Some were not seriously
considered to have any discriminating capabilities, but they were included to provide a rationale for a few of the derived parameters. This section discussed the general nature of the parameters used, while the specific parameters tested are described in the next section, Parameter Definitions.

By way of introduction, it may be noted that many of the parameters can be considered as geometric means of other parameters, but that no corresponding algebraic mean of parameters is a linear function of the parameters, but that a geometric mean is a non-linear function. Since the discriminant analysis technique generates linear functions, any arithmetic mean will normally be generated if it is a significant discriminator. However, the corresponding geometric mean would not be generated due to the non-linearity inherent in the process of finding it. It should also be noted that the parameters to be tested were not rigorously and mathematically derived, but were generated as a result of observation and intuition.

Since the comparison list plays an important part in the generation of parameters, a brief description is in order. (More detail on its construction may be found in Chapter IV, in the section Vocabulary Profile). The comparison list is a list of words whose frequency of occurrence in a user's corpus, (i.e., the set of abstracts or documents judged relevant by a user) is sufficiently high so that the words can be considered to be indicative of the user's interests. Since the comparison list, by its nature, does not contain all the words in the user's corpus, the failure of a word in an abstract to be found on the comparison list does not mean that it can never be used in relevant documents, only that it may be infrequently used. The ratio of the
number of tokens represented by the words in the comparison list to the total number of tokens in the user's corpus is $p$.

The parameters themselves can be categorized into four basic types with an artificial fifth type which consists of ad hoc modification to two of the first four types. The four basic types of parameters are:

1) the counting parameters,
2) the parameters which measure the degree of correspondence between the vocabulary of the abstract and the comparison list,
3) the parameters which compared the repetition of words in the abstract to the expected repetition of the words in the comparison list, and
4) the chi-square parameters.

The counting parameters are simply the four parameters which one may obtain by counting the words in an abstract ($W$), the tokens in an abstract ($T$), the number of words common to the abstract and the comparison list ($M$), and the number of tokens in the abstract accounted for by the $M$ words ($C$).

Concerning the second parameter type, there are at least two ways of estimating the degree to which the vocabulary of the abstracts corresponds to the comparison list. One can assume that the ratio of tokens in the abstract accounted for by the matched words, $C$, to the total number of tokens, $T$, in the abstract will be approximately the same as the ratio, $p$, of tokens represented by the words in the comparison list to the total number of tokens in the user's corpus (i.e., $C/T$ is approximately $p$). Alternatively, one can consider that the words in the
vocabulary profile are those words which should appear frequently in any relevant abstract. Since, in an abstract, the number of tokens is fairly close to the number of words (e.g., in the ERIC files, the ratio of tokens to words was approximately 90:65), the ratio $M/W$ should also be fairly close to, although smaller than, the ratio $p$, of tokens represented in the comparison list to tokens in the corpus. Both estimates, however, have certain inherent problems. The ratio $C/T$ may be unduly biased if certain words have a higher token count than expected. This problem can arise when an abstract is concerned with a survey article in which many topics are covered in a general manner. In this case a few words are used frequently, but the majority of words are used only once. If the frequently used words are on the vocabulary profile, then $C/T$ may be very high relative to $M/W$. Conversely, if only the words occurring once are on the vocabulary profile, then $M/W$ may be high relative to $C/T$.

Since normally $C/T$ and $M/W$ will be fairly close in value, either ratio could serve as an estimate of the degree to which the vocabulary in the abstract corresponds to the vocabulary profile. However, if they differ markedly, it seems reasonable to assume that some type of average value might be a better estimator than either separately. The obvious type of average to check is the linear average, the weighted mean. There is, also, another type of average which is simple to calculate, the geometric mean, which has at least one advantage in the context of the experiment. The discriminant analysis program is not capable of generating non-linear functions of parameters. Yet it seems likely that non-linear combinations of parameters have some significance in the discriminating process. Since the geometric mean is a non-linear function of two or more parameters, it was included in the group of potential discriminators.
Once having made a decision to use a geometric mean, several other parameters could be generated, based on the degree of correspondence or degree of non-correspondence. Thus, the degree of non-correspondence, \((W-M)/W\)\(*((T-C)/C)\), becomes a new parameter, as does the relative correspondence, which is the difference between the degrees of correspondence, \(CM/TW\), and the degree of non-correspondence. Still another possibility involves scaling. Since geometric means are well ordered, their squares are also well ordered. However, squaring is a non-linear function. Thus, when parameters of the geometric-mean type were computed, their square were also computed in order to determine which would be the best discriminator. In each case where the square was used, it was a better discriminator than the original mean. Since the means, differences in means, the products of means were all less than one in absolute value, it was also decided to invert some of the parameters.

The third parameter type measures word redundancy. The second parameter, \(W/T\), is a measure of the redundancy of all the words in the abstract, while the first parameter, \(M/C\), is an indication of the redundancy of the matched words. For example, if \(W/T = 50/100\), then one could say that on the average, every word occurred twice. If \(M/C\) were \(25/75\), then it would be clear that half the words comprise three-fourths of the tokens in the abstract, and, more significantly, that those words are on the comparison list.

The fourth basic parameter type, the chi-square \(\chi^2\) parameters, is derived as follows:

Let \(X = \sum_{j=1}^{r} \frac{(E_j - E(E_j))^2}{E(E_j)}\)
where \( r \) is the number of categories (cells), \( Z_j \) is the number of events assigned to the \( j \)th category, and \( E(Z_j) \) is the expected number of events assigned to the \( j \)th category. If the hypothesis under which the categories expected values of occupancy is not false, then \( X \) should be distributed as a \( \chi^2 \) variable. Let \( X \) be defined as above, and define

\[
f(g, y) = \frac{y^{g-2} e^{-y/2}}{2^{g/2} \Gamma(g/2)}
\]

where \( g \) is fixed, \( y \) is a continuous variable and \( \Gamma \) is the gamma function.

Then \( P(X) = \int f(g, y) dy \) is the probability that the computed value of \( X \) would have been less than or equal to its actual value, if \( X \) were indeed a \( \chi^2 \) variable (i.e., the hypothesis of expected values is correct.)

The three \( \chi^2 \) variables to be generated, then, are \( X \), \( P(X) \), and \( f(g, x) \), the value of the function when \( x = y \). A fourth variable was generated by taking the reciprocal of \( X \). The rationale was that for very many degrees of freedom, \( X \) would be greater than one, and the reciprocal would provide a rapid way to convert it to a \((0, 1)\) scale while inverting the order of the individual parameter values. Similarly, with very few degrees of freedom, where \( X \) is less than one, the reciprocal would have a value greater than one, but again the order would be inverted.

In the list of parameters defined in the next section are found what may appear to be a fifth group of parameters. These are modifications of those in the second and third groups above, and were generated as a result of initial experiences with very short abstracts. When an abstract has fewer than twenty tokens, with only a dozen words represented, the effect of a single word has a significant effect on the ratios of \( C \) and \( M \), since
it affects both C and M. This is not true of long abstracts, since the ratios tend to stabilize (e.g., 50/100 is not appreciably different from 51/101 or 40/99, whereas 5/6 is appreciably different from 6/7 or 4/5.) The problem was alleviated by the addition of a constant to every denominator of each parameter is groups two and three.

**Parameter Definitions**
(* denotes the selection of the parameter as one of the final twenty-seven discriminators.)

1. W Number of words in the abstract.
2. * T Number of tokens in the abstract.
3. * MAX (.5,M) The number of words common to both the abstract and the comparison list. If this were zero, then a value of .5 was used to prevent dividing by zero in the generation of other parameters.
4. MAX (.5,C) The number of tokens in the abstract accounted for by the M words. Again, .5 was used when C was zero.
5. C/T If one assumes that a given document is relevant, and that its vocabulary is similar to that of the comparison list, then it seems reasonable that C/T would be close to the value used for cutoff in constructing the comparison list. This variable was chosen to see if this would make a good discriminator.
6. M/W This variable was chosen on the same basis as variable 5.
7. * W/T This variable was selected to see if some measure of redundancy in the text of the abstract could be used to provide a correction factor for extremely redundant abstracts.
8. \( \frac{M}{C} \)  
In an abstract in which a few words are used numerous times, some ratio variables (e.g., variables 6 and 7) may have rather high values even though \( M \) is relatively small. Like variable 7, this variable represents an attempt to correct this problem.

9. \( \frac{C}{T+100} \)  
Very short abstracts normally contain primarily high frequency words, thus potentially producing very high values to ratios (e.g., variables 5-8). The addition of 100 to all denominators tended to reduce this effect.

10.\( \frac{M}{W+100} \)  
Variable 6 modified as explained for variable 9.

11.\( \frac{M}{C+100} \)  
Variable 8 modified as explained for variable 9.

12. \( \frac{M}{W} \times \frac{C}{T} \)  
The variables \( \frac{M}{W} \) and \( \frac{C}{T} \) (variables 5, 6) tended to be fairly close in value, but \( \frac{C}{T} \) was usually larger than \( \frac{M}{W} \). By multiplying the two, and taking the square root, a geometric mean was found. This mean could be considered as measure of average goodness of fit between the abstract vocabulary and the vocabulary profile. However, geometric means are well ordered (in the range 0-1), and their squares are also well ordered. Therefore, the square root was not taken.

13. \( \left( \frac{M}{W} \times \frac{C}{T} \right)^{1/2} \)  
This variable, the actual geometric mean goodness of fit measure. Taking the square root was a good method for increasing the value of variable 12 in a non-linear manner.

14.\( \frac{M}{(W+100)} \)  
Variable 12 modified as explained for variable 9.

15. \( \left( \frac{M}{(W+100)} \right)^{1/2} \)  
Parameter 13 modified as explained for parameter 9.
16. \( \frac{(W-M)}{W} \times \frac{(T-C)}{T} \) If parameter 12 could be considered a mean value for degree of correspondence of a document's vocabulary to the comparison list, this variable could be considered a mean value for degree of non-correspondence.

17. \( \left( \frac{(W-M)}{W} \times \frac{(T-C)}{T} \right)^{\frac{1}{2}} \) The geometric mean of the degree of non-correspondence measures.

18.* \( \frac{(MC/WT)}{(MC/WT) - \left( \frac{W-M}{W} \times \frac{(T-C)}{T} \right)} \) The first portion of the variable (i.e., \( \frac{MC}{WT} \)) is variable 12. The remaining portion is the difference between goodness and poorness of correspondence of the abstract to the comparison list and can be considered as a measure of relative correspondence. The difference reduces to \( \frac{(M/W+C/T-1)}{} \), and this is the form which will be used in later explanations. Effectively, then, this measures variable correspondence as modified by a factor which measures relative correspondence.

19.* \( \left( \frac{(W-M)}{W} \times \frac{(T-C)}{T} \right) \times \left( \frac{M}{W+C/T-1} \right) \) This variable is the degree of non-correspondence analogue of variable 18.

20. \( \frac{M}{(W+100)} + \frac{C}{(T+100)} - 1 \) The second portions of variables 18, 19 modified as explained for variable 9.

21.* \( \frac{(MC/WT+100)}{(C/(T+100)+M/(W+L))} - 1 \) Variable 18 modified as explained for variable 9.

22.* \( \frac{(W-M)}{(W+100)} \times \frac{(T-C)}{(T+100)} \) This is the degree of non-correspondence variable modified as explained for variable 9.

23. \( \frac{M}{W+C/T-1} \) This is the difference between the correspondence parameter and the non-correspondence parameter.
24. \((M/W+C/T-1)^{-1}\) This parameter represents an attempt to scale parameter 23 to values greater than \((-1, 1)\).

25.* \((M/W+100)+C/(T+100)-1\) This is parameter 24 modified as explained for parameter 9.

26. \((WC)/(TM)\) This was an outgrowth of speculation that there might be a relationship of the form \(W/T = K(M/C)\) which could take repetitiveness on an abstract into account when deciding that there was a fit with the comparison list.

27. \(((W-M)/W*(T-C)/T)-1\) This was an attempt to scale parameter 16 to a scale other than \((0, 1)\).

28.* \(W/T-M/C\) Parameter 28 and 29 represented two methods of solving the problem of a few words representing a disproportionate number of tokens in an abstract.

29.* \(C/M-T/W\) Similar to earlier parameter, this represents an attempt to get a geometric mean on repetitiveness.

30. \(M/T/C\) This parameter has a rationale similar to that of variable 26. In this case the generating equation was \(K(C/M) = (T/W)\), and the parameter to be calculated was \(K\).

31.* \(((T-C)/C)*(W-M/M)\) This variable is similar to variable 16 in that it is a rating of non-correspondence.

32. \(T/M/C\) Similar to parameters 28 and 29, this was a method of estimating repetitiveness.

33.* \(C/T-M/W\) If, in parameter 32, \(K = 1\), then \(C = TM/W\). This parameter was a check on this possibility.

34.* \(T/M/W\) This variable was an outgrowth of the previous variable. If the geometric mean of \(M/W\) and \(C/T\) could be considered
an indication of how well the abstract vocabulary
corresponded to the comparison list, then the
geometric mean of \((W-M)/W\) and \((T-C)/T\) could be
considered an indication of how poor the mismatch
was, as could its square. Subtracting the mismatch
factor from the match factor might give some
indication of the relativematch significance. The
mismatch factor is variable 8.

36.* \(WM/(TC+100)\) This parameter is parameter 30 modified as explained
for parameter 9.

37.* \(WC/(MT+100)\) This parameter is parameter 26 modified as explained
for parameter 9.

38.* \(TM/WC\) This parameter was the reciprocal of parameter 32.

Parameters 39 through 46 are \(X\) values using two different assumptions.

The first assumption was that in a relevant abstract, 75 per cent of the
tokens in the abstract would represent words appearing in the comparison list.
The second assumption was that not only was the first assumption correct,
but that additionally the token occurrences would be in the same ratio as in
the corpus of relevant documents. Parameters 39 through 42 pertain to the
first assumption, while parameters 43 through 46 pertain to the second.

39.*, 43.* \(P(X)\) the probability that the computed \(X\) value less than
or equal to its actual value if the hypothesis is
correct.

40.*, 44.* \(f(g, x)\) This parameter is the chi-square probability
distribution function, with \(g\) degrees of freedom,
evaluated at \(X\).
where \( n \) equals the number of categories and represents the actual number of cases in each category. As \( n \) increases, the limiting distribution for \( X \) is the chi-square distribution.

This parameter value is a method of changing the scale while inverting the order of the values.

**Evaluation Criteria**

Success, in document retrieval, is normally expressed in terms of precision and recall, although other measures have been proposed by Jackson (23), Salton (34), and others. Recall is the ratio of the number of relevant documents retrieved to the number of relevant documents in the collection. Precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved. There are several limitations concerning the use of these measuring parameters. First, although there is no clearly defined mathematical relationship between the two measures, experimental results suggest that the precision-recall relationships are inverse. Such a relationship requires the user to evaluate any retrieval system on the basis of optimization of the tradeoffs involved in a given situation. Thus, some users may be satisfied with a precision of .2 if the recall is .9, while others may use evaluation criterion that are considerably different (18). Another limitation is that to measure recall, the entire collection must be evaluated for each user. This is normally not practical, so one of the measures is at best a guess.

The major emphasis of this research was to find parameters which could be used to pseudoclassify documents. However, the value of the
parameters identified would be a function of the success attained in the pseudoclassification process. The method used for measuring the degree of success was to compute precision ratios for each pseudoclassification, but not to attempt to compute recall. The results obtained from the pseudoclassification processes would be compared to the results obtained from an index term search of the file, and subjective conclusions would be derived. Recall ratios could not be estimated, but the number of relevant abstracts retrieved by both approaches could be compared, and this comparison would be a measure of recall difference, since the ratios (if computed) would both have the same denominator. Because of these limitations of the recall and precision measures mentioned above, it was felt that any statistical tests performed on pseudoclassification and abstract retrieval could only be performed if a large number of users were involved, so that a broad spectrum of success evaluations could be obtained. Since only eight users were actually involved in the research, a statistical evaluation of system success was not attempted.

The evaluation criteria for determining which parameters can best discriminate between relevant and non-relevant abstracts were to be similarly subjective. Using the discriminant analysis program it would be simple to determine which parameters were the most frequently selected and the stage at which they were selected. Groups of parameters which would be consistently used in the first few steps would generally be considered to be good discriminators. Additionally, if certain parameters were consistently selected in later steps, but not
in the first few steps, they would be considered to be fair discriminators, especially if there appeared to be some way to improve upon them.
CHAPTER IV. IMPLEMENTATION OF THE TECHNIQUE

Corpus Generation

The first step in the research, after the data base selection and user recruitment was completed, was to obtain a corpus of relevant abstracts for each user. Although other approaches to acquiring a corpus of abstracts relevant to a user's needs were possible, the QUERY retrieval system, a software package available from Computer Resources Corporation (30) for searching the ERIC files, was used for this purpose. QUERY provides the capability of searching the text of the abstract and the index terms assigned to each document in addition to titles, authors and other data elements of potential interest.

In this study, each user's interests were identified by a set of index terms and a match function. When the appropriate combination of index terms specified by the match function was found, an abstract was retrieved from the file being searched. The retrieved abstracts were then analyzed by the users, and the search profiles (i.e., the set of search terms and the match function) were modified if necessary to improve the quality of retrieval. Normally a good search required at least three iterations before the precision and recall were acceptable.

In general, at least two iterations were required for the information user to be able to define his interests sufficiently well to find the proper index terms and combinations which would retrieve the relevant documents. This bears out Meadow's comment that users frequently don't know what they do want, only what they don't want (29). In only one case was it necessary to base the search entirely on the text of the abstract instead of on index terms (no index terms were available to
satisfactorily define the user's interest). Two of the searches did augment the index term search by searching abstracts for key words or phrases.

Although there were only eight users, there were twelve corpora generated because of multiple topics for several users, for which there was a lack of obvious overlap in the subject matter being processed. For clarity, all discussion will be presented as if there were twelve users instead of eight. The search profiles used for generating the twelve corpora (final iterations only) are shown in Table 1. The format of the searches is that defined in the QUERY User's Manual (30). (In the queries, "\+" means AND, "/" means OR and "\#" means NOT. For example, search OOGI asks for any abstract in which at least one of the words Arithmetic, Mathematics, Algebra, Geometry, or Calculus appear. See Chapter II, Retrieval Operations.)

Relevance Criteria

Relevance of documents to a particular information need has been studied at length by Cuadra and coworkers (15, 16) with few firm results because of the subjective nature of the evaluations involved. Because many aspects of this research were intertwined with the concept of relevance, some criteria for its evaluation had to be established.

Since the users were constructing corpora rather than searching for answers to specific questions, the following criterion was suggested for determining relevance, "Is it reasonable to assume that this document might be of interest in the future?" If the answer was "yes", then the document was to be deemed relevant. Although this criterion was broad enough to resolve many relevance decisions, some problems did arise
TABLE 1

USER SEARCH PROFILES

OOB1 (TALLY); O (+) QUESTIONING (TXT), (+ #) QUESTIONNAIRE (TXT), (+ #) QUESTION-ANSWER INTERVIEW (TXT), (+ #) TEST CONSTRUCTION (TXT), (+ #) QUESTIONABLE (TXT), (+ #) QUESTIONED (TXT), (+ #) EDUCATOR (TXT), (+ #) VOTATIONA (TXT), (+ #) DISTRIBUTIVE EDUCATION (TXT), (+ #) JUNIOR COLLEGE (TXT), (+ #) MEDICAL EDUCATION (TXT);

OOB2 (TALLY); J (+) ENVIRONMENTAL EDUCATION (TXT), ( ) CONSERVATION EDUCATION (TXT), ( ) OUTDOOR EDUCATION (TXT), ( ) ECOLOGY (TXT);

OOW1 (TALLY); J (AND) COMPUTERS (TXT), ( ) COMPUTER ASSISTED INSTRUCTION (TXT), ( ) COMPUTER BASED LABORATORIES (TXT), ( ) COMPUTER GRAPHICS (TXT), ( ) COMPUTER ORIENTED PROGRAMS (TXT), ( ) COMPUTER PROGRAMS (TXT), ( ) COMPUTER SCIENCE (TXT), (+) MATHEMATICS (TXT), ( ) MATHEMATICAL LOGIC (TXT), ( ) MATHEMATICAL CONCEPTS (TXT), ( ) MATHEMATICAL ENRICHMENT (TXT);

OOW2 (TALLY); J (AND) COMPUTERS (TXT), ( ) COMPUTER ASSISTED INSTRUCTION (TXT), ( ) COMPUTER BASED LABORATORIES (TXT), ( ) COMPUTER GRAPHICS (TXT), ( ) COMPUTER ORIENTED PROGRAMS (TXT), ( ) COMPUTER SCIENCE (TXT), (+) PROGRAMMING LANGUAGES (TXT), ( ) SCIENCES (TXT), ( ) SCIENCE PROJECTS (TXT), ( ) SCIENCE CURRICULUM (TXT), ( ) SCIENCE EDUCATION (TXT), ( ) SCIENCE INSTRUCTION (TXT), ( ) SCIENCE LABORATORIES (TXT);

OOSH (TALLY); J (+) COMPUTER (TXT), (+) MATHEMATICS (TXT), ( ) RESEARCH (TXT), ( ) LEARNING (TXT), ( ) INSTRUCTION (TXT), ( ) COGNITIVE (TXT), (+ #) TEACHER (TXT), (+ #) COST (TXT), (+ #) DISADVANTAGED (TXT), (+ #) LIBRARY (TXT), (+ #) SCHEDULE (TXT), (+ #) HANDICAP (TXT), (+ #) TELEPHONE (TXT), (+ #) MEDICAL (TXT), (+ #) MODULAR (TXT), (+ #) GRAPHIC (TXT);

OOST (TALLY); J (+) LOGICAL ADVANCE (TXT), ( ) SCIENCE HISTORY (TXT), ( ) TYPICAL ATTITUDES (TXT), ( ) TYPICAL ENTERPRISE (TXT), ( ) TYPICAL METHODOLOGY (TXT), ( ) TYPICAL LITERACY (TXT), ( ) UNDERSTANDING SCIENCE (TXT), ( ) REACTION INVENTORY (TXT), ( ) FACTS ABOUT SCIENCE (TXT), O ( ) NATURE OF SCIENCE (TXT), ( ) ATTITUDES TOWARDS SCIENCE (TXT), ( ) ATTITUDES ABOUT SCIENCE (TXT), J (+) TURAL RESOURCE (TXT), ( ) COMPUTER (TXT), ( ) AUTOMATION (TXT), ( ) SOCIAL PROBLEM (TXT), ( ) SOCIOECONOMIC INFLUENCES (TXT), ( ) SOCIAL CHANGE (TXT), ( ) SCIENCE HISTORY (TXT), ( ) SCIENTIFIC ATTITUDES (TXT), ( ) SCIENTIFIC ENTERPRISE (TXT), ( ) SCIENTIFIC LITERACY (TXT), ( ) SCIENTIFIC METHODOLOGY (TXT), K ( ) UNDERSTANDING SCIENCE (TXT), ( ) REACTION INVENTARY (TXT), ( ) FACTS ABOUT SCIENCE (TXT), O ( ) NATURE OF SCIENCE (TXT), ( ) ATTITUDES TOWARDS SCIENCE (TXT), ( ) ATTITUDES ABOUT SCIENCE (TXT);
TABLE 1 - Continued

OOH1 ((TALLY)); K (+) BIOLOGICAL SCIENCE (TXT), (//) CHEMICAL (TXT), (//) SCIENCE CURRICULUM (TXT), (//) PHYSICAL SCIENCE (TXT), (//) PROJECT PHYSICS (TXT), (//) ELEMENTARY SCIENCE STUDY (TXT), (//) INDIVIDUALLY PRESCRIBED (TXT), (//) MINNESOTA MATHEMATICS (TXT), (//) SCHOOL MATHEMATICS (TXT), (//) PROCESS APPROACH (TXT), (//) INQUIRY ROLE (TXT), (//) TIME (TXT), (//) BIOSPHERE (TXT), (//) MATTER AND ENERGY (TXT), (//) SCHOOL SCIENCE (TXT), (//) ASTRONOMY PROJECT (TXT), (//) CONCEPTUALLY ORIENTED (TXT), (//) STRUCTURED LEARNING (TXT), (//) COMPUTER ORIENTED (TXT), (//) GENERAL SCIENCE (TXT), (//) IMPROVED SCIENCE (TXT), (//) MAN MADE (TXT), (//) ID EA (TXT);

OOM1 ((TALLY)); J (+) ASTRONOMY (TXT), (//) CLIMATOLOGY (TXT), (//) EARTH SCIENCE (TXT), (//) GEOLOGY (TXT), (//) GEOPHYSICS (TXT), (//) METEOROLOGY (TXT), (//) OCEANOGRAPHY (TXT), (//) OCEANOLOGY (TXT), (//) PALEONTOLOGY (TXT), (//) PHYSICAL GEOGRAPHY (TXT), (//) PLANETARIUM (TXT), (//) SEISMOLOGY (TXT), (//) SOIL SCIENCE (TXT);

OOMR ((TALLY)); J (+) COLLEGE SCIENCE (TXT), (//) SECONDARY SCHOOL SCIENCE (TXT), (//) ABILITY GROUP (TXT), (//) ACHIEVEMENT (TXT), (//) DIFFER ENCE (TXT), (//) PLACEMENT (TXT), (//) EDUCATIONAL OBJECTIVES (TXT), (//) PREDICT (TXT), (//) TEST (TXT); G (/) JONES, K (TXT), (/) MAYER, H (TXT), (/) NAIBERT, Z (TXT), (/) WOODWARD, D, C (TXT), (/) HANSON, R (TXT); K (/) ANDERSON CHEMISTRY TEST (TXT), (/) TEST ON UNDERSTANDING SCIENCE (TXT); J (+ #) GRADE 7 (TXT), (+ #) GRADE 8 (TXT), (+ #) GRADE 9 (TXT), (+ #) TEACHING METHODS (TXT), (+ #) TEACHER CHARACTERISTICS (TXT);

OOG1 ((TALLY)); J (+) ARITHMETIC (TXT), (//) MATHEMATICS (TXT), (//) ALGEBRA (TXT), (//) GEOMETRY (TXT), (//) CALCULUS (TXT);

OOG2 ((TALLY)); J (+) CONSERVATION (TXT), (+ #) CONSERVATION EDUC (TXT), (+ #) HEARING CONS (TXT), (+ #) SOIL CONS (TXT);

OOG3 ((TALLY)); J (+) STATISTICS (TXT), (+) MATHEMAT (TXT), (//) MODEL (TXT), (//) SIMULATION (TXT), (//) PROBABILITY (TXT);

a See "QUERY--A Universal Search System" (30) for details on query construction and syntax.
(e.g., the problem of partial or marginal relevance). As a result, several users introduced a second criterion: to wit, "would it disturb me if I didn't see this document?" In these cases, only those documents passing the second test were considered relevant for this study. (All output was saved until conflicting judgments about the relevance of a given document were resolved.)

This approach did not, perhaps, give due consideration to the future needs of the users, but was essential in constructing the basic document collection from which the vocabulary profiles were generated. (Later processing did, in fact, retrieve many of the semi-relevant documents (i.e., those not passing the second test mentioned above), indicating that the earlier decisions had not be irrevocable.)

**Vocabulary Profiles**

Once the corpora of relevant documents were generated, the second step involved construction of the word comparison lists for each corpus. This was accomplished by decomposing all abstracts for a corpus into word and token counts and combining the results. As a means of excluding some low information words from the comparison list (e.g., is, as, of), an additional constraint was placed in the definition of a word; to wit, a word had to be at least three characters long. A stoplist was also used to delete fifty-two additional low information words (Table 2) from the word and token counts. Originally it was assumed that there was no need to introduce stoplist processing. However, early results in the research showed that for abstracts containing fewer than 125 tokens, the presence of the common words caused numerous non-relevant abstracts to be pseudoclassified as relevant. Since the processing of the longer
TABLE 2

STOPLIST USED FOR PSEUDOCALSSIFICATION OPERATIONS

<table>
<thead>
<tr>
<th>Also</th>
<th>Off</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>And</td>
<td>One</td>
<td>When</td>
</tr>
<tr>
<td>Any</td>
<td>Only</td>
<td>Where</td>
</tr>
<tr>
<td>Are</td>
<td>Seven</td>
<td>Which</td>
</tr>
<tr>
<td>Between</td>
<td>She</td>
<td>Will</td>
</tr>
<tr>
<td>But</td>
<td>Six</td>
<td>With</td>
</tr>
<tr>
<td>Did</td>
<td>Some</td>
<td></td>
</tr>
<tr>
<td>Eight</td>
<td>Ten</td>
<td></td>
</tr>
<tr>
<td>Five</td>
<td>Than</td>
<td></td>
</tr>
<tr>
<td>For</td>
<td>That</td>
<td></td>
</tr>
<tr>
<td>Four</td>
<td>The</td>
<td></td>
</tr>
<tr>
<td>From</td>
<td>Their</td>
<td></td>
</tr>
<tr>
<td>Had</td>
<td>There</td>
<td></td>
</tr>
<tr>
<td>Have</td>
<td>They</td>
<td></td>
</tr>
<tr>
<td>How</td>
<td>This</td>
<td></td>
</tr>
<tr>
<td>Its</td>
<td>Too</td>
<td></td>
</tr>
<tr>
<td>Less</td>
<td>Three</td>
<td></td>
</tr>
<tr>
<td>Made</td>
<td>Two</td>
<td></td>
</tr>
<tr>
<td>Make</td>
<td>Upon</td>
<td></td>
</tr>
<tr>
<td>More</td>
<td>Very</td>
<td></td>
</tr>
<tr>
<td>Much</td>
<td>Was</td>
<td></td>
</tr>
<tr>
<td>Nine</td>
<td>Well</td>
<td></td>
</tr>
<tr>
<td>Not</td>
<td>Were</td>
<td></td>
</tr>
</tbody>
</table>
abstracts seemed to be insensitive to the effect of common words, the need for a stoplist when processing full text documents is still debatable.

Construction of the comparison lists (hereafter called vocabulary profiles) was simple. A vocabulary profile is the minimal set of words whose tokens comprise some fixed per cent of all tokens used in the corpus of relevant documents. If there is a tie on frequency of token occurrences when the cutoff percentage is attained, all words with the same token occurrence will be included in the profile, with a resulting increase in the actual per cent of the corpus represented. For example, if "physical" and "physics" both occurred five times in the corpus, and a cutoff percentage of 75 per cent was attained on "physical", then "physics" and all other words occurring five times in the corpus would be included in the vocabulary profile. Three vocabulary profiles were generated for each user with cutoffs of 50 per cent, 65 per cent, and 75 per cent. The 65 per cent profile was discarded immediately, since it was identical in all but three cases to the 75 per cent cutoff. (Table 3 shows the number of words and tokens in each corpus, the number of words in both the 50 per cent and 75 per cent vocabulary profiles, and the actual per cent token coverage for each of the vocabulary profiles). For five of the corpora, the numbers of words and tokens were so small that the 75 per cent cutoff actually created a vocabulary profile consisting of the entire corpus because of the tie rule explained above. These corpora were augmented with additional abstracts from another file and the vocabulary profiles still comprised the entire corpus. Rather than augment again, the research was continued to determine what effect 100 per cent coverage would have on the final results. Portions
### TABLE 3
SUMMARY DATA ON VOCABULARY PROFILES

<table>
<thead>
<tr>
<th>Data Base Identifier</th>
<th>Number of Abstracts Used in Generating Vocabulary Profile</th>
<th>Number of Words in Corpus</th>
<th>Number of Tokens in Corpus</th>
<th>75% Cutoff Words/Actual Per Cent of Tokens Covered</th>
<th>50% Cutoff Words/Actual Per Cent of Tokens Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>27</td>
<td>1146</td>
<td>2748</td>
<td>480/75.8</td>
<td>181/51.4</td>
</tr>
<tr>
<td>B2</td>
<td>92</td>
<td>1991</td>
<td>6946</td>
<td>660/76</td>
<td>220/50.4</td>
</tr>
<tr>
<td>W1</td>
<td>58&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1667</td>
<td>5240</td>
<td>754/82.5</td>
<td>190/52.2</td>
</tr>
<tr>
<td>W2</td>
<td>25&lt;sup&gt;a&lt;/sup&gt;</td>
<td>980&lt;sup&gt;b&lt;/sup&gt;</td>
<td>2285</td>
<td>980/100</td>
<td>197/57.7</td>
</tr>
<tr>
<td>SH</td>
<td>67</td>
<td>1957</td>
<td>6324</td>
<td>865/82.7</td>
<td>203/50.4</td>
</tr>
<tr>
<td>ST</td>
<td>80</td>
<td>2184</td>
<td>7140</td>
<td>1026/83.8</td>
<td>293/54.2</td>
</tr>
<tr>
<td>H1</td>
<td>105</td>
<td>2220</td>
<td>9745</td>
<td>617/76</td>
<td>199/51</td>
</tr>
<tr>
<td>M1</td>
<td>86</td>
<td>2008</td>
<td>6993</td>
<td>639/75.4</td>
<td>228/52.1</td>
</tr>
<tr>
<td>MR</td>
<td>49&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1232</td>
<td>4431</td>
<td>420/77.1</td>
<td>155/52.5</td>
</tr>
<tr>
<td>G1</td>
<td>283</td>
<td>3514</td>
<td>24142</td>
<td>726/76.3</td>
<td>207/50.7</td>
</tr>
<tr>
<td>G2</td>
<td>31&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1157</td>
<td>3107</td>
<td>527/79.7</td>
<td>221/56.6</td>
</tr>
<tr>
<td>G3</td>
<td>22&lt;sup&gt;a&lt;/sup&gt;</td>
<td>801&lt;sup&gt;b&lt;/sup&gt;</td>
<td>1721</td>
<td>801/100</td>
<td>180/57.1</td>
</tr>
</tbody>
</table>

<sup>a</sup> These corpora were augmented with abstracts from another file in order to provide a large enough vocabulary to construct a profile, since the number of retrieved relevant abstracts was initially so low.

<sup>b</sup> In these cases, the vocabulary profiles still included all the words in the corpora. Rather than augment again, the profiles were constructed and the results so obtained were used.

<sup>c</sup> The database identifier codes are the same as those shown in Table 1, User Search Profiles.
of the vocabulary profile sorted by decreasing frequency is shown in Table 4. In practice the profile is alphabetically ordered.

**Decomposition of Abstract File**

While the vocabulary profiles were being constructed, the entire abstract file was reduced to word and token counts for each abstract, eliminating strings of fewer than three characters and using a stoplist to eliminate common words. The resulting words were sorted and combined so that each abstract was represented by a list of words together with their frequencies of occurrence within the document. Since all the parameters to be tested were based on word occurrence, this step eliminated a large amount of repetitive text searching.

**Final Parameter Selection**

Before it would be possible to determine which set of parameters would satisfactorily pseudoclassify the abstract file, it was necessary to generate an initialization corpus for each user corpus.

The first step in generating the initialization corpus was to evaluate each of the parameters for each abstract in the user corpus. Next, a set of abstracts known to be non-relevant to each user also had to be selected, and each parameter had to be evaluated for each abstract in the non-relevant set. The combination of the two sets was the initialization corpus, the need for which is due to the fact that the discriminant analysis technique requires parameter values for each element in each class. In other words, it is not sufficient to describe relevant documents alone; it is also necessary to describe non-relevant documents.
<table>
<thead>
<tr>
<th>WORD</th>
<th>TOKEN OCCURRENCE IN CORPUS</th>
<th>RELATIVE FREQUENCY IN CORPUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics</td>
<td>522</td>
<td>.02162</td>
</tr>
<tr>
<td>Program</td>
<td>234</td>
<td>.00969</td>
</tr>
<tr>
<td>Students</td>
<td>209</td>
<td>.00865</td>
</tr>
<tr>
<td>School</td>
<td>198</td>
<td>.00820</td>
</tr>
<tr>
<td>Mathematical</td>
<td>193</td>
<td>.00799</td>
</tr>
<tr>
<td>Materials</td>
<td>192</td>
<td>.00795</td>
</tr>
<tr>
<td>Teachers</td>
<td>166</td>
<td>.00687</td>
</tr>
<tr>
<td>Study</td>
<td>163</td>
<td>.00675</td>
</tr>
<tr>
<td>Student</td>
<td>143</td>
<td>.00592</td>
</tr>
<tr>
<td>These</td>
<td>140</td>
<td>.00579</td>
</tr>
<tr>
<td>Group</td>
<td>139</td>
<td>.00575</td>
</tr>
<tr>
<td>Programed</td>
<td>139</td>
<td>.00575</td>
</tr>
<tr>
<td>Teaching</td>
<td>136</td>
<td>.00563</td>
</tr>
<tr>
<td>Used</td>
<td>135</td>
<td>.00669</td>
</tr>
<tr>
<td>Education</td>
<td>124</td>
<td>.00513</td>
</tr>
<tr>
<td>Developed</td>
<td>121</td>
<td>.00501</td>
</tr>
<tr>
<td>Teacher</td>
<td>121</td>
<td>.00501</td>
</tr>
<tr>
<td>Project</td>
<td>111</td>
<td>.00459</td>
</tr>
<tr>
<td>Number</td>
<td>110</td>
<td>.00455</td>
</tr>
<tr>
<td>Test</td>
<td>108</td>
<td>.00447</td>
</tr>
<tr>
<td>Accelerated</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Activities</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Applied</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Apply</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Articulartion</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Assessed</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Association</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Basic</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Basis</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Best</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Calculated</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Can</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Changes</td>
<td>3</td>
<td>.00067</td>
</tr>
<tr>
<td>Characteristics</td>
<td>3</td>
<td>.00067</td>
</tr>
</tbody>
</table>

---

*a* See Table 1 for corpus identifier.

*b* Stoplist shown in Table 2 was used.
Once the twelve initialization corpora were generated, the discriminant analysis program was run against each initialization corpus, the results were analyzed. Of the forty-six parameters tested, eight were never used in the discriminating process, and another eight were used only once. Five parameters, however, were used in every case. Table 5 shows the frequency with which parameters were chosen, and Table 6 shows which parameters were chosen as discriminators for each corpus.

By eliminating all parameters which had been used fewer than three times, a set of twenty-seven parameters remained for further testing. The elimination criteria were arbitrary. It was felt that if a parameter was not chosen as a discriminator at least three times, it was probably not very good at discriminating. Additionally, every parameter occurring fewer than three times was highly correlated with a parameter which occurred more than three times. Thus, it was felt that deleting those parameters would not affect the final results in any significant manner.

When the twenty-seven final parameters were identified, the discriminant analysis program was used again, and the frequencies of the individual parameter selections were observed. Table 7 shows which of the twenty-seven parameters were selected for each corpus, together with their frequency of selection when only the twenty-seven parameters were used as compared to their frequencies of selection when the full set of forty-six parameters were used.

To check that the elimination of nineteen parameters had not degraded the discriminating power of the discriminant function, comparative results obtained from pseudoclassification of the initialization corpora
<table>
<thead>
<tr>
<th>Frequency Of Selection</th>
<th>Parameter Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13, 15, 16, 17, 23, 26, 32, 35</td>
</tr>
<tr>
<td>1</td>
<td>4, 8, 9, 12, 20, 24, 27, 30</td>
</tr>
<tr>
<td>2</td>
<td>1, 5, 6</td>
</tr>
<tr>
<td>3</td>
<td>10, 34</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>2, 7, 14, 22, 28, 42</td>
</tr>
<tr>
<td>6</td>
<td>38</td>
</tr>
<tr>
<td>7</td>
<td>3, 19, 29</td>
</tr>
<tr>
<td>8</td>
<td>37, 40</td>
</tr>
<tr>
<td>9</td>
<td>41</td>
</tr>
<tr>
<td>10</td>
<td>18, 31, 33, 36, 43</td>
</tr>
<tr>
<td>11</td>
<td>46</td>
</tr>
<tr>
<td>12</td>
<td>21, 25, 39, 44, 45</td>
</tr>
</tbody>
</table>
**TABLE 6**

**FREQUENCY OF PARAMETER SELECTION IN DISCRIMINANT ANALYSIS (46 PARAMETERS)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B1</th>
<th>B2</th>
<th>W1</th>
<th>W2</th>
<th>SH</th>
<th>ST</th>
<th>H1</th>
<th>M1</th>
<th>MR</th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Selected as one of the 27 discriminators
TABLE 6 - Continued

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B1</th>
<th>B2</th>
<th>W1</th>
<th>W2</th>
<th>SH</th>
<th>ST</th>
<th>H1</th>
<th>M1</th>
<th>MR</th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>11 *</td>
</tr>
<tr>
<td>36</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>8  *</td>
</tr>
<tr>
<td>37</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>6  *</td>
</tr>
<tr>
<td>38</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>12 *</td>
</tr>
<tr>
<td>39</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>9  *</td>
</tr>
<tr>
<td>40</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>8  *</td>
</tr>
<tr>
<td>41</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>5  *</td>
</tr>
<tr>
<td>42</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>10 *</td>
</tr>
<tr>
<td>43</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>12 *</td>
</tr>
<tr>
<td>44</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>12 *</td>
</tr>
<tr>
<td>45</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>11 *</td>
</tr>
</tbody>
</table>

* Selected as one of the 27 discriminators.
TABLE 7
FREQUENCY OF PARAMETER SELECTION
IN DISCRIMINANT ANALYSIS (27 PARAMETERS)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B1</th>
<th>B2</th>
<th>W1</th>
<th>W2</th>
<th>SH</th>
<th>ST</th>
<th>H1</th>
<th>M1</th>
<th>MR</th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>F27</th>
<th>F46</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>21</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>22</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>28</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>29</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>31</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>33</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>34</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>36</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>37</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>38</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>39</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>40</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>41</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>42</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>43</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>44</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>45</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>46</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

Number of Parameters Used: 19 18 18 19 18 18 17 19 18 17 20 18

---

a The frequency of selection of these parameters when all forty-six parameters were used is provided for comparison purposes.

b Parameter was selected as valid, then discarded as combinations of other parameters made original selection redundant.
### TABLE 8

**INITIALIZATION CORPORA PSEUDOCATEGORIZED BY 27 AND BY 46 PARAMETERS**

*(COMPARATIVE RESULTS)*

<table>
<thead>
<tr>
<th>Corpus&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Actual</th>
<th>Pseudoclassified 27 Parameters</th>
<th>Pseudoclassified 46 Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relevant</td>
<td>Non Relevant</td>
<td>Relevant</td>
</tr>
<tr>
<td>B1</td>
<td>Relevant</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>1</td>
<td>249</td>
</tr>
<tr>
<td>B2</td>
<td>Relevant</td>
<td>81</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>9</td>
<td>241</td>
</tr>
<tr>
<td>W1</td>
<td>Relevant</td>
<td>53</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>4</td>
<td>246</td>
</tr>
<tr>
<td>W2</td>
<td>Relevant</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>SH</td>
<td>Relevant</td>
<td>63</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>6</td>
<td>244</td>
</tr>
<tr>
<td>ST</td>
<td>Relevant</td>
<td>72</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>5</td>
<td>245</td>
</tr>
<tr>
<td>H1</td>
<td>Relevant</td>
<td>94</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>15</td>
<td>235</td>
</tr>
<tr>
<td>M1</td>
<td>Relevant</td>
<td>75</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>12</td>
<td>238</td>
</tr>
<tr>
<td>MR</td>
<td>Relevant</td>
<td>46</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>1</td>
<td>249</td>
</tr>
<tr>
<td>G1</td>
<td>Relevant</td>
<td>258</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>38</td>
<td>212</td>
</tr>
<tr>
<td>G2</td>
<td>Relevant</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>1</td>
<td>249</td>
</tr>
<tr>
<td>G3</td>
<td>Relevant</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>1</td>
<td>249</td>
</tr>
</tbody>
</table>

<sup>a</sup> See Table 1 for corpus identifiers.
using the complete set of forty-six parameters and the subset of twenty-seven parameters are shown in Table 8. As can be seen, there were only minor changes in the pseudoclassification results.

Pseudoclassification of the Abstract File

After pseudoclassifying the initialization corpora with the full and the reduced parameter sets, the coefficients of the reduced-parameter-set pseudoclassification equations were punched for each corpus. These coefficients were used to pseudoclassify the contents of one entire tape from the ERIC file (12,764 abstracts) for each corpus. The method used was described in the Discriminant Analysis section of Chapter III. Briefly, two functions were computed,

\[ f_1 = \sum_{i=1}^{27} c_i x_i + k_1, \]

and

\[ f_2 = \sum_{i=1}^{27} d_i x_i + k_2, \]

where the \( c \) and \( d \) were the coefficients generated under the assumption of class membership, and the \( x_i \) were the actual parameter values. If \( f_1 > f_2 \), then the abstract were assigned to the class associated with \( f_1 \), otherwise it was assigned to the class associated with \( f_2 \). In this research, \( f_1 \) represented the class of relevant abstracts, \( f_2 \) the class of non-relevant abstracts. Table 9 summarizes the results of these twelve runs. The results are discussed in the next Chapter.
TABLE 9

FINAL RESULTS

MASTER ABSTRACT FILE PSEUDOCALSSIFIED BY 27 PARAMETERS

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Actual</th>
<th>Pseudoclassified</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relevant</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>19</td>
<td>12714</td>
</tr>
<tr>
<td>B1</td>
<td>Relevant</td>
<td>119</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>169</td>
<td>12465</td>
</tr>
<tr>
<td>B2</td>
<td>Relevant</td>
<td>29</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>56</td>
<td>12676</td>
</tr>
<tr>
<td>W1</td>
<td>Relevant</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>2</td>
<td>12748</td>
</tr>
<tr>
<td>W2</td>
<td>Relevant</td>
<td>93</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>176</td>
<td>12491</td>
</tr>
<tr>
<td>SH</td>
<td>Relevant</td>
<td>116</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>218</td>
<td>12322</td>
</tr>
<tr>
<td>ST</td>
<td>Relevant</td>
<td>144</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>171</td>
<td>12438</td>
</tr>
<tr>
<td>H1</td>
<td>Relevant</td>
<td>103</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>249</td>
<td>12401</td>
</tr>
<tr>
<td>M1</td>
<td>Relevant</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>48</td>
<td>12683</td>
</tr>
<tr>
<td>MR</td>
<td>Relevant</td>
<td>414</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>501</td>
<td>11821</td>
</tr>
<tr>
<td>G1</td>
<td>Relevant</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>12</td>
<td>12729</td>
</tr>
<tr>
<td>G3</td>
<td>Relevant</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Non Relevant</td>
<td>0</td>
<td>12753</td>
</tr>
</tbody>
</table>

a See Table 1 for corpus identifiers.
CHAPTER V. RESULTS

Introduction

In general, the extent to which it was possible to pseudoclassify abstracts using word frequency techniques and appropriate similarity measures was encouraging. Twenty-seven of the forty-six variables described in Chapter III were identified as being of at least marginal importance in discriminating between relevant and non-relevant documents. The variables used fall into four groups, each of which will be discussed in separate sections to follow. The last three sections of the chapter deal respectively with a summary of the parameter selections; an alternative form of the retrieval (pseudoclassification) equation; and retrieval (pseudocalssification) results.

Counting Parameters

Of the four counting parameters described in Chapter III (parameters 1, 2, 3 and 4), two were selected as potential discriminators together with an estimator of a third. The two that were selected were T (parameter 2) and M (parameter 3). The selection of M as a discriminator is not too surprising since, other things being equal, the higher the number of matches, the more likely that the abstract is relevant. The reason for the inclusion of T is less obvious. The value of T is used in many of the parameter calculations as a means of scaling, that is, it accounts for differences in the lengths of abstracts. Apparently, it is also used as a means of correcting for those abstracts whose lengths differ markedly from the average length.

Although the counting parameter C (parameter 4) was not selected as a discriminator, its estimator, TM/W (parameter 34), was selected.
Vocabulary Correspondence Parameters

The vocabulary correspondence parameters (numbers 5, 6, 9, 10, 12 - 25, 27, 31, 35, Chapter III) furnished the best single discriminator, and also one of the more interesting discriminators. In all, eight parameters from this group (numbers 10, 14, 18, 19, 21, 22, 25, 31) were selected as discriminators. When the twenty-seven parameter discriminant analysis runs were made, parameter 18, \((MC/WT) \times (MC/WT) - (W-M)/W(\frac{t-C}{T})\), was selected as the best discriminator (i.e., was used in the first step) for eight of the twelve runs. Table 10 shows the order in which each parameter was chosen for discrimination proposed in the forty-six parameter discriminant analysis runs, while Table 11 shows the same data for the twenty-seven parameter runs. Table 12 summarizes Table 11 in that it shows for each block of five steps (seven, for the last block) the number of times each parameter was chosen.

Parameter 14, MC/WT, was the most interesting discriminator because it was chosen as the best discriminator four times in the twenty-seven parameter runs, was chosen as the third best discriminator twice, and was never chosen after the third step.

Overall, one or the other of the above two parameters (numbers 14 and 18), which provide a measure of the correspondence between the vocabulary of the abstract and the user vocabulary profile was always the best single discriminator. The parameter measuring the degree of non-correspondence, \((W-M)/W(\frac{t-C}{T})\)/T (parameter 22), was never selected as one of the five best discriminators, but was selected as a discriminator eight times.
TABLE 10
ORDER OF PARAMETER SELECTION FOR DISCRIMINATION OF EACH
INITIALIZATION CORPUS (46 PARAMETERS)

<table>
<thead>
<tr>
<th>Step</th>
<th>Initialization Corpora²</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18 14 18 18 18 18 14 14 14 5 18 18</td>
</tr>
<tr>
<td>2</td>
<td>29 39 3 12 21 21 6 39 19 39 39 39</td>
</tr>
<tr>
<td>3</td>
<td>46 6 34 19 39 7 39 36 39 45 40 14</td>
</tr>
<tr>
<td>4</td>
<td>25 36 46 31 42 33 3 21 44 18 42 19</td>
</tr>
<tr>
<td>5</td>
<td>7 18 39 39 7 29 2 22 46 19 44 43</td>
</tr>
<tr>
<td>6</td>
<td>3 44 42 46 46 46 46 45 46 31 29 46</td>
</tr>
<tr>
<td>7</td>
<td>21 29 11 20 22 2 22 2 18 28 43 4</td>
</tr>
<tr>
<td>8</td>
<td>24 33 25 22 33 11 21 41 9 41 41 2</td>
</tr>
<tr>
<td>9</td>
<td>44 43 29 21 29 41 38 33 33 31 25 41</td>
</tr>
<tr>
<td>10</td>
<td>43 19 37 41 41 27 31 25 29 43 5 21</td>
</tr>
<tr>
<td>11</td>
<td>11 31 10 44 3 45 46 45 3 46 22 45</td>
</tr>
<tr>
<td>12</td>
<td>36 45 41 34 25 19 44 19 45 44 21 25</td>
</tr>
<tr>
<td>13</td>
<td>1 10 31 40 45 44 43 11 41 2 34 7</td>
</tr>
<tr>
<td>14</td>
<td>33 1 43 25 36 25 40 40 21 3 3 3 37</td>
</tr>
<tr>
<td>15</td>
<td>45 21 44 10 44 42 25 38 28 36 33 36</td>
</tr>
<tr>
<td>16</td>
<td>31 40 8 37 37 39 36 37 37 21 45 38</td>
</tr>
<tr>
<td>17</td>
<td>38 28 45 36 31 28 30 43 36 33 29 33</td>
</tr>
<tr>
<td>18</td>
<td>37 25 28 45 43 44 40 25 31 46</td>
</tr>
<tr>
<td>19</td>
<td>40 21 38 25 37 38 44</td>
</tr>
<tr>
<td>20</td>
<td>33 43 40 7</td>
</tr>
<tr>
<td>21</td>
<td>42 36</td>
</tr>
</tbody>
</table>

a See Table 1 for corpus identifiers.
TABLE 11
ORDER OF PARAMETER SELECTION FOR DISCRIMINATION OF EACH
INITIALIZATION CORPUS (27 PARAMETERS)

<table>
<thead>
<tr>
<th>Step</th>
<th>Initialization Corpora</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B1</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
</tr>
<tr>
<td>3</td>
<td>46</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>44</td>
</tr>
<tr>
<td>13</td>
<td>45</td>
</tr>
<tr>
<td>14</td>
<td>31</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>36</td>
</tr>
<tr>
<td>17</td>
<td>40</td>
</tr>
<tr>
<td>18</td>
<td>29</td>
</tr>
<tr>
<td>19</td>
<td>28</td>
</tr>
<tr>
<td>20</td>
<td>42</td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>

The minus (-) represents deletion of the parameter when new parameters rendered inclusion of original redundant.
### TABLE 12

**PARAMETER VERSUS RANK AS DISCRIMINATOR**  
*(DERIVED FROM TABLE 11)*

<table>
<thead>
<tr>
<th>Parameter Number</th>
<th>Steps 1-5</th>
<th>Steps 6-10</th>
<th>Steps 11-15</th>
<th>Steps 16-20</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>+2,-1&lt;sup&gt;a&lt;/sup&gt;</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>+1,-1&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>21</td>
<td>6</td>
<td>2</td>
<td>+1,-1&lt;sup&gt;a&lt;/sup&gt;</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>31</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>33</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>34</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Parameter Number</td>
<td>Steps 1-5</td>
<td>Steps 6-10</td>
<td>Steps 11-15</td>
<td>Steps 16-20</td>
<td>Total</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>39</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>41</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>44</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>46</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
</tbody>
</table>

The minus (-) represents deletion of the parameter when new parameters rendered inclusion of the original redundant.
Redundancy Parameters

The redundancy parameters (numbers 7, 8, 11, 26, 28 - 30, 32, 33, 36 - 38, Chapter III) measure the repetition of words in an abstract. Eight parameters from this group (parameters 7, 11, 28, 29, 33, 36 - 38) were selected as discriminators. Although one of them, C/T-M/W (parameter 33), was used in every run, none of these parameters was consistently in the first five discriminators. However, as a group, the parameters were used sufficiently often to indicate that a parameter which measures redundancy is necessary.

Chi-Square Parameters

As a group, the Chi-square parameters (parameters 39 - 46, Chapter III) performed best in discriminating between relevant and non-relevant documents. One of them (parameter 39), the probability that words in the abstract were chosen from the same universe as words in the vocabulary profile, was the second best discriminator five times and was also one of the five best discriminators nine times. It was one of several parameters chosen all twelve times. Parameter 43, the probability that the words in the abstract not only were chosen from the same universe as the words in the vocabulary profile, but also appeared with the same relative frequency as in the user's corpus, was also selected twelve times. However, it was not consistently selected as a good discriminator. The second set of $X^2$ variables (parameters 43 - 46) was undoubtedly affected by the lack of stem truncation and word variant form transformations. Overall, however, they were consistently chosen as discriminators.
Parameter Selection Summary

The parameters selected as discriminators for this pseudo-classification study, and their rank-orders as discriminators have already been summarized in Table 12. If one uses as a criterion for discriminating power the step in which a parameter is selected as a discriminator (i.e., the lower the step, the greater the discriminating power), Table 12 shows that only four parameters (numbers 14, 18, 21 and 39) were selected as discriminators six or more times in the first five steps, and a total of seven parameters (numbers 14, 18, 19, 21, 33, 34 and 39) were selected as discriminators six or more times in the first ten steps. Of the best four, three (parameters 14, 18, and 21) were of degree of correspondence parameters, while one was a chi-square parameter. The remaining three parameters were the estimator of C (parameter 34), a degree of non-correspondence parameter (number 19), and a measure of redundancy parameter (number 33).

A total of nine parameters (numbers 18, 21, 25, 33, 36, 39, 43, 44 and 46), however, were chosen ten or more times. Four of these parameters (numbers 39, 43, 44 and 46) were chi-square, three of which (parameters 43, 44 and 46) were based on the assumption that the occurrence of words in a relevant abstract would have the same relative frequency as the observed frequency of words in the user's corpus. The remaining five which were chosen as discriminators ten or more times were three (parameters 18, 21 and 25) degree of correspondence parameters and two (parameters 33 and 36) redundancy parameters.
Alternate Form of the Pseudoclassification Equation

The general form of the pseudoclassification equation was described in Chapter IV in the section Pseudoclassification of the Abstract File. An alternate form of the pseudoclassification equations is also possible if one converts some of the parameter forms to alternate forms. For example, the parameter for degree of non-correspondence of vocabulary in the abstract to the vocabulary profile (parameter 16) is

$$((T-C)/T)((W-M)/W).$$

This can be rewritten as $1-C/T-M/W+CT/MW$, or

$$1-(\text{Parameter 5})-(\text{Parameter 6})+(\text{Parameter 5})\times(\text{Parameter 6}).$$

Considering only the non-chi-square variables, the alternative form of the linear function finally obtained was:

$$f=k_1g^2+k_2a^2+k_3a/g^2+k_4/g^2+k_5/a+k_6a+k_7((N/W)^2+(C/T)^2)+k_8M/W +$$

$$k_9C/T+k_{10}WC/TM+k_{11}TM/WM+k_{12}TM/W+k_{13}W/T+k_{14}M/C +$$

$$k_{15}T/W+k_{16}WM/TC+k_{20}$$

where $a$ is the arithmetic mean of $M/W$ and $C/T$,

g is the geometric mean of $M/W$ and $C/T$,

$k_i$ are the coefficients for each term, and

$M, W, C, \text{ and } T$ are the original matching variables.

The first six terms in the alternative form of the equation represent combinations of the geometric and arithmetic means of the two estimators of abstract vocabulary fit to vocabulary profile. The seventh through the eleventh terms represent additional combinations of the two parameters. Terms twelve through fourteen are weighted sums of either the original variables or estimators of those variables,
and terms fifteen through nineteen represent combinations of the redundancy factor. (The first two terms in the equation actually are an alternative method of writing a weighted sum of parameters fourteen and eighteen, the two parameters which always furnished the best single parameter for discrimination).

Several observations can be made from this form of the equation. First, when a parameter is selected as a discriminator, its reciprocal is also frequently selected. Second, the number of terms representing each type of parameter tends to indicate that an optimal parameter of each type was not found. (However, the results of the pseudoclassification runs, see next section, indicate that acceptable parameters were found).

Retrieval Results

The results of the pseudoclassification runs were encouraging. Table 13 compares the number of abstracts retrieved by the QUERY system and the pseudoclassification system. Table 14 compares the precisions of retrieval by the two systems.

In ten cases, there was an increase in the number of relevant documents retrieved. Measurement of recall for retrieval results from an abstract file of the size used is difficult, since an evaluation of every abstract in the file must be made by each user. However, if one considers only the numerator of the recall fraction (the denominator is constant), recall results for two different systems operating on the same abstract file will be ordered in the same manner as the number of relevant documents retrieved.
TABLE 13
COMPARISON OF NUMBER OF RELEVANT ABSTRACTS
RETRIEVED BY THE QUERY³ SYSTEM
AND BY PSEUDOCLASSIFICATION

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>27</td>
<td>29</td>
<td>2</td>
<td>14.8</td>
<td>7.4</td>
</tr>
<tr>
<td>B2</td>
<td>92</td>
<td>119</td>
<td>11</td>
<td>41.3</td>
<td>11.9</td>
</tr>
<tr>
<td>W1</td>
<td>23</td>
<td>29</td>
<td>3</td>
<td>39.2</td>
<td>13.1</td>
</tr>
<tr>
<td>W2</td>
<td>14</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SH</td>
<td>67</td>
<td>93</td>
<td>4</td>
<td>44.7</td>
<td>5.97</td>
</tr>
<tr>
<td>ST</td>
<td>80</td>
<td>116</td>
<td>8</td>
<td>55.0</td>
<td>10.0</td>
</tr>
<tr>
<td>H1</td>
<td>105</td>
<td>144</td>
<td>11</td>
<td>47.6</td>
<td>10.4</td>
</tr>
<tr>
<td>M1</td>
<td>86</td>
<td>103</td>
<td>11</td>
<td>32.6</td>
<td>12.8</td>
</tr>
<tr>
<td>MR</td>
<td>28</td>
<td>32</td>
<td>1</td>
<td>17.8</td>
<td>3.68</td>
</tr>
<tr>
<td>G1</td>
<td>283</td>
<td>414</td>
<td>28</td>
<td>56.3</td>
<td>9.89</td>
</tr>
<tr>
<td>G2</td>
<td>16</td>
<td>22</td>
<td>1</td>
<td>43.9</td>
<td>6.25</td>
</tr>
<tr>
<td>G3</td>
<td>11</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

a See Bibliography citation 30.

b See Table 1.
TABLE 14

COMPARISON OF PRECISIONS OF RETRIEVAL OBTAINED BY THE QUERY\(^a\) SYSTEM AND THE PSEUDOCATEGORIZATION TECHNIQUE

<table>
<thead>
<tr>
<th>Corpus</th>
<th>QUERY Results</th>
<th>Pseudocategorization Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Retrieved</td>
<td>Relevant</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>43</td>
<td>27</td>
</tr>
<tr>
<td>B2</td>
<td>99</td>
<td>92</td>
</tr>
<tr>
<td>W1</td>
<td>36</td>
<td>23</td>
</tr>
<tr>
<td>W2</td>
<td>34</td>
<td>14</td>
</tr>
<tr>
<td>SH</td>
<td>152</td>
<td>67</td>
</tr>
<tr>
<td>ST</td>
<td>118</td>
<td>80</td>
</tr>
<tr>
<td>H1</td>
<td>127</td>
<td>105</td>
</tr>
<tr>
<td>M1</td>
<td>94</td>
<td>86</td>
</tr>
<tr>
<td>MR</td>
<td>47</td>
<td>28</td>
</tr>
<tr>
<td>G1</td>
<td>459</td>
<td>283</td>
</tr>
<tr>
<td>G2</td>
<td>21</td>
<td>16</td>
</tr>
<tr>
<td>G3</td>
<td>17</td>
<td>11</td>
</tr>
</tbody>
</table>

See Bibliographic citation 30.

See Table 1
Disregarding two sets of the results (i.e., W2, G3), the precision figures are not too surprising. The increase in recall was paid for by a drop in precision, the largest drop being in those cases where the QUERY precision was the highest. In those cases (e.g., B2, M1) the high QUERY precision was obtained with only a few test queries, and relevance assessments appeared to be fairly straightforward. In the cases where QUERY precision was low (e.g., SH, G1), the relevance assessments were apparently difficult, and four or more test queries had to be generated to get as high precision as was finally obtained.

When problems of this type arise, the reason may be that the information user requires information which is not readily partitioned by the existing index. That is, the index was designed from a point of view which is not completely compatible with the user's needs. In these cases in particular, the pseudoclassification technique did retrieve a significant number of abstracts which had not been retrieved by QUERY with only a small drop in precision.

Five of the initialization corpora and vocabulary profiles were generated in a slightly different manner from the remaining seven, with interesting effects on the final results. For the five corpora in question, the original searches retrieved only a few relevant documents. The corresponding vocabulary profiles contained 100 per cent of all the words in the retrieved documents. (This occurred even though a 75 per cent cutoff was used because of the tie-handling rule discussed in Chapter IV under Vocabulary Profile.) In order to get larger initialization corpora, each corpus was augmented by additional relevant documents retrieved by running QUERY against an additional abstract file.
In two cases, (G3 and W2), there were not enough additional abstracts retrieved to lower the percentage of corpus words in the vocabulary profile. Nevertheless, these two corpora were retained for processing to see what the results of pseudoclassification would be. The results were not too surprising, since many of the parameters were based on the presence or absence in the vocabulary profile of words in the abstract being classified. For these two initialization corpora, many of the parameters had variance of zero for the group of known relevant documents, so that the discriminant analysis program could have terminated after one iteration with all documents in the initialization corpora properly pseudoclassified. However, the program actually continues until certain F ratios have decreased to a predetermined value, a process which requires more than fifteen iterations in each case. In the case of G3, when the entire abstract master file was pseudoclassified, only the previously identified relevant abstracts were retrieved.

The three other cases where the vocabulary profiles and the initialization corpora were augmented gave satisfactory results, since additional relevant abstracts were retrieved in the pseudoclassification runs against the master file.

In one additional case (i.e., B1) the number of relevant documents retrieved by QUERY was fairly small (i.e., 28), but it was decided not to augment the vocabulary of the corpus. Since only three additional relevant abstracts were retrieved by pseudoclassification, it could not be determined whether this was a result of the original small corpus or because there were no more relevant abstracts in the file. However, based on a subjective interpretation of the results obtained from the
pseudoclassification runs in which the vocabulary profile was generated from fewer than thirty abstracts, the conclusion reached is that the minimum number of abstracts required for creating effective profiles for this process exceeds thirty.

Two questions arise from even a cursory glance at the pseudoclassification results on the master abstract file. What caused non-relevant documents to be retrieved? What caused the failure to retrieve known relevant documents?

The first question is the easiest to answer. The retrieved abstracts were similar in nature, but not similar in emphasis to the original corpus. For example, from a general point of view, an abstract referring to a document describing the establishment of a foreign language curriculum is similar to an abstract concerning the establishment of a mathematics curriculum, in that both articles deal with the topic of curricula. However, to a user interested in mathematics curricula, the first abstract is, at most, of marginal relevance, while the second is definitely relevant. This type of relationship was responsible for most of the irrelevant retrievals. The second type of irrelevant retrieval was caused by abstracts which were so general that they could refer to anything. An example is the abstract referring to a collection of articles or to a comprehensive bibliography.

The known relevant abstracts which were not retrieved usually were of the marginal type. That is, either some of the index terms for the document seemed to have little direct bearing on the text of the abstract, or the general subject of the abstract was not the general subject in which the information user was interested. An example of
this is a case in which the user judged an abstract in music education because the technique described seemed interesting from his point of view. Although many of these tangential abstracts were retrieved, it seems probably that more were not.

One method by which any new system for retrieving documents or abstracts can be evaluated is to consider the number of documents or abstracts retrieved by the new system, but not retrieved by a standard system. Additionally, one should consider the number of documents or abstracts which were not retrieved by the new system but which were retrieved by a standard system. In this experiment QUERY was the standard system with which the word frequency pseudoclassification system was compared. Generally, about 10 per cent of the known relevant abstracts were not retrieved by word frequency pseudoclassification (Table 13). However, there was a corresponding increase of from 15 per cent to 55 per cent relevant abstracts retrieved by the word frequency pseudoclassification, but as mentioned earlier, this was paid for by a drop in precision which was expected due to the inverse relationship of recall and precision.

The twelve corpora could be easily divided into two groups, those for which there were fewer than thirty abstracts retrieved by QUERY, and those for which there were more than sixty-five abstracts retrieved. Overall, the results of word frequency pseudoclassification for the first group (i.e., B1, W1, W2, Mr, G2, G3) do not appear to be as good as for the second group (i.e., B2, SH, ST, H1, M1, G1). The two exceptions to these results (i.e., W2, G3) were the two cases where the vocabulary profiles consisted of all the words in the corpus and thus
affected most of the parameter values. In all other cases, more relevant abstracts were retrieved than were missed, as far as could be determined. The four corpora which had the smallest number of retrievals both in quantity and percentage (i.e., B1, W2, MR, G3) were among those which had the smallest number of relevant abstracts retrieved by QUERY. The three corpora which had the largest number of retrievals were those which had the largest number of relevant abstracts retrieved by QUERY. There are several, non-exclusive, possible explanations for this. First, it seems reasonable to assume that the greater the number of abstracts from which the vocabulary profile is constructed, the better the chances of getting a representative vocabulary profile for a field of interest. Second, if the number of relevant abstracts is large, the interests of the user may be very broad. If so, a larger number of abstracts are marginally relevant. Conversely, if the number of relevant abstracts is low, the relevance criteria are probably fairly narrow, and the number of marginally relevant abstracts would be drastically reduced. Finally, it may be that some fields are referred to so seldom that any reference to them in a document causes an index term to be generated, whereas other fields are so common that a document must be primarily oriented towards them to warrant an index term assignment.

Finally, some consideration should be given to the general type of abstracts retrieved by word frequency pseudoclassification in assessing the final results. Few of the abstracts appeared to be misindexed. Thus, few of the abstracts were in the category of abstracts which should have been retrieved by QUERY but were not because a particular
index term was not assigned. Most of the relevant new abstracts were in the marginally relevant category. That is, the primary subject of the abstract was not the user's area of interest, but there was a high degree of overlap with the user's interests. For example, one of the users was interested in conservation and ecology. Many of the abstracts retrieved were concerned with forestry, and about one-third of these were relevant.

Although the retrieval of marginally relevant abstracts may not be desirable for all users, there is one class for whom it is important. These are the users whose areas of interest cannot be easily described by the index. These users may require that any abstract with any potential relevance be retrieved in order that they can be reasonably sure of getting a satisfactory number of relevant abstracts. There were two corpora of this type (i.e., SH and ST), and in both cases the number of new retrievals was sufficiently high to show that the word frequency pseudoclassification could be used with satisfactory results.
CHAPTER VI. CONCLUSIONS

Introduction

The immediate goal of the research was to show that some combination of statistical measures and word frequency techniques could be used to pseudoclassify a document collection satisfactorily for individual users. The experimental results showed that such an approach is feasible, in that good recall is obtained, but at the cost of low precision. The following questions then arise:

1) Under what circumstances should such a system be used?

2) How can the recall be maintained while increasing precision, if necessary?

The answers to both questions are not simple, since they involve factors such as cost; user interaction required to generate an initialization corpus as well as to review the results of such a process; adequacy of the files being searched; and the uses to which retrieved documents (or surrogates) are to be put. Each of the above factors will be discussed in the next few sections followed by some suggestions for future research and possible applications of the technique.

Cost Factor

Comparisons of cost can be misleading when one considers the hidden penalties inherent in missing relevant documents and the time costs of preparing and, possibly, modifying the match function for index term searches (e.g., as for QUERY). These costs are separate from the financial aspects of actually performing the document retrieval function. Because these hidden costs can vary so much for each user, and because the financial costs of performing an automated search using standard
programs fluctuates greatly with the complexity of the match function used, no attempt can be made to compare the total costs of using the pseudoclassification approach with the costs of any existing system. Instead, some figures will be given on the actual running time of the programs used together with some comments on methods of improving these times.

There were six basic programs used in the research, five of which were written in PL/1 as a result of this research (see Appendix B), while the other was a program already available on campus (see Chapter III, section on Discriminant Analysis). The main program, which performed the pseudoclassification and retrieval process, required approximately six minutes to handle 12,762 abstracts on an IBM System 370/165 for each user. Interestingly, the size of the vocabulary profile, a measure of the number of word comparisons required for each abstract did not significantly affect the run time. The run time difference between using a profile of 420 words and using a profile of 1,026 words was about thirty seconds.

Several comments are in order about the above data. First, the program seems to be input/output bound. This seems to be supported by the fact that the running time was only about two minutes longer when the program was run on an IBM System 360/75. (However, the use of different timing algorithms on the 360/75 and the 370/165 makes any such comparisons a little less meaningful). Second, six minutes seems to be too long a time to process approximately 13,000 abstracts for a single user. A method of reducing the processing time and reducing the impact of the input/output operations would be to program the system in a symbolic language. This would have the advantages of eliminating
inefficiencies inherent in a compiler oriented language and would allow better control over the read/write operations. Additional efficiencies would result if several pseudoclassifications could be batched in a single run. If the programs were rewritten as suggested, the run time should be less than two minutes for a single pseudoclassification run, with only slight increases in run time for each additional pseudoclassification performed in the same batch.

Additional processing costs were a function of the processing times required for generating the user vocabulary profiles, for generating the classification functions' coefficients, and for decomposing the abstracts in the collection to word and token counts. The time required to generate the vocabulary profiles was about three minutes for all twelve profiles handled in batch mode. The generation of coefficients required about forty-five seconds for an initialization corpus of 500 abstracts and twenty-seven parameters. The time required to reduce the abstract file to word and token counts for each abstract was approximately twenty minutes for a file of about 13,000 abstracts.

The first two costs would be incurred whenever the vocabulary profile was out of date or whenever it was felt that adding sufficient number of new abstracts in the initialization corpus would help refine the retrieval criteria. These are subjective evaluations and are discussed further in the section entitled Precision-Recall Considerations. The third cost is a one-time file processing cost, since the resulting file can be used for any additional pseudoclassifications desired. Overall, if the main pseudoclassification and retrieval programs were to be rewritten as suggested, the overall costs involved would not seem to be
too unreasonable compared to the operating costs of other systems now available.

User Interaction

The second factor, user interaction, is as critical for the success of the pseudoclassification process as it is for other retrieval systems. As mentioned previously, an initialization corpus is required for the approach, and the quality of the results is a function of the user's efforts in creating the corpus. Initially, the user must define his interests in terms of a minimum number of relevant abstracts. Then, he must also identify a sufficient number of non-relevant abstracts, since the combination of the two constitutes the initialization corpus.

The purpose of identifying relevant abstracts is obvious, but the identification of non-relevant abstracts is equally important because of the nature of the retrieval operations. For example, it is unlikely that an abstract of a document concerning school construction will be retrieved on the basis of pseudoclassification when the initialization corpus pertained to mathematics curriculum development. However, it is not unlikely that an abstract pertaining to the development of an English curriculum will be retrieved. If the user were not interested in the latter topic, some abstracts of this type should be identified as irrelevant in the initialization corpus in order to reduce the likelihood of their retrieval. The balance is delicate, however, because if he uses too many abstracts on a particular irrelevant topic in the initialization corpus, he may reduce his recall while striving to increase his precision.
The type of judgments required and the effort involved are no different from those required for other retrieval systems if one wishes to retrieve satisfactory results. The main difference is that with conventional systems, the judgments are always made a posteriori to an initial search query, and the feedback is used to modify the query, if necessary. In the pseudoclassification technique, a minimum number of such judgments must be made a priori in order to generate a user's vocabulary profile. In either case, the user must interact with the system if he wants to obtain satisfactory results.

The main advantage of the pseudoclassification technique is that it could preclude the need for an intermediary who must be familiar with both the query language (see QUERY Profiles, Table 1) and the index vocabulary of the system. For example, although the user corpora for this research were obtained via a QUERY search, they possibly could have been obtained equally well by having a user make such judgments on an issue or two of RIE and use the results of those judgments as his initialization corpora.

Adequacy of Files Being Searched

As suggested in Chapter III (section on Data Base and User Selection), file type and availability are factors which definitely affect the feasibility of the pseudoclassification approach described in this thesis. The research results suggest that abstracts can be pseudoclassified with reasonable effectiveness in the absence of full-text computer-readable files. As to how effectively the technique would work on files consisting only of standard bibliographic data (i.e., authors, titles, and journal references), even if augmented by index terms, this is difficult
to answer in the absence of further research. Since some problems were encountered with short abstracts, for which the use of stoplists had to be introduced, it is reasonable to expect that other problems would have to be dealt with on files containing only standard bibliographic data. The abstract file experiments conducted thus far suggest that a minimum of fifty words per abstract is necessary to produce reasonably effective results without having to incorporate refinements such as those to be discussed in the section on **Precision-Recall Considerations**.

Other file characteristics such as file size, subject matter content and index quality merit the usual consideration that applies for other automated retrieval systems. For example, if the only files available are so small that the user can manually scan all entries in a few minutes, the use of the pseudoclassification technique or any other automated search technique would hardly be worthwhile. In other words, files should be fairly large, that is, on the order of thousands of abstracts rather than hundreds before considering the use of an automated retrieval system. Also, there should be a high probability that the file will contain a sufficient number of relevant references to make the retrieval process worthwhile for individual users. As an extreme example, if the only available data base is concerned with chemistry, it is unlikely that one will find very many abstracts concerning school construction and space utilization.

Finally, for those files which are large enough to consider searching by automated techniques, it remains for the individual user to assess for himself the time-money tradeoffs involved in conducting a reasonably efficient search on any given topic whether it be done manually by
conventional subject heading (or keyword) matching techniques, automatically by conventional subject heading (or keyword) matching techniques, or by the automatic pseudoclassification technique described in this thesis. Some of the more pertinent time-money tradeoff factors influencing the user's decision in such a situation were touched upon in the previous two sections of this chapter.

Use of Retrieved Abstracts

There are several obvious ways in which the results of the word frequency pseudoclassification process could be used which are essentially the same as for other systems. First, if a user wished to create a file pertaining to his own interests for later reference, he could partition a large file (e.g., 20,000 abstracts) into a smaller file (e.g., 1,000 abstracts) which would be easier to work with. If the file were partitioned so as to contain items of marginal interest (i.e., to obtain high recall), he could then perform any type of text search he wished knowing that, at worst, he could only retrieve 1,000 abstracts. Alternatively, the file could be partitioned more narrowly (i.e., with higher precision and lower recall) to eliminate both the non-relevant items and the items of marginal interest.

Another way in which the results of pseudoclassification could be used is for current awareness searches. For example, pseudoclassification run could be made for each user whenever new updates to the abstract file became available. In this situation, the number of retrieved abstracts would be sufficiently small so that the user might not object to the time required to study the results, even if there were low precision, as long as he were sure of getting high recall. However, if
the user were being charged for every retrieved abstract, relevant or not, he may wish to lower recall in order to improve precision.

(Techniques to improve precision are discussed in the next section.)

Economics of charging aside, the word frequency pseudoclassification approach would seem to be a feasible approach to current awareness searches. Retrospective searches are also possible and feasible only if an initialization corpus can be generated on a subset of the complete abstract or document file, following which the entire file would be searched for relevant items by the word frequency pseudoclassification technique. Also, as more files are made available to a user, the pseudoclassification technique can be used to examine them for additional relevant documents.

**Precision-Recall Considerations**

As discussed in Chapter V, the retrieval results obtained by the pseudoclassification method showed an increase in recall at a cost of lower precision when compared to the corresponding results obtained by the QUERY system. There are several methods which can be used to alleviate the serious problem of low precision while holding the recall at a high level. First, some type of stem truncation could be attempted. This would decrease the size of the vocabulary profile, since most variant word forms would be grouped as one word. Consequently, the chi-square variables should also reflect more closely the degree of correspondence between the vocabulary profile and the text of the abstract under consideration, thus providing a better discriminating function. In addition, the decrease in size of the vocabulary profile would have the potential advantage of reducing the time required for parameter evaluation.
The approach to stem truncation should probably be algorithmic in nature (e.g., use only the first six characters of a word), so that an inordinate amount of processing is not required. Although any algorithmic approach will create artificial homographs and will not combine all word form variations, overall there should be a net benefit.

The second method for improving precision is to raise the retrieval criteria. In this experiment, all abstracts which had a fifty per cent probability of being relevant were retrieved. However, the probability was calculated on the basis of the parameter values of the initialization corpus. In practice, the probability that a document was relevant was less than the computed value. Thus, one could retrieve only those abstracts whose computed probability of relevance was greater than some arbitrary value (e.g., 75 per cent). An alternative approach would be to select only a fixed number of abstracts whose computed probability of relevance was highest. This approach would reduce the recall ratio, however, and its implementation should be used with discretion after discussion with the information user.

A third method for improving the precision ratio is to use a feedback from initial pseudoclassification runs. If, as was the case for this research, the initialization corpus for the first pseudoclassification run were generated from the results of the text search retrieval, the portion of the initialization corpus pertaining to non-relevant abstracts might not give a true picture of abstracts which are similar in content to relevant abstracts. However, the results of the pseudoclassification run could be used to augment this portion of the initialization corpus in order to produce a more accurate representation of non-relevant abstracts. The potential pitfall in this approach is
that the fine dividing line between relevant and non-relevant abstracts could result in an overcorrection whereby the recall ratio would decrease too much. For this reason, augmentation should proceed slowly, in small steps, and the user should be consulted at every point.

Although augmentation seems like a time-consuming process, it does not differ in concept from the usual procedures of going through several iterations to develop an acceptable text search strategy, but with the important distinction that such iterations would normally be eliminated. The approach would be to run a text search, use the output to create the vocabulary profile, establish an initialization corpus, generate the parameter coefficients and pseudoclassify the abstract file. The feedback iterations could then take place when the user was more aware of the contents of the abstract file, and was thus in a better position to determine what was available.

The last method for improving precision would be to make use of titles and index terms (if they are available). Although the research reported in this thesis was oriented towards showing that word frequency techniques based only on the text of abstracts themselves could be used to retrieve relevant abstracts or documents from a file, from an information retrieval point of view it seems reasonable that any additional information about a document should be used if it is available. Thus, concatenating index terms to an abstract would be making use of an additional resource, which represents a supplementary and sometimes independent intellectual evaluation of a document's content. Similarly, augmenting the abstract with the document title would seem to be taking advantage of the author's evaluation of the document's content. It may
also be desirable to provide some weighting of the index words and the words in the title. For example, the occurrence of a word in either or both of these locations could count as two token occurrences.

Possible Applications

Several applications of this automatic pseudoclassification technique have already been covered in the section on Use of the Retrieved Abstracts. These embody the more customary applications of creating private document files, current awareness searching and retrospective searching. Another possible application suggested in Chapter II would be to use the technique to help in the document assignment or subclassification problem. Consider information processing systems such as ERIC, Chemical Abstracts, and the American Institute of Physics, etc. which process large numbers of documents pertaining to broad areas of knowledge. Each of these broad areas are partitioned into more specific subject areas (i.e., nineteen for ERIC, eighty for Chemical Abstracts, and forty-nine for the American Institute of Physics) for printed publications. These partitionings could conceivably be accomplished by a variation of the pseudoclassification technique described herein, particularly in view of the fact that most of these organizations are now processing their documents (abstracts) in computer-readable form. This particular application would not specifically be one of pseudoclassification, since more than two classes would be involved in the partitioning process. However, since some of the same general principles are involved, the approach used for pseudoclassification would appear to be applicable for these situations as well. In a somewhat related application, information centers involved in storage
and retrieval of large files of documents or abstracts may wish to set up sub-collections for subsequent processing in order to reduce the total amount of processing required.

Finally, as computer-aided typesetting becomes utilized by more organizations, and full text documents become more readily available in computer readable form, one might envision possible utilization of the pseudoclassification technique for purposes such as the above within the framework of a completely automated document analysis system which includes some form of automatic abstracting and automatic indexing as well (13, 14, 26, 32).

Suggestions for Future Research

Although suggestions for future research have been interspersed throughout earlier sections of this thesis, many of them are discussed again in this section simply to provide a cohesive treatment of the topic. The suggestions are grouped into three generally overlapping areas comprising assumption testing, parameter refinement, and implementation considerations.

One of the basic assumptions made in this research was that the vocabulary of a user's field of interest would tend to stabilize after his corpus reached a certain size. However, no assumption was made concerning the number of tokens required before stability would be attained. Consequently an investigation of the rate of convergence to a stable vocabulary would be worthwhile for several reasons. First it would help to determine how many documents or abstracts are required in order to obtain effective pseudoclassification results. If too small a corpus is used, the vocabulary profile will have word
occurrence frequencies that have not yet approached the frequency distributions of a stable profile thereby leading to a situation where either everything or nothing can be retrieved. Second, such an investigation might show how stability is attained. For example, it is reasonable to assume that the most frequently used words (e.g., those whose tokens comprise the top 25 per cent of the entire corpus vocabulary will attain stability faster than the remaining portion of the vocabulary profile. Investigations of this type may give some indications for methods of algorithmically selecting a vocabulary profile which encompasses only the stable portion of the corpus vocabulary. Finally, in many fields the vocabulary is changing dynamically as new words are introduced and old words change their meanings or are discarded. If this change is to be rapidly reflected in the vocabulary profile, which is a function of the number of tokens in the corpus of relevant documents, it follows, then, that the size of the corpus should be small enough (but not too small for reasons cited above) so that the new terminology can exert its effects on the retrieval operations as rapidly as possible.

A second assumption, that stoplists were unnecessary, was discarded early in the research when it was noted that some common words seemed to affect the retrieval of short abstracts (i.e., those containing fewer than 100 tokens). The retrieval of long abstracts was apparently unaffected. Since eliminating the short abstracts would have effectively reduced the abstract file by about 50 per cent, a stoplist was used. Nevertheless, some research could be directed towards determining the degree to which a stoplist is really required in document pseudo-classification.
The limits within which various forms of document representation can be effectively used for the pseudoclassification process should be more fully examined. For example, in the absence of full-text documents, computer-readable abstracts had to be used in this research, and the results appeared to be reasonably effective. However, since a greater number of computer-readable files are available which contain only standard bibliography data (i.e., authors, titles, and journal references) it remains to be seen how effectively titles could be used for the pseudoclassification process. Obviously, truncation techniques and additional stoplist words would be necessary before any attempt should be made on the use of titles, particularly in view of the results experienced with short abstracts.

Parameter refinement is another general area for study. The chi-square variables were, as a group, used more than any other type variables, although they were frequently in the marginal category (i.e., ranked eleventh or lower as discriminators). Several approaches could be used to improve their discriminatory ability and thus the efficiency of the pseudoclassification process. First, stem truncation (mentioned above) should improve the computed probabilities, since most of the forms of a word would be combined into one word, with a corresponding adjustment of the expected and actual frequencies of tokens. Second, a method for combining contingency table cells in the computed chi-square values for relative frequency of occurrence of tokens should be implemented. Since the set of chi-square variables used in this research proved to be satisfactory, no adjustment for expected cell sizes of less than five elements were made, although such adjustments
should improve the quality of the parameters as discriminators.

The non-chi-square variables also provide some indications for the direction of possible future research. These parameters which were all of the continuous type, were formed by taking simple ratios and, in some cases, by forming products or sums of ratios. However, the creation of step functions, binary variables, logarithmic variables or more complex arithmetic functions seem to be obvious extensions of the parameters generated. For example, the redundancy computations in the experiment (see Chapter V) were simple ratios; but, perhaps, they could be improved upon by using methods similar to those of Shannon (36). However, this would introduce a problem in the determination of expected probabilities, since redundancy in the context of the experiment was merely an estimate of the repetitiveness of tokens. Similarly, the introduction of step functions or binary values instead of using computed geometric means might be advantageous. Certainly, these, and other, modifications could be explored.

The final area for future research is a study of implementation considerations. This overlaps in part with the first area, in that the size of a vocabulary profile can be a problem if it gets to be too large, simply because of the core storage requirements. Other considerations might be to combine some or all of the PL/1 programs used and reprogram them in a symbolic language. Certainly, the economic feasibility of the approach described in this research would seem to demand this type of activity.

Another implementation consideration is the method of constructing the original corpus of relevant abstracts or documents. In this research,
the corpus was selected by a keyword search. An alternative approach would be to give the user a collection of abstracts and let him choose a corpus in that manner. This would seem to be a method of decreasing the user's possible bias towards some abstracts which can be caused by examining the index terms assigned to that abstract.

The final suggestion for research is that some method be found whereby the user could initiate this system without recourse to a programmer. The system developed during the research is somewhat unwieldy from a user's point of view. If elimination of an intermediary in the document retrieval process is desirable, then systems must be designed which are easily implemented by a user who has no desire to learn a complex procedure.

Of the three general areas for research, the first two (i.e., assumption testing, parameter refinement) seem to be the most critical. A great deal of work needs to be done, and many tests with users need to be performed, before implementation in a fully operational sense is considered.

Summary

In attempting to retrieve information from a collection of documents, a major problem is to describe one's interests in a manner which coincides with the way the documents are described in the retrieval sources (i.e., to identify the set of terms used to describe the documents in an index) (Chapter I). Since word-occurrence methods based on Zipf's law have been applied somewhat successfully to the automatic indexing and classification of documents (see Chapter II), a possible solution to the
above problem would be to utilize similar techniques to identify a user's topics of interest, thereby increasing the degree of coincidence between his descriptors and those used to describe the items in the collection. Such a technique was explored in this thesis. The approach used was to measure the similarity between known relevant documents in order to establish document retrieval criteria, which could then be used to retrieve all documents in a collection falling within the established criteria. The end result of the process was that each document in the collection being searched was classified as relevant or non-relevant to a particular user, a process called pseudoclassification.

The major research effort was directed towards: 1) developing a set of similarity measures (Chapter III); 2) determining which combination of the measures in the set could best discriminate between relevant and non-relevant documents (Chapters III and IV); 3) establishing retrieval criteria (Chapter IV); and 4) evaluating the results of experimental trials (Chapters V and VI). The similarity measures for each particular user reflected the degree of correspondence between the composite vocabulary of his set of known relevant documents and the vocabulary of each document in the collection. The different types of measures chosen were combined using the stepwise linear discriminant analysis technique on sets of known relevant and non-relevant documents. The retrieval criteria so established were then used to pseudoclassify an abstract collection for a group of users and the results were compared with those obtained from index term searches of the same collection. The pseudoclassification results obtained with the similarity measures used showed a higher recall but lower precision
than the results obtained from the corresponding index term searches. This result was expected because of the inverse relationship between recall and precision. Methods for increasing precision while holding recall constant by use of feedback or implementing artificial retrieval constraints were discussed, as were possible areas for future research. Applications of the proposed system in the context of automated document analysis or collection partitioning are also discussed.
APPENDICES
APPENDIX A

GLOSSARY

The following terms occur with some frequency in this dissertation and are defined here in order to eliminate possible misunderstandings.

a) **Classification**: assignment of a document to a subject class on the basis of the rules in a classification schedule.

b) **Classification schedule**: a set of rules which effectively partition all knowledge into a set of subject classes, each of which has a hierarchical structure relating subclasses, sub-subclasses, etc.

c) **Common words**: those words whose frequency of occurrence are relatively constant in all fields of interest (e.g., "the", "and", "or", "if").

d) **Corpus** (pl., corpora): a set of documents having some common property.

e) **Document**: a full text journal article, a book, or a report.

f) **Indexing**: assignment of a document to a subject class.

g) **Match**: the co-occurrence of a word in a document and in a vocabulary profile.

h) **Precision**: the ratio of the numbers of relevant documents retrieved to the total number of documents retrieved.

i) **Pseudoclassification**: the process of partitioning a document collection on the basis of relevance judgments.

j) **Recall**: the ratio of the number of relevant documents retrieved to the total number of relevant documents in a collection.

k) **Stoplist**: a list of common words which are to be extracted from all documents prior to any processing.

l) **Thesaurus**: a type of dictionary which defines the allowable terms used in an index (also called Authority List).

m) **Token**: an occurrence of a word.
n) **Vocabulary control**: a process of defining the set of allowable index terms used for a particular indexing system (see Thesaurus).

o) **Vocabulary profile**: a minimal set of words which represents some portion of all token occurrences in a corpus.

p) **Word**: a unique string of alphabetic characters preceded and followed by a non-alphabetic symbol (see Token).
APPENDIX B

Introduction

There are six basic programs used in the pseudoclassification process. Five are written in PL/1, and their descriptions follow. The sixth, Stepwise Discriminant Analysis (BMD07M) of the Biomedical Computer Program Package (BMD), is available from the UCLA Health Services Computing Facility (6). (If the BMD programs are not available, routines DMTX and DSCR in PL/1SSP may be used.) The basic programs use a total of seven permanent tape files, several work files, the IBM SORT/MERGE package with PL/1 entry point IHESRTA, the IBM PL/1 Scientific Subroutine Package (PL/1SSP) with entry point CDTR, and a minimum core requirement of 126K. Since simultaneous mounting of all the tapes required for some of the programs is not always feasible, the JCL listings for some of the programs make use of the AFFINE (AFF) parameter in the UNIT field. This will force a dismount message to be passed to the computer operator and will allow a new tape to use the same tape drive. Also, since the programs are input/output bound, the BUFNO parameter is used in the DCB field to utilize all available storage. Blocking factors for some files are based on IBM 2314 disc track length. Modification should be made for any other direct access device, although this may cause the core requirements for the programs to be modified.

The next section shows the relationship of the files and programs. Additional sections contain program descriptions, file descriptions, control card formats, program listings, and sample JCL listings.
Program Descriptions

SETUP

Control Cards: C - 5

Input File: ORIGINAL

Output Files: FREQUENC, ABSTRACT

Purpose of Program: To decompose individual abstracts in the collection (ORIGINAL) into word and token frequency counts and to assign identification numbers to the abstract and the decomposed abstract for cross reference purposes.

Program Description: Each abstract in the collection (ORIGINAL) is read into core, all non-alphabetic characters are replaced by blanks, and individual words are identified. (In this program a word is considered a unique string of three or more alphabetic characters preceded and followed by one or more blanks.) The words are sorted, and multiple occurrences of words (tokens) are combined. The words are then compared to a stoplist, and deletions in the wordlist are made if a word or words appear in the stoplist. The number of tokens remaining are counted and the relative frequency of occurrence of each word in the abstract is computed. Then, two output files are created. The first (ABSTRACT) is essentially the original abstract file with a modified header character string on each record (i.e., the abstract). The header contains an identification number so that the two output files can be cross-referenced. The second output file (FREQUENC) contains the same header information and the word and token counts for each abstract in the original file.
This program is run only once for each collection, and requires approximately twenty minutes to process approximately 13,000 abstracts.

**SELECT**

Control Cards: C - 1

Input Files: ORIGINAL

Output File: SELECTED

Purpose of Program: To select, from the abstract collection (ORIGINAL), the abstracts a user has determined are relevant, decompose them into word and token counts, and produce files to be used as input to MRG and/or NEWVAR.

Program Description: The first operation performed by the program is to read the stoplist into core. Then the corpus identifier and the specific abstract identifiers are read. (The corpus identifier is on a separate card from the abstract identifiers because the input (ORIGINAL) may be the results of a SEARCH run. If so, the SEARCH user code may be different from the user code used elsewhere in this system.) An abstract from ORIGINAL is read into core and examined to see if it is one of the desired abstracts. If so, it is decomposed in the same manner as in SETUP, and an output record is formed. Then a new abstract is read in and the process is repeated.

**MRG**

Control Cards: C - 2

Input Files: CORPUS, SELECTED

Output Files: CORPUS, DBASES
Purpose of Program: To generate, or update, vocabulary profiles.

Program Description: The program reads into core all the word and token counts from SELECTED which pertain to a given user. These counts are written on a temporary file. Then, the word and token counts for the same user's corpus are read from CORPUS and also written onto the temporary file. This file is then sorted alphabetically, and the records are merged so that each word appears only once. The combined token counts and relative frequencies for each word are computed, and the results are written onto a new temporary file which is then sorted by relative frequency. This results of the sort are then used to create the vocabulary profile file (DBASES) and update CORPUS. At this point, the process repeats itself until all vocabulary profiles have been updated.

CALCPR

Control Cards: C - 3

Input Files: DBASES, FREQUENC, ABSTRACT

Output Files: None

Output: Abstracts (printed)

Purpose of Program: To pseudoclassify and retrieve relevant abstracts.

Program Description:

The flow of the program is:

1) the code representing the user for whom the run is being made is read into core.

2) the vocabulary profile for the user is read into core.
3) the coefficients, etc., for pseudoclassification are read into core.

4) the word and token count for an abstract are read into core from the file FREQUENC.

5) the words in the abstract are compared to the words on the vocabulary profile. If a match is obtained the token count from the abstract is recorded. The number of matches is also kept.

6) when all comparisons are made, the parameter values are computed, and the document is pseudoclassified.

7) if the document is found to be relevant, identification data is printed, and also written onto a temporary file. If more abstract word and token counts remain to be processed, repeat steps 4 - 7. Otherwise,

8) read in an abstract from ABSTRACT and an identification number from the temporary file containing the identification data for relevant abstracts. If the identification numbers are the same, print the abstract and read in the next identification number from the temporary file. Then read in the next abstract and repeat process. When all relevant documents have been processed, terminate the program.

NEWVAR

Control Cards: C - 6

Input Files: SELECTED, DBASES

Output File: INITDB

Purpose of Program: To prepare the initialization corpora.
Program Description: Using SELECT, a set of $n$ relevant abstracts for a given user are decomposed into word and token counts, and output onto SELECTED (this may be the same data file used in MRG). Then $m$ non-relevant abstracts are selected, and decomposed, the results being concatenated onto SELECTED. SELECTED is input data for NEWVAR. An additional input file for NEWVAR is DBASES. The program is similar to CALCPR insofar as the matching and calculation of the parameter values. However, the output consists of $(n+m)$ records on a temporary file. The format of each record is four bytes identifying the user for whom the initialization corpus is generated and twenty-seven ten byte fields containing the values for each parameter is fixed decimal notation. This data file is used as input data for the BMD07M program.

File Descriptions

ORIGINAL: This is the file containing the abstract or document collection. It is used as input for SELECT and SETUP.

ABSTRACT: This file, produced by SETUP contains the abstract or document collection contained in ORIGINAL with one modification. Each record (abstract) has added to it a thirteen character header prefix. The first nine bytes of the header data are 'DOCUMENT-', and the following four bytes contain a fixed binary identification number. The purpose of the number is to relate abstracts with the word and token count file (FREQUENC) also produced by SETUP. This file is used as input to the program CALCPR and is the source
of retrieved abstracts in that program. The records in the file are variable length.

FREQUENC: This file, produced by SETUP contains the word and token counts for each abstract in ORIGINAL. Although a variable length record file according to the JCL definitions, each record contains up to 250 subrecords each of length twenty-eight bytes. The format of the subrecords is twenty characters, a four byte fixed binary number, and a four byte floating point number. The first subrecord contains in the first thirteen bytes the same information and the same format as the header information in ABSTRACT. The next seven bytes are blank. The four byte fixed binary number contains the number of tokens in the abstract. The floating point number contains the number of words in the abstract. The remaining subrecords (up to 249) each have in the first twenty bytes a word which appeared in the abstract. (The words are sorted into alphabetic order for convenience in later processing.) The four bytes contain the number of token occurrences of the word within the abstract. The last four bytes give the relative frequency of the token occurrences. This file is used as input to the CALCPR program and is the file which provides the data inputs for the pseudoclassification process.

SELECTED: This file is produced by SELECT, and contains the word and token counts for the users corpora. It is used as input to the MRG program, and, when non-relevant abstracts are concatenated
onto it, to the NEWVAR Program. The format is the same as
FREQUENC with one exception. Bytes 17-20 in the first sub-
record contain the corpus identifier.

DBASES: This file is produced by MRG and is used as input to
CALCPR and NEWVAR. The file contains the vocabulary profiles
for all the users. The format consists of twenty-eight byte
records. The first record of each vocabulary profile contains:

a) eight bytes containing the word DATABASE.
b) eight bytes containing blanks.
c) four bytes containing the vocabulary profile code,
   the same as the user's code.
d) four bytes containing blanks.
e) a four byte floating point number containing the
   number of words in the vocabulary profile.

The format for each succeeding record (after the first record)
in a specific profile is:

a) twenty bytes containing the specific word.
b) four bytes containing a fixed binary number showing
   the number of occurrences of the word in the user's
   corpus.
c) a four byte floating point number indicating the
   relative frequency of the word within the user's corpus.

The words are sorted alphabetically within the profile for ease
in usage in later programs. The vocabulary profiles are in
alphabetical order based on user codes.
CORPUS: This file is created by, and used as input to, MRG. It contains the word and token counts of the user's corpus, sorted alphabetically. Since there is no way of knowing, in advance, the number of words in the user's corpus, the first and last records in each corpus are identified. The first record contains 'DATABASE' in the first eight bytes and the user code in bytes 17-20. Since this code is usually a combination of alphabetic and numeric characters, there is little chance that a word ending would be confused with the corpus code. The last record contains 'XXXX' in the first four bytes, bytes 21-24 contain fixed binary number giving the number of tokens in the corpus, and bytes 25-28 contain a floating point number giving the number of words in the corpus. The corpora will appear in the file in alphabetic order based on the user's codes, thus all updates to the file must appear in the same order.

INITDB: This file is created by NEWVAR and is used by BMD07M. Each record contains the four byte user code and the parameter values derived by comparing the user vocabulary profile with the individual abstract word and token counts. The parameter values are output in fixed decimal form to facilitate using the BMD07M program.

Control Cards
C-1: This is the control deck for SELECT, and contains the identifier for the corpus being updated or created and the identifiers of
the abstracts being selected for the update. The format of
the abstract identifiers is that of the ERIC Accession Number,
and would have to be changed to process other files.

Cards 1-75:
Columns 1-20 Stoplist words, left-justified and in
alphabetic order.

Card 76:
Columns 1-4 Corpus identifier.
Column 5 Blank.
Columns 6-80 Not used or processed by program.

Card 77 through N: (N-1 abstracts to be selected)
Columns 1-4 Not used by program. May be used for
card identification by programmer or user.
Columns 5-6 'ED', ERIC Accession Number identified.
Columns 7-12 nnnnnn, the ERIC Accession Number.
Columns 13-80 Not used or processed by program.

Card N+1: Same as card 1 for next corpus to be updated or created.

Following cards have same format as cards 2 through N.

C-2: This is the control deck for MRG, and consists of only one
card. This card contains the cutoff value for the vocabulary
profile. This value, p, is any positive number less than or
equal to one. The number can be placed on the card in any
position, but it must include the decimal point.

C-3: This is the control deck for CALCPR, the program which
performs the actual pseudoclassification and retrieving of
documents. Card 2 in the control deck is essentially the GPLABL card used in the BMD07M program.

Card 1
Columns 1-4 User identifier

Card 2
Columns 7-12 Title to be given to column listing relevant abstract numbers in summary data (e.g., 'BlREL').
Columns 13-18 Title to be given to column listing non-relevant abstract numbers in summary data (e.g., 'B2NREL')

Card 3
Columns 1-10 Constant to be added to classifying equation assuming abstract is relevant.
Columns 12-22 Constant to be added to classifying equation assuming abstract is non-relevant.

Card 4-n
Columns 1-2 Parameter number.
Columns 4-14 Coefficient for the parameter assuming abstract is relevant.
Columns 20-30 Coefficient for the parameter assuming abstract is non-relevant.

C-4: The control deck for the BMD07M program is described in Bibliographic reference BMD: Biomedical Computer Programs (6).
C-5: This is the control deck for SETUP, and consists of the stoplist.

Cards 1-75

Columns 1-20 Contain the stoplist words, left justified and in alphabetic order.

C-6: This is the control deck for NEWVAR, and consists of one card. It has the four character user identifier code in the first four columns of the card.
**PROCEDURE SETUP**

**SETUP:**
```plaintext
PROC OPTIONS (MAIN) REORDER;

/* THIS PROGRAM DECOMPOSES THE ABSTRACT MASTER FILE*/
/* <ORIGINAL> INTO WORDS, ELIMINATES SOME COMMON */
/* WORDS, GETS WORD-TOKEN COUNTS FOR EACH ABSTRACT, */
/* AND PRODUCES TWO NEW FILES, <FREQUENCY> AND */
/* <ABSTRACT>. IT IS ORIENTED TOWARDS THE ERIC */
/* FORMAT, BUT THOSE BLOCKS OF CODE WHICH ARE */
/* DEPENDENT ON THAT FORMAT WILL BE IDENTIFIED. */

DCL STOP(75) CHAR(20),
(FI, FB) FIXED BIN (15),
ERIC CHAR (7000) VAR,
CD CHAR (2),
SUM FIXED BIN (31),
(FREQUENCY,
ABSTRACT,
ORIGINAL) FILE RECORD,
(T, TT) CHAR (20),
(TU, IL)(10) FIXED BIN (31),
WORDS (500) CHAR (20),
ABSTL CHAR(23) INIT (',;:-()*/<>?"',=+/#$@0123456789"'),
KEPT CHAR(23) INIT (',;:-()*/<>?"',=+/#$@0123456789"'),
1 ITRMS (250),
2 WOS CHAR(20),
2 FRQ FIXED BIN (31),
2 RFR,
1 RCD,
2 W CHAR(20),
2 F FIXED BIN (31),
2 R,
BUF CHAR(13),
DTA CHAR (7000) DEFINED ITRMS,
CHK(7000) CHAR (1) DEFINED ITRMS,
CODE CHAR (2),
NUMB CHAR (4) DEFINED ICNT,
ICNT FIXED BIN (31);
GO TO 19;
OPEN FILE (SYSPRINT) OUTPUT LINESIZE (128);
```

10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
PROCEDURE SETUP

/*********************************************/
/* READ IN STOPLIST */
/*********************************************/
DO I=1 TO 75 ;
GET EDIT(CARD)(A(80)) ;
STOP(I)=SUBSTR(CARD,1,20) ;
END ;
/*********************************************/
/* INITIALIZE ICNT. THIS IS THE UNIQUE IDENTIFIER */
/* WHICH RELATES THE SPECIFIC ABSTRACTS AND THE */
/* WORD-TOKEN COUNTS FOR EACH ABSTRACT. */
ICNT=13000 ;
OPEN FILE(FREQUENCY)OUTPUT,FILE(ABSTRACT)OUTPUT ;
L1:
READ FILE (ORIGINAL) INTO (ERIC) ;
LR=1 ;
/*********************************************/
/* FROM M2 TO M3 DEPENDS ON THE ERIC FORMAT. FI IS */
/* THE FIELD IDENTIFIER CODE, FB IS THE LENGTH OF */
/* THE FIELD. 44 IS CODE FOR ABSTRACT. */
M2:
IF FI=44 THEN DO ;
DATA=SUBSTR(ERIC,LR+4,FB-4) ;
GO TO M3 ;
END ;
ELSE DO ;
LR=LR+FB ;
GO TO M2 ;
END ;
M3:
/*****************************/
/* PUT HEADER ON ABSTRACT */
/* *************************************/
BUF=DOCUMENT- || NUMB ;
ERIC=BUF||ERIC ;
/*****************************/
/* WRITE FILE <ABSTRACT> */
/* *************************************/
WRITE FILE(ABSTRACT)FROM (ERIC) ;
K=0 ;
PROCEDURE SETUP

/* REMOVE NON-ALPHABETIC CHARACTERS */
DTA=TRANSLATE(DTA,REPT,ABSTL);
J=0;
DO I=1 TO FB;
/* IDENTIFY WORDS. A WORD IS THREE OR MORE CONSECUTIVE ALPHABETIC CHARACTERS BOUNDED BY BLANKS. */
IF CHK(I)=' '
THEN
DO;
IF I-J<4
THEN
DO;
J=I;
GO TO L3;
END;
K=K+1;
WORDS(K)=SUBSTR(DTA,J+1,I-J-1);
J=I;
END;
/* SORT WORDS USING QUICKSORT CODED IN PL/1 */
L3:
END;
M,I=1;
J=K;
S5:
IF I>=J
THEN
GO TO S70;
S10:
IK=1;
IJ=(I+J)/2;
T=WORDS(IJ);
IF WORDS(I)<=T
THEN
GO TO S20;
WORDS(IJ)=WORDS(I);
WORDS(I)=T;
T=WORDS(IJ);
S20:
L=J;
PROCEDURE SETUP

IF WORDS(J) >= T
THEN
  GO TO S40;
WORDS(IJ) = WORDS(J);
WORDS(J) = T;
T = WORDS(IJ);
IF WORDS(I) <= T
THEN
  GO TO S40;
WORDS(IJ) = WORDS(I);
WORDS(I) = T;
T = WORDS(IJ);
GO TO S40;
S30:  
WORDS(L) = WORDS(IK);
WORDS(IK) = T;
S40:  
L = L - 1;
IF WORDS(L) > T
THEN
  GO TO S40;
T = WORDS(L);
S50:  
IK = IK + 1;
IF WORDS(IK) < T
THEN
  GO TO S50;
IF IK <= L
THEN
  GO TO S30;
IF (L - I) <= (J - IK)
THEN
  GO TO S60;
IL(M) = I;
IU(M) = L;
I = IK;
M = M + 1;
GO TO S80;
S60:  
IL(M) = IK;
IU(M) = J;
J = L;
M = M + 1;
GO TO S80;
S70:  
M = M - 1;
IF M = 0
PROCEDURE SETUP

123
PROCEDURE Setup

1:24
PROCEDURE SETUP

EFIC='';
ITOP=0;

REGROUP WORDS, GET NEW TOKEN COUNTS, COMPUTE REL*;

DO !=i TO N;
   IF ITRMS(I).WDS=''
      THEN
         GC TO L8;
         IMAX=(I-1)*28+1;
         EFIC=ERIC||SUBSTR(DTA,IMAX,28);
         SUM=SUM+ITRMS(I).FRQ;
         ITOP=ITOP+1;

L9:
   END;
END;

CREATE HEADER RECORD. MOVE RECORDS DOWN ONE *;

DO I=N+1 TO 2 BY -1;
   ITRMS(I)=ITRMS(I-1);
END;

ITRMS(1)=RCD;
PUT EDIT (ICNT,F,N)(X(3),F(5),X(1),F(3),X(1),
   F(3));

END SETUP;
PROCEDURE MRG

MRG:
PROC OPTIONS (MAIN) RECORD;

40

*/ THIS PROGRAM GENERATES OR UPDATES THE VOCABULARY*/
*/ PROFILES. INPUT FILES ARE <CORPUS> AND */
*/ <SELECTED>, OUTPUT FILES ARE <CORPUS> AND */
*/ <DRASES>. THE ABSTRACT WORD-TOKEN COUNTS WHICH */
*/ ARE TO BE USED TO UPDATE THE CORPORA COUNTS MUST */
*/ BE GROUPED BY CORPUS. THE ORDER OF THE GROUP MUST*/
*/ BE THE SAME AS THE ORDER ON <CORPUS> AND */
*/ <DRASES>.
*/
DCL

IHESRTA ENTRY (,, FIXED BIN (31),,,),
(ICO,]
(II) FIXED BIN (31),
(II, DRASES,)
SORTIN,
SORTCUT) FILE RECORD,
(CORPUS,}
SELECTED,
TEMP,
NSPD,
HOLD) FILE RECORD,
1 RCO,
2 W CHAR (20),
2 F FIXED BIN (31),
2 R,
1 RCO,
2 DW CHAR (20),
2 DF FIXED BIN (31),
2 DR,
CODE CHAR (4),
(IAD,)
(IAL) FIXED BIN (31) INIT (0),
SFD1 CHAR (50),
RFLD1 CHAR (50),
SFD2 CHAR (50),
RFLD2 CHAR (30),
;
BEGIN ;

DCL

1 INDEX (250),
2 WD CHAR (20),
2 FW FIXED BIN (31),
2 RW,
1 WORD,
PROCEDURE MRG

2 WW CHAR (20),
2 WF FIXED BIN (31),
2 WR,
(SUM,
ITGT) FIXED BIN (31),
ERIC CHAR (7000) VAR,
INPT CHAR (7000) DEFINED INDX;

SFO1=' SORT FIELDS=(1,20,CH,A) ' ;
SFD2=' SORT FIELDS=(21,4,RI,9,1,20,CH,A) ' ;
RFLD1=' RECORD TYPE=F,LENGTH=(25) ' ;
RFLD2=' RECORD TYPE=F,LENGTH=(28) ' ;
WW=' ' ;
SFD1=' SORT FIELDS=(1,20,CH,A) ' ;
RFLD1=' RECORD TYPE=F,LENGTH=(28) ' ;
CODE=' ' ;
ON ENDFILE (CORPUS)
  GO TO CI ;
ON ENDFILE (SELECTED)
  GO TO D2 ;
OPEN FILE (NSPD) OUTPUT ;
OPEN FILE (TEMP) OUTPUT TITLE ('SORTIN') ;
SUM=0 ;

/***********************************************************/
/* READ IN THE ABSTRACT WORD-TOKEN COUNTS FOR THOSE*/
/* ABSTRACTS WHICH ARE TO BE USED TO UPDATE THE */
/* GIVEN CORPUS COUNTS. CONTINUE UNTIL ALL COUNTS FOR A */
/* GESN CORPUS ARE INPUT. WRITE THESE COUNTS ON */
/* <SORTIN>. WHEN THE NEW COUNTS ARE INPUT, THEN */
/* READ IN THE COUNTS FOR THE SAME CORPUS FROM */
/* <CORPUS>, AND WRITE THESE COUNTS ON <SORTIN>. */
/* SORT THE COMBINED OUTPUT ALPHABATICALLY, THEN */
/* COMBINE TOKEN COUNTS FOR EACH WORD OCCURRING MORE*/
/* THAN ONCE, COMPUTING RELATIVE FREQUENCIES OF */
/* OCCURRENCE. OUTPUT GOES TO <NSPD>, THE NEW */
/* <CORPUS> FILE. */
/***********************************************************/
CI:
READ FILE(SELECTED) INTO (ERIC) ;
INPT=ERIC ;
IF CODE=' ' ,
  THEN
    CODE=SUBSTR(INDX(1),WD,17,4) ;
CB:
IF CODE=SUBSTR(INDX(1),WD,17,4)
  THEN
    DO ;
PROCEDURE MRG

RDC=INDX(1);  
SUM=SUM+DF ;

/****Running Token Count****/  
K=DR ;
DO I=2 TO K+1 ;

RDC=INDX(I) ;
WRITE FILE (TEMP) FROM (RDC) ;
END ;
GO TO C1 ;
END ;

D3:
IF WW=""
THEN
GO TO D8 ;
PUT SKIP DATA(WW,WF,WR) ;
READ FILE (CORPUS) INTO (WORD) ;

D8:

/*If the wrong corpus is read in (ie, no */
/* updates, and precedes the corpus being searched */
/* for ), it is written directly onto <NSPD>.*/

IF SUBSTR (WW,1,4)<CODE
THEN
DO ;

WRITE FILE (NSPD) FROM (WORD) ;

D9:

READ FILE (CORPUS) INTO (WORD) ;
WRITE FILE (NSPD) FROM (WORD) ;
IF SUBSTR (WW,1,4)="XXXX"
THEN

GO TO D9 ;
PUT SKIP DATA(WW,WF,WR) ;
WW="" ;
GO TO D3 ;
END ;

IF SUBSTR (WW,1,4)=CODE
THEN
DO ;

D7:

READ FILE (CORPUS) INTO (WORD) ;

/*THE last record for each corpus contains 'XXXX'*/
/*In the first four bytes*/
/* ****Running Token Count****/
PROCEDURE MRG

IF SUBSTR(WW,1,4)='XXXX'
THEN
  DO;
    SUM=SUM+WF;
    WRITE FILE(TEMP) FROM (WORD);
    GO TO D7;
  END;
  WW='';
  GO TO D4;
END;
D4:
  X=SUM;
  CLOSE FILE (TEMP);
  CALL IHERSTA(SFDI,RFLDI,25000,ICD,'SORT');
  IF ICD=0
  THEN
    DO;
      PUT SKIP LIST ('SORT ERROR');
      STOP;
    END;
  OPEN FILE (TEMP) INPUT TITLE ('SORTOUT');
  OPEN FILE (NSPD) OUTPUT;
  ON ENDFILE (TEMP)
  GO TO D6;
  WW='DATABASE '||CODE;
  F=SUM;
  R=0;
  SUM=0;
  PUT SKIP DATA(W,F,R);
  WRITE FILE (NSPD) FROM (RCD);
/* COMBINE TOKEN COUNTS, COUNT WORDS, COMPUTE REL. */
/* FREQS. LAST RECORD CONTAINS WORK-TOKEN COUNTS. */
READ FILE (TEMP) INTO (RCD);
ITOT=1;
SUM=F;
D5:
READ FILE (TEMP) INTO (RDC);
IF W=WD
THEN
  DO;
    F=F+DF;
    SUM=SUM+DF;
    GO TO D5;
  END;
  R=FLOAT(F)/X;
PROCEDURE MRG

WRITE FILE (NSPD) FROM (RCO) ;
SUM=SUM+OF ;
ITQT=ITQT+1 ;
RCO=RCO+1 ;
GO TO D5 ;

D6:
R=FLOAT(F)/X ;
WRITE FILE (NSPD) FROM (RCO) ;
W='XXXX '11CODE ';
R=ITQT ;
F=SUM ;
SUM=0 ;
WRITE FILE (NSPD) FROM (RCO) ;
PUT SKIP DATA(W,F,R) ;
IT=0 ;
CLOSE FILE (TEMP) ;
OPEN FILE (TEMP) OUTPUT TITLE ('SORTIN') ;
IF IAD=1 THEN DO ;
IF IAL=1 THEN GO TO D10 ;
GO TO D8 ;
END ;
CODE=SUBSTR (INDEX(1),WD,17,4) ;
GO TO C8 ;

/**********************************************************************************/
/* DEPENDING UPON WHICH ENDFILE CONDITION WAS */
/* RAISED, AND STATUS WHEN RAISED, FINISH CREATING */
/*<NSPD>, <NSPD>, THE NEW <CO-PUS> FILE IS THE INPUT*/
/* FILE FOR CREATING THE VOCABULARY PROFILES. */
/**********************************************************************************/

D1:
IAL=1 ;
IF IAC=1&IT=1 THEN GO TO D4 ;
IF IAD=1 THEN GO TO D10 ;
GO TO D4 ;

D2:
IAD=1 ;
IT=1 ;
IF IAL=1 THEN
PROCEDURE MRG

GO TO D4 ;
GO TO D3 ;
D10:
CLOSE FILE (SELECTED) ;
CLOSE FILE (TEMP) ;
CLOSE FILE (NSPD) ;
CLOSE FILE (CORPUS) ;
ERIC=' ' ;
ON ENDFILE(NSPD)
  GO TO OUT1 ;
OUT1:
CLOSE FILE (NSPD) ;
CLOSE FILE (CORPUS) ;
END ;
ON ENDFILE (CORPUS)
  GO TO OUT ;
/**************************************************************************/
/* READ IN VOCABULARY PROFILE CUTOFF, INITIALIZE */
/* COUNTERS. */
/**************************************************************************/
GET LIST (CUTOFF_PROBABILITY) ;
OPEN FILE(DBASES)OUTPUT ;
L2:
OPEN FILE(SORTIN)OUTPUT ;
P1,P2=0 ;
K,N1=0 ;
M:
  READ FILE (NSPD) INTO (RCD) ;
  READ FILE(CORPUS) INTO (RDC) ;
/**************************************************************************/
/* IDENTIFY CORPUS, WRITE WORDS AND TOKEN COUNTS ON*/
/* <SORTIN>, SORT WORDS BY FREQUENCY. */
/**************************************************************************/
IF SUBSTR(DW,1,0)='DATABASE'
  THEN
    DO ;
      PUT SKIP DATA (DW,DF,DR ) ;
      IJK=0 ;
    M1:
      READ FILE (NSPD) INTO (RCD) ;
      IF SUBSTR(W,1,4)='XXXX'
        THEN
          DO ;
            IJK=IJK+1 ;
            WRITE FILE (SORTIN) FROM (RCD) ;
          GO TO M1 ;
        END ;
    END ;
PROCEDURE MRG

DR=IJK;
CLOSE FILE(SORTIN);
call THESRTA(SF02,RFLD2,45000,ICD,'SORT');
if ICD>0
then
do;
   put skip list('ERROR IN SORT 1');
go to out;
end;
/** **********************************************************
/* FIND THE MINIMAL SET OF WORDS TO COMPRIZE */
/* VOCABULARY PROFILE. WRITE OUT ON <SORTIN> SO THAT*/
/* THE SET CAN BE ALPHABETIZED. COPY SORTED PROFILE */
/* ON <DBASES> AND TRANSFER TO 'L2' TO REPEAT */
/* PROCESS. */
/** **********************************************************
open file(SORTIN)output;
do i=1 to IJK;
read file(SORTOUT)into (RCD);
/** **********************************************************
/* HAS CUTOFF BEEN ATTAINED? */
/** **********************************************************
if cutoff_frequency>pi
then
do;
   write file(SORTIN)from(RCD);
   ni=ni+1;
k=f;
   pl=pl+r;
goto l;
end;
/** **********************************************************
/* TIE HANDLING RULE. */
/** **********************************************************
if f=k
then
do;
   write file(SORTIN)from(RCD);
   pl=pl+r;
   ni=ni+1;
goto l;
end;
else
   do;
   l:
   end;
end;
PROCEDURE MRG

PUT SKIP DATA (DW,DF,DR );  3310
PUT SKIP LIST ('ERROR IN D/B' );  3320
GO TO M ;  3330
END ;  3340
P2=R ;  3350
PUT DATA(P1) ;  3360
PUT DATA(NI) ;  3370
CLOSE FILE(SORTIN) ;  3380
CLOSE FILE (SORTOUT) ;  3390
CALL IHESRAT(SF01,RFLD1,45000,ICD,'SORT') ;  3400
IF ICD >0 THEN 3410
DO ;  3420
PUT SKIP LIST('ERROR IN SORT 2') ;  3440
GO TO OUT ;  3450
END ;  3460
ON ENDFILE(SORTOUT) 3470
GO TO L1 ;  3480
DR =NI ;  3490
WRITE FILE (DBASES) FROM (RCD) ;  3500
/** Output Summary Data */ 3510
/* Output Summary Data */ 3520
PUT SKIP DATA (DW,DF,DR ) ;  3530
J1:
READ FILE(SORTOUT) INTO (RCD) ;  3540
WRITE FILE (DBASES) FROM (RCD) ;  3550
GO TO J1 ;  3560
L1:
CLOSE FILE (SORTOUT) ;  3570
GO TO L2 ;  3580
OUT:
END MRG ;  3590
PROCEDURE SELECT

SELECT:
PROC OPTIONS (MAIN) REORDER:

/*****************************/
/* PORTIONS OF THIS PROGRAM ARE DEPENDENT ON THE */
/* FORMAT OF THE ERIC FILES, THOSE PORTIONS WILL BE */
/* IDENTIFIED. THE PURPOSE OF THE PROGRAM IS TO */
/* SELECT FROM THE ERIC FILE THOSE ABSTRACTS KNOWN */
/* TO BE RELEVANT TO A SPECIFIC USER, REDUCE THE */
/* ABSTRACTS TO WORD/TOKEN COUNTS, AND OUTPUT THOSE */
/* COUNTS IN THE FORMAT USED BY <MRG>, THE PROGRAM */
/* USED TO GENERATE CORPUS WORD/TOKEN COUNTS AND */
/* VOCABULARY PROFILES. IT IS ASSUMED THAT */
/* WORD/TOKEN COUNTS ARE NOT AVAILABLE FOR INPUT TO */
/* THIS PROGRAM, THUS THE REQUIREMENT FOR ABSTRACT */
/* DECOMPOSITION, OUTPUT FROM THIS PROGRAM MAY BE */
/* USED TO CREATE OR UPDATE THE CORPUS WORD/TOKEN */
/* COUNTS OR TO CREATE INITIALIZATION CORPORA. */
/*****************************/

CCL
(T,
(T),CHAR (20),
(IU,
(II)(10)FIXED BIN (31),
STOP(75)CHAR(20),
ICARD CHAR(50),
CARD CHAR(12)DEFINED ICARD,
 ICC CHAR (80),
 CD CHAR(4)DEFINED ICD,
 (FI,
(FR)FIXED BIN (15),
ERIC CHAR (7000)VAR,
WORDS (500)CHAR (20),
ABSTL CHAR (29) INIT (",.;:-()"/<>¿±º½$0123456789"),
REPT CHAR(29) INIT (',
1 ITMS (250),
 2 WDS CHAR(20),
 2 FRQ FIXED BIN (31),
 2 PCD,
 2 W CHAR(20),
 2 F FIXED BIN (31),
 2 R,
 DTA CHAR (7000)DEFINED ITMS,
CHK(7000)CHAR (1)DEFINED ITMS,
DOC CHAR(9),

134
PROCEDURE SELECT

/* SKIP THE FIELD AND REPEAT THE PROCESS. */
FI=UNSPEC(SUBSTR(ERIC,LR+2,2)) ;
FB=UNSPEC(SUBSTR(ERIC,LR,2)) ;
IF FI=44 THEN DO ;
OTA=SUBSTR(ERIC,LR+4,FB-4) ;
GO TO M3 ;
END ;
ELSE DO ;
LR=LR+FB ;
GO TO M2 ;
END ;
M3:
K=0 ;
OTA=TRANSLATE(OTA,REP'T,ABSTL) ;
J=0 ;
DO I=1 TO FB ;
IF CHK(I)=" " THEN DO ;
OTA=SUBSTR(OTA,I+1,I-J-1) ;
WORDS(K)=SUBSTR(OTA,J+1,I-J-1) ;
J=I ;
END ;
K=K+1 ;
END ;
L3:
END ;
/* REMOVE NON-ALPHABETIC CHARACTERS AND REPLACE */
** A WORD CONSISTS OF THREE OR MORE ALPHABETIC **
** CHARACTERS BOUNDED BY BLANKS. **
IF I-J<4 THEN DO ;
J=I ;
GO TO L3 ;
END ;
K=K+1 ;
WORDS(K)=SUBSTR(OTA,J+1,I-J-1) ;
J=I ;
END ;
M,I=1 ;
PROCEDURE SELECT

J=K ;
S5:
IF I>=J
THEN
GO TO S70 ;
S10:
IK=I ;
IJ=(I+J)/2 ;
T=WORDS(IJ) ;
IF WORDS(I)<=T
THEN
GO TO S20 ;
WORDS(IJ)=WORDS(I) ;
WORDS(I)=T ;
T=WORDS(IJ) ;
S20:
L=J ;
IF WORDS(J)>=T
THEN
GO TO S40 ;
WORDS(IJ)=WORDS(J) ;
WORDS(J)=T ;
T=WORDS(IJ) ;
IF WORDS(I)<=T
THEN
GO TO S40 ;
WORDS(IJ)=WORDS(I) ;
WORDS(I)=T ;
T=WORDS(IJ) ;
GO TO S40 ;
S30:
WORDS(L)=WORDS(IK) ;
WORDS(IK)=T T ;
S40:
L=L-1 ;
IF WORDS(L)>T
THEN
GO TO S40 ;
TT=WORDS(L) ;
S50:
IK=IK+1 ;
IF WORDS(IK)<T
THEN
GO TO S50 ;
IF IK<=L
THEN
GO TO S30 ;
PROCEDURE SELECT

IF (L-I) <= (J-IK)
    THEN
        GO TO S60;
    ILL(M) = I;
    IU(M) = L;
    I = IK;
    M = M + 1;
    GO TO S60;
S60:
    ILL(M) = IK;
    IU(M) = J;
    J = L;
    M = M + 1;
    GO TO S80;
S70:
    M = M - 1;
    IF M = 0
        THEN
            GO TO Z;
    I = ILL(M);
    J = IU(M);
S90:
    IF J - I >= 11
        THEN
            GO TO S10;
    IF I = II
        THEN
            GO TO S5;
    I = I - 1;
S90:
    I = I + 1;
    IF I = J
        THEN
            GO TO S70;
    T = WORDS(I + 1);
    IF WORDS(I) <= T
        THEN
            GO TO S90;
    IK = I;
S100:
    WORDS(IK + 1) = WORDS(IK);
    IK = IK - 1;
    IF T < WORDS(IK)
        THEN
            GO TO S100;
    WORDS(IK + 1) = T;
    GO TO S90;
PROCEDURE SELECT

Z:
N=1 ;
I=1 ;
SUM=0 ;
L5:
W=WORDS(I) ;
IF I=K
THEN
GO TO L6 ;
FORMANCE COUNTS (TOKENS) TOGETHER. */
/*
L6:
F=K-I+1 ;
ITRMS(N)="CD" ;
W="DOCUMENT"||DOC||CD ;
/*
I=1 ;
J=1 ;
L2:
IF STOP(J)>ITRMS(I)*WDS
THEN
DO ;
I=I+1 ;
IF I>N
THEN
GO TO L7 ;
GO TO L2 ;
END ;
IF STOP(J)<ITRMS(I)*WDS
THEN
DO ;
PROCEDURE SELECT

J = J + 1 ;
IF J > 75
THEN
   GO TO L7 ;
GO TO L2 ;
END ;
IF STOP(J) = ITRMS(I).WDS
THEN
   ITRMS(I).WDS = '* ' ;
   J = J + 1 ;
   I = I + 1 ;
   IF J > 75 | I > N
THEN
      GO TO L7 ;
GO TO L2 ;
END ;

L7:
   ERIC = * ;
   DTOP = 0 ;
   /* PULL OUT REMAINING WORDS, GET NEW TOKEN COUNTS, */
   /* COMPUTE RELATIVE FREQUENCIES OF TOKENS REMAINING */
   /* IN ABSTRACT. */
   DO I = 1 TO N ;
      IF ITRMS(I).WDS = '* '
      THEN
         GO TO LB ;
      IMAX = (I - 1) * 29 + 1 ;
      ERIC = ISTRDTA, IMAX, 23 ) ;
      SUM = SUM + ITRMS(I).FRQ ;
      DTOP = DTOP + 1 ;
   END ;
   DTA = ERIC ;
   DO I = 1 TO DTOP ;
      ITRMS(I).FR = FLOAT(ITRMS(I).FRQ) / SUM ;
   END ;
   F = SUM ;
   R = DTOP ;
   N = DTOP ;
   /* MOVE WORD/TOKEN COUNTS BY ONE SUBRECORD SO THAT */
   /* HEADER DATA CAN FILL FIRST SUBRECORD. */
   /* 3230 */
   DO I = N + 1 TO 2 BY -1 ;

/* PULL OUT REMAINING WORDS, GET NEW TOKEN COUNTS, */
/* COMPUTE RELATIVE FREQUENCIES OF TOKENS REMAINING */
/* IN ABSTRACT. */
PROCEDURE SELECT

ITRMS(I)=ITRMS(I-1) ;
END ;
ITRMS(1)=RCD ;
L=2*(N+1) ;
ERIC=SUBSTP(DTA,1,L) ;
/* WRITE OUT THE ABSTRACT WORD/TOKEN COUNT */
/* GET NEXT SELECTED ABSTRACT */
/* WRITE FILE(SELECTED)FROM (ERIC) ;
/*****/
GO TO M1 ;
NEXT:
PUT SKIP LIST('ENDFILE QTAPE') ;
IF IEND=1
THEN
GO TO M1 ;
CLOSE FILE(ORIGINAL) ;
IEND=1 ;
GO TO LL ;
OUT:
END SELECT ;
PROCEDURE NEWVAR

NEWVAR:
  PROC OPTIONS('MAIN')REORDER ;
/****************************/**
/* THIS PROGRAM CREATES THE INITIALIZATION DATABASE*/
/* TO BE USED BY THE DISCRIMINANT ANALYSIS PROGRAM. */
/* THE INPUT FILE <SELECTED> IS GENERATED BY THE */
/* <SELECT> PROGRAM. THE OUTPUT FILE <INITD3> IS THE*/
/* INPUT TO <BMDQ7M>. */
/****************************/**
  DCL  
    1 RCD, 
      2 WD CHAR (20), 
      2 FR FIXED BIN (31), 
      2 RR, 
    1 RDC, 
      2 WP CHAR (20), 
      2 DF FIXED BIN (31), 
      2 DR, 
    1 RESULT, 
      2 HEADER CHAR (20), 
      2 FFS (25) FIXED DECIMAL (10,5), 
  REC (28), 
    K, 
  DOC_COUNT, 
  KIN, 
  KOUT, 
  N, 
  J, 
  MATCH, 
  CUMFREQ) FIXED BIN (31), 
    (A, 
    B, 
    C, 
  D) FLOAT BIN, 
  CDT ENTRYFLOAT BIN, FLOAT BIN, FLOAT BIN, FLOAT BIN, 
  ITERM (100) FLOAT, 
  NTERM (100) FLOAT, 
  1 ABSTR_WORD_COUNT (250) BASED (P), 
    2 W CHAR (20), 
    2 F FIXED BIN (31), 
    2 R, 
( CODE, 
  CD) CHAR(4), 
;
PROCEDURE NEWVAR

BEGIN ;

DCL

I VOCAB_PROFILE(1200),
2 TERM CHAR(20),
2 TIMES FIXED BIN(31),
2 EXPECTED_FREQ,
;

ON_ERROR

BEGIN ;

PUT LIST('ERROR ON DOC=',DOC_COUNT ) ;
go to ENDIT ;
END ;

READ CORPUS IDENTIFIER */
GET LIST (CODE) ;
ON ENDFILE(DBASES)
go to LOOP4 ;

READ VOCABULARY PROFILE FILE (DBASES). PROFILES */
ARE IDENTIFIED IN THE FIRST RECORD BY 'DATABASE' */
IN THE FIRST EIGHT BYTES AND BY THE USER */
IDENTIFICATION CODE IN THE TENTH THRU THIRTEENTH */
BYTES. THE LAST FOUR BYTES IN THE RECORD CONTAIN */
A FLOATING POINT NUMBER REPRESENTING THE NUMBER */
OF WORDS IN THE PROFILE. */
LOOP:
READ FILE (DBASES) INTO (RCD) ;
IF SUBSTR(WD,1,8)=='DATABASE'
THEN
  GO TO LOOP ;
CD=SUBSTR(WD,17,4) ;
IF CD=CODE
THEN
  GO TO J2 ;
go to LOOP ;
J2:
N=RR ;
DO I=1 TO N ;

READ FILE(DBASES) INTO(RDC) ;
VOCAB_PROFILE(I)=RDC ;
END ;

CLOSE FILE(DBASES) ;
CLOSE FILE(SYSIN) ;
DOC_COUNT=0 ;
PROCEDURE NEWVAR

ON ENDFILE (FREQUENCY)
    GO TO ENDT ;
/* ******************************************* */
/* READ IN WORD AND TOKEN COUNTS FOR AN ABSTRACT. */
/* IDENTIFY THE WORDS COMMON TO BOTH THE ABSTRACT */
/* AND THE VOCABULARY PROFILE. KEEP TRACK OF THE */
/* NUMBER OF TOKENS AND WORDS AFFECTED */
/* ******************************************* */
BIG:

READ FILE (FREQUENCY) SET (P) ;
DOC_COUNT=DOC_COUNT+1 ;
K=ABSTR_WORD_COUNT(I).R+1 ;
M=ABSTR_WORD_COUNT();.F ;
ITERM=O ;
NTERM=O ;
MATCH,CUMFRQ=O ;
J=1 ;
I=2 ;

L1:

IF VOCAB_PROFILE(J).TERM<ABSTR_WORD_COUNT(I)
   .W
   THEN
   DO ;
   J=J+1 ;
   IF J>N
       THEN
       GO TO J1 ;
   GO TO L1 ;
   END ;

IF VOCAB_PROFILE(J).TERM>ABSTR_WORD_COUNT(I)
   .W
   THEN
   DO ;
   I=I+1 ;
   IF I>K
       THEN
       GO TO J1 ;
   GO TO L1 ;
   END ;

IF VOCAB_PROFILE(J).TERM=ABSTR_WORD_COUNT(I)
   .W
   THEN
   DO :
   MATCH=MATCH+1 ;
   ITERM(MATCH)=VOCAB_PROFILE(J)
   .EXPECTED_FREQ ;
   NTERM(MATCH)=ABSTR_WORD_COUNT(I).F ;
PROCEDURE NEWVAR

CUMFRQ=CUMFRQ+ABSTR_WORD_COUNT(I).F ;
I=I+1 ;
J=J+1 ;
IF [I>K] | J>N THEN
    GO TO JL ;
END ;

GO TO LI ;

/*************************************************************************/
/* COMPUTE PARAMETER VALUES FOR THE ABSTRACT */
/* VALUES ARE PLACED IN ARRAY NAMED REC. EXTERNAL */
/* REFERENCE CDTR IS IN PL/I SCIENTIFIC SUBROUTINE */
/* PACKAGE (CHI SQUARE DISTRIBUTION ROUTINE) */
/*************************************************************************/

JL:

XWORDS=K-1 ;
XCUMFRQ=CUMFRQ ;
XTOKENS=M ;
XMATCH=MATCH ;
IF MATCH=0 THEN
    XMATCH=.5 ;
IF CUMFRQ=0 THEN
    XCUMFRQ=.5 ;
REC(1)=XTOKENS ;
REC(2)=XMATCH ;
REC(3)=XWORDS/XTOKENS ;
REC(4)=XMATCH/(XWORDS+100) ;
REC(5)=XMATCH/(XCUMFRQ+100) ;
REC(6)=XMATCH*XCUMFRQ/(XWORDS*XTOKENS+100) ;
REC(7)=(XMATCH*XCUMFRQ)/(XWORDS*XTOKENS) ;
*(XMATCH/XWORDS+XCUMFRQ/XTOKENS-1) ;
REC(8)=(XWORDS-XMATCH)/XWORDS*(XTOKENS-XCUMFRQ)/XTOKENS) ;
*(XMATCH/XWORDS+XCUMFRQ/XTOKENS-1) ;
REC(9)=(XMATCH*XCUMFRQ)/(XWORDS*XTOKENS+100) ;
*(XCUMFRQ/(XTOKENS+100)+XMATCH/(XWORDS+100)-1) ;
REC(10)=(XWORDS-XMATCH)/(XWORDS+100) *(XTOKENS-XCUMFRQ)/(XTOKENS+100) ;
REC(11)=1/(XCUMFRQ/(XTOKENS+100)+XMATCH/(XWORDS+100)-1) ;
REC(12)=XWORDS/XTOKENS-XMATCH/XCUMFRQ ;
REC(13)=XCUMFRQ/XMATCH-XTOKENS/XWORDS ;
REC(14)=(XTOKENS*XMATCH+XCUMFRQ*XWORDS-XTWORDS*XTOKENS)/(XMATCH*XCUMFRQ) ;
PROCEDURE NEWVAR

BEGIN

END NEWVAR.
PROCEDURE NEWVAR

IF CH2>0
    THEN
        REC(27)=1/CH2 ;
    ELSE
        REC(27)=9999.9999 ;
    DD I=1 TO 28 ;
    IF REC(I)>99999.99999
        THEN
            REC(I)=99999.99999 ;
        IF REC(I)<-999.9998
            THEN
                REC(I)=-999.9998 ;
        END ;
        /*************************************************************/
        /* PUT IN IDENTIFIER INFORMATION: DOCUMENT NUMBER */
        /**************************************************************/
        /* HEADER=ABSTRACT_WORD_COUNT(1).*W ;
        SUBSTR(HEADER,17,4)=CODE ;
        RFS=REC ;
        /* WRITE FILE <INITOBJ> */
        /* WRITE FILE (<INITOBJ>) FROM (RESULT) ;
        GO TO RIG ;
        END ;
    END IT:
END ;

END PROCEDURE NEWVAR
PROCEDURE CALCPR

CALCPR:
PROC OPTIONS (MAIN) REORDER ;
DCL
  (REC,
   COEFIN,
   COEFSOUT)(28),
  1 RES,
    2 RFC FIXED BIN(31),
    2 CLASS CHAR(9),
    2 PRIN,
    2 PROUT,
  (TITLE1,
   TITLE2)CHAR(6),
  1 RCD,
    2 WD CHAR (20),
    2 RR FIXED BIN (31),
  1 RDC,
    2 WR CHAR (20),
    2 DF FIXED BIN (31),
  2 PR,
  (I,
   DCC_COUNT,
   KIN,
   KOUT,
   K,
   M,
   N,
   J,
   MATCH,
   CUMFREQ)FIXED BIN (31),
  (A,
   B,
   C,
   D)FLOAT BIN,
  CTR ENTRY(FLOAT BIN,FLOAT BIN,FLOAT BIN,
          FLOAT BIN),
  ITERM (100)FLOAT,
  NTERM (100)FLOAT,
  1 ABSTP_WORD_COUNT (250)BASED (P),
    2 WC CHAR (30),
    2 F FIXED BIN (31),
    2 R,
  (CDEF,
   CD)CHAR(4),
;
BEGIN ;
PROCEDURE CALCPVR

DCL
   1 VOCAB_PROFILE(1200),
   2 TERM CHAR(20),
   2 TIMES FIXED SIN(31),
   2 EXPECTED_FREQ,
;
ON ERROR
BEGIN
   PUT LIST('ERROR ON DOC=', DOC_COUNT);
   GO TO ENDIT;
END;
ON ENDFILE(SYSIN)
GO TO INITLP2;
******************************************************************************
/* READ CORPUS IDENTIFIER */
******************************************************************************
GET LIST (CODE);
ON ENDFILE(FILES)
GO TO LOOP4;
******************************************************************************
/* READ VOCABULARY PROFILE FILE (FILES). PROFILES */
/* ARE IDENTIFIED IN THE FIRST RECORD BY 'DATABASE' */
/* IN THE FIRST EIGHT BYTES AND BY THE USER */
/* IDENTIFICATION CODE IN THE TENTH THRU THIRTEENTH */
/* BYTES. THE LAST FOUR BYTES IN THE RECORD CONTAIN */
/* A FLOATING POINT NUMBER REPRESENTING THE NUMBER */
/* OF WORDS IN THE PROFILE. WHEN THE PROFILE HAS */
/* BEEN BROUGHT INTO CORE STORAGE, THE COEFFICIENTS, */
/* CONSTANTS, AND PSEUDOCATEGORY TITLES ARE */
/* READ IN AND OUTPUT HEADINGS ARE PRINTED. FINALLY, */
/* COUNTERS ARE INITIALIZED. */
******************************************************************************
LOOP:
   READ FILE (FILES) INTO (RCD);
   IF SUBSTR(WD,1,8)='DATABASE'
      THEN
         GO TO LOOP;
      CD=SUBSTR(WD,17,4);
      IF CD=CODE
         THEN
         GO TO J2;
      GO TO LOOP;
J2:
   N=RR;
   DO I=1 TO N;
      READ FILE(FILES) INTO(RDC);
      VOCAB_PROFILE(I)=RDC;
   ENDDO
PROCEDURE CALCPR

. W
THEN
DO;
I=I+1;
IF I>K
THEN
GO TO J1;
GO TO L1;
END;
IF VOCAB_PROFILE(J).TERM=ABSTR_WORD_COUNT(I)
.W
THEN
DO:
MATCH=MATCH+1;
TERM(MATCH)=VOCAB_PROFILE(J)
.*EXPECTED_FREQ:
TERM(MATCH)=ABSTR_WORD_COUNT(I).F;
CUMFRQ=CUMFRQ+ABSTR_WORD_COUNT(I).F;
I=I+1;
J=J+1;
IF (J)J>N
THEN
GO TO J1;
GO TO L1;
END;

**------------------------------------------------------------------------**
/* COMPUTE PARAMETER VALUES FOR THE ABSTRACT. */
/* VALUES ARE PLACED IN ARRAY NAMED REC. EXTERNAL */
/* REFERENCE CDTR IS IN PL/1 SCIENTIFIC SUBROUTINE */
/* PACKAGE (CHI SQUARE DISTRIBUTION ROUTINE) */
**------------------------------------------------------------------------**
J1:
XWORDS=K-1;
XCMFRQ=CUMFRQ;
XTCKENS=M;
XMATCH=MATCH;
IF MATCH=0
THEN
XMATCH=.5;
IF CUMFRQ=0
THEN
XCMFRQ=.5;
REC(1)=XTCKENS;
REC(2)=XMATCH;
REC(3)=XWORDS/XTCKENS;
REC(4)=XMATCH/(XWORDS+100);
REC(5)=XMATCH/(XCMFRQ+100);
PROCEDURE CALCP R

REC (6) = XMATCH * XCUMFRQ / (XWORDS * XTOKENS + 100) ; 1910

REC (7) = (XMATCH * XCUMFRQ) / (XWORDS * XTOKENS) ; 1920

* (XMATCH / XWORDS + XCUMFRQ / XTOKENS - 1) ; 1930

REC (8) = (XWORDS - XMATCH) / (XWORDS * XTOKENS) ; 1940

* (XCUMFRQ / (XTOKENS + 100) + XMATCH / (XWORDS + 100) - 1) ; 1950

REC (10) = (XWORDS - XMATCH) / (XWORDS + 100) * (XTOKENS -

XCUMFRQ) / (XTOKENS + 100) ; 1960

REC (11) = 1 / (XCUMFRQ / (XTOKENS + 100) +

XMATCH / (XWORDS + 100) - 1) ; 1970

REC (12) = XWORDS / XTOKENS - XMATCH / XCUMFRQ ; 1980

REC (13) = XCUMFRQ / (XMATCH + XWORDS) ; 1990

REC (14) = (XTOKENS * XMATCH + XCUMFRQ * XWORDS -

XWORDS * XTOKENS) / (XMATCH * XCUMFRQ) ; 2000

REC (15) = XCUMFRQ / XTOKENS - XMATCH / XWORDS ; 2010

REC (16) = XTOKENS * XMATCH / XWORDS ; 2020

REC (17) = XWORDS / XMATCH / (XTOKENS + XCUMFRQ + 100) ; 2030

REC (18) = XWORDS / XCUMFRQ / (XTOKENS + XMATCH + 100) ; 2040

REC (19) = XTOKENS * XMATCH / (XWORDS * XCUMFRQ) ; 2050

A = CH1 ; 2060

B = 1 ; 2070

CALL CDTR (A, B, C, D) ; 2080

REC (20) = C ; 2090

IF D > 1 THEN

D = 1 ; 2100

REC (21) = 0 ; 2110

REC (22) = CH1 ; 2120

IF CH1 > 0 THEN

REC (23) = 1 / CH1 ; 2130

ELSE

REC (23) = 999.99 ; 2140

IF MATCH = 0 THEN

END ; 2150

DO ; 2160

X = SUM (ITER4) ; 2170

Y = 0.75 * XTOKENS / X ; 2180

ITERM = ITERM * Y ; 2190

CH2 = 0 ; 2200

DJ I = 1 TO MATCH ; 2210

CH2 = CH2 + ((ITERM(I) - ITERM(I)) * 2) / ITERM(I) ; 2220

END ; 2230
PROCEDURE CALCULATION

1. Initialize:
   - MAX(SUM1, SUM2, SUM3)
   - COUNT(COUNT1, COUNT2, COUNT3)
   - REC(REC1, REC2)

2. Process:
   - Read another transaction and repeat the process.

3. Increment the INCREMENT counter if the transaction is made.

4. Less than the appropriate COUNT and print the document if the RELEVANCE is true.

5. Add the results of the COUNT and SWEEP the data.

6. Compute linear function values for both.

7. END

8. 2540

9. 2650

10. 2760

11. 2870

12. 2980

13. 3090

14. 3200

15. 3310

16. 3420

17. 3530

18. 3640

19. 3750

20. 3860

21. 3970

22. 4080

23. 4190

24. 4300

25. 4410

26. 4520

27. 4630

28. 4740

29. 4850

30. 4960

31. 5070

32. 5180

33. 5290

34. 5400

35. 5510

36. 5620

37. 5730

38. 5840

39. 5950

40. 6060

41. 6170

42. 6280

43. 6390

44. 6500

45. 6610

46. 6720

47. 6830

48. 6940

49. 7050

50. 7160

51. 7270

52. 7380
PROCEDURE CALCPR

SUMIN = SUMIN - X ; 2950
SUMCUT = SUMCUT - X ; 2960
CONIN = EXP(SUMIN) ; 2870
CONOUT = EXP(SUMCUT) ; 2880
PRIN = CONIN / (CONIN + CONOUT) ; 2890
PROUT = CONOUT / (CONIN + CONOUT) ; 2900
IF PRIN > PROUT THEN
  DO ;
    KIN = KIN + 1 ;
    CLASS = TITLE1 ;
    DOC = DOC_COUNT ;
    PUT SKIP EDIT(RES) (F(5), X(5), A, 2F(15, 5)) ; 2970
    WRITE FILE(RESU) FROM(RES) ;
    GO TO BIG ;
  END ;
ELSE
  DO ;
    KOUT = KOUT + 1 ;
    GO TO BIG ;
  END ;
LOOP4:
  PUT LIST(
    'DATA BASE NOT FOUND OR HAS INCORRECT COUNT'
  ) ;
GO TO ENDIT ;
OUT:
/* PRINT FINAL TOTALS */ 3120
/* PUT SKIP EDIT(KIN,' DOCUMENTS IN ' ,TITLE1)
   (F(5), A, A) ;
   PUT SKIP EDIT(KOUT,' DOCUMENTS IN ' ,TITLE2)
   (F(5), A, A) ;
CLOSE FILE (RESU) ;
END ;
/* EACH RECORD IN RESU CONTAINS AN IDENTIFICATION */ 3210
/* NUMBER. EACH ABSTRACT IN THE ABSTRACT FILE */ 3220
/* CONTAINS A HEADER WHICH ALSO HAS AN */ 3230
/* IDENTIFICATION NUMBER. THIS ESTABLISHES THE */ 3240
/* NECESSARY CROSS-REFERENCE FOR RETRIEVAL OF */ 3250
/* ABSTRACT ON THE BASIS OF THE EARLIER PORTION OF */ 3260
/* THE PROGRAM. HEADER OR IDENTIFIER INFORMATION IN */ 3270
/* THE ABSTRACT FILE WHICH WAS GENERATED IN EARLIER */ 3280
/* PROCESSING PROGRAMS IS REMOVED PRIOR TO PRINTING */ 3290
/* THE RETRIEVED ABSTRACTS, AT THE END OF THE */ 3310
PROCEDURE CALCPR

/* PROCESSING, THE COUNT OF PRINTED ABSTRACT IS */
/* PRODUCED. */

BEGIN ;

DCL

ERIC CHAR(7000)VAR,
 1 IMP BASEC(P),
 2 CASE FIXED BIN(31),
 2 CLASS CHAR(8),
 2 PRIN,
 2 PROUT,

I=0 ;
ON ENDFILE(ABSTRACT)
  GO TO L5 ;
ON ENDFILE(RESU)
  GO TO L5 ;
READ FILE(ABSTRACT) INTO(ERIC) ;
READ FILE(RESU) SET(P) ;

L3:
UNSPEC(J)=UNSPEC(SUBSTR(ERIC,10,4)) ;
L4:
IF CASE>J
  THEN
    DO ;
      READ FILE(ABSTRACT) INTO(ERIC) ;
      GO TO L3 ;
      END ;
    IF CASE<J
      THEN
      DO ;
        PUT SKIP LIST(' DOCUMENT NOT FOUND ','CASE,
          CLASS) ;
        READ FILE(RESU) SET(P) ;
        GO TO L4 ;
      END ;
    IF CASE=J
      THEN
      DO ;
        ERIC=SUBSTR(ERIC,21) ;
        I=I+1 ;
        LIN=LENGTH(SYSPRINT) ;
        IF LIN>1&LIN<40
          THEN
            LIN=LIN+5 ;
          ELSE:
            LIN=1 ;
PROCEDURE CALCPR

PUT DATA(CODE,J)LINE(LIN); 3790
PUT SKIP; 3800
PUT LIST(ERIC); 3810
PUT SKIP(2); 3820
READ FILE(RESU)SET(P); 3830
READ FILE(ABSTRACT)INTO(ERIC); 3840
GO TO L5; 3850
END; 3860

L5:
CLOSE FILE(RESU); 3870
CLOSE FILE(ABSTRACT); 3880
DO ; 3890
PUT SKIP(4); 3900
PUT LIST(I,"DOCUMENTS RETRIEVED FOR 'CODE'); 3910
END; 3920
END ; 3930
END ; 3940
END ; 3950
END ; 3960
END ; 3970
/* JCL FOR <SELECT> */

/* SELECT EXEC */
PROCL=PLRUN,TIME=5,PARM.CMP='OPT=2'
/* CMP.SYSIN DD */

/* PROGRAM DECK */

/* GO.SYSIN DD */

/* DECK C-1 */

/* GO.ORIGINAL DD */
DSN=ORIGINAL,
UNIT=2400,
VOL=SER=ZZZZZZ,
DISP=OLD

/* GC.SELECTED DD */
DSN=SELECTED,
UNIT=2400,
VOL=SER=WWWWWW,
DCB=(RECFM=VA,LRECL=7000),BLKSIZE=7004,
BUFNO=4),
DISP=(NEW,KEEP)
//* JCL FOR <SETUP>

//SETUP EXEC PROC=PLRUN,TIME=20,PARM.CMP='OPT=2'
//CMP.SYSIN DD *

//* PROGRAM DECK

//GO.FREQUENCY DD DSN=FREQUENCY,
//UNIT=2400,
//VOL=SER=XXXXXX,
//DCB=(RECFM=VB,LRECL=7000,BLKSIZE=7004,
//BUFNO=4),
//DISP=(NEW,KEEP)

//GO.ABSTRACT DD DSN=FREQUENCY,
//UNIT=2400,
//VOL=SER=YYYYYY,
//DCB=(RECFM=VB,LRECL=7000,BLKSIZE=7004,
//BUFNO=4),
//DISP=(NEW,KEEP)

//GO.ORIGINAL DD DSN=ORIGINAL,
//UNIT=2400,
//VOL=SER=ZZZZZZ,
//DISP=OLD

//GO.SYSIN DD *

//* DECK C-5

//
/* JCL FOR <CALCPR> */

//CALCPR EXEC PL?UN,TIME=10,PARM.CMP='OPT=2'
//CMP.SYSIN DD *

/* PROGRAM DECK */

//LKED.SYSLIB DD DSN=SYS1.PLILIB,DISP=SHR
// DD DSN=SYS1.PLISSP,DISP=SHR
//GO.DBASES DD DSN=DBASES,
// UNIT=2400,
// VOLL=SER=NNNNNNNN,
// DISP=OLD
//GO.FREQUENCY DD DSN=FREQUENCY,
// UNIT=2400,
// VOLL=SER=XXXXXXX,
// DISP=OLD
//GO.ABSTRACT DD DSN=ABSTRACT,
// UNIT=AFF=CBASES,
// VOLL=SER=YYYYYYYY,
// DISP=OLD
//GO.SYSIN DD *

/* DECK C-3 */

//GO.RESU DD DSN=RESU,
// UNIT=SYSDA,
// SPACE=(TRK,(1,1)),
// DCB=(RECFM=FB,LRECL=20,BLKSIZE=1600),
// DISP=(NEW,DELETE)
// *  JCL FOR NEWVAR *

//NEWVAR EXEC PROC=PLRUN, TIME=5, PARM.CMP='OPT=2'
//CMP.SYSIN DD *

//** PROGRAM DECK

//LKEDSYSLIB DD DSN=SYS1.PLIB, DISP=SHR
// DD DSN=SYS1.PLLSSP, DISP=SHR
//GO.DBASES DD DSN=DBASES,
// UNIT=2400,
// VOL=SEP=NNNNNNM,
// DISP=OLD
//GO.SELECTED DD DSN=SELECTED,
// UNIT=2400,
// VOL=SEP=WWWWWW,
// DISP=OLD
//GO.INITDB DD DSN=INITDB,
// UNIT=2400,
// VOL=SEP=RRRRRR,
// DCB=(RECFM=FB, LRECL=300, BLKSIZE=7200),
// DISP=(NEW,KEEP)
//GO.SYSIN DD *

//** DECK C-6

//
//* JCL FOR <MRG> *

//MRG EXEC PROC=PLRUN,TIME=10,PARAM,CMP='CPT=2'
//CMP.SYSIN DD *

//* PROGRAM DECK *

//GO.SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//GC.CORPUS DD DSN=CORPUS,
//UNIT=2400,
//VOL=SER=VVVVV,
//DCB=(RECFM=FB,LRECL=29,BLKSIZE=7000),
//DISP=(OLD,KEEP)
//GO.NSPD DD DSN=CORPUS,
//UNIT=2400,
//VOL=(,RETAIN,SER=UUUUU),
//DCB=*,CORPUS,
//DISP=(NEW,KEEP)
//GO.SORTIN DD UNIT=SYSDA,
//SPACE=(CYL,(5,5)),
//DCB=(RECFM=FB,LRECL=29,BLKSIZE=1736),
//DISP=(NEW,DELETE)
//GC.SORTOUT DD UNIT=SYSDA,
//SPACE=(CYL,(5,5)),
//DCB=*,SORTIN,
//DISP=(NEW,DELETE)
//GO.SORTWKO1 DD UNIT=SYSOA,SPACE=(CYL,(5,5))
//GO.SORTWKO2 DD UNIT=SYSOA,SPACE=(CYL,(5,5))
//GO.SORTWKO3 DD UNIT=SYSOA,SPACE=(CYL,(5,5))
//GO.SORTWKO4 DD UNIT=SYSOA,SPACE=(CYL,(5,5))
//GO.SORTWKO5 DD UNIT=SYSOA,SPACE=(CYL,(5,5))
//GO.SORTWKO6 DD UNIT=SYSOA,SPACE=(CYL,(5,5))
//GO.DBASES DD DSN=DBASES,
//UNIT=AFF=CORPUS,
//VOL=SER=NNNNNN,
//DCB=*,SORTIN,
//DISP=(NEW,KEEP)
//GOSELECTED DD DSN=SELECTED,
//UNIT=2400,
//VOL=SER=WWWWWW,
//DISP=OLD
//GO.SYSIN DD *

//* DECK C-2
BIBLIOGRAPHY


6. BMP: Biomedical Computer Programs, Health Sciences Computing Facility, Department of Preventive Medicine and Public Health, College of Medicine, University of California, Los Angeles, 1964.


