SIMULATION

OF

DOCUMENT TITLE

DATA BASE

Tasneem M. Syed

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada

April, 1978

ABSTRACT

SIMULATION OF DOCUMENT TITLE DATA BASE

Tasneem M. Syed

Simulation is studied in order to determine its effectiveness as a tool for the evaluation of a document title retrieval system. Based on the statistical behaviour of the word frequency distribution as stated in Zipf's law, a model has been built to simulate an information retrieval system by generating pseudo terms, pseudo documents, pseudo queries, pseudo relevance judgements for relevant and non-relevant documents, and pseudo relevance rating for terms in context and terms out of context.

In order to avoid storage problems the method does not require generation of the whole data base; instead only the documents required to process the user query terms are stored. Use of such a storage scheme has allowed the model to be tested with a data base as large as 50,000 documents and 400,000 terms. Furthermore it is possible to simulate even larger data bases by use of a relatively small amount of memory.

Finally experiments are described to indicate the use of the model in simulation of a user-system interaction in which a feedback mechanism is used to produce successive improvements in retrieval effectiveness through user controlled question modification.

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to Professor H.S. Heaps who provided me all the support and valuable advice, constantly during the course of my thesis research. I am also thankful to my friends who helped me to put the final draft of the thesis together.

Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

Introduction

1.1 The general effectiveness problem in information retrieval

Effectiveness and Efficiency are the traditional measures of performance of information retrieval systems. The effectiveness of a document retrieval system is usually expressed in terms of recall and precision ratios. The recall ratio is defined as the proportion of relevant documents actually retrieved, whereas the precision ratio is the proportion of retrieved documents actually relevant. The two ratios thus provide some measure of user satisfaction with the system. On the other hand, efficiency is often defined as a measure of effectiveness for a given cost level. For example, one information retrieval system might be said to be more efficient than another if it achieves the same effectiveness at a lower cost. The principal cost factors that affect the efficiency of an information retrieval system are the initial system design and building cost, the maintenance, servicing, and updating costs, and the user utilization cost.

The most straightforward strategy for evaluation of an information retrieval system involves starting with a model of the system followed by examination of the key variables and parameters that characterize the performance of the system. A general model of an information retrieval system usually contains the following components.

a) Data base containing the documents for retrieval
b) Man-machine communication system
c) Organization of data
d) Software structure.

The above four units of any information retrieval model contain the important and necessary parameters that influence the performance of the system. Since evaluation is an analytical procedure, it is important to determine how far the system satisfies the user requirements. It is also necessary to find the possible sources of system failure and the means to remedy them.

The above components of the model can be decomposed into simpler and smaller units directly related to the performance criteria. Important parameters are:

1) The relevance of documents in the database.
2) The ability of the system to retrieve relevant documents.
3) The ability of the system to withhold non-relevant documents.
4) The amount of user effort involved in the search process.
5) The type of output obtained from the system.
6) The processing capabilities for treatment of user search requests.
7) The type of user (naive, skilled, browsing, etc.).

8) The structure of data in the data base.

The objective in the design of an information retrieval system is the creation of a system that closely satisfies the needs of each user. Since information retrieval systems should be user oriented, the prime objective is to retrieve all relevant, and only relevant, information in response to a user query. Such an ideal system, however, does not exist in practice.

There are numerous factors that affect the retrieval performance. However the effectiveness of a literature search or reference retrieval, is often characterized by the recall ratio R (Number of retrieved relevant documents divided by the total number of relevant documents) and the precision ratio P (Number of retrieved relevant documents divided by the total number of retrieved documents). The ideal information retrieval system is characterized by both R=1 and P=1. However, experience suggests that R and P are not usually equal to 1 simultaneously.

In order to make an evaluation study of an information retrieval system it is important to describe the evaluation environment. Such environment includes the type of information retrieval system, for example whether it is a document retrieval system, a fact retrieval system, a management information system, or some other system. Specification of the environment also includes characterization of the type of users that will use the system. Such users might include computer scientists, librarians, managers, students, and so forth. All these factors are very important to know prior to the start of any information retrieval evaluation study. This is because there is considerable variation in the required standards of effectiveness and efficiency for different environments and user classes.

Once the environment is determined and the goals are set, the traditional methodology for treatment of the problem is to design the information retrieval system model, write the test programs, and then run the programs to determine the effect of the individual parameters on the degree of effectiveness and efficiency that is achieved. The experiments must be repeated until sufficient performance data is collected to adequately describe the behaviour of the system. The procedure is usually carried out on existing data bases, and often with operational information retrieval systems.

The process of evaluation is usually costly. The total cost of evaluating an information retrieval system could be divided into the following three major parts [1,2].

1) Cost of building the system.
2) Cost of maintaining the system including update, maintenance, and servicing the clients.
3) The user cost in terms of using the system.

These costs, when added together, are sometimes unacceptably high or lead to an unacceptably low resulting effectiveness. There may, indeed, be levels of effectiveness that are not attainable at any cost. This occurs when insufficient knowledge is available for a proper development of the search techniques, or when the organizational environment does not allow proper implementation of the required techniques. However, determination of a suitable tradeoff between the cost and the desired effectiveness is one of the important design considerations.

An alternative technique for performance evaluation, and one that is more commonly used in hardware branches of computer science, is the method of simulation. A literature search for information about simulation studies for document title retrieval has indicated that very little work has been done. Consequently no information is available for assesment of the value of simulation as a tool for the evaluation of information retrieval systems. The results reported in the present thesis, dealing with simulation of the occurrence and retrieval of document titles, suggest that simulation may prove to be a useful tool. It is conjectured that simulation may prove as fruitful in studies of information retrieval systems as it has in many branches of engineering [3,4].

It may also be remarked that in experiments with a large data base it is necessary to use a large amount of file storage, although the processing of any given question requires access to a very small proportion of the total file space. In use of the simulation technique described in the present thesis it is sufficient to generate only those portions of the files that are actually accessed for a given question. Thus storage is required only for these generated portions of the files. Since the amount of storage required is dependent on the number of documents to be output, rather than on the size of the data base, the present procedure may be used to simulate searches on very large data bases without the need for a large amount of storage.

1.2 The Simulation Model

In the design of the model the Zipf law of word frequencies is used as the basis for simulation of the word frequency distribution in the data base. In fact, it is well known that such an empirical law is satisfactory for application to many document data bases that involve textual data.

Document titles in the simulated data base are generated by means of pseudo random numbers. A user with a certain interest is associated with a "user relevance probability" C, which is the probability that a document selected randomly from the document collection is relevant to his interest. It should be emphasized that the documents relevant to his interest are not necessarily the documents that satisfy his question profile. If M is the total number of documents in the data base then the expected number of relevant documents for the particular user is CM. The model simulates term-document associations by generating pseudo random numbers between 0 and 1. The simulator classifies a document as relevant if the generated random number is less than or equal to the user's relevance probability C; otherwise it is classified as non-relevant to the user. To calculate the number of relevant and non-relevant documents associated with a particular term the model considers that certain terms tend to associate more with relevant documents than with non-relevant documents. Certain measures of the association are designated as "relevance ratios". Similarly, other terms with no given relevance ratios are classified into "content" and "non-content" terms according to some given probabilities. The relevance ratios of such terms are computed on the basis of their relative frequencies, the calculation being made by an automatic procedure in the simulator.

The input to the simulation model is the size of the data base, the user relevance probability, the relevance ratios, and finally the probabilities for content and non-content terms. A user query consisting of a set of index terms combined with boolean operators (AND,OR,NOR) is supplied to the simulator, which subsequently produces an output in the following form.

a) A list of the total number of relevant documents associated with the query.

b) The total number of documents retrieved.

c) The recall-precision ratios.

A list that indicates the term content of each retrieved document may also be obtained. The resulting recall-precision ratios may be changed by reformulating the query. The reformulation may be repeated a number of times until sufficiently optimal recall and precision ratios are obtained. The values of recall and precision are regarded as acceptable if the user is satisfied with the retrieved documents. The process is represented in Fig. 1.1, and is intented to simulate the behaviour of a terminal user who requests a search, examines the resulting output, reformulates his query, requests a further search, and so forth. Of course a user would not necessarily know the values of the recall ratios, and so each manual reformulation of the question should be made without reference to the computed values of this ratio.

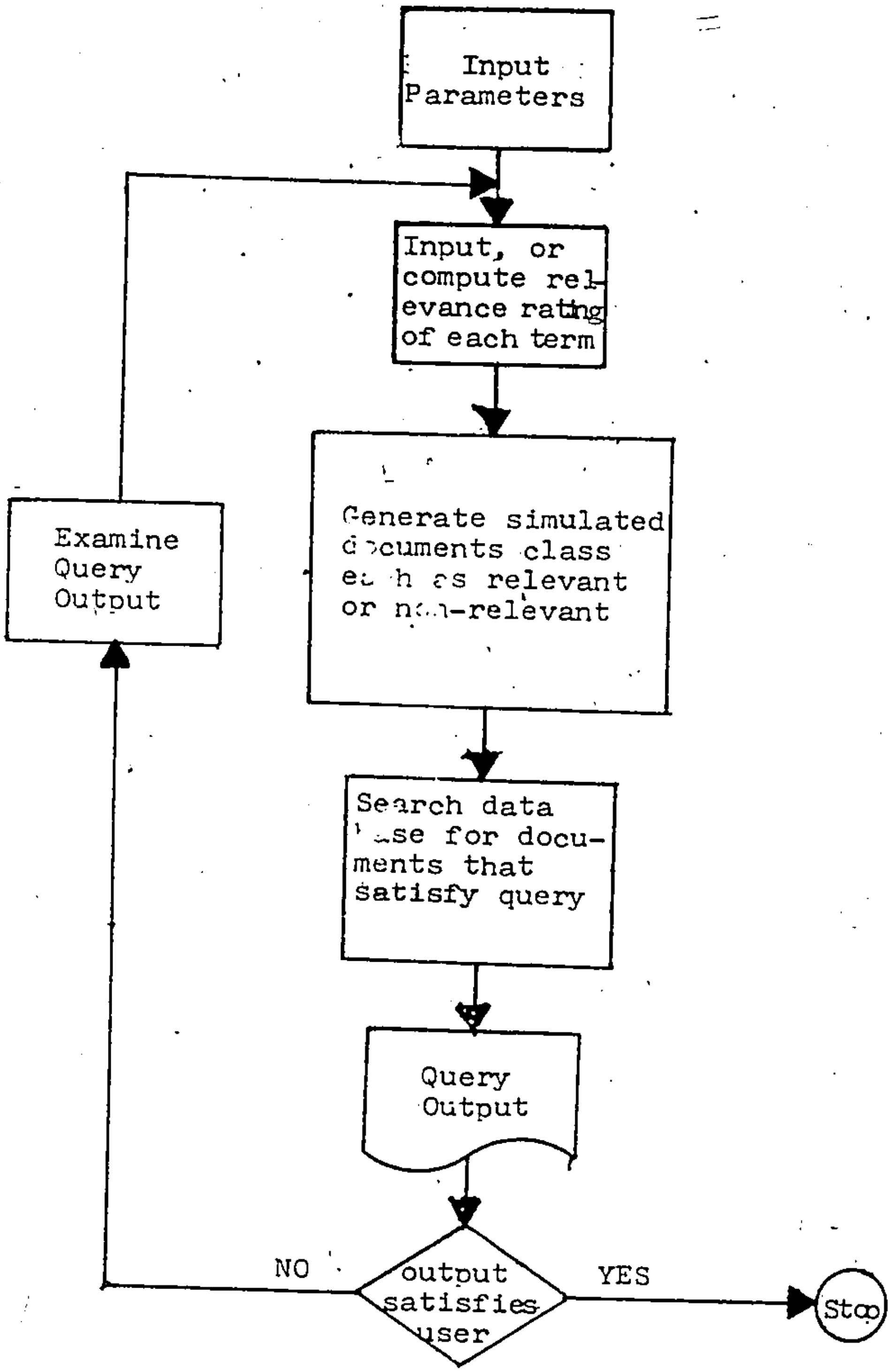


Fig. 1.1 The general simulation model

Any evaluation technique must be reliable and dependable. It should give accurate estimates of performance and be meaningful in relation to a real system. The simulation results should be stable and accurate enough to form a basis for the design of future real systems. The simulation procedure should be cost effective. These are the general expectations of any simulation technique.

The simulation model described in the present thesis allows a number of factors to be taken into account. These factors include the following:

a) Size of the data base.
b) Whether the user of the retrieval system has chosen a data base that contains documents in fields appropriate to his interest.
c) The efficiency of the indexing of documents in the data base with respect to the interest of the particular user. The efficiency depends on the number of words of high indexing value, on the number of hononyms, and on the manner in which the index terms have been chosen at creation of the data base [2,5].
d) The user's ability to select good search terms.
e) Feedback to the user, so that by examination of the output documents he may modify his query.
f) The effect of using different samples of a data base.

It is realized that the statistical model described in the present thesis has its limitations. More study is needed in order to determine the full nature of these limitations, to reduce processing time, and in order to increase the applicability of the model. In particular the following questions may be asked:

1) How reliable is the value of C?
2) What properties should be satisfied by the list of retrieved documents in order for it to be acceptable to the user without the need for further reformulation of the question?
3) Is the recall-precision ratio biased?

Since simulation of document title retrieval systems is a new approach to system evaluation, it cannot yet be considered as a proven evaluative technique until there is sufficient evidence of its general applicability. The method described in the present thesis is dependent on a satisfactory description of certain statistical properties of document title data bases. The universality of certain such properties is discussed in the light of past observation. It is realized however that, in the absence of further experience with simulation of document data bases, the results of any particular study, such as the present one, must be accepted with caution.

CHAPTER II

Properties of bibliographic data bases

2.1 Zipf's law

In the study of natural language text it is well known that a comparatively small percentage of words account for a very large percentage of the total words used in the text. Studies of word distributions and language statistics by linguists and others have led to the so-called Zipf law to describe the frequencies of occurrence of words in general English text [2] [6-8]. The Zipf law may be stated as follows. Suppose that the number of occurrences of each different word in a text is counted and the words are then arranged in a table in which the first word is the most frequent, the second word is the second most frequent, and so forth. The order of any word in the list is called its rank r, and the number of occurrences of that word is called its frequency f. The law then states that

$$rf=c$$

where c is a constant for each particular text.

The size of the vocabulary in either a natural language or in a document data base is usually determined by the environment in which the language or data base has grown. Important factors of the environment are, for example, the quality of the human mind, type restrictions on the use of words, and the limitations of subject matter.

In many instances, as new terms are added the vocabulary continues to satisfy the Zipf law. This law may be illustrated by logarithmic scales.

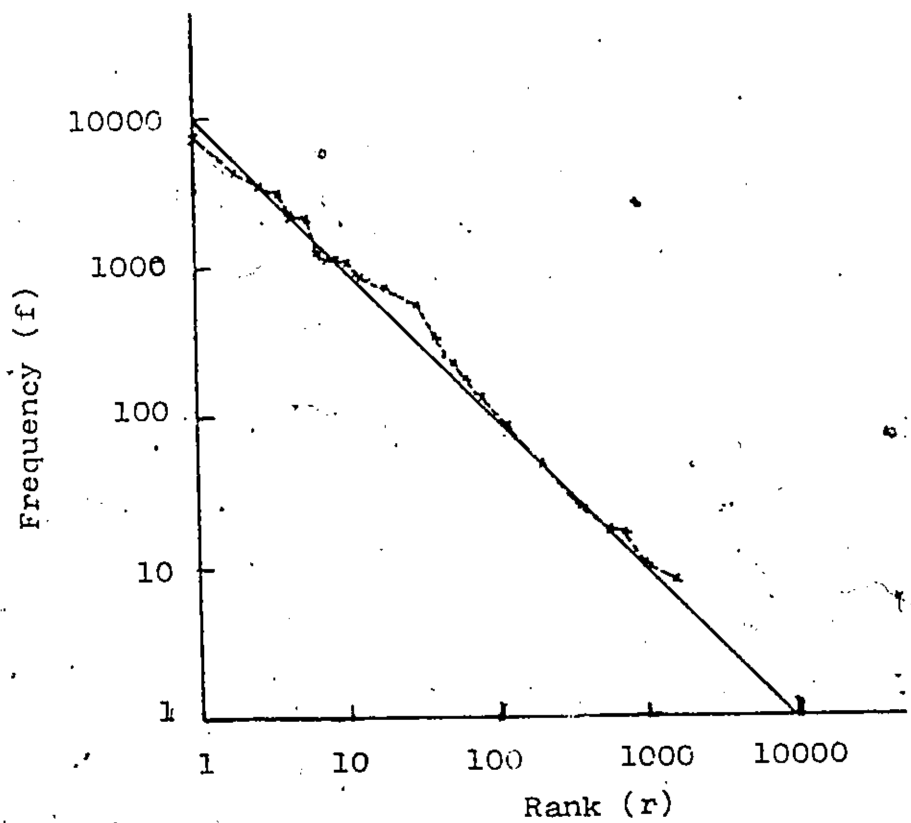


Fig. 2.1

The solid line is the ideal representation of the Zipf law in which log(f) is plotted against log(r). The dotted line represents a typical observed variation. If the equation were satisfied exactly then the points in Fig. 2.1 would lie on the solid line.

The Table 2.1 shows the word frequency distribution for words contained in the Chemical Abstracts titles present on some issues of tapes issued by the American Chemical Society [9] .

Word	Rank	Frequency
OF	1	107,687
AND	2	37,578
THE	3	36,318
IN	4	32,868
ON	5	10,984
BY	6	10,727
A	7	10,252
DI	8	8,419
WITH	9	7,964
FOR	10	6,509
METHYL	20	4,030
ACIDS	30	2,697
5	40	2,236
AN	50	1,890
PER	100	1,210
SULFIDE	200	736
8	300	503
METALLIC	400	395
FLUORESCENCE	500	316
TOXIN	1,000	147
OXYTOCIN	2,000	67
DECREASE	3,000	36
GERMINATING	4,000	20
RESONATOR	5,000	14

Table 2.1 Word Frequency Distribution

A typical Zipfian distribution of word usage in a technical information system is illustrated by Fig. 2.2 in which the cummulative percentage of term frequencies is plotted against cummulative percentage of terms contributing to the total.

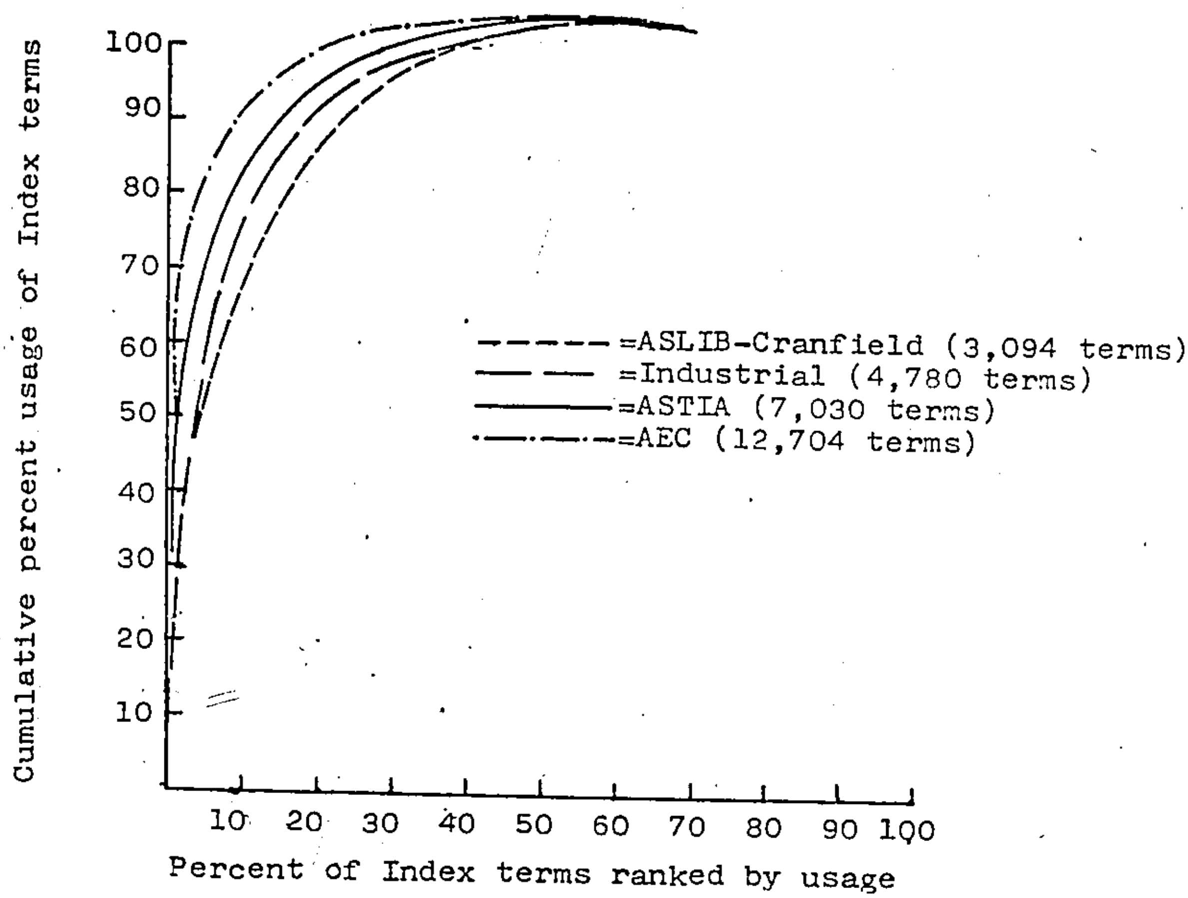


Fig. 2.2 Distribution of term usage

The distribution illustrated in Fig. 2.2 is based on results from the Arthur D. Little report [10] on an industrial information system, an information system of the U.S. Atomic Energy Commission AEC, the Defence Documentation centre ASTIA, and the experimental corpus established in the Cranfield study [11]. In the Cranfield study the most frequent 10 percent of the terms accounted for 68 percent of all the postings (the number of times the descriptor has been used for indexing) and the most frequent 30 percent of the terms accounted for close to 90 percent of the postings. After the 30 percent level of the index terms the curve flattens out.

Observation of occurrences of terms indicates that in practice the Zipf law is not obeyed exactly. However, in the study of document title data bases the frequencies of word occurrences are usually found to be in approximate agreement with the Zipf law. It may be noted that with a slight modification of the first law, the number of words occuring once, twice, and in general n times, where n is a small integer, can be calculated as follows [12, 13]:

Let Np(1) indicate the number of occurrences of word of rank 1

Np(2) indicate the number of occurrences of word of rank 2

Np(r) indicate the number of occurrences of word rank r

where p(r) is the probability of occurrence of the word of rank(r). If N is the total number of words, and D the total number of different words, then the Zipf law may be interpreted as predicting the rth word to occur once if

$$1.5 > Np(r) >= 0.5$$

Zipf's first law states that:

$$p(r) = k/r$$

where k is the constant for the particular text, so that

$$1.5 > N*k/r >= 0.5$$

Therefore r has a value between $r_{max}=k*N/.5$

and $r_{min}=k*N/1.5$

The number of words occurring once, I1, is the difference of r_{max} and r_{min} and hence is as follows:

$$r_{max}-r_{min}= I1 = (4/3)*k*N$$

For a word that occurs n times the similar condition is

$$(n+0.5) > Np(r) >= (n-0.5)$$

from which it follows that

$$In = k*N/(n^2 - 1/4)$$

The second equation is a modified form of the first Zipf law and is sometimes called the second Zipf law.

The ratios In/I1 can always be calculated for any text sample. The predicted value of the ratios may then be compared with the calculated values in order to describe the extent to which the particular text or data base satisfies the second form of Zipf's law. Alternatively, in reference to the terms used to index the documents of a data base, the second form of the Zipf law may be regarded as a useful tool for the prediction of the number of index terms that occur with a given low frequency.

In the present attempt to simulate a document data base the occurrence of terms is generated by means of pseudo-random numbers. In the simulation model the Zipf law is interpreted in the following form

$$M_i=A*N/i$$

where A is the constant chosen to satisfy

$$A*N/1+A*N/2+A*N/3+ -----+A*N/D=N$$

to ensure that the sum of all the different word frequencies is equal to the total number of word occurrences.

Hence

$$A*N(1+1/2+1/3+ ----+1/D)=N$$

which implies

$$A=1/(\log_e D+\Psi)$$

where

Ψ is Euler's constant 0.5772...

M_i is the word frequency for the i-th term

N is the total number of terms in the data base

D is the total number of different terms

i is the rank of the term

It may be observed that Np(r) is equal to the word frequency f. Thus the following relationship holds between the quantities of the Zipf law and the variable names in the model:

M_i is equal to f

A/I is equal to p(i)=k/i

The Fig. 2.3 illustrates how well the simulated terms were found to have a rank fréquency distribution close to that of the ideal solid line of the Zipf curve.

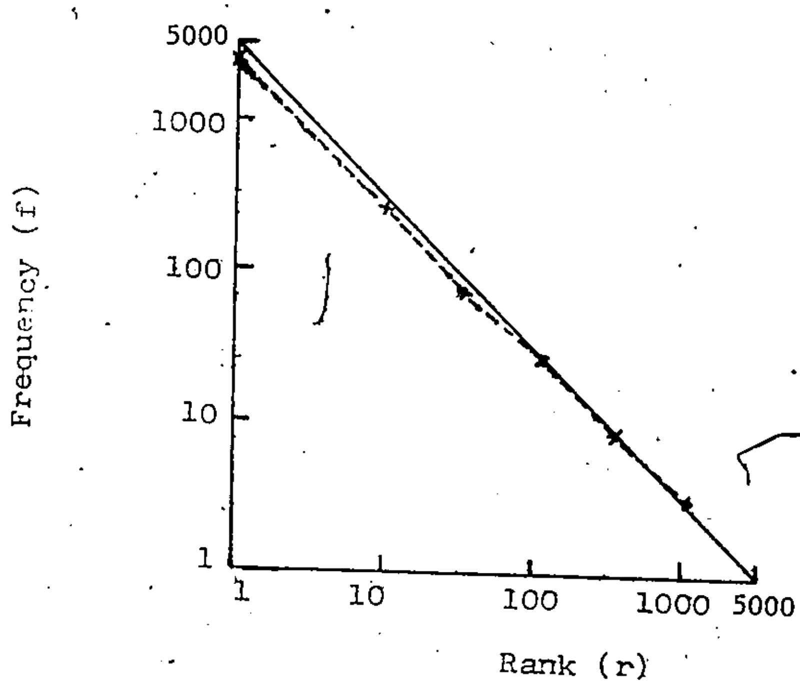


Fig. 2.3 Simulated term distribution

In Fig. 2.3 the dotted line indicates the variation of log(f) with log(r). The graph indicates the relationship between the position of occurrence of a particular word in the simulated data base and its frequency of occurrence.

Table 2.2 contains a list of randomly generated word positions with their rank and frequencies.

Word Position	Rank	Frequency
1	1	4241
2	2	2120
3	3	1413
4	4	1060
5	5	848
50	50	84
51	51	83
52	52	81
53	53	80
54	54	78
100	100	42
101	101	41
102	102	41
103	103	41
104	104	40
500	500	8
501	501	8
502	502	8
503	503	8
504	504	8
1000	1000	4
1001	1001	4
1002	1002	4
1003	1003	4
1004	1004	4

Table 2.2 Rank Frequency distribution of generated random numbers

A close examination of Table 2.2 indicates that the distribution of index terms generated by means of the pseudo-random numbers follows the pattern of word distribution in accordance with the Zipf law. The agreement is important in the simulation model. Otherwise the vocabulary distribution, its growth, and its use may result in indexing that is at variation with the results found in real data bases.

2.2 Vocabulary growth

In bibliographic data bases various components of bibliographic information (author, title, subject, etc.) are represented by sets of index terms. When a document is indexed according to one or more criteria it is assigned to a set of classes, depending upon the subject matter, as represented in Fig. 2.4 [2,14-16].

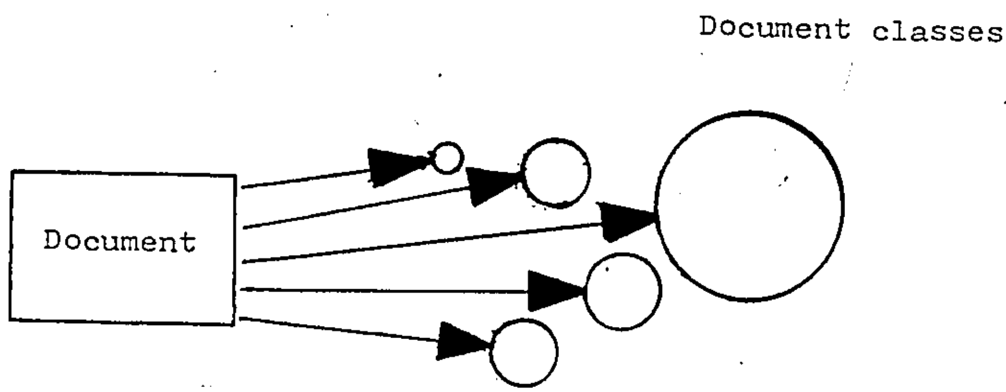


Fig. 2.4 Formation of document classes by indexing process

The subject matter is not necessarily the same as the subject area. It is assumed that the documents to be indexed belong to the same subject area but contain different subject matter. The narrow difference is related to the characteristics that distinguish different documents in the same field. The names given to these classes are generally known as index terms, and the complete set of these index terms is called the index language. It may be observed that the index terms carry the same meaning as the classes. To retrieve documents from a data base a search request is formulated in order to determine the document classes most likely to contain items relevant to the given search request. The classes are then examined and some documents are retrieved.

In the previous chapter there was some discussion of the criterion of effectiveness and the parameters that affect this criterion. It is generally observed that the effectiveness of an information retrieval system is closely related to the number of document classes [2]. If the number of document classes is large it is easier to find a large number of documents that relate to a particular topic. This results in a high recall, but makes it more difficult to retrieve only relevant documents and hence to attain high precision. This means, in fact, that due to the large number of document classes there is a high probability that many relevant documents are retrieved. However, at the same time the probability of retrieving too many non-relevant documents is also high, and this may reduce the precision value. The situation is reversed as the number of document classes is reduced. For an effective retrieval system it is therefore necessary to exercise some control on the growth of the number of index terms.

In information retrieval terminology a set of controlled index terms is usually called a controlled vocabulary. In the process of indexing the controlled vocabulary forms the indexing language. The two major aspects of a controlled vocabulary relate to

1) Indexing
2) Searching

The procedure of assigning vocabulary terms to describe documents with the help of a given vocabulary is usually called indexing. The searching procedure, on the other hand, involves a matching algorithm that acts on a question that may have full or partial specifications. The specification aspect refers to the extent to which the searcher is specific in formulating his query. For example the use of a truncated question term such as COMPUT* is less specific than use of the single query term COMPUTER. Furthermore, the searching process may also be regarded as a search of document classes for retrieval of documents in response to a search request. On the basis of the occurrence of the index terms present in the search request some of the documents may be retrieved and others may be discarded. This indicates that the vocabulary also has a suggestive role to play in the search process. It suggests the language that the searcher should use, since as a result of examining the documents corresponding to his request he may be directed from the use of non-accepted terms to accepted terms. This suggestive role in searching that is played by the organization of vocabulary helps the user to formulate the best possible strategy in terms of system user need (acceptable recall and precision ratios)[17-19].

The matching algorithm is basically independent of the indexing language. However the index language is designed with the intent to bring the vocabulary of the indexer and the vocabulary of the searcher into agreement.

The vocabulary for indexing and searching usually cannot remain static, and therefore it continues to grow as the data base grows. Clearly, the growth of vocabulary for indexing the documents that belong to a certain subject area is much greater when the data base is first created, and it gradually decreases after a certain number of papers have been indexed [2]. Furthermore, the vocabulary growth within different subject areas is different. For example, the rate of growth of vocabulary in chemical abstracts may, or may not, be greater than in management science. Similarly the size of the chemical abstracts vocabulary may, or may not, be greater than that of management science.

Besides the dependence of size and growth of vocabulary on the subject area, the specificity of vocabulary is also closely related to the vocabulary size. Specificity of vocabulary usually means the ability of vocabulary to express precisely the subject topic of a search document [20]. In many instances the vocabulary of the discipline is not sufficiently specific to allow the indexer to index all documents in such a manner as to make them uniquely identifiable. This may result in some documents being indexed under more general terms, and eventually the document class becomes included within the broader class. One mechanism that may be used to achieve more specificity in the vocabulary is to combine some index terms in order to form new document classes. For example, two index terms such as COMPUTER and DESIGN that belong to the class COMPUTER and the class DESIGN may be joined together to form a new class COMPUTER DESIGN. The individual index terms that represent broad concepts, and thus which are general in nature, are joined together to specialize the concept in order to make the new vocabulary term more specific. For example

Search Request: Design of Computers

Search Strategy: COMPUTERS AND DESIGN

may retrieve a document that is not relevant since it deals, not wit: the design of computers, but with the design of aircraft by use of computers. The search strategy has caused an incorrect term-relationship [5] . Suppose, for a further example, that the vocabulary and the search strategy had contained the term COMPUTER DESIGN. Then the retrieved document may have been relevant. This means that a specific vocabulary is convenient and important for the retrieval of relevant documents in general.

Basically there are two ways in which the system vocabulary may be used to index and retrieve documents. One way is to create an index term that uniquely identifies a specific class. For example, if the index term COMPUTER ARCHITECTURE is adopted then the term itself establishes a relationship between the basic words COMPUTER and ARCHITECTURE [21] . In other words, the system vocabulary includes a label to identify a class that is the logical product of two other classes. Such a vocabulary is generally known as a pre coordinate vocabulary. In such terminology the term COMPUTER ARCHITECTURE is th pre coordination of the term COMPUTER and the term ARCHITECTURE.

To consider another extreme it may be mentioned that there are systems completely free from any pre coordinate vocabulary. In such systems the indexer is allowed to use only basic words, or single words, for document indexing. In this case the vocabulary terms consist only of individual words. Following the above example, the topic COMPUTER ARCHITECTURE is indicated by assigning to the document the separate index terms COMPUTER and ARCHITECTURE. This type of vocabulary is known as a post coordinate vocabulary. In conducting searches on such a system the user is allowed to manipulate the classes in order to derive their logical product, logical sum, or logical complement [22].

The above mechanism of term relationship, assigned at the time of vocabulary growth, is known as precordinate indexing. Similarly there is the alternate procedure whereby the index terms in the vocabulary are single words and two or more words are not allowed to be merged together to create a new class, but the searcher, at search time, manipulates the terms by using logical combinations of the terms of the vocabulary. This is known as post coordinate indexing. It allows manipulation of the classes at the time of searching. In the case of pre coordinate indexing there is no facility for manipulating classes. Instead, the class relationships are built into the language. The size of a precordinate vocabulary is larger than that of a uniterm vocabulary (a vocabulary that contains index terms consisting of only single words). It is obviously so because, as new classes are introduced in pre coordinate indexing, the vocabulary size is increased.

Consider, for example, in a pre coordinate vocabulary a descriptor AIRCRAFT ENGINE NOISE. Since AIRCRAFT, ENGINE, NOISE, AIRCRAFT ENGINE, ENGINE NOISE etc. are all descriptors that are likely to appear in subject headings the terms must all appear in the vocabulary list as independent classes. On the other hand, it is known that in the uniterm vocabulary there is no term coordination or term intersection mechanism. It may be observed that AIRCRAFT ENGINE NOISE is the coordination of AIRCRAFT, ENGINE, NOISE. As such, in the uniterm vocabulary only the basic terms such as AIRCRAFT, ENGINE etc. can appear and this reduction on the pre coordination causes the number of descriptors in the vocabulary to be reduced [2].

An information retrieval system can be subdivided into a number of subsystems. For example:

1) The vocabulary subsystem
2) The indexing subsystem
3) The searching subsystem
4) The user-system interface subsystem

Among the above four subsystems of an information retrieval system, the vocabulary subsystem is considered to be one of the most frequent sources of failure of an information retrieval system. Failures in retrieval attributable to the vocabulary subsystem may be due to lack of specificity in the index terms or due to ambiguous relationships between index terms in pre coordinate indexing. Lack of specificity will always cause precision failures, but need not cause recall failures as long as the appropriate references are included in the vocabulary. If no specific term exists, and no references are made in the form of "see reference" or "use" etc., in relation to the specific terms, then both recall and precision failures are likely to occur in a search on the specific topic. Thus, specificity of vocabulary is by far the most important factor that affects the precision capability of a retrieval system [2,14].

Vocabulary growth can also be viewed in terms of precision capability of the retrieval system. In this context term coordination plays the important role [5].

Term coordination involves the logical intersection of classes. If $term_A$ is coordinated with $term_B$ then by relating the two together the effect will be to reduce the size of the class under consideration since the result is "$term_A$ in relation to $term_B$ rather than $term_A$ ALONE". This relating procedure is known as term coordination. It has the property of reducing the number of classes, achieving greater specificity and improving precision. Classes may be coordinated at the time of indexing (pre coordinate vocabulary) or at the time of searching (post coordinate vocabulary).

It appears that class coordination is a powerful precision device, but it tends to have the effect of reducing recall. Consider, for example, in a post coordinate system a search on the topic "STARS and NUCLEAR REACTION". Astrophysics and nuclear reaction may be the two potential classes, and the search strategy might use the question:

Nuclear reaction AND any term in the vocabulary for star.

Thus two conceptual classes have been coordinated. It may be observed that the precision is likely to be high, but recall could be low in view of the very general specification of the second index term. Similarly at a lower coordination level, which is a reduced exhaustivity of indexing, the above question could achieve a better recall but a poor precision. In general, index languages should provide some facility of coordination of classes either at the time of indexing, or at the time of searching. Many information retrieval systems are partly equipped with precoordinate vocabulary which can be further coordinated in searching operations.

Coordinate indexes are also referred to as manipulative indexes and belong to the post coordinate index group. It has been observed that with a manipulative index vocabulary there exists a relationship similar to Zipf's law between the index entries and the distribution of term usage.

In some studies [7,23] of the distribution of manipulative term usage it has been found possible to predict the probability of any index term with a given number of postings as a function of the total number of index entries in the system. This procedure is very helpful in calculation of the growth rate of the vocabulary and its size.

CHAPTER III

Simulation procedure

3.1 Simulation for analysis of data base

The general characteristics of a bibliographic data base, and the use of simulation as a tool to evaluate its performance, have been discussed in the previous chapters. In any study of the use of simulation techniques it is important to describe the separate steps in detail and to divide the task into sequential phases such as, for example:

1) Problem formulation
2) Construction of a theoretical model to represent the system under study
3) Derivation of the solution from the model
4) Test of the model

These four phases may vary in detail according to the type of information retrieval system being studied. However they represent the basic steps required for any simulation study [24].

An information retrieval system includes numerous variables and numerous subsystems. It is practically impossible to undertake a study that can monitor all the variables and all the subsystems simultaneously. The subsystem under study in this thesis is that which is required for the performance evaluation of a document title retrieval system. The rules necessary to define the simulation model have already been outlined in general terms. They will now be discussed in detail. The objects being simulated are represented in the model as:

1) Term file
2) Document file
3) Query formulation

In order to understand the real motivation of simulation as an evaluative tool, it is necessary to understand the objects being simulated. It is assumed that the three objects listed above are the necessary components refered to by any simulation study that attempts to analyze the performance of a document title data base system. The term file contains all the terms in the data base, the document file contains the document records of the data base, and the query formulation is for a real time search simulation. The three objects combined together determine the system performance as a whole.

3.2 Random number generator

The simulation model uses set rules to create a collection of pseudo terms, pseudo documents, pseudo queries, pseudo term-document associations, and pseudo relevance ratings for certain terms by generating pseudo random numbers. The totality of such pseudo quantities will constitute a pseudo data base for literature searching.

Since the simulation studies are undertaken on the Concordia University CDC-CYBER 172 computer it was appropriate to use the CDC library function RANF for generation of the pseudo random numbers. This, however, does not restrict the experimentation to only CDC machines, since by minor modifications of the pseudo random number generator routine the simulation programs could be executed on any machine. All the programs are written in the FORTRAN language.

The function RANF is a Fortran external routine. It accepts a dummy argument and returns a floating-point result uniformly distributed between 0. and 1. exclusive. The dummy argument has no effect on the result.

The method used by RANF to generate the pseudo random numbers is the multiplicative congruential method modulo 2^{48}. It makes use of the following recursive congruence:

$$X_{n+1} \equiv K * X_n (\mathrm{MOD} m)$$

where K is a constant multiplier and m is an integer chosen as indicated below. If X_o is the initial value of the sequence it follows that

$$X_1 \equiv K*X_o (MODm)$$

The multiplier should be chosen to give a long cycle of different pseudo random numbers. All odd integers are of the form 8n+1, 8n-1, 8n+3, 8n-3 (n is an integer), and studies in number theory have shown that choice of K in the form 8n+3 or 8n-3 yields the maximum cycle of 2^{b-2} where b is the word length of the computer [25]. The modulo m is an integer such that $m=2^b$. Such a choice implies that division by m is merely a shifting of position. Different values of the seed X_o can be utilized to produce different sequences of pseudo random numbers, thus giving a considerable flexibility to the method.

The function RANF on the CDC machine is represented as

$$x(n+1) = a*x(n) \pmod{2^{48}}$$

The reason for raising 2 to the power 48 instead of to the word length of 60 bits is due to the fact that the numbers generated are real (in FORTRAN sense) in the exclusive range of 0. and 1., and in the CDC realisation of FORTRAN such numbers are stored in the lower order 48 bits of the computer word. The multiplier a is a constant which has the octal value 20001207264271730565b. It can be shown that the multiplier passes the Coveyou-Macpherson test as well as other statistical test of randomness including the auto correlation test.

Basically the random number generator uses the initial value of the seed X_o equal to the octal value 17171274321477413155b. However, if a different seed is required it is possible to reset the seed by means of another library function called RANSET. This is also a Fortran external function. It accepts a floating point argument and returns the new address of the suggested seed. This function is used extensively in the simulation experiments as the model is so designed that different sets of pseudo random numbers are needed to simulate the three objects in the simulation model. For example, the following program generates a sequence of twenty random numbers.

```
      PROGRAM RAND (INPUT,OUTPUT)
I is the term number
X, PHI are the arbitrary constants
used for initialising different seeds for different
sequences of pseudo random numbers
      I=50
      X=.00001
      PHI=(SQRT(5.)+1)/2
      XX=(I+X)*PHI
set the seed of random number generator
      CALL RANSET (XX)
      DO 10 J = 1, 20
```

Fortran function RANF generates random numbers in the range 0. and 1.

```
        Y=RANF(DUM)
   10   KK=Y*5000+1
        STOP
        END
```

Following is the list of random numbers, and their Integer conversion respectively:

Random Number	Integer Equivalent
.4158828441667	2080
.9404661544677	4703
.2915424651987	1458
.7823937084672	3912
.3350111033781	1676
.279777314636	1399
.483501697939	2418
.8993786876637	4497
.6569218702714	3285
.006483786570588	33
.5478338038623	2740
.733820955941	3670
.4339856255594	2170
.4628215151311	2315
.2760265341333	1381
.3501366874212	1751
.4499985630042	2250
.6938284217629	3470
.6163425940888	3082
.790813718125	3955

3.3 Description of the simulation model

To allow comprehensive experimentation on the simulation objects, the model is divided into the following components.

1) User query

2) Document generator

3) Search routines

4) Evaluation routines

In the previous chapter the applicability, and the statistical importance, of Zipf's law for word frequencies was discussed. As one of the operating rules it is assumed that the Zipf law holds for document title data bases in real systems. In use of the Zipf law the model considers that the terms are ranked according to decreasing frequency of their occurrence. Let the data base contain a total of N terms (including repetitions) and M document titles, so that the average number of terms in each document title is N/M. If there are D different terms then the first term will have the highest frequency and the lowest rank; similarly the D-th term will have the lowest frequency and highest rank. In general, the number of occurrences of the i-th term is M_I=A*N/i. The constant A is equal to $1/(\log_e D+\Psi)$ where Ψ is Euler's constant equal to 0.5772 ...

Let c denote the probability that a document is relevant to the interest of a particular user. The expected number of relevant documents in the data base is therefore cM.

As discussed earlier, the prime objective of a document retrieval system is to retrieve relevant, and only relevant, documents in response to a user query. This goal is defined in the simulation model in terms of relevance, rather than recall and precision. It is by no means true that the evaluation routines do not consider recall and precision as performance measures, however the approach is based on choosing as a basic parameter the value of c which describes the probability of the relevancy of a document. In addition to the value of c, term relevancy is another important measure that will be discussed later in this chapter.

	Retrieved	Not Retrieved	Total
Relevant	X	Y	X+Y
Not-Relevant	Z	M-X-Y-Z	M-X-Y
Total	X+Z	M-X-Z	M

Table 3.1 2x2 contingency table of relevance

From table 3.1, which is a 2x2 contingency table for relevance and retrieval, it may be noted that the recall-precision measure depends on a prior knowledge of what is relevant in the total collection of documents. Therefore the value of Y in the table can be predicted only through user experiments conducted over a sufficiently long period of time. Such a predicted value may be supposed to be close to the true value. It may appear to be a very subjective type of assessment since a document may be relevant to one user but not relevant to another user. In real systems the relevance judgements are often made by sets of users, and in case of disagreement the majority opinion is taken. Alternatively, experts in the subject field may decide on the doubtful cases. Taking these facts into account it may be possible to estimate that for a particular document data base and type of user it is possible to simulate the value of X+Y in terms of the value of the probability c. It may be noted that the value of c is predicted on the basis of sufficient experience of experimental results [26]. Alternatively, c may be varied throughout a wide range of values in order to simulate a widely varying sample of users.

With a chosen value of c it is possible to calculate the approximate number of relevant documents in the particular data base described in the model. Since the value of c is not chosen at random, but is supposed to be a close approximation to the true value, it is possible to simulate the total number of relevant documents in the data base under study. However, if there is any error in the choice of the value of c it can be changed without effecting any other part of the simulation model.

Let r (1), r (2), ---, r (m) be a sequence of pseudo-random numbers uniformly distributed in the range $0 < r\ (m) < 1$.

The m-th document will be regarded as of interest to the user if r(m) < =c, and non-relevant to the interest of the user if r(m) > c. An example of the simulation of the first 29 relevant document numbers is given below. Each document number, as m increases over 1,2,3, etc., is rejected unless r(m) < = c. In the listed results the value of c was chosen as 0.01.

Following is the program unit to generate m and r(m)
MTIT is the total number of documents in the data base
X, C, PHI are used to initialize the seed
Constant C is also used to determine if a document is relevant or non-relevant to the user interest

```
                LL1=0
                DO 11 II=1,MTIT
        set the value of the seed XX
               XX=(II+X)*C*PHI
               CALL RANSET(XX)
               Y=RANF(DUM)
               IF(Y .GT. C) GOTO 11
               LL1=LL1+1
               RELDOC(LL1) = II
   11          CONTINUE
```

m	r(m)
132	.0004159548091103
162	.005271679218122
235	.004826607042457
520	.006272738395548
606	.00362234010321
692	.0009719418108709
726	.008478064512222
898	.003177267927544
984	.0005268696352054
1001	.00213996549294
1069	.009646088194291
1148	.004253362599105
1241	.006995689901952
1320	.001602964306766
1388	.009109087008117
1413	.004345291609614
1560	.006458688715778
1585	.001694893317275
1653	.009201016018626
1732	.003808290423439
1825	.006550617726287
1904	.001157892131101
1972	.008664014832451
2074	.009961599442239
2099	.007579701742987
2257	.002186976147801
2393	.009693098849151
2418	.0073112011499
2737	.007042700556813

In order that a document qualify for retrieval, whether relevant or not, it must contain specific index terms as formulated in the user query. The restriction $r(m) <= c$ for a document to be relevant allows specification of the extent to which the documents in the whole data base are relevant to a particular subject area or user interest. The relative positions of the relevant documents in the data base have no association with their relevancies; this is a consequence of the simulation procedure and is also observed in practice [27-30].

It is likewise possible to simulate a set of documents relevant to the interests of a different user by use of a different sequence of pseudo random numbers r(m). Such different sequences may be associated with either the same or different value of c.

The next step in the simulation procedure is the generation of term-document associations. However, before proceeding to this step it is appropriate to describe a measure of the relevance of certain terms and how relevance is simulated in the model.

Suppose that some terms tend to associate more with relevant documents than with non-relevant documents. A measure of association in the model is expressed in terms of two relative frequencies. The first relative frequency f is defined as

$$f_i = \frac{\text{number of relevant documents that contain i-the term}}{\text{number of relevant documents}}$$

The second relative frequency is

$$M_i/M = \frac{\text{number of documents that contain i-the term}}{\text{number of documents in the data base}}$$

The ratio of the two relative frequencies is thus:

$$R_i = f_i/M_i/M = f_iM/M_i$$

Therefore: $$f_i = R_iM_i/M$$

The value of Ri is termed the relevance rating of the i-th term. It may be noted that the relevance rating R of certain terms may be known prior to the simulation experiments.

This knowledge may have been gained by experience with term usage and the observation that occurrence of certain terms is more likely in the relevant documents than in the non-relevant documents of the data base. It may have been observed, for example, that certain terms used in document indexing are frequently found to be in the relevant documents belonging to a particular subject area [31-33]. Thus it is possible to manually specify, or at least estimate, the relevance rating for certain terms. On the other hand, there are likely to be other terms of unknown relevance rating, and it will be described how the simulation model uses built-in rules to determine the relative importance, or non-importance, of other index terms.

From the above definition of Ri it may be concluded that if no relevant document contains the i-th term then the ratio R is equal to O. Alternatively if the same proportion of relevant and non-relevant documents contain the i-th term then Ri is equal to 1. Finally, if all the relevant documents contain the i-th term then Ri is equal to M/M_i. Therefore Ri will have a value in the range determined by the inequality

$$0 <= R_i <= M/M_i$$

A value of R greater than 1 indicates the extent to which the relevant documents are more likely to contain the i-th term than are non-relevant documents. Similarly, a value of R less than 1 indicates the extent to which the relevant documents are less likely to contain the i-th term.

The total number of documents that contain the i-th term is the value of Mi that is given by the Zipf law. The total number of relevant documents that contain the i-th term is $f_i cM = cRiM_i$, and therefore the number of non-relevant documents that contain the i-th term is $(1-cRi)M_i$. The proportions of relevant documents and non-relevant documents that contain the i-th term are therefore

$$cRiM_i/cM \text{ and } (1-cRi)M_i/(1-c)M \text{ respectively.}$$

The criteria used to choose the Ri may now be defined formally. It may be noted that the value of Ri must be chosen to satisfy both of the inequalities.

$cRiM_i \leq$ number of relevant documents

and

$cRiM_i \leq M$

It may also be noted that the expected number of relevant documents is cM, but the actual number may not be exactly cM. In fact, the probability of there being exactly q relevant documents is

$$\frac{M!}{q!(M-q)!} \quad c^q(1-c)^{M-q}$$

which is a maximum when q=cM.

Although the number of relevant documents is approximately equal to cM, the restriction on Ri should be applied in the form

$$R_i <= \frac{\text{number of relevant documents}}{cM}$$

It follows that $Ri \leq M/M_i$ which may be regarded as a useful approximation when deciding on manual choice of some of the Ris.

Similarly, the Ri must be choosen so that $(1-cRi)M_i \leq$ number of non-relevant documents. Since the number of non-relevant documents is likely to be very close to M, the restriction may be written as

$$(1-cRi)M_i <= M$$

and therefore

$$Ri >= \frac{1}{c}(1-M/M_i) \ \&(M/M_i \geq 1 \text{ is always true.})$$

Values of Ri that have been observed in practice are discussed in section 3.4.

The above discussion has explained the criteria for determination of whether a document is relevant, the document numbers of all relevant documents in the data base, and the relevance ratings of the i-th term. It is now appropriate to describe the rules for simulation of the term-document associations and for simulation of the relevant and non-relevant documents that contain the i-th term.

For a given value of i and Ri, where i is in the range $1 \le i \le D$, let

r (1), r (2), ---

denote a sequence of different pseudo random numbers with values in the range $0 <= r(j) <= M$. It may be noted that real pseudo random numbers are converted into integers by multiplication followed by use of the modulo function.

For the first cR_iM_i values of j for which

$$r_c[r_i(j)] <= c$$

the i-th term will be regarded as present in the r(j)-th document, which is already known to be a relevant document. Also, for the first $(1-cR_i)M_i$ values of j for which

$$r_c[r_i(j)] > c$$

the i-th term will be regarded as present in the r(j)-th document which is already known to be a non-relevant document. This procedure is implemented by generating a real pseudo random number Y, then by applying the modulo function to convert it into an integer, and using this integer as the seed to generate another pseudo random number which is finally checked for the relevancy of the document. In the experiments the sequence was generated by the following statements in which the seed X and the variable PHI are as explained in section 3.2.

Generate document number KK in the range 1 to M

M1 is the number of relevant documents for the i-th term

M2 is the number of non-relevant documents for the i-th term

```
     XX=(I+X)*PHI
     RICMI=c*Ri*Mi
     MIMRICM=(1-c*Ri)*Mi
     MTITLE=Mi
     M1=M2=INDEX=0
     CALL RANSET (XX)
10   Y=RANF(DUM)
     KK=Y*5000+1
```

The function RANGET below stores the value of the seed in the variable Y

This function is used to initialize a different value of the seed by using the document number KK.

Subsequently the original value of the seed is reinitialized by using Y.

```
     CALL RANGET(Y)
     XX=(KK+X)*C*PHI
     CALL RANSET(XX)
     RCKK=RANF(DUM)
```

Test document for relevance

```
     IF(RCKK.LE.C)GOTO 50
     M2=M2+1
     IF(K2.LE.MIMRICM)GOTO 50
     GOTO 65
50   M1=M1+1
     IF(M1.GT.RICMI)GOTO 65
60   INDEX=INDEX+1
```

```
      RNUM(INDEX)=KK
65    IF(INDEX.EQ.MTITLE) EXIT
      CALL RANSET(Y)
      GOTO 10
```

Following is a list of 30 relevant and non-relevant documents simulated for a given i-th term. Documents containing the i-th term are classified as relevant if the random number RCKK <=c, otherwise the documents are classified as non-relevant since RCKK>c.

Document Number	RCKK
2080	.3988155477646
4703	.52155875392
1458	.1532444928818
3912	.5215639092786
1676	.8678079374478
1399	.3123002151477
2418	.0073112011499
4497	.1527493469919
3285	.8576064980985
33	.2427412641067
2740	.701469674718
3670	.766911381924
2170	.5477147490368
2315	.6385017942218
1381	.9791765252132
1751	.2213365983801
2250	.4872206952504
3470	.7342364928263
3082	.4465946112756
3955	.6857438565466
4630	.95733390948
496	.1524296123649
1326	.7793108609516
4964	.04522210636846
901	.4647951881363
2477	.7697383517989
1929	.6803041202965
4656	.8711541386999
3292	.688662755432
2826	.02982956925395

The above procedure ensures that exactly cRiMi relevant documents contain the i-the term and exactly (1-cRi)Mi non-relevant documents contain the i-th term. It may be observed that if all R_is=1 then the set of relevant documents has the same relative term frequencies as the entire data base. This means that the number of relevant documents that contain the i-th term is then exactly cM_i.

Specification of the complete set of the R_i may describe the data base in more detail than is needed or is possible in many instances, since it requires a knowledge of the relevance rating of every term. It would, in fact, be a very artificial situation in which so much is known about the contents of the data base. A realistic approach that has been considered in this study requires an estimation of only some of the relevance ratings.

For the terms for which R_i is specified the procedure is followed as described above. However, for many other terms in the data base the relevance rating Ri is unknown. Such terms may be divided into two categories that contain "content terms" and "non-content terms" defined as follows. Content terms are those which, when used in the query formulation, could help to distinguish the relevant documents from the other documents in the data base. In contrast, terms whose occurrence in documents is unrelated to the relevance of the documents are regarded as non-content terms. For example, terms for which R is specified equal to 1 may be regarded as non-content terms since their occurrence in documents is in no sense related to the relevance of the documents. They should therefore not be used in questions intended to retrieve relevant documents. Many terms, such as THE, AND, WHICH, ON, etc., are likely to be non-content terms for all sets of relevant documents, but many other terms will act as content terms for some sets of relevant documents but not for other sets.

Therefore, prior to the formulation of a question for retrieval of relevant documents it may be known that certain terms are likely to be content terms but their relevance rating R is unknown except in statistical terms. For example it might have been found from experience that a certain proportion of content terms give rise to values of R as great as 100. Also, it might be believed that a particular term is a content term but of unknown relevance rating believed to be in the range of 10 to 100.

Let the extent to which the data base is rich in content terms be described by the value of the probability Pc of the occurrence of content terms. Similarly, let the extent to which the data base is rich in non-content terms be described by the probability Pn of the occurrence of non-content terms. The terms that have accidental associations with relevant documents therefore occur with the probability 1-Pc-Pn. It is believed that the inclusion of terms with accidental associations is important in the simulation procedure since it allows for variations in different samples of a data base, and it allows for the fact that a data base sample may contain some titles whose word usage is not typical of the data base as a whole.

A factor delta (Δ) may be defined as the amount in excess of 1 that may be assumed by any Ri for a content term. It should be noted that delta (Δ) could, in fact be a large number greatly in excess of 1. If delta (Δ) is large the content term is more significant than another content term that has a small value of delta (Δ).

The model simulates the value of R by generating pseudo-random numbers of the following sequences

$r_R(1)$, $r_R(2)$, $r_R(3)$, ---, $r_R(D)$

and $r_\Delta(1)$, $r_\Delta(2)$, $r_\Delta(3)$, ---, $r_\Delta(D)$

Uniformly distributed in the range $0 < r_R(i)$ or $r_\Delta(i) < 1$.

In the experiments the seeds choosen for these sequences were respectively:

```
XX=(I+Pc)*C*PHISQ
XX=(I+DELTA)*C*PHISQ
```

where PHISQ is the constant used to initialize the seed and is computed as

```
PHISQ=PHI**2
```

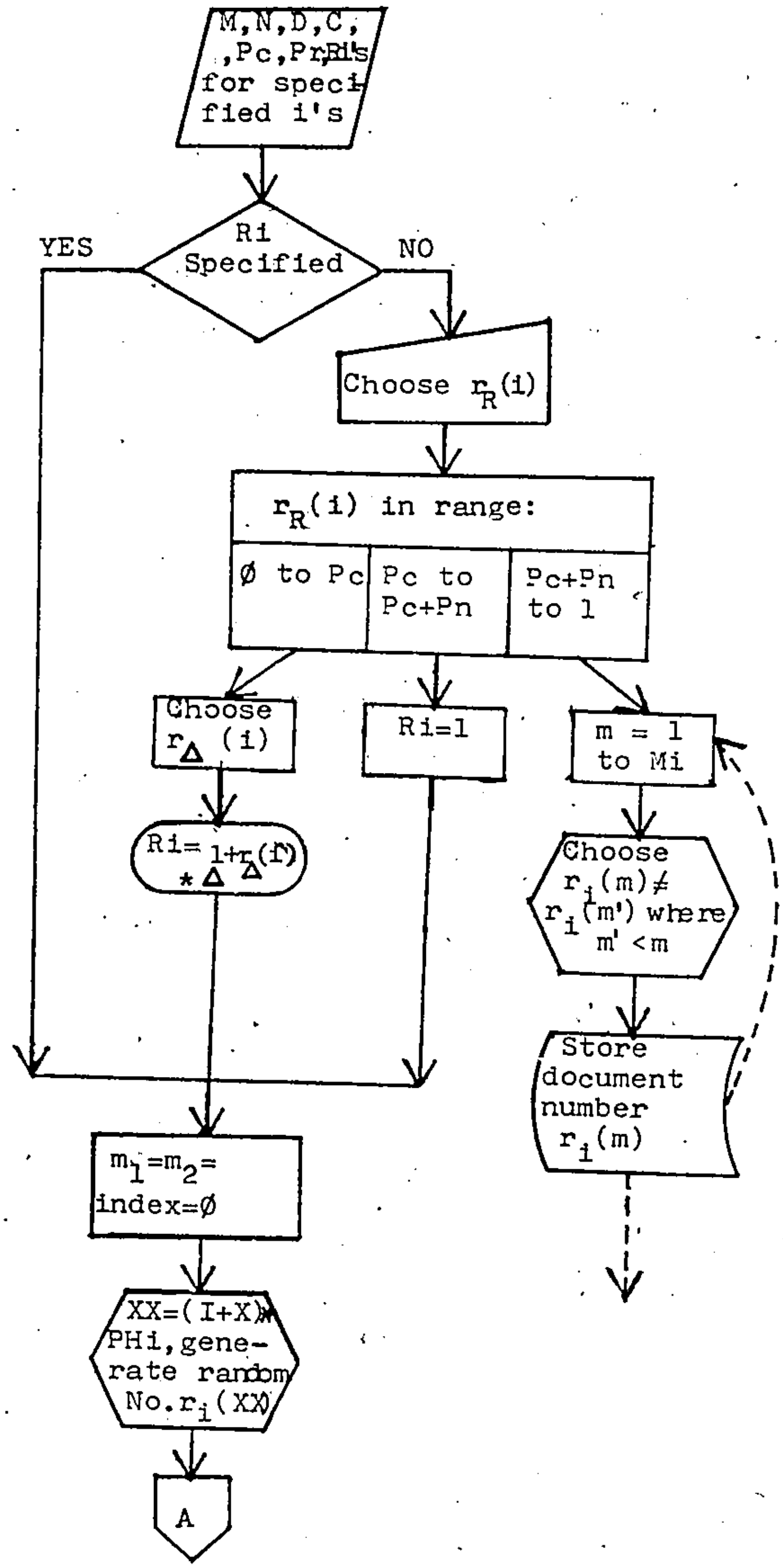
and where PHI is as defined previously

For each value of i, the i-th term is associated with the documents in the following manner:

1) If $0 < r_R(i) <= Pc$ then $Ri = 1+r_\Delta(i)*\Delta$ (unless $1+r_\Delta(i)*\Delta > M/M_i$ in which case Ri=M/Mi), alternatively if $Pc < r_R(i) <= Pc+Pn$ then Ri=1. In either instance the association of the i-th term with documents is determined as explained above for the instance of terms with specified Ri's.

2) If $Pc+Pn < r_R(i) \leq 1$ the i-th term is regarded as present in the documents numbered: r(1), r(2), r(3), ---, r(Mi) of which the expected number of relevant documents is cMi. The assignment of relevance or non-relevance to a document may lead to accidental associations between terms and relevant documents.

The simulation model described above can be represented by Fig. 3.2.


M,N,D,C,
,Pc,Pr,Ri's
for speci-
fied i's
Ri
Specified
YES
NO
Choose $r_R(i)$
$r_R(i)$ in range:
Ø to Pc
Pc to
Pc+Pn
Pc+Pn
to 1
Choose
r_Δ (i)
Ri=1
m = 1
to Mi
Ri= 1+r_Δ(i)
* Δ
Choose
$r_i(m) \neq$
$r_i(m')$ where
m' < m
Store
document
number
$r_i(m)$
$m_1=m_2=$
index=Ø
XX=(I+X)*
PHi, gene-
rate random
No. r_i(XX)
A

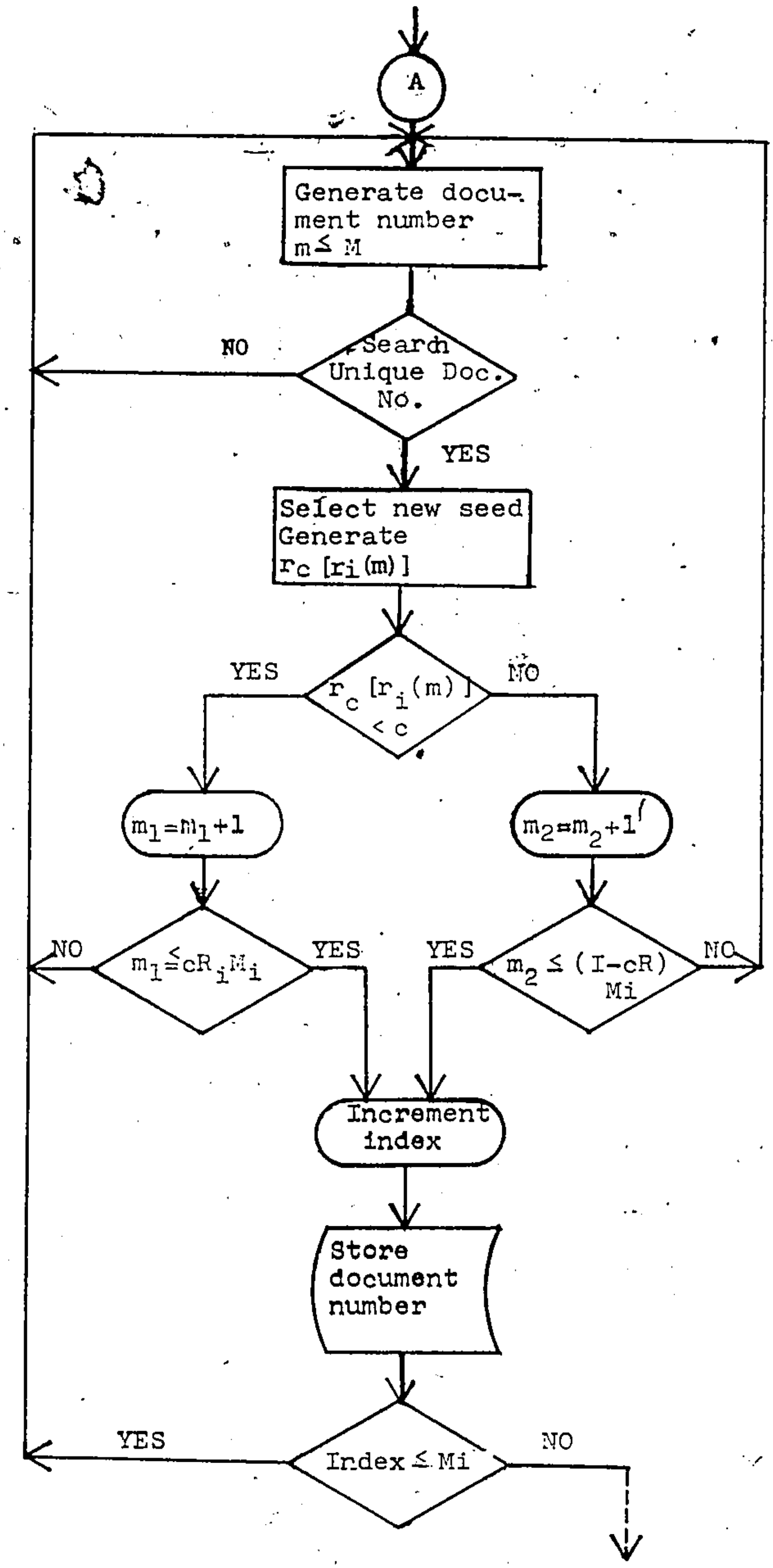


Fig. 3.2 Simulation of inverted file list of document numbers for the i-th term.

It may be observed that the flowchart illustrates the following important points:

1) Generation of documents by means of pseudo random numbers that specify the terms used to index each document.
2) Generation of relevance ratings for those terms that appear in the query but whose relevance ratings are not specified manually
3) Determination of the total number of relevant and non-relevant documents indexed by the i-th term
4) Determination of the relevancy of a document with respect to the i-th term

In addition to the program represented in the flow chart of Fig. 3.2 above there is a query handling process, sorting process, evaluation process, and query reformulation process. These procedures are implemented through programming the individual processes as described separately in Section 3.5.

3.4 Relative frequencies (r) and relevance ratings (Ri)

In a general document retrieval environment it is possible to determine the frequencies of different terms in the total collection. The determination of term frequency is sometimes used in order to derive a measure of term importance for indexing as well as for searching purposes. In fact, in many studies it has been observed that use of information about the known frequencies of terms in different subject categories allows prediction of the relevance of documents to subject areas by examination of word frequencies in the document text. For example, in a study by Heaps [34] a relative frequency r of each term is defined as the frequency of occurrence of the term in a particular document text divided by its frequency of occurrence in general text.

The following is a set of document abstracts taken from [34].

Doc. 1.:

H.E. Stiles, The Association Factor in Information Retrieval. Journal of ACM Vol. 8, 271-279(1961)

For this document N=3188 and D=777. The abstract is as follows.

Title: The Association Factor in Information Retrieval.

Abstract: This paper describes an all computer document retrieval system which can find documents related to a request even though they may not be indexed by the exact terms of the request, and can present these documents in the order of their relevance to the request. The key to this ability lies in the application of a statistical formula by which the computer calculates the degree of association between pairs of index terms. With proper manipulation of these associations (entirely within the machine) a vocabulary of synonyms, near synonyms and other words closely related to any given term or group of terms is derived. Such a vocabulary related to a group of request terms is believed to be a much more powerful tool for selecting documents from a collection than has been available heretofore. By noting the number of matching terms between this extended list of request terms and the terms used to index a document, and with due regard for their degree of association, documents are selected by the computer and arranged in the order of their relevance to the request.

Doc.2.:

C.D. Lowenstein, and V.C. Anderson, Quick Characterisation of the Directional Response of Point Array. Journal of the Acoustical Society of America, Vol.43, 32-46 (1958)

For this document N = 1404 and D = 383. The abstract is as follows:

The directional response of a two- or three-dimensional point array is a function of two independent directions and also a function of frequency A suitable mapping of these three parameters into a pseudodirection and a pseudofrequency allows the examination of the major and minor lobe structure of the array response with only a two-parameter computation. This method has been applied during the design of a 32-element planar array, to permit adjustment of the element positions for a minimum and uniform minor-lobe structure.

Doc.3.: W.J. Holtslander, and G.R. Freeman, Competition Between Scavengers in the Vapor-Phase Radiolysis of Hydrocarbons. Journal of Physical Chemistry, Vol. 71, 2582-2584 (1967).

For this document N = 1275 and D = 368.

The abstract is as follows:

When the electron scavengers S_1 and S_2 are present in a radiolysis system, the reaction $S_1^- + S_2$ $S_1 + S_2^-$ can occur if S_2 has a greater electron affinity then does S_1 and if S_1^- has a long enough lifetime to enable it to encounter an S_2 and react with it. In 380 torr of methylcyclohexane vapor at 110°, the half-lives of N_2O^- and SP_6^- with respect to decomposition, are 10^{-4} sec and 10^{-7} sec respectively. The electron affinities of the three electron scavengers used apparently decrease in order DI $> SF_6 > N_2O$.

Doc.4.: G.A. Needham, Advanced Integrated Circuit Packaging. SCP and Solid State Technology, 22 to 29 (June 1965).

For this document N = 1515 and D = 500.

The abstract is as follows:

Current research and development in the field of integrated circuit packaging is treated in this article. It is pointed out that the package as we know it today may be completely obsolete in the next three or four years if present research bears fruit. The discussion includes a survey of past, present, and possible future methods of lead attachment; the "flip-chip," or up-side-down mounting technique; elimination of the individual chip package which has the potential of a twenty to one saving in space; and the inclusion of many circuit functions within a single chip. Finally three dimensional packaging as opposed to the planar package is discussed.

Doc.5.: The Chip
This is the title of an anonymous article published in a popular magazine and related in subject matter to Doc. 4. N = 1018

Doc.6.: G. Orwell, Why I Write.
This is contained in "A Collection of Essays by George Orwell", Doubleday, 1954, p.313-to 320. N = 2758 and D = 947.

Dec.7.: L. Zanelly, The Land of King Arthur. Yachts and Yachting, 292 to 233 (February 2, 1968).

This article is about diving in the Welsh Lakes. N = 1324 and D = 567. There is no abstract. It may be noted that the title of the article is a poorer indication of the subject matter than are the underlined words in Table 3.2.

Table 3.2 : Relative frequencies r_i of terms in text of certain documents

STILES:	r_i
INDEXED	5300
PROFILES	1800
DOCUMENT	770
DOCUMENTS	680
REQUEST	300
TERMS	190
TERM	140
GENERATION	120
ASSOCIATION	59
INDEX	47
THIN	41
LIST	38
RELATED	37
INFORMATION	13
NUMBER	13
SECOND	10

LOWENSTEIN:	r_i
DB	5000
LOBE	4300
BETA	3900
LATTICE	2800
ARRAY	2300
PI	2100
ALPHA	1400
STEERED	1400
TRIANGULAR	1100
DISTORTION	810
PARAMETERS	710
ARRIVAL	280
FIG	220
DIRECITONS	210
ELEMENT	150
DIRECTION	82
FUNCTION	57
LEVEL	30
SIDE	22

HOLTSLANDER:	r_i
MCH	15000
SF	14000
SCAVENGERS	7800
HD	7100
SCAVENGER	6300
BINARY	5400
MIXTURES	1800
MOLE	1600
YIELDS	1000
DI	860
N	560
MIXTURE	310
O	290
ELECTRON	240
H	150
YIELD	150
G	143
REACTION	81
FORMED	71
D	60
C	55
S	40
ADDITION	38
RESULTS	36
ADDED	31
EFFECT	30
PRESENT	17

NEEDHAM:	r_i
CIRCUITS	3000
INTEGRATED	1100
CIRCUIT	870
PACKAGE	600
CHIP	430
DIMENSIONAL	420
LEADS	120
INDIVIDUAL	22
FIGURE	19
COST	17
FORM	12

The Chip:	r_i
CHIPS	2300
CIRCUITS	1200
CIRCUIT	300
ELECTRONIC	74
SIZE	51
RADIO	50
ALREADY	22

ORWELL:	r_i
WRITE	41
WRITING	28
BOOK	26
AGE	13

ZANELLY	r_i
LLYN	5300
LLYDAW	5300
GLASLYN	3800
LAKES	670
DEPTH	100
LAKE	98
BOAT	53
S	44
BOTTOM	43
AREA	16
FEET	16
AGO	15
WATER	12

Table 3.2, taken from [34], lists sets of terms that appear in the above document abstracts together with their relative frequencies. The values of r suggest that for searching and query formulation purposes the relative frequencies r may prove to be important quantities for subject identification, and hence for retrieval of relevant documents.

With regard to the simulation model and the relevance ratings described in the previous section, it may be noted that the relevance rating Ri of a term measures the extent to which the term is more likely to appear in relevant document than in non-relevant document. This is analogous to the meaning of the relative frequencies r described above. Examination of the above abstracts suggests that many terms useful for determination of subject matter or for retrieval purposes may have Ri values of at least 100.

3.5 The simulation experiment

The present section deals with the programming aspect of the simulation model and the interrogation of the resulting output.

After the simulation model has been chosen the next step is to decompose the entire retrieval process into a set of smaller independent subproblems. The decomposition process may be performed in reference to the following modules.

i) Control program

ii) Query processing and syntax checking

iii) Document generation and storage

iv) Search and document retrieval

v) System evaluation

The first step in the experimentation is to input to the system the following data base parameters:

1) The seeds which may be varied to simulate different data bases with the same simulation parameters.

2) M, N, D to describe the size of the data base

3) c=probability that a document is relevant

4) Some specified values of Ri's

5) Delta, where $1+\Delta$ is the maximum value allowed for any unspecified value of Ri. This, in fact, implies that $1 < Ri < 1+\Delta$.

6) Pc, Pn which are the probabilities of occurrence of a content and non-content term respectively.

The above parameters are input to the control program and are manipulated by the control program only. It is supposed that some user interest profile relating to a particular subject is known, and that certain index terms pertinent to the subject matter have some specified relevance rating. Obviously in the query formulation such index terms are likely to be preferred over index terms whose relevance ratings are unknown.

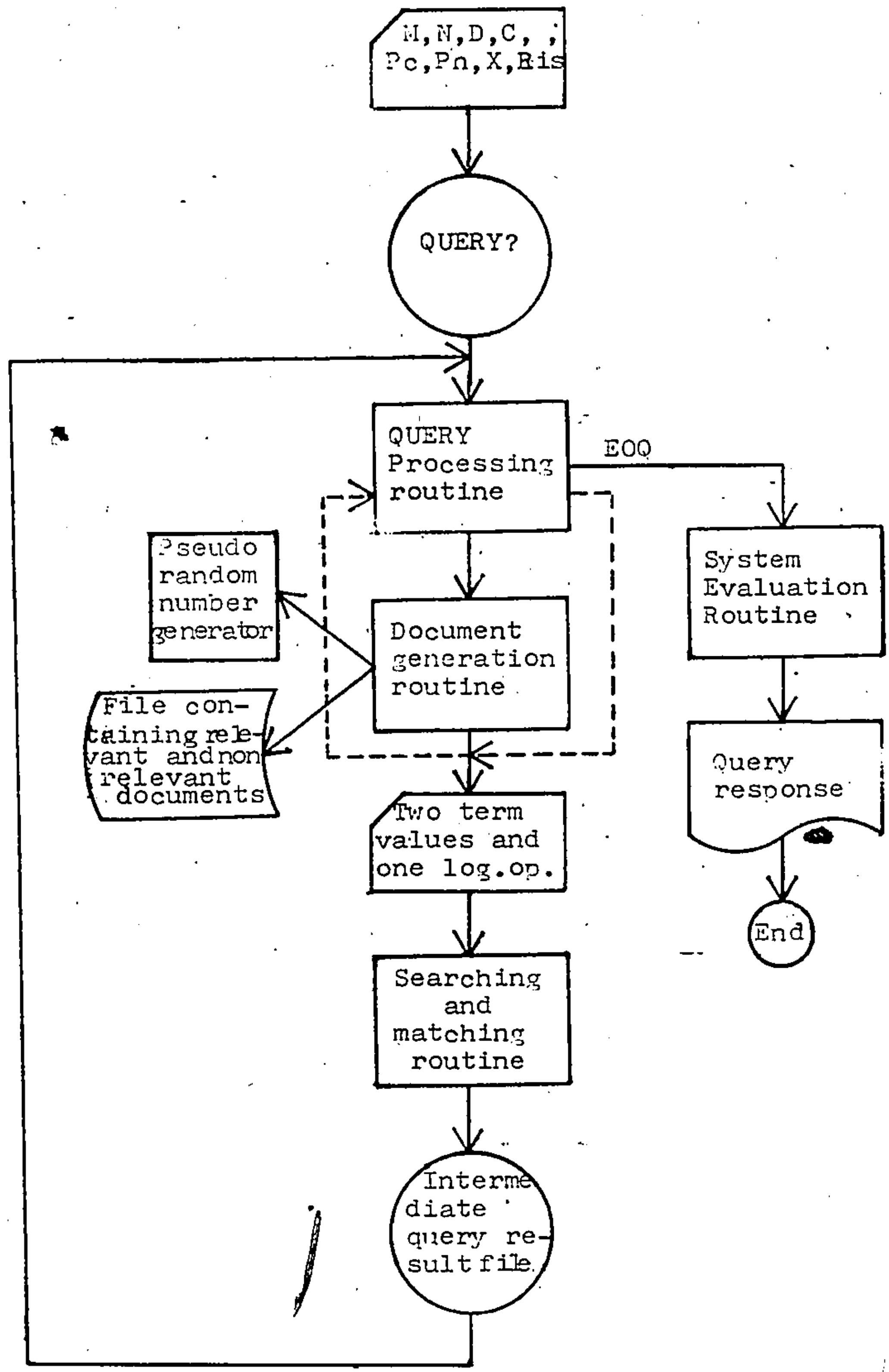
To start the experimentation process a query is formulated initially in the form of index terms and some associated logic operators. In the simulation each index term is specified by means of its rank rather than by a string of alphabetic characters. The query response may be examined with respect to the number of relevant documents retrieved. Also, by examining the terms present in retrieved documents, there may be discovered some content terms which were not present in the initial query. This subsequent determination of other content terms in the output documents may lead to a question reformulation and a corresponding improvement in retrieval effectiveness.

Following input of the query the query processing routine is invoked. The parser contained in the routine checks the syntactic correctness of the query according to the described BNF form of the query language. End of query is indicated by a $ sign. Since the query is read character by character in alphanumeric format, numerical symbols representing term numbers are decoded into their numerical values. It may be mentioned here that there is no need to generate the whole data base, but only those terms contained in the query generated documents. This is an important feature of the simulation technique since it allows economy of storage [35].

Whenever there is a numerical symbol in the query, it is decoded and stored in the term stack. Next, the document generation routine is invoked to determine the numbers of documents that contain the particular term. As the module is invoked it calculates Mi, which is the number of documents that contain the i-th term according to Zipf's law. For each term with a specified relevance rating a flag is set to ON in the control program. The flag is checked for the given i-th term to determine whether the relevance rating is specified or is to be simulated according to the given rules. If no relevance rating is specified for the i-th term it is calculated subsequently as described earlier, and its value is used to determine the number, cRiMi, of documents relevant, and the number, (1-cRi)Mi, of documents non-relevant, to the i-th term.

The term number I is used to initialize the seed XX for the pseudo random number generator that generates a sequence of real numbers which are converted into integers by using the modulo function. The resulting integers are selected as the document numbers associated with the i-th term. The document number is, in turn, used as a seed for the next sequence of pseudo random numbers which determine whether the document is relevant or non-relevant. The iterative procedure is repeated for the first cRiMi relevant documents and the first (1-cRi)Mi non-relevant documents. The document numbers are then stored in a random access file by using system mass storage routines.

The algorithm is designed so that as soon as the document numbers related to the i-th term are calculated, and cRiMi relevant documents and (1-cRi)Mi non-relevant documents are stored in the random access primary file, the control is passed over to the matching algorithm provided at least two values are in the term stack and at least one operator is in the operator stack. It may be observed that it is possible to have just one term and one intermediate result, or even two intermediate results, in the term stack. This means that in order to invoke the matching algorithm the input to the module should be two operands and one operator. In the matching algorithm the analysis is based on the logical operator (+,-,*) in process. The partial results are stored in an intermediate file and the control is transfered to the query processing and syntax check module. The query pointer is reset during the processing, which follows the same procedure described above, until a $ sign is encountered. When the query processing is complete the stored documents are input to the system evaluation routine for calculation of the performance measures. The block diagram of Fig. 3.3 represents the above processing.



EOQ: End of Query

Fig. 3.3, Block diagram of Query Processing

The resulting output consists of the total number of documents relevant to the user's interest as simulated by use of the probability c, the total number of retrieved documents, and the first fifty retrieved document numbers (in the case that more than fifty documents are retrieved) each with the set of terms that the document contains. Terms of ranks 1 to 49 are not included in the list of the first fifty or fewer, retrieved documents. Performance measures, consisting of the recall and precision values, are also output.

Initial system evaluation results may be used to upgrade the evaluation measure. The procedure of upgrading involves manual analysis of the term-document output. Terms which seem to appear more frequently in the documents, but have not been used in the query, may then be used in the updated version of the query. Alternatively, if the question was very general in nature, and the precision value was low, the query could be restated in a more constrained form. With such modifications to the question it is possible to upgrade the performance measure of the system in the same manner as with a real, in contrast to simulated, system. Particular examples are discussed in the next section.

3.6 Analysis of search output

The simulation procedure and experimentation process described previously have been tested on the CDC cyber 172 under NOS 1.2. The programming language used was FORTRAN IV. In the paragraphs that follow, some sample search request are displayed in order to illustrate the performance of the model. The data base for experimentation contained N=40,000 terms, M=5000 documents and D=7000 different terms. The probability for content terms in the data base was assumed to be Pc=.002, and for non-content terms Pn=.30. The value of delta =100., the value of user interest probability c=.01, and the assumed value of Ris=50 for i=50, 100, 150.

The simulated system was thus for a user whose interest could be satisfied by 50 documents of the collection and who knows that 3 particular terms are of indexing value (with Ri=50) for determination of relevant documents. It is supposed, that the data base contains 10 further unspecified terms of indexing value unspecified (but of Ri in the range 1 to 101). In the example 1 the initial search request and the resulting output was as follows:

```
N=40000 M=5000 D=7000

FOR I=50,100,150  RI= 50

PC= .002 RN= .300 DELTA=100.0 X=.00001 C=.010

        PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
        1. USE SIMPLE INTEGERS FOR TERM VALUES
        2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
        3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
        4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
        5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
        6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                              USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? 50-100-150$   example 1

THE FOLLOWING ARE THE RELEVANT DOC'S:
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
   33  123  132  162  235  263  370  421  496  520
  527  605  606  638  692  726  755  898  901  984
 1001 1069 1148 1241 1308 1320 1326 1381 1388 1399
 1458 1509 1527 1560 1584 1585 1597 1653 1676 1703
 1732 1751 1777 1825 1904 1910 1917 1929 1963 1972
 2003 2006 2062 2063 2074 2080 2099 2170 2250 2257
 2315 2373 2392 2393 2418 2477 2534 2603 2737 2740
 2748 2762 2787 2826 2847 2923 2966 3055 3059 3081
 3082 3096 3106 3285 3292 3348 3425 3438 3470 3514
 3522 3561 3586 3609 3653 3670 3741 3824 3912 3930
 3955 4198 4199 4202 4292 4317 4342 4367 4392 4435
 4497 4562 4584 4630 4639 4656 4703 4709 4906 4955
 4964 4980
 THE PRECISION PR= .37
 THE RECALL RC= .94
```

The above query is very general in that it requests a search for all documents that contain $term_{50}$ or $term_{100}$ or $term_{150}$. It is supposed that $R_{50}=R_{100}=R_{150}=50$.

In all there are 122 documents retrieved out of which 63 percent are non-relevant. On the other hand, only 6 percent of the relevant documents in the data base are not retrieved by the given query. It may be observed that 37 percent of the retrieved documents are relevant, and the retrieved documents include 94 percent of all documents relevant to the user's interest. Since, the query is formulated very broadly it leads to a low precision value. This, in fact, means that some constraints must be imposed on the original query if the precision value is to be upgraded. However, it is also desired to at least maintain the recall value if not to upgrade it.

A new query is shown below together with the resulting search output.

```
USER QUERY
? 50+100+150$   example 2
 830 72.27454309067
 949 72.36577458697
 1343 18.30904940893
 1581 18.40028090523
 2901 93.91300185194
 2920 93.81043808767
 3415 93.79941058197
 3891 93.890337316
 4367 93.9815688123
 4386 25.48000269163
 4862 21.19186650487
 4881 55.33600500093
 4934 63.37506184989
 5247 63.30106996833
 6199 63.39230146463
 6237 63.28973770036

THE FOLLOWING ARE THE RELEVANT DOC'S!
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY!
  132  726 1320 1388 1972 2257 2393 3930
 THE PRECISION PR=1.00
 THE RECALL RC= .17

DOC.NO.  132 CONTAINS THE TERMS
   50  100  150  830 1902

DOC.NO.  162 CONTAINS THE TERMS
   50   69  100 2623 3891

DOC.NO.  235 CONTAINS THE TERMS
  100  513  949 1217

DOC.NO.  520 CONTAINS THE TERMS
   50  251 3650 6888

DOC.NO.  606 CONTAINS THE TERMS
   50   72  100  867 4616

DOC.NO.  692 CONTAINS THE TERMS
   50   87  117  381  620 6255

DOC.NO.  726 CONTAINS THE TERMS
   50   54  100  150  353 1601

DOC.NO.  898 CONTAINS THE TERMS
   50  221  333  577

DOC.NO.  984 CONTAINS THE TERMS
   50  100 2200 4490 5261

DOC.NO. 1001 CONTAINS THE TERMS
   50   75  100  554 6199

DOC.NO. 1069 CONTAINS THE TERMS
  149  150  192  741 6003
```

DOC.NO. 1148 CONTAINS THE TERMS
50 130 6870

DOC.NO. 1241 CONTAINS THE TERMS
50 113 365 3415 4591

DOC.NO. 1320 CONTAINS THE TERMS
50 69 100 150 313 965 2444 6237

DOC.NO. 1388 CONTAINS THE TERMS
50 78 100 150 423 830 949 1850 3790 4881

DOC.NO. 1413 CONTAINS THE TERMS
102 132 1584 4823

DOC.NO. 1560 CONTAINS THE TERMS
50 546 841 1966 4022

DOC.NO. 1585 CONTAINS THE TERMS
50 150 301

DOC.NO. 1653 CONTAINS THE TERMS
50 715 1340 4934

DOC.NO. 1732 CONTAINS THE TERMS
50 87 100 526

DOC.NO. 1825 CONTAINS THE TERMS
50 62 82 117 616 3002 6689

DOC.NO. 1904 CONTAINS THE TERMS
50 56 150 1638

DOC.NO. 1972 CONTAINS THE TERMS
50 100 150 742 2920 5247

DOC.NO. 2074 CONTAINS THE TERMS
50 338 2067 4224

DOC.NO. 2099 CONTAINS THE TERMS
50 100 181 260 830 847 949

DOC.NO. 2257 CONTAINS THE TERMS
50 100 150 770 6242

DOC.NO. 2393 CONTAINS THE TERMS
50 100 150 225 830 5801

DOC.NO. 2418 CONTAINS THE TERMS
50 94 100 2901

DOC.NO. 2737 CONTAINS THE TERMS
50 55 332 6729

DOC.NO. 2762 CONTAINS THE TERMS
50 51 163 345 3553 6694

DOC.NO. 2787 CONTAINS THE TERMS
50 98 100 1997

DOC.NO. 2923 CONTAINS THE TERMS
50 100 362 1634

DOC.NO. 3081 CONTAINS THE TERMS
50 88 901 913

The situation that results is somewhat the reverse of that of Example 1. The search request has retrieved a relatively few number of documents; however all of them are relevant.

For a true simulation of a user's search for relevant documents the list of relevant documents would not be displayed to the user. Instead he should receive only the list of retrieved documents togehter with an indicator of which of these retrieved documents are relevant. The additional output in Example 2 is included to illustrate the simulation procedure as well as the output available to the user.

Consider the following procedure that a user might follow after examination of the above outputs of retrieved documents in Examples 1 and 2.

By analysing the list of documents with their different terms it may be observed that $term_{50}$ appears in almost all the documents. Similarly $term_{100}$ and $term_{150}$ appear frequently,,but not always, in the relevant retrieved documents. These terms are certainly to be regarded as content terms. After examination of the two queries and their respective results it is desired to formulate another query that is neither too general nor too constrained. One formulation is a $term_{50}$ and either of the other two terms. With such a reformulation of the query the following results are obtained.

```
DOC.NO. 3106 CONTAINS THE TERMS
  150  360 3515

DOC.NO. 3425 CONTAINS THE TERMS
   50  100  167

DOC.NO. 3561 CONTAINS THE TERMS
   50   53  753 1709

DOC.NO. 3586 CONTAINS THE TERMS
   50

DOC.NO. 3880 CONTAINS THE TERMS
   59   68  195  214 1487

DOC.NO. 3905 CONTAINS THE TERMS
   80  444 1828

DOC.NO. 3930 CONTAINS THE TERMS
   50   53   60  100  150  594 5839

DOC.NO. 4292 CONTAINS THE TERMS
   50   77  100  706

DOC.NO. 4317 CONTAINS THE TERMS
   50  150  355 1343

DOC.NO. 4342 CONTAINS THE TERMS
   50

DOC.NO. 4367 CONTAINS THE TERMS
   50   75  190  256

DOC.NO. 4392 CONTAINS THE TERMS
   50   57   70  100 2216 5325

DOC.NO. 4639 CONTAINS THE TERMS
   50   58 4647

DOC.NO. 4955 CONTAINS THE TERMS
   50  150  362  572 6663

DOC.NO. 4980 CONTAINS THE TERMS
   50 1715 3520 4367

PLEASE SPECIFY PRINT OPTIONS
? 0000


USER QUERY
? 50+(100-150)$  example 3

THE FOLLOWING ARE THE RELEVANT DOC'S:
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
  132  162  606  726  984 1001 1320 1388 1585 1732
 1904 1972 2099 2257 2393 2418 2787 2923 3425 3930
 4292 4317 4392 4955
 THE PRECISION PR=1.00
 THE RECALL RC= .50
```

The results of Example 3 show significant improvement in recall in comparison to Example 2, and they show significant improvement in precision over Example 1. The simulation procedure allows the user to attempt to upgrade the system performance by changing the logic of the search request without changing any other query parameter. Addition of one more term into the list of search terms leads to the following results.

```
USER QUERY
? 50+(100-150-55)$   example 4

THE FOLLOWING ARE THE RELEVANT DOC'S!
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY!
   33  132  162  606  726  984 1001 1320 1388 1585
 1732 1904 1972 2099 2257 2393 2418 2737 2787 2923
 3425 3930 4292 4317 4392 4955
 THE PRECISION PR= .96
 THE RECALL RC= .52

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? /
    42.739 CP SECONDS EXECUTION TIME
/bye

KEMSI72   LOG OFF    14.42.29.
KEMSI72   SRU     85.139 UNTS.
```

It shows a 4 percent drop in the precision value and a 2 percent upgrade of the recall value as compared to the previous precision and recall values. Clearly, there is little to choose between the queries of Examples 3 and 4.

A more detailed discussion of an attempt to improve a question in order to optimize precision and recall values is given in Chapter VI.

CHAPTER IV

Retrieval function

4.1 The Query Language

For any document retrieval system, it is essential to use a query language that conveys a description of the user's interest as accurately as possible to the search processor. The process of transformation of the user's interest into a search query is based on the use of certain logic operations contained implicitly in the query language. In fact, the core of the search logic and the query language is the set of logic operators used to formulate the search requests [35]. Hence, it is extremely important that the syntax of the query language be well defined. Conventionally, the present description is in terms of the Backus Normal Form. Table 4.1 is the truth table for the BNF of the query language designed for use in the simulation experiments.

READ / READ NEXT	TERM	(AND) +	(OR) −	(NOT) *	(	)	ƀ	$	/
TERM	F	T	T	T	T	F	T	EOQ	EOJ
(AND) +	T	F	F	F	F	T	T	EOQ	EOJ
(OR) −	T	F	F	F	F	T	T	EOQ	EOJ
(NOT) *	T	F	F	F	F	T	T	EOQ	EOJ
(		T	T	T	F	F	T	EOQ	EOJ
)		F	F	F	F	T	T	EOQ	EOJ
ƀ	T	T	T	T	T	T	T	EOQ	EOJ
$	T	F	F	F	F	T	T	EOQ	EOJ
/		F	F	F	F	F	F	F	EOJ

Table 4.1 Truth table of Query Language

EOQ= End of Query
EOJ= End of Job

Table 4.2 below gives the explanation of the BNF symbols.

Symbol	Meaning
< >	Variable name or expression
: : =	is defined to be
|	exclusive OR
b̸	blank

Table 4.2 Interpretation of BNF Symbols.

The following specifications in the BNF represent the syntax of the query language developed for use in the present simulation experiments.

<query> : : = < search parameter >< end symbol >|< stop symbol >

<search parameter > : : = <term>|<in bracket >

<term>. : : = <numeric term> <logical operator><term>|
<numeric term ><logical operator>
<in bracket>|< numeric term>
<logical operator>< numeric term>

<logical operator > : : = < and >|<or >|<not >

<in bracket> : : = < bracket open >< term >< bracket close >|
< bracket open>< term ><bracket close>
<operator>

<operator > : : = <logical operator>< numeric term>|
<logical operator>< term >|<logical
operator >< in bracket>

<bracket open> :: = (

<bracket close>:: =)

<numeric term> :: = 0/1/2/3/4/5/6/7/8/9

<and> :: = +

<or> :: = −

<not> :: = *

<end Symbol> :: = $

<stop Symbol> :: = /

It may be noted that the query language itself is not the main topic of discussion in the present thesis, but since the aim is to simulate the performance of a document title retrieval system it is believed that a clear specification of a reasonably detailed query language is necessary in order to provide the basis for a subsequent development of an automatic question and answer system based on the present work.

To implement the query language two routines are needed to perform the following functions:.

1) Query translation
2) Search and match

The query language is keyword based and the query syntax, while providing much flexibility for the searcher, nevertheless requires him to specify exactly what he wishes to search for and the precise search alternatives. The searcher is allowed to use any combination of the primary logical operators AND, OR, NOT.

The boolean operator AND, when used in the query formulation, signifies the logical intersection of the following and the preceeding keywords. For example $term_1$ AND $term_2$ would represent all documents containing $term_1$ and $term_2$. A question may also be expressed in terms of OR logic, which signifies the disjunction of the following and the preceeding terms. Thus $term_1$ OR $term_2$ would represent all documents that contain either $term_1$ or $term_2$ or both. On the other hand the logical operator NOT, when used in the query formulation, signifies the documents that do not contain the term that immediately follows the NOT operator. Thus $term_1$ NOT $term_2$ would represent all documents that contain $term_1$ but do not contain $term_2$.

Other operators, such as adjacency ADJ and precedence PRE, are not included in the query language. These operators, since they represent the physical position of keywords in the document, are difficult to simulate in the framework of the simulation model since it uses a non-positional inverted file representation of the data base. Also, the simulation of synonyms and thesaurus dictionaries have been avoided. The basic reason for not including these features is because insufficient statistical information about synonyms and thesaurus is available.

Within the given scope of the model the essence of query translation is to communicate the stated information requirement of a searcher to the index. It may be inconvenient, or impossible, to use identical languages for queries and indexes. However the system considered in the present investigation consists of index searching operations which imply a common language between the search terms, or keywords, and the terms stored in the document records. In such a type of a system it is guaranteed that the indexer and the system users have a common vocabulary, and it is important that the users be well acquainted with the list of vocabulary terms in order to avoid errors in retrieval.

The basic operation in index searching is a table lookup in which a query term is input and searched for in an index file or dictionary. It should be noted that in an automatic retrieval system a query usually consists of more than a single search term. The search processor is implemented to perform one search request at a time.

The organization of the index data is based on use of an inverted file list, in which each record of the file contains one index term and a list of document numbers associated with that term. To find all documents that contain a particular term it is necessary only to search the index file for the record that contains the term, and then to retrieve all the listed documents. If two terms are required, for example $term_1$ AND $term_2$, then two records must be retrieved, their reference sets intersected and the result used to satisfy the query. The index file is stored in a mass storage random access file, and question processing procedures are specified in Fig. 4.3 and explained below.

Index File	Processing of Lists
$term_A$ Document numbers: 15 17 49 58 65	1. Query: $term_A$ Result Document numbers: 15 17 49 58 65
$term_B$ Document numbers: 13 17 38 65 107	2. Query: $term_A$ and $term_B$ Result Document numbers: 17 65 15 13 17 17 49 38 58 65 65 107

Fig. 4.3 Inverted Index File

Details of the query processing and search procedure are as follows. For each term of the query there must be created in core a list of document numbers from the inverted file. The parsing algorithm has two stacks called TSTACK and OSTACK which form respectively the term stack and the operator stack. Whenever there are at least two values in the TSTACK, and one value in the OSTACK, the control is passed to a matching algorithm. In the matching routine the two records of the index file that correspond to the respective index terms are selected and are merged in a manner dependent on the form of the logical operator input to the matching routine. The merged file is stored in an intermediate file and the control is given back to the parsing routine for further processing.

4.2 Query enhancement

Generally, query enhancement refers to an automatic procedure by which a query is changed for the purpose of improving the resulting response. For example, in one form of query enhancement a query term is replaced by a set of alternative terms. The requestor may program the replacement, and expect the query to find the information needed to modify itself. Alternatively the use of truncation operators, such as * and $, to signify unlimited and limited truncation mode respectively, may be described as a query modification scheme.

In the present approach the emphasis is on query enhancement by means of query reformulation through a technique that makes use of the user analysis of the results of a search request. A query is thus formulated by consideration of the output from a sequence of reformulated queries. Such techniques may be regarded as useful in general, since users are often not sufficiently familiar with the subject matter of the data base contents, the indexing vocabulary, and so forth to make useful modifications of the question without first observing the results of several search requests. As in relevance measurements, one of the most effective means of query modification is to let the requestor decide whether the desired relevance has been achieved. The major disadvantage in this technique is that it requires the user to be provided with fast computer response to the queries as is obtained with an interactive processing facility. If the results from the search are obtained only after a considerable time delay the application of such a technique is tedious and hence of limited value.

In the query enhancement experiments performed in the present investigation it was found that analysis of document records forms an excellent basis for query modification. For example, if very few documents are retrieved in response to a search request the one of the various actions that might be useful in the query reformulation technique is generalization of the query logic. Logic can be generalized by changing all, or some, AND (+) operators to OR (-) operators. Similarly, if too much is retrieved then the first of the OR logic operators may be changed to AND. Although this method may be regarded as rather a random process for performance upgrading of an information retrieval system it is fast and logically simple. It is the method likely to be used by a searcher who has on-line access to a document retrieval system.

CHAPTER V

Mechanisms for question modification

5.1 Use of Thesaurus

In information retrieval the term thesaurus usually refers to a representation of words or descriptors with an indication of certain groupings by subject category. Traditionally the use of a thesaurus or a synonym list has been found to be an effective means for improvement of the efficiency of an information retrieval system. It has been observed in many studies, including the one by salton [37], that the implementation of a thesaurus provides synonym recognition and may therefore be expected to be useful in retrieving some documents that cannot be retrieved easily by a keyword matching procedure alone [38]. Before considering the role of a thesaurus in the simulation of a document retrieval system it is appropriate to discuss briefly the important features involved in the construction and use of a thesaurus.

The purpose of a thesaurus in information retrieval is to provide vocabulary normalization by reduction of some query terms into equivalent representations contained in the synonym dictionary. Therefore, in construction of a thesaurus various considerations should be taken into account. These include consideration of the type of words to be included in the thesaurus, and the type of synonym category to be used (for example broader terms only, or narrow terms only, or both etc.)

Consider the choice of words to be included in the synonym dictionary. In general, words that are content bearing terms in the given subject area and that are present in the document collection are likely to be selected. In fact, there is no single rule for determination of the type of words. However, in the existing thesauri, such as Salton's thesaurus, Harris 2, Harris 3 thesaurus etc., it is observed that non-content terms, such as terms that represent articles and prepositions, are not included [37]. The choice of words might be based on the frequency count of words in the data base, or perhaps on some properties of word distribution for the given data base. Terms that occur with high frequency are usually included in the synonym dictionary. For example, if the data base deals with operating systems then terms such as computer, program, input, output might have reasonably high frequency but may not help to retrieve more relevant documents than non-relevant documents. It is therefore desirable that both very rare terms and very common high frequency terms should be excluded from the synonym dictionary. Individual high frequency terms might be replaced by a combination of two or more terms [39].

Selection of the type of synonym category may be dependent on the operating environment of an information retrieval system. If the users are interested only in broad retrieval which may result in high recall and low precision, then broader terms could be selected for the dictionary. Alternatively, if the users are interested in the retrieval of rather few, but all relevant, documents, and hence in high precision, then narrow terms are preferred for inclusion in the thesaurus.

Th implementation of a thesaurus in an information retrieval system may contribute to the efficiency of the whole system in several ways. The following are a few advantages of the use of a thesaurus:

1. A thesaurus is usually helpful in allowing a check of the acceptability of terms used by indexers and terms used by searchers. In some instances the indexers and searchers may be allowed to use any of several synonyms recognized in the thesaurus.

2. It is possible to maintain statistics of term frequency and usage. In fact, if the frequency of the query terms are displayed it may help the searcher to formulate an effective query. Also, statistics of the frequency of assignment in indexing and searching may be maintained for

the vocabulary control.

3. By the use of a thesaurus the scope of the retrieval procedure can be extended to collections in different subject areas, since thesaurus construction is complementary to the retrieval process.

4. For an on-line user a thesaurus is a very effective mechanism for question modifications. For example, given a set of thesaurus entries, it may be desired to display all related entries that appear under the same concept category. Alternatively, a display of the complete hierarchical structure of a set of query terms could facilitate the question formulation for performance of exhaustive searches.

In fact a thesaurus could be studied, in itself, as a function of an information retrieval system environment. Thus standard rules for thesaurus construction cannot be applied in every environment. In fact in some special cases a new set of operating rules may be needed in order to adjust to a particular environment [40].

It may be noted that very little statistical knowledge is needed for simulation of a thesaurus. In general, the following properties may be required for the simulation of a thesaurus and of its use.

a) The objective of using a thesaurus is to improve the efficiency of an information retrieval system by means of various search and data manipulation mechanisms. To simulate the thesaurus the whole environment of the retrieval system should be structured according to properties of simulated documents and simulated queries.

b) A simulated thesaurus could optimize a user search request on the basis of term frequencies in the data base in order to direct an optimal path for the search.

c) Displaying synonyms from simulated thesaurus entries could help an on-line user to formulate an optimal query. In order to implement this option the simulation model could generate some similarity rule between the simulated queries and the simulated documents. The rule could be based on word association mechanisms. [41,42]

d) The size of vocabulary, the number of classes in the vocabulary, and the means of calculating the association of two words are the important features to be considered in construction of a simulated thesaurus.

In the present study the queries are not simulated by an automatic procedure. Therefore, no attempt has been made to construct a simulated thesaurus. However, a further study could be made to determine the important features required for simulation of a thesaurus that could be embedded in the present simulation model. The use of a thesaurus in the simulation model might improve the performance of the retrieval system.

5.2 Citation indexing

As discussed in the previous section the use of a thesaurus in a document retrieval system may be expected to produce considerable improvement in the effectiveness of retrieval. For the same reason citation indexing may be used to supplement the conventional subject indexing in order to achieve better retrieval performance.

The use of citation indexing in a document retrieval environment recognizes the fact that some documents are connected to some other documents by means of bibliographic citations. For example document 1 may cite a set of documents 2, 3, 4. The documents 2, 3, 4 in turn may cite (or be cited by) another set of documents. In addition, the bibliographic citations also play a role of content identifiers [37]. In many studies it is observed that documents related by similarities in bibliographic citations also provide a large number of common subject identifiers. Finally the most important feature of citation indexing in context of the present study and, in particular, as a mechanism for question modification is the fact that bibliographic citations are usually not used directly as content indicators for retrieval purposes. Instead, they are incorporated as feedback information during the search process in an attempt to retrieve additional information similar to that being identified in the search [43-46].

Specifically, in a feedback mechanism an initial search is made leading to the retrieval of a number of documents. The output is scanned manually or automatically and document authors, citations made by the documents, and authors of these citations are returned to the system to be incorporated into an improved search formulation. The use of this bibliographic feedback process may lead to better retrieval results. In one of the studies [47] it was found that, by adding citations data with standard subject terms in a feedback environment, improvements of up to 10 percent in retrieval effectiveness were obtained above the results produced by use of subject terms only.

To study citation indexing by means of simulation of a document retrieval system requires a careful study of the properties of document citations. A general statistical study of documents that deal with a certain subject area and their citation pattern should be made. It is observed in the literature that such statistics are either not available or they are not sufficiently complete to allow formulation of a mathematical or a statistical model for a simulation study.

In any attempt to simulate citation indexing it is important to consider the following parameters in addition to those described in the present model.

1. A conditional probability parameter (X) that a document to be generated will cite a randomly chosen relevant document in the data base.

2. A second conditional probability (Y) that is interpreted in the same manner as (1) except that the cited document is non-relevant.

Based on the above two conditional probabilities it might be possible to simulate the probability that N documents from a collection of M documents in the data base are cited by the generated pseudo documents. However, the validity of this conclusion could be very dependent on the citation pattern of the data base, the frequency of documents cited in the data base, and the variation in the number of documents cited by different documents etc.

If citation indexing is included in the simulation model it might prove helpful for simulation of an on-line user in a feedback mechanism environment of an information retrieval system. In the next section some existing feedback mechanisms are discussed briefly.

5.3 Feedback mechanisms

A feedback mechanism is designed primarily for on-line users of an information retrieval system. In such an environment some user interaction with the system is important and necessary in order to implement the system effectively and to improve the retrieval performance. The following three examples are a few of the relevance feedback mechanisms implemented in the SMART system by Salton [48].

1. <u>The automatic dictionary process:</u>

In this process a system is considered in which a communication link enables the user to influence the search process by making it possible for him to choose certain terms to be added or deleted from the original search request. To implement this process a thesaurus is used in several ways as, for instance, to display related entries under a certain concept category, to show a hierarchical arrangement of terms or concept classes, to display a statistical term-term association matrix so that for a given set of terms it is possible to find all the related entries that exhibit a tendency to co-occur in many documents, and finally based on the set of documents retrieved from the original search request the user may add to the terms originally specified in the initial query all those terms which occur in several of the retrieved documents but do not occur in the initial request. In addition, the automatic dictionary process may display the frequencies with which the various terms are assigned to the documents of the data base.

2. Request optimization using relevance feedback:

The previous feedback mechanism, known as a vocabulary feedback mechanism, is different from request optimization in that with request optimization the user plays a smaller role since most of the optimization is performed automatically by the system.

The process consists of initiation of an initial search, and presentation to the user of a certain number of retrieved documents. The user examines some of the documents and classifies them as either relevant or non-relevant. These relevance judgements are returned to the system which adjusts the initial search request in such a way that the query terms present in the relevant documents are enhanced by increasing their weight, whereas terms that occur in non-relevant documents are similarly devalued by decreasing their weights. The degree of improvement to be obtained from the user feedback information is dependent on the user supplied relevance judgements.

3. Automatic modification of the relevance process:

The feedback mechanism involved in this process deals with the qualitative judgement of the user instead of relevance judgement as in the previous case. On the basis of examination of retrieved documents the user makes a qualitative assessment of the output. For example, he may realize from examination of the retrieved documents that his query was interpreted too broadly or too narrowly. The feedback mechanism provides this information to the system and results in selective changes in the document and request analysis process.

The next example of a feedback mechanism is different from the above three in the way that the user initiates a search request and subsequently the system uses an automatic feedback mechanism to retrieve the best possible results.

4. <u>An automatic optimum iterative feedback system:</u>

This example of feedback mechanism consists of three phases:

1) Pre-search phase
2) The search phase
3) Post search phase.

In the pre-search phase, the user formulates a search request with the help of a set of index terms and their associated weights which can be abstracted automatically from the data base. The search phase is responsible for deciding the degree of relevance that a document has in relation to the search request. The relationship between document relevance to the search request is in fact a relevance measure which the system automatically calculates based on some statistical criterion. Subsequently the calculated relevance measure provides the document with a relevance value which could be greater than, or equal to the system pre-determined cutoff value. A document selected as relevant at this stage is classified as provisionally relevant and the set of all such documents is arranged in descending order of relevance. Finally, in the post search phase of the analysis the members of the set of provisionally relevant documents are checked to determine whether some pre-defined relevance criterion are met. In case of failure to meet the relevance standard the search request is modified automatically and the control is given back to the search phase. such analysis and modification is repeated until the required relevance criterion are met.

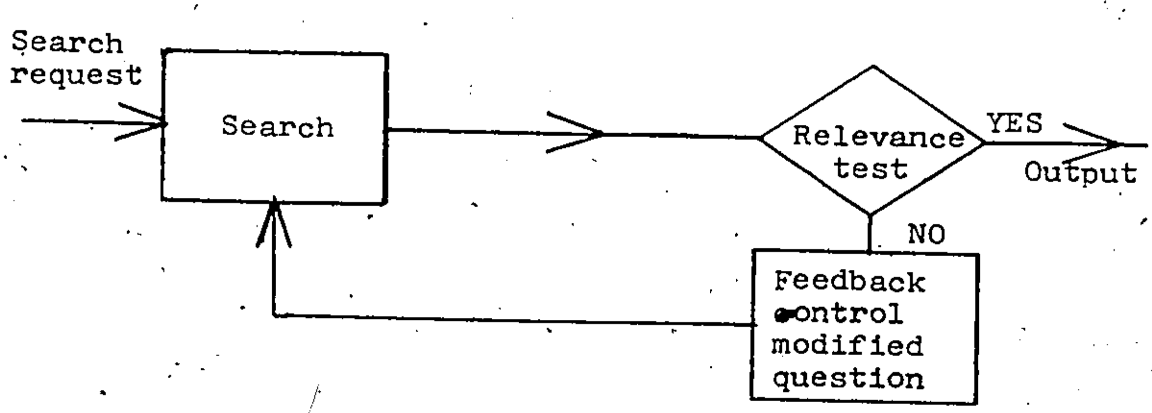


Fig. 5.1 is the representation of the system described above.

In concluding this chapter, it may be noted that term manipulation and measures of term importance are important factors that affect the overall efficiency of an information retrieval system [49].

CHAPTER VI

Illustration and conclusions

6.1. Determination of Simulation Parameters

The purpose of simulation of an information retrieval system is to provide an economical means of conducting experiments with a view to determination of the factors that affect the efficiency of an information retrieval system. It is also hoped that simulation will provide a means whereby retrieval procedures may be improved in order to increase the effectiveness of the retrieval process.

Consider an experiment in which it is desired to simulate the situation in which a user interrogates a data base on-line in order to search for a set of documents relevant to his interest. The queries are chosen by the user, but all other quantities that affect retrieval effectiveness are properties of the particular data base. Thus in simulating the responses to the user the following questions should be asked (see Fig. 6.1).

a) What is the size of the data base?

The answer to this question determines N, D, and M.

b) What is the value of X?

The value of X may be chosen arbitrarily but it allows the simulation of different data bases with similar statistical properties. Consideration of data bases where parameters are identical except for the value of X allows simulation of different issues of a data base by a particular supplier.

c) Is it proposed to simulate a data base that contains many or few, documents relevant to the user's interest? Both cases are important since the proper search technique may well depend on whether the data base contains many, or few, relevant documents. In Fig. 6.1 it is supposed that many documents is the number 50 and that few documents is the number 5. The corresponding values to be chosen for the parameter c are then c=50/M and c=5/M.

d) How well does the indexing scheme of documents in the data base relate to the interest of the user?

All possibilities are of interest since the optimum search strategy may well depend on the correlation between indexing terms and the interest of the user. In the simulation model of the present thesis the suitability of the indexing scheme is characterized by:

1) <u>Number of content terms.</u>

In Fig. 6.1 a number of 20 is regarded as high, and a number of 5 is regarded as low. The number determines the value chosen for Pc.

2) <u>Relevance rating of content terms.</u>

If the content terms are good descriminators of relevant documents then delta should be chosen large (say = 100). If the content terms are poor descriminators then delta should be chosen small (say = 5).

3) <u>Number of accidental term associations</u>.

Accidental term associations might arise through inconsistent indexing or the occurrence of hononyms. In Fig. 6.1it is supposed that the occurrence of 5 such terms is low, and the occurrence of 50 such terms is high. The number n of such accidental terms determines the value of Pn since

$$Pc + Pn + n/D = 1.$$

It may be noted that the occurrence of a few content terms with large relevance rating, or delta, characterizes a data base in which relatively short questions are sufficient. In contrast, the occurrence of a large number of content terms, but with small delta Δ, characterizes a data base whose terms are less specific but may be combined into relatively large questions to produce satisfactory output of relevant documents. On the other hand if there are few content terms, and a low value of Δ , then the indexing scheme does not correlate well with the interests of the particular user.

The above quantities concern the suitability of the data base. There is also the following question concerning the users initial choice of question terms.

e) Has the user made a good initial choice of question terms?

In Fig.6.1 it is supposed that R values of 50 apply to well-chosen question terms, whereas R values of 5 apply to poorly chosen question terms. It is clearly of interest to know how the efficiency of a feedback process is affected by the initial choice of question terms.

The above discussion indicates that it is possible to treat a user as being either well-informed or ill-informed with regard to the terms to be used in his initial question formulation. Similarly it is possible to simulate a well-indexed or poorly-indexed data base.

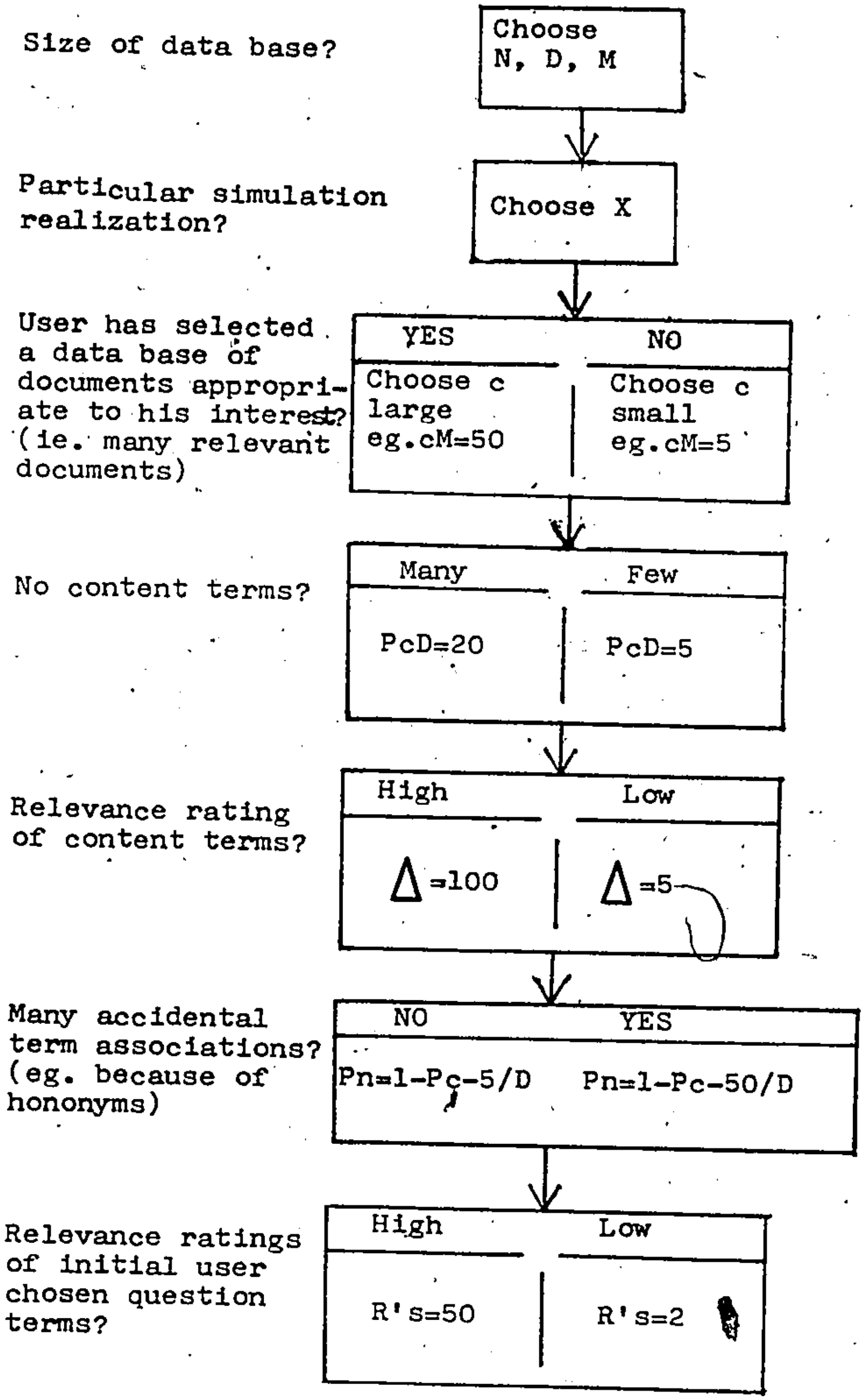


Fig. 6.1 Choice of Simulation Parameters

6.2 Illustration of simulation for on-line user

This section deals with an approach that an on-line user may adopt to update his queries. In fact some of its aspects have already been discussed in chapter iii. Consider the following data base parameters:

M=5000, D=7000; N=40,000

with values of Pc=.02, c=.01, and X=.00001.

Suppose an on-line user with some specific subject interest profile issues an arbitrary search request

50+100+150$ (Q1)[1]

For the above user query it is found that 8 documents are retrieved and the precision value of 1.00 and the recall value of 0.17 is output. The system simulates the relevance of a document by comparing the list of retrieved documents with the list of relevant documents in the data base already determined. The list of relevant documents in the data base in fact represents the system's criteria of relevancy based on the values of the input parameters. These values may have been determined previously through user experiments. However, the on-line user has no idea of these parameters. Also, the user is not shown any details of the relevant documents. He sees only the retrieved documents and the values of precision and recall. The user may analyse distribution of terms in the output

1 For results see Appendix "C"

documents and may reach the following two conclusions.

(1) The initial query was interpreted by the system as very constrained and therefore should be generalized.

(2) The three terms of rank 159,307 and 427 appear very frequently in the retrieved documents and therefore some of these terms might be included in the subsequent queries.

Suppose the user decides to generalize Q1 to

50-100-150$ (Q2)[1]

The system retrieves 122 documents and outputs a precision value of .37 and recall .94. This situation is somewhat contrary to the first one. However, by analyzing the retrieved documents the user finds that terms 159,307 and 427 are once again occurring quite frequently, and in addition, terms 308,477 and 774 are also frequent in the retrieved documents. Thus the output of Q1 and Q2 suggests the query (Q3)

$$50+(100-150-159-307-308-477-774)^2\$$$

The system response to Q3 results in retrieval of 40 documents with a precision value of 1.00 and a recall of .83. Q3 could be judged as a clear improvement over both Q1 and Q2. The output for Q3 fails to retrieve only 17 percent of the relevant documents.. However, it may be noticed that term 427 was not used in the query in spite of its frequent occurrence in documents output for Q1 and Q2. This ultimately suggests the following query (Q4)

1,2 For results see Appendix "C"

(50+(100-150-159-307-308-477-774))-(307+427+(100-150))[1]$

The system response to Q4 results in retrieving 42 documents with the same value of precision as in Q3, but the recall value is increased to .88.

The above feedback approach for an online user suggests how one can develop a sequence of questions to obtain the optimal query response. Starting with the user's initial query the system provides some feedback to the user which he may use to update the subsequent queries. In fact, the feedback mechanism process should continue until the user is satisfied with the output.

6.3 Summary of several experiments

Several experiments conducted in the present investigation are summarized in the TABLE 6.2. Several data bases have been considered in order to prove the consistency and reliability of the simulation rules and the stability of the simulation results. The results show that if different data bases are chosen for a particular type of user group, or a set of user groups, the simulation rules remain the same.

In table 6.2 the column for the value of X indicates different data bases for different values of X. Since X is used as a seed for pseudo random numbers representing pseudo documents, any change in the value of X generates a new set of documents and therefore a different data base.

1 For results see Appendix "C"

Similarly, the value of P_c is varied to include the possibility of different percentages of content-terms in the data base. These changes are important as they do occur in real data bases.

Similar tests have been conducted with a large data base of the following size

$$M=50{,}000, \; D=20{,}000, \text{ and } N=400{,}000$$

Except for the increased processing time the results were found to be equally stable as in case of the smaller data base.

M=5000, D=7000, N=40000,
Δ=100, Pn= .3, Ri=50 i=50, 100, 150

Pc	C	X	Query	Pre.	Rec.
.002	.01	.00001	50+100+150 =Q1	1.00	.17
			50-100-150 =Q2	.37	.94
			50+(100-150)=Q3	1.00	.50
			50+(100-150-159-307-308-477-774)=Q4	1.00	.50
			Q4-(307+427+(100-150)=Q5	1.00	.50
.02	.01	.00001	Q1	1.00	.17
			Q2	.37	.94
			Q3	1.00	.50
			Q4	1.00	.83
			Q5	1.00	.88
.002	.01	.01	Q1	1.00	.04
			Q2	.39	.98
			Q3	1.00	.45
.02	.01	.01	Q1	1.00	.04
			Q2	.39	.98
			Q3	1.00	.45
			Q4	1.00	.71
			Q5	1.00	.71
.002	.01	.02	Q1	1.00	.10
			Q2	.37	.92
			Q3	1.00	.50
.02	.01	.02	Q1	1.00	.10
			Q2	.37	.92
			Q3	1.00	.50
			Q4	1.00	.70
			Q5	1.00	.70
M=50000, D=20000, N=400000 Δ=100, Pn=.3, Ri=50 i=50, 100, 150					
.002	.001	.00001	Q1	1.00	.13
			Q2	.03	.96
			Q3	.73	.41

Table 6.2 Summary of experiments

Results of online user experiments with the larger data base.

```
N=400000 M=50000 D=20000

FOR I=50,100,150  RI= 50

PC= .002 PN= .300 DELTA=100.0 X=.00001 C=.001

        PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
        1. USE SIMPLE INTEGERS FOR TERM VALUES
        2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
        3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
        4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
        5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
        6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                               USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 0100

USER QUERY
? 50+100+150$

THE FOLLOWING ARE THE RELEVANT DOC'S:
   320   554  3759  3965  6075  7400  7812  9407 10938 11762
 12533 13357 14181 16007 17371 18426 19197 19840 21257 21488
 23136 24447 24678 26095 26326 29516 29747 31164 31395 32706
 32937 34123 34354 37544 37775 39192 39423 39785 40247 43206
 43543 43668 46627 46964 47089 49923

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
   320  7400  7812 12533 18426 21488
 THE PRECISION PR=1.00
 THE RECALL RC= .13

DOC.NO.   320 CONTAINS THE TERMS
    50    82   100   335   429  1123  5173 11611

DOC.NO.   554 CONTAINS THE TERMS
    69   119  1060  2416 13218

DOC.NO.  3759 CONTAINS THE TERMS
    50    60  4395  4701  5018  7388

DOC.NO.  3965 CONTAINS THE TERMS
    50   330  1793  3616  3885  4211  5026

DOC.NO.  6075 CONTAINS THE TERMS
    50   165   184  1667  3511 15913
```

```
DOC.NO.  7400 CONTAINS THE TERMS
    50    68   100   164 19374

DOC.NO.  7812 CONTAINS THE TERMS
    50    63    74   100   150   222   428   656  1176  4352

DOC.NO.  9407 CONTAINS THE TERMS
    50    59    64    68   491   679  7046 13586

DOC.NO. 10938 CONTAINS THE TERMS
    50    75   100   429   776  2396  7065 10318

DOC.NO. 11762 CONTAINS THE TERMS
    50   100   584   871  4374  8818

DOC.NO. 12533 CONTAINS THE TERMS
    50    83   100   128   256   554   587   596

DOC.NO. 13357 CONTAINS THE TERMS
    50   972

DOC.NO. 14181 CONTAINS THE TERMS
    50   158   364

DOC.NO. 16007 CONTAINS THE TERMS
   100   150   429   657  5010

DOC.NO. 17371 CONTAINS THE TERMS
    50    67   150   557  2451

DOC.NO. 18426 CONTAINS THE TERMS
    50    83   100   150   461  1221  2309

DOC.NO. 19197 CONTAINS THE TERMS
   100   129  1368

DOC.NO. 19840 CONTAINS THE TERMS
    50   150   429   543   587   606   661   729  1128 18751

DOC.NO. 21257 CONTAINS THE TERMS
    50    64   231   429   452  1126  1450

DOC.NO. 21488 CONTAINS THE TERMS
    50   100   150   417

DOC.NO. 23136 CONTAINS THE TERMS
   100   416 12747

DOC.NO. 24447 CONTAINS THE TERMS
    50   797   903

DOC.NO. 24678 CONTAINS THE TERMS
    50   100   587  8471

DOC.NO. 26095 CONTAINS THE TERMS
    50    71   167 17232

DOC.NO. 26326 CONTAINS THE TERMS
    72   100   587  1320  9297

DOC.NO. 29516 CONTAINS THE TERMS
    50   587 14809

DOC.NO. 29747 CONTAINS THE TERMS
    50    64    71   604
```

```
DOC.NO. 31164 CONTAINS THE TERMS
    50   230   248  2266

DOC.NO. 31395 CONTAINS THE TERMS
    50

DOC.NO. 32706 CONTAINS THE TERMS
    50    61   150   235  2212 11421 14836

DOC.NO. 32937 CONTAINS THE TERMS
    62    79   100   252  1092 12386 15537

DOC.NO. 34123 CONTAINS THE TERMS
    50    95   252  2172  4373  5445 17417

DOC.NO. 34354 CONTAINS THE TERMS
    50   100   150  1520

DOC.NO. 37544 CONTAINS THE TERMS
    50   100   150

DOC.NO. 37775 CONTAINS THE TERMS
    50   172  2265  3955

DOC.NO. 39192 CONTAINS THE TERMS
   171

DOC.NO. 39423 CONTAINS THE TERMS
   273   324   587  1176  1630  2010

DOC.NO. 39785 CONTAINS THE TERMS
    50   123   150   527  1644

DOC.NO. 40247 CONTAINS THE TERMS
    50    86   974  2439  5517

DOC.NO. 43206 CONTAINS THE TERMS
    50    87   100   266   428

DOC.NO. 43543 CONTAINS THE TERMS
    50    56  1176  5252

DOC.NO. 43668 CONTAINS THE TERMS
    50    58    66   100   150   764  9764

DOC.NO. 46627 CONTAINS THE TERMS
    50   100   101   687 18193

DOC.NO. 46964 CONTAINS THE TERMS
    50   150   163   667   808  1195 14556

DOC.NO. 47089 CONTAINS THE TERMS
    50    56    76   150   509

DOC.NO. 49923 CONTAINS THE TERMS
    50    73   162   712
   118.802 CP SECONDS EXECUTION TIME
```

```
N=400000 M=50000 ?=20000

FOR I=50,100,150  RI= 50

PC= .002 PN= .300 DELTA=100.0 X=.00001 C=.001

        PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
        1. USE SIMPLE INTEGERS FOR TERM VALUES
        2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
        3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
        4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
        5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
        6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                              USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPE[illegible]
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? 50-100-150$

THE FOLLOWING ARE THE RELEVANT DOC'S:
   320   554  3759  3965  6075  7400  7812  9407 10938 11762
 12533 13357 14181 16007 17371 18426 19197 19840 21257 21488
 23136 24447 24678 26095 26326 29516 29747 31164 31395 32706
 32937 34123 34354 [illegible]544 37775 39192 39423 39785 40247 43206
 43543 43668 46627 46964 47089 49923

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
    16     25     29     53    128    134    211    260    320    325
   344    347    384    400    401    442    448    530    556    582
   585    586    617    641    643    670    687    713    769    812
   885    895    920    924    941    972   1009   1037   1109   1146
  1181   1213   1224   1332   1349   1416   1468   1525   1533   1580
  1587   1615   1761   1794   1831   1917   1918   1922   2004   2029
  2094   2177   2202   2212   2233   2249   2257   2261   2323   2408
  2458   2479   2505   2509   2511   2549   2625   2666   2750   2754
  2768   2784   2814   2878   3027   3042   3130   3143   3261   3306
  3323   3444   3477   3480   3487   3516   3543   3551   3582   3596
  3606   3639   3664   3682   3691   3697   3710   3730   3743   3748
  3759   3770   3796   3846   3884   3925   3965   3985   4041   4056
  4075   4124   4131   4209   4211   4234   4277   4301   4320   4357
  4377   4391   4541   4548   4557   4654   4676   4777   4782   4829
  4876   4939   4970   4987   4994   5033   5044   5046   5104   5193
  5248   5261   5353   5360   5367   5403   5435   5449   5454   5459
  5467   5468   5564   5581   5621   5637   5647   5682   5718   5727
  5760   5769   5792   5835   5904   5912   5957   5982   6041   6059
```

6075	6086	6089	6100	6172	6177	6237	6327	6378	6403
6410	6555	6600	6758	6765	6818	6977	7089	7096	7258
7259	7350	7363	7400	7416	7425	7448	7484	7527	7535
7549	7556	7605	7662	7760	7793	7812	7889	7906	7949
7984	7994	8072	8109	8126	8136	8156	8178	8252	8362
8403	8428	8500	8525	8549	8591	8597	8607	8648	8697
8727	8779	8858	8918	9001	9101	9103	9132	9137	9171
9212	9228	9245	9247	9315	9385	9407	9422	9424	9598
9614	9694	9716	9747	9764	9796	9797	9811	9815	9837
9863	9890	9945	9979	10239	10339	10353	10383	10387	10399
10424	10474	10488	10573	10581	10641	10668	10727	10790	10025
10844	10888	10923	10938	10956	10964	10994	11113	11124	11143
11161	11172	11215	11311	11350	11400	11421	11445	11555	11507
11604	11641	11762	11877	11900	11904	11916	11973	11981	12010
12052	12125	12133	12145	12158	12164	12227	12236	12246	12324
12346	12353	12359	12383	12401	12440	12483	12519	12520	12528
12533	12535	12540	12574	12610	12673	12708	12723	12800	12831
12852	12909	12915	12937	12954	13044	13048	13050	13051	13078
13094	13153	13200	13240	13258	13293	13307	13339	13357	13437
13473	13525	13566	13610	13642	13649	13677	13762	13773	13802
13821	13826	13904	13953	13988	13989	14071	14093	14181	14187
14252	14379	14414	14430	14466	14471	14477	14500	14507	14519
14542	14557	14564	14578	14603	14668	14860	14872	14902	14929
15051	15064	15070	15085	15158	15205	15237	15262	15268	15279
15363	15405	15426	15429	15432	15447	15479	15528	15544	15555
15573	15589	15685	15609	15705	15711	15822	15838	15845	15862
15907	15908	15921	15933	15967	15989	16007	16028	16046	16069
16092	16136	16169	16212	16240	16247	16250	16316	16403	16482
16560	16693	16751	16770	16796	16849	16853	16880	16913	16966
16989	17028	17052	17069	17103	17140	17150	17186	17191	17371
17463	17500	17507	17527	17587	17667	17676	17715	17765	17788
17799	17801	17803	17899	17908	17977	18003	18094	18117	18132
18164	18230	18263	18267	18269	18283	18284	18296	18313	18330
18350	18405	18411	18426	18454	18466	18480	18753	18843	18894
18895	18951	19041	19082	19091	19121	19122	19131	19161	19197
19215	19288	19292	19297	19322	19334	19401	19421	19437	19449
19453	19525	19532	19541	19552	19576	19591	19608	19621	19660
19699	19706	19730	19765	19813	19829	19840	19850	19895	19915
19921	19970	19973	19996	20029	20059	20067	20124	20142	20197
20201	20405	20406	20418	20480	20541	20558	20579	20614	20622
20686	20720	20761	20770	20795	20806	20817	20834	20850	20868
20906	20914	20976	20984	20987	20989	20998	21190	21257	21392
21396	21449	21460	21488	21572	21589	21603	21652	21661	21674
21695	21700	21723	21730	21791	21833	21835	21885	21890	21988
22049	22065	22081	22144	22156	22173	22238	22241	22310	22326
22346	22396	22402	22436	22446	22457	22500	22524	22580	22601
22620	22666	22676	22693	22775	22900	22969	22993	23055	23073
23075	23081	23083	23128	23136	23142	23226	23238	23255	23297
23319	23329	23375	23363	23391	23410	23601	23604	23607	23724
23740	23742	23778	23887	23913	23944	23974	24036	24128	24142
24162	24168	24176	24202	24205	24256	24346	24348	24427	24447
24488	24508	24529	24537	24627	24660	24678	24700	24718	24768
24885	24912	24942	25007	25145	25182	25305	25313	25336	25346
25421	25469	25514	25549	25652	25754	25765	25785	25794	25827
25851	25853	25911	26009	26029	26033	26041	26095	26102	26139
26178	26326	26376	26385	26445	26539	26541	26568	26635	26645
26680	26701	26767	26770	26850	26872	26951	26977	26979	27003
27011	27072	27125	27163	27212	27214	27251	27252	27284	27333
27392	27399	27471	27505	27524	27549	27551	27556	27560	27642
27643	27651	27658	27797	27820	27833	27864	27882	27888	28005
28024	28062	28230	28256	28316	28465	28489	28490	28512	28516
28589	28702	28855	28857	28870	28954	28990	29021	29056	29068
29112	29154	29206	29356	29373	29476	29480	29516	29599	29621
29653	29662	29677	29684	29718	29743	29747	29779	29781	29812
29836	29849	29870	29871	29905	29906	30028	30090	20199	30232
30287	30290	30307	30413	30469	30549	30566	30585	30601	30615

```
30287 30298 30387 30413 30469 30549 30566 30585 30601 30615
30750 30764 30818 30903 30914 30944 30954 31024 31164 31193
31205 31341 31364 31395 31401 31437 31499 31545 31565 31577
31657 31664 31667 31697 31785 31807 31816 31830 31967 32009
32069 32078 32093 32154 32218 32321 32342 32400 32440 32558
32660 32687 32706 32787 32794 32847 32870 32872 32901 32920
32937 32994 33006 33023 33045 33170 33263 33374 33380 33437
33463 33465 33475 33477 33512 33531 33557 33575 33607 33699
33821 33835 33941 33979 34014 34022 34078 34123 34129 34183
34210 34263 34268 34308 34310 34354 34380 34479 34574 34578
34605 34621 34640 34675 34692 34707 34724 34737 34774 34837
34866 34892 34893 34921 34968 34973 35000 35036 35054 35060
35114 35116 35123 35133 35180 35195 35196 35209 35216 35329
35372 35377 35400 35412 35453 35459 35506 35563 35568 35577
35632 35679 35697 35734 35825 35844 36043 36074 36090 36117
36167 36294 36314 36326 36330 36423 36441 36514 36530 36670
36681 36692 36771 36791 36797 36833 36912 36923 36930 36936
36939 37059 37213 37233 37249 37254 37296 37402 37439 37468
37544 37550 37555 37570 37580 37584 37666 37775 37807 37828
37877 38103 38115 38158 38199 38234 38240 38339 38342 38378
38431 38461 38466 38527 38547 38549 38552 38629 38644 38648
38653 38748 38804 38807 38825 38844 38950 38968 39120 39135
39144 39192 39264 39293 39307 39333 39489 39492 39509 39541
39549 39565 39605 39624 39628 39740 39774 39785 39800 39831
39867 39894 39922 40060 40071 40148 40192 40232 40237 40247
40268 40322 40336 40351 40383 40404 40426 40454 40457 40460
40478 40494 40520 40568 40577 40602 40659 40708 40732 40746
40755 40785 40834 40874 40890 40897 40944 40952 40981 40993
41095 41098 41112 41118 41140 41186 41189 41324 41351 41375
41446 41498 41519 41545 41571 41663 41702 41715 41785 41791
41811 41821 41839 41909 41922 41928 41972 41984 41987 42015
42018 42038 42114 42160 42235 42244 42302 42325 42367 42379
42386 42425 42535 42643 42687 42703 42728 42760 42779 42832
42848 42865 42894 42916 42973 43027 43035 43038 43045 43130
43132 43140 43206 43218 43233 43314 43315 43318 43345 43447
43543 43550 43657 43668 43687 43696 43774 43786 43807 43813
43825 43885 43936 44020 44067 44117 44209 44222 44233 44252
44277 44304 44335 44341 44351 44403 44431 44449 44497 44556
44588 44659 44683 44704 44709 44728 44786 44800 44829 44846
44860 44874 44936 44938 44952 44969 44970 45019 45036 45113
45127 45146 45167 45174 45243 45361 45372 45373 45436 45469
45475 45578 45591 45615 45623 45630 45636 45644 45682 45735
45788 45832 45847 45860 45912 45934 46005 46027 46029 46064
46151 46165 46192 46237 46262 46295 46352 46417 46473 46486
46546 46551 46627 46655 46762 46797 46920 46921 46964 46972
47024 47088 47089 47135 47150 47167 47237 47259 47275 47308
47379 47380 47400 47423 47450 47452 47514 47532 47644 47710
47753 47824 47846 47871 47992 48017 48020 48074 48101 48124
48141 48157 48243 48274 48314 48355 48360 48377 48494 48525
48619 48641 48724 48732 48736 48762 48773 48782 48789 48820
48840 48854 48905 48929 48947 48979 48983 49012 49019 49056
49061 49111 49205 49327 49336 49342 49421 49469 49540 49554
49566 49592 49633 49637 49638 49675 49687 49781 49785 49836
49840 49874 49876 49881 49923
THE PRECISION PR= .03
THE RECALL RC= .96
   48.118 CP SECONDS EXECUTION TIME
```

```
N=400000 M=50000 D=20000

FOR I=50,100,150  RI= 50

PC= .002 PN= .800 DELTA=100.0 X=.00001 C=.001

          PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
          1. USE SIMPLE INTEGERS FOR TERM VALUES
          2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
          3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
          4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
          5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
          6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                              USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? 50+(100-150)$

THE FOLLOWING ARE THE RELEVANT DOC'S:
   320   554  3759  3965  6075  7400  7812  9407 10938 11762
 12533 13357 14181 16007 17371 18426 19197 19840 21257 21488
 23136 24447 24678 26095 26326 29516 29747 31164 31395 32706
 32937 34123 34354 37544 37775 39192 39423 39785 40247 43206
 43543 43668 46627 46964 47089 49923

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
   320  5467  7400  7812 10938 11762 11900 12533 12540 14430
 17371 18426 19292 19840 20850 21488 24678 31164 32706 34354
 37544 39785 43206 43543 43668 43786
 THE PRECISION PR= .73
 THE RECALL RC= .41
   27.943 CP SECONDS EXECUTION TIME
```

```
N=400000 M=50000 D=20000

FOR I=50,100,150  RI= 50

PC= .002 PN= .800 DELTA=100.0 X=.00001 C=.001

          PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
          1. USE SIMPLE,INTEGERS FOR TERM VALUES
          2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
          3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
          4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
          5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
          6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                                        USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? 50+(100-150-159-307-308-477-774)$

THE FOLLOWING ARE THE RELEVANT DOC'S:
   320   554  3759  3965  6075  7400  7812  9407 10938 11762
 12533 13357 14181 16007 17371 18426 19197 19840 21257 21488
 23136 24447 24678 26095 26326 29516 29747 31164 31395 32706
 32937 34123 34354 37544 37[illegible]5 39192 39423 39785 40247 43206
 43543 43668 46627 46964 47089 49923

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
   320  1181  4391  4541  5467  7400  7812 10938 11762 11900
 12533 12540 14430 17371 18426 19292 19840 20850 21488 22969
 23742 24128 24678 31164 32706 34354 36833 37544 39785 43206
 43318 43543 43668 43786 45578 46027 49638
 THE PRECISION PR= .51
 THE RECALL RC= .41
  35.195 CP SECONDS EXECUTION TIME
```

APPENDIX A

Instructions for use of simulation program

As stated previously, the simulation process is programmed in the FORTRAN language. Before a user attempts to use the simulation program it is important that he knows the query syntax of the language as well as the print features embedded in the program to control the output.

The BNF of the query language is explained in Chapter IV. However the important features and implementation limitations of the query language may be summarized as follows:

1) A query consists of a string of a maximum of 80 alphanumeric and special characters.

2) Query terms are represented by numerical values (term rank) rather than by a string of alphabetic characters. This could, of course, be changed by allowing terms to be specified in alphabetic form and having the program search for the terms in a dictionary that lists the corresponding term rank.

3) Logical operators AND, OR, NOT are represented by the signs +, -, * respectively.

4) A user can choose any level of bracket nesting provided he does not have two or more openbrackets following one another. However any number of closing brackets may follow one after another.

5) For query evaluation the terms represented by numerical values.that appear outside brackets are processed from left to right. Those that appear inside brackets are processed right to left.

Thus the order of processing terms in the query:

$$t_1-(t_2+t_3+(t_4-t_5)+t_6)-t_7-t_8\$$$

is:

$$t_5-t_4=R_1$$
$$t_6-R_1=R_2$$
$$R_2+t_3=R_3$$
$$R_3+t_2=R_4$$
$$R_4-t_1=R_5$$
$$t_7-R_5=R_6$$
$$t_8-R_6=R_7$$

6) A term must be followed by either a logical operator or one of the three symbols consisting of a blank, a closing bracket, a ($) sign.

7) End of query is indicated by a ($) sign.

8) End of query processing is marked by a (/) sign.

In view of the print options available in the simulation program the user has considerable flexibility to control the output of the search request. There are basically four print options that a user can specify. The user has the freedom to use these options either separately or in combination. Following is the list of possibilities to control the output.

a) Print OPTION(1) ... This option directs a print of the list of content terms and their respective relevance rating.

b) Print OPTION(2) ... This option directs a print of the terms in each relevant document in the data base.

c) Print OPTION(3) ... This option directs a print of the terms in each retrieved document.

d) Print OPTION(4) ... This option directs a print of the terms in each retrieved and in each relevant document in the data base.

Note: In case (c) and case (d), where the term document list is printed the number of documents to be printed is restricted to a maximum of 50 documents. Also terms 1 to 49 are not included in any of the term-document lists. This is because it is believed that the most frequent 49 terms are likely to be useless for search purposes.

When a user executes the simulation procedure the system responds with a message "PLEASE SPECIFY PRINT OPTIONS" and it indicates the input request for the four print options. Since the options are read in FORTRAN 4I1 format the user should indicate either a binary 1 to execute an option or a a binary 0 to suppress an option. The order in which the options are applied is the same as shown above in case a-d. Thus 0110 is the user specification for both of options b) and c).

After the print option input the system responds with another message "USER QUERY" to indicate that it is ready for a query request. Now the user is expected to state his query. The given query is checked for syntactic correctness, and only then does processing begin. In the case of an error in the query syntax an appropriate message is issued. Finally, the user is provided with the system response to his search request in the form of output of the document numbers of the relevant documents in the data base, the document numbers of the retrieved documents for the given search request, the precision ration, and the recall ratio.

In case the user has issued any of the print options the corresponding print option will be executed and the results will be presented in addition to the other output as described above.

A sample query session is included below to illustrate the experimentation process of the simulation program.

```
N=40000 M=5000 D=7000

FOR I=50,100,150  RI= 50

PC= .002 PN= .300 DELTA=100.0 X=.00001 C=.010

        PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
        1. USE SIMPLE INTEGERS FOR TERM VALUES
        2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
        3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
        4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
        5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
        6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                              USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 1100

USER QUERY
? 50+(100-150)$
 830 72.27454309067
 949 72.36577458697
 1343 18.30904940893
 1581 18.40028090523
 2901 93.91300185194
 2920 93.81043808767
 3415 93.7991058197
 3891 93.890337316
 4367 93.9815688123
 4386 25.48800269163
 4862 21.19186650487
 4881 55.33608580093
 4934 63.37506184989
 5247 63.30106996833
 6199 63.39230146463
 6237 63.28973770036
```

PART I

```
THE FOLLOWING ARE THE RELEVANT DOC'S:
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
  132  162  606  726  984 1001 1320 1388 1585 1732
 1904 1972 2099 2257 2393 2418 2787 2923 3425 3930
 4292 4317 4392 4955
```

PART II

THE PRECISION PR=1.00
THE RECALL RC= .50

```
DOC.NO.  132 CONTAINS THE TERMS
   50  100  150  830 1902

DOC.NO.  162 CONTAINS THE TERMS
   50   69  100 2623 3091

DOC.NO.  235 CONTAINS THE TERMS
  100  513  949 1217

DOC.NO.  520 CONTAINS THE TERMS
   50  251 3650 6880

DOC.NO.  606 CONTAINS THE TERMS
   50   72  100  867 4616

DOC.NO.  692 CONTAINS THE TERMS
   50   87  117  381  620 6255

DOC.NO.  726 CONTAINS THE TERMS
   50   54  100  150  353 1601

DOC.NO.  898 CONTAINS THE TERMS
   50  221  333  577

DOC.NO.  994 CONTAINS THE TERMS
   50  100 2200 4490 5261

DOC.NO. 1001 CONTAINS THE TERMS
   50   75  100  554 6199

DOC.NO. 1069 CONTAINS THE TERMS
  149  150  192  741 6003

DOC.NO. 1148 CONTAINS THE TERMS
   50  130 6870

DOC.NO. 1241 CONTAINS THE TERMS
   50  113  365 3415 4591

DOC.NO. 1320 CONTAINS THE TERMS
   50   69  100  150  313  965 2444 6237

DOC.NO. 1388 CONTAINS THE TERMS
   50   78  100  150  423  830  949 1850 3790 4881

DOC.NO. 1413 CONTAINS THE TERMS
  102  132 1584 4823

DOC.NO. 1560 CONTAINS THE TERMS
   50  546  841 1966 4022

DOC.NO. 1585 CONTAINS THE TERMS
   50  150  301

DOC.NO. 1653 CONTAINS THE TERMS
   50  715 1340 4934

DOC.NO. 1732 CONTAINS THE TERMS
   50   87  100  526
```

PART III

```
DOC.NO. 4367 CONTAINS THE TERMS
   50    75   190   256

DOC.NO. 4392 CONTAINS THE TERMS
   50    57    70  100 2216 5325

DOC.NO. 4639 CONTAINS THE TERMS
   50    58 4647

DOC.NO. 4955 CONTAINS THE TERMS
   50   150   362   572 6668

DOC.NO. 4980 CONTAINS THE TERMS
   50 1715 3520 4367

PLEASE SPECIFY PRINT OPTIONS
? 0)00

USER QUERY
? /
     10.120 CP SECONDS EXECUTION TIME
```

In the above sample session the print options (1) and (2) are executed. Part I of the output is the result of executing the print option (1).

The user query is interpreted as $TERM_{50}$ AND ($TERM_{100}$ OR $TERM_{150}$). Part II of the output is the standard form of output for any search request. Finally part III of the output is the result of executing the print option (2).

The system response also includes another request for a new query. The user may either continue using the same procedure as described above or may discontinue by indicating zeroes for all the four print options and a slash (/) sign for the query request.

APPENDIX B

Program listing

```
      PROGRAM DOCSIM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3)
C  SIMULATION OF DATABASE OF DOCUMENT TITLES(THESIS. PROJECT)
C  RANDOM NUMBERS ARE GENERATED FOR THE DOC. TITLES TO FORM
C  TERM DOCUMENT ASSOCIATION.THIS IS ACHIEVED BY SELECTING
C  RELEVENT AND NON-RELEVANT DOC'S BASED ON PROBABILITY
C  CRITERIA DISCUSSED IN ROUTINE RANMOD
C
C  FOR DEBUGGING PURPOSES THE DATABASE CONSIST OF THE FOLLOWING
C  SIZE
C  M=5000   : NO. OF DOCUMENT TITLES
C  N=40000  : NO. OF TERMS IN THE DATA BASE
C  D=7000   : NO. OF DIFFERENT TERMS IN THE DATABASE
C
      COMMON MTITLE,RNUM(5000),MDOC(7001)
      COMMON/BLK1/ RCKK,RATIO,SAVE1,MN(110),MATTERM(110,20),NPRINT(4)
      COMMON/BLK2/ A,NDOC,MTIT,C,X,PHI,PHISQ,DELTA,PC,PN,RFLAG,MASK1,
     1MASK2
      DIMENSION ITERM(80),STORE(200),RELDOC(110)
      INTEGER RNUM,STORE,DINDEX,RELDOC,SAVE1
      LOGICAL EFLAG,RFLAG,RERROR
      DATA SAVE1/50/,
     1MASK1,MASK2/20000000000000000000B,40000000000000000000B/
C
C  APPLYING ZIPF'S LAW THE NO. OF OCCURRENCE OF THE I-TH TERM
C  IS GIVEN AS M(I)=A*N/I WHEREBY A IS DEFINED AS A CONSTANT
C  =1/(ALOG(D)+SIE) AND SIE=.5772 DEFINED AS EULER'S CONSTANT
C
C                  OPENMS IS A SY..EM ROUTINE WHICH OPENS THE MASS STORAG. .ANDOM
C                  FILE AND INFORMS THE RECORD MANAGER THAT THE FILE IS WO.
C                  ADDRESSABLE THE ARRAY USED FOR WRITING IN AND READING O. ,IS
C                  CLEARED BY OPENMS BEFORE IT IS (ARRAY) CALLED BY EITHER .RITMS
C                  OR READMS ROUTINES.
C                  THE ACTUAL PARAMETERS HAVE THE FOLLOWING MEANING
C                  U=3 IS THE UNIT DESIGNATO WHERE THE ARRAY IS STORED AS .  .NDOM
C                  FILE
C                  IX=MDOC IS THE 1-ST WORD ADDRESS IN CM OF THE ARRAY CON. .INING
C                  INDEX
C                  LNGTH=201 IS THE MAX. LENGTH OF INDEX(NUMBER OF RECOR.S IN FILE+1
C                  T=0 MEANS THE FILE IS REFERENCED BY NUMBER INDEX
      CALL OPENMS(3,MDOC,7001,0)
      C=.01
      X=.00001
      PONE=.9
      PTWO=.8
      A=1/(ALOG(7000.)+.5772)
      NDOC=40000
      MTIT=5000
      IDTERM=7000
      PHI=(SQRT(5.)+1)/2
      PHISQ=PHI**2
      DELTA=100.
      PC=.02
      PN=.30
C                  CALCULATE THE TOTAL NUMBER OF RELEVANT DOCUMENTS IN THE DATA BASE
      LL1=0
      DO 11 II=1,MTIT
      XX=(II+X)*C*PHI
      CALL RANSET(XX)
```

```
      Y=RANF(DUM)
      IF(Y .GT. C) GOTO 11
      LL1=LL1+1
      RELDOC(LL1)=II
11    CONTINUE
      WRITE(6,994) NDOC,MTIT,IDTERM
994   FORMAT(1X,' N='I5,' M=',I4,' D=',I4,/)
      WRITE(6,996) SAVE1
996   FORMAT(1X,'FOR I=50,100,150  RI=',I3,/)
      WRITE(6,997) PC,PN,DELTA,X,C
997   FORMAT(1X,'PC=',F5.3,' PN=',F5.3,' DELTA=',F5.1,' X='F6.5,' C=',
     1F4.3,/)
      WRITE(6,999)
999   FORMAT(10X,'PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INST
     1RUCTIONS'/10X,'1. USE SIMPLE INTEGERS FOR TERM VALUES'
     2/10X,'2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
     3'/10X,'3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LO
     4GIC'/10X,'4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES
     5AND RIGHT TO LEFT INSIDE'/10X,'5. AT THE END OF EACH QUERY TYPE DOL
     6LAR SIGN ($) AND RETURN'/10X,'6. TO EXIT FROM QUERY PROCESSING
     7TYPE A SLASH SIGN (/) AND RETURN'//40X ,'USER PRINT OPTIONS'/)
      PRINT 2002
2002  FORMAT('USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:'/
     $'1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONT
     $ENT TERMS ONLY'/'2) OPTION (2) PRINT DOC.TERM LIST FOR RELEVANT DO
     $CUMENTS IN THE DATA BASE'/'3) OPTION(3).. PRINT DOC.TERM LIST FOR
     $RETRIEVED DOCUMENTS'/'4) OPTION(4).. PRINT DOC.TERM LIST FOR RETRI
     $EVED AND RELEVANT DOCUMENTS IN THE DATA BASE'
     $//'PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTI
     $ONS'/'A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORR
     $ESPONDING OPTION LOCATION'
     $/'B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE
     $ EXECUTED'/'C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTIONS
     $ TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED'
     $/'NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'
     $S AND OR 0'S.')
C
C  ROUTINE RELDOC DETERMINES THE RELEVANCY DEGREE OF A DOC.
C  WRT THE ITH TERM
C
C
C                                       THE USER QUERY IS READ IN CHARACTER FORMAT
C                                       THE MAX.SIZE OF A QUERY CAN BE EQUAL TO A CARD LENGTH
C                                       EVERY QUERY ENDS WITH A DOLLAR SIGN
C                                       TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN
      ICR=00010000000000000000B
5     PRINT 2001
2001  FORMAT(/,'PLEASE SPECIFY PRINT OPTIONS')
      READ(5,2000) NPRINT
2000  FORMAT(4I1)
      PRINT 998
998   FORMAT(/,'USER QUERY')
      READ(5,1000) ITERM
1000  FORMAT(80R1)
      IF(ITERM(1) .EQ. 1R/) STOP
      DINDEX=0
      CALL QUPARS(ITERM,STORE,DINDEX,EFLAG,RELDOC,LL1)
6     IF(EFLAG)10,7
7     IF(DINDEX .EQ. 0) GOTO 50
      IF(NPRINT(1).EQ.0.A.NPRINT(2).EQ.0.A.NPRINT(3).EQ.0 .A.
     1NPRINT(4) .EQ. 0) GOTO 609
      JK=DINDEX
      IF(DINDEX .GT. 50) JK=50
      LL3=LL1
      DO 19 IJK=1,5000
19    RNUM(IJK)=0
      DO 21 IJK=1,LL1
```

```
      MN(IJK)=0
21    RNUM(RELDOC(IJK))=RELDOC(IJK) .OR. MASK1
      DO 200 IJK=1,JK
      IF(RNUM(STORE(IJK)) .GT. 0) GOTO 200
      LL3=LL3+1
      RELDOC(LL3)=STORE(IJK)
      RNUM(RELDOC(LL3))=STORE(IJK) .OR. MASK1
200   CONTINUE
      J=LL3
      DO 20 I=50,7000
      RATIO=-1.
      IF(I.EQ.10.OR.I.EQ.50.OR.I.EQ.100.OR.I.EQ.150) RATIO=SAVE1
      RFLAG=.F.
      RERROR=.T.
      IF(RATIO .GE. 0) RFLAG=.T.
      MTITLE=A*NDOC/I
      IF(MTITLE .LT. 1) MTITLE=1
20    CALL RANMOD(I,RELDOC,LL1,STORE,J,RERROR)
      DO 6900 JJ=1,LL3
6900  RNUM(JJ)=RELDOC(JJ)
609   LL=DINDEX-1
      LL2=LL1-1
      IF(DINDEX .NE. 1) CALL SORTING(STORE,LL)
      IF(LL1 .EQ. 0) GOTO 1006
      IF(LL1 .NE. 1) CALL SORTING(RELDOC,LL2)
      WRITE(6,1004)(RELDOC(JJ),JJ=1,LL1)
1004  FORMAT(/,'THE FOLLOWING ARE THE RELEVANT DOC'S:'/10(1X,I4))
1006  WRITE(6,1001)(STORE(I),I=1,DINDEX)
1001  FORMAT(/,'THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:'
     1/10(1X,I4))
      CALL MEASUR(STORE,RELDOC,LL1,DINDEX)
      IF(NPRINT(2) .EQ. 1) GOTO 201
      IF(NPRINT(3) .EQ. 1) GOTO 202
      IF(NPRINT(4) .EQ. 1) GOTO 203
      GOTO 5
201   J=LL1
      GOTO 204
203   J=LL3
204   KK1=1
208   DO 6901 JJ=KK1,J
      LK=MN(JJ)
6901  WRITE(6,6902) RNUM(JJ),(MATTERM(JJ,LL),LL=1,LK)
6902  FORMAT(/,'DOC.NO. ',I4,' CONTAINS THE TERMS'/10(1X,I4))
      GOTO 5
202   DO 206 IJK=1,JK
      DO 205 JKL=1,LL1
      IF(STORE(IJK) .EQ. RNUM(JKL)) GOTO 207
205   CONTINUE
      GOTO 206
207   LK=MN(JKL)
      WRITE(6,6903) RNUM(JKL),(MATTERM(JKL,LL),LL=1,LK)
6903  FORMAT(/,'DOC.NO. ',I4,' CONTAIN THE TARMS'/10(1X,I4))
206   CONTINUE
      IF(LL1 .EQ. LL3) GOTO 5
      KK1=LL1+1
      J=LL3
      GOTO 208
10    WRITE(6,1002)
1002  FORMAT(1X,'REENTER THE QUERY')
      GOTO 5
50    WRITE(6,1003)
1003  FORMAT(1X,'NO DOCUMENTS ARE SELECTED')
      GOTO 5
60    WRITE(6,1005)
1005  FORMAT(1X,'RATIO R(I) IS NOT IN RANGE: R(I) IS .GT. M/M(I)')
      GOTO 5
```

```
      END
      SUBROUTINE QUPARS(ITERM,STORE,DINDEX,EFLAG,RELDOC,LL1)
      COMMON MTITLE,RNUM(5000),MDOC(7001)
      COMMON/BLK1/ RCKK,RATIO,SAVE1,MN(110),MATTERM(110,20),NPRINT(4)
      COMMON /BLK2/ A,NDOC,MTIT,C,X,PHI,PHISQ,DELTA,PC,PN,RFLAG,MASK1,MASK2
      INTEGER COUNT,TSTACK,OSTACK,BOPEN,STORE,PVAL,OP,DINDEX,RNUM,OPP,
     1RELDOC,SAVE1,STIT,AUX,DINDEX1
      DIMENSION TSTACK(25),OSTACK(25),STORE(1),ITERM(1),RELDOC(1),
     1STIT(25),AUX(200)
      LOGICAL FLAG,EFLAG,EXIT,RFLAG,RERROR,FFLOP,SETF
C
C                                 THIS ROUTINE PARSES A GIVEN QUERY.AFTER SCANNING THE QUERY
C                                 TERM VALUES AND AN OPERATOR IS PASSED ON AS PARAMETERS TO
C                                 A ROUTINE QUPROC WHERE THE DOCUMENTS ARE SELECTED ACCORDING
C                                 TO THE QUESTION LOGIC.THE SCANNING IS DONE LEFT TO RIGHT OUTSIDE
C                                 THE PARENTHESIS ELSE RIGHT TO LEFT
C
      INPO=OP=1
      IP=NN=BOPEN=KFLAG=NEG=DINDEX1=0
      SETF=FFLOP=EFLAG=EXIT=.FALSE.
      OSTACK(OP)=1R#
C
C                                 NUMBERS HAVE DISPLAY CODE BETWEEN 33B - 44B
C                                 + ,- ,* HAVE DISPLAY CODES 45B ,46B ,47B RESPECTIVELY
C                                 ( AND ) HAVE DISPLAY CODE 51B AND 52B RESPECTIVELY
C                                 THE PARSER IN PRINCIPLE CHECKS FOR EITHER NUMBER(0-9),OPERATOR
C                                 (+,-,*)OR PARENTHESIS'(',')',IF NONE OF THESE IS PRESENT AND
C                                 ALSO NOT ANY OF DELIMETER CHARACTERS COMMENTED ABOVE IS THERE
C                                 THE ROUTINE RETURNS BACK TO MAIN WITH AN ERROR FLAG BIT ON.
C                                 OTHERWISE THE PARSER SCANS THE QUERY LEFT TO RIGHT,USES TWO
C                                 STACKS NAMED TSTACK FOR STORING THE TERM VALUES AND OSTACK FOR
C                                 STORING THE OPERATORS OR PARENTHESIS.THE ROUTINE QUPROC IS
C                                 ONLY CALLED WHEN TSTACK HAS ATLEAST TWO VALUES AND OSTACK POINTER
C                                 HAS ATLEAST THE VALUE 2(SINCE OSTACK(1)=# WHICH IS EOQ PROCESSING
C                                 MEANING THE STACKS ARE EMPTY.
      DO 40 K=1,80
      IF(ITERM(K) .EQ. 1R#) GOTO 700
      IF(ITERM(K) .EQ. 1R ) GOTO 40
      IF(KFLAG .EQ. 0) GOTO 100
      KFLAG = KFLAG-1
      GOTO 40
100   IF(ITERM(K) .GE. 33B .AND. ITERM(K) .LE. 44B) GOTO 200
C
C  THIS PATH OF PROGRAM INDICATES THAT EITHER OPERATOR + (AND)
C  -(OR), OR PARENTHESIS HAS BEEN FOUND.
C
      IF(ITERM(K) .GE. 45B .AND. ITERM(K) .LE. 47B) GOTO 250
      IF(ITERM(K) .EQ. 51B .OR. ITERM(K) .EQ. 52B) GOTO 300
      EFLAG=.T.
      GOTO 500
C
C  LABEL 800 RETURNS THE PROGRAM FLOW TO THE MAIN FOR ERROR
C  RHANDLING AND REINITIALISATION
C
300   IF(ITERM(K) .EQ. 51B) GOTO 350
      IF(SETF)33,34
33    INPO=INPO+1
34    IF(INPO .GT. 1) NEG=-1
      OP=OP+1
      OSTACK(OP)=ITERM(K)
      GOTO 400
350   BOPEN=BOPEN+1
      IF(FFLOP) 351,352
351   DINDEX1=0
      INPO=1
352   FFLOP=.F.
```

```
      OP=OP+1
      OSTACK(OP)=ITERM(K)
      GOTO 40
250   OP=OP+1
      OSTACK(OP)=ITERM(K)
400    FLAG=.FALSE.
      IF(IP .GE. 2 .AND. BOPEN .EQ. 0) GOTO 375
      IF(ITERM(K) .EQ. 52B) GOTO 385
      GOTO 40
385   IOP=OP
750   IOP=IOP-1
      IF(OSTACK(IOP) .EQ. 51B) GOTO 395
      FLAG=.T.
      OPP=IOP
      GOTO 275
395   OP=IOP-1
      BOPEN=BOPEN-1
      IF(BOPEN .NE. 0) GOTO 40
      SETF=.T.
      NEG=0
      GOTO 40
375   IF(OSTACK(OP-1) .GE. 45B .AND. OSTACK(OP-1) .LE. 47B) GOTO 425
      OPP=OP
      GOTO 275
425   OPP=OP-1
275   K1=TSTACK(IP)
      K2=TSTACK(IP-1)
C
C                              THE ROUTINE QUPROC IS CONSEQUITIVELY CALLED PROCESSING THE
C                              QUERY EVERY TIME PARTIALLY UNTIL THE STACKS ARE EMPTY.AFTER
C                              THE LOGIC PROCESSING THE TSTACK POINTER IS REDUCED BY ONE
C                              AND THIS LOCATION IS ZEROED.(NOTE:THE RESULT IS ALREADY STORED
C                              IN THE ARRAY STORE,WHICH IS THEN USED FOR THE NEXT PHASE)
C
      CALL QUPROC(DINDEX,K1,K2,OSTACK,STORE,OPP,IP,STIT,
     1FFLOP,NEG,DINDEX1,AUX)
      IF(FLAG)397,398
397   IP=IP-1
      TSTACK(IP)=0
      GOTO 750
398   OP=OP-1
      IP=IP-1
      TSTACK(IP)=0
      OSTACK(OP)=ITERM(K)
      IF(EXIT)500,40
200   IF((ITERM(K+1) .GE. 33B .AND. ITERM(K+1) .LE. 44B) .AND.
     1(ITERM(K+2) .GE. 33B .AND. ITERM(K+2) .LE. 44B) .AND.
     2(ITERM(K+3) .GE. 33B .AND. ITERM(K+3) .LE. 44B)) GOTO 196
      IF((ITERM(K+1) .GE. 33B .AND. ITERM(K+1) .LE. 44B) .AND.
     1(ITERM(K+2) .GE. 33B .AND. ITERM(K+2) .LE. 44B)) GOTO 201
      IF(ITERM(K+1) .GE. 33B .AND. ITERM(K+1) .LE. 44B) GOTO 202
C
C                              HERE THE INTEGER CHARACTERS ARE CONVERTED INTO INTEGER NUMBERS
C                              BY DECODING THEM USING THE PROPER I-FORMAT.
C
      COUNT=10
      DECODE(COUNT,1005,ITERM(K))I1
1005  FORMAT(9X,I1)
      GOTO 60
196   KFLAG=3
      COUNT=40
      DECODE(40,1002,ITERM(K))I1,I2,I3,I4
1002  FORMAT(4(9X,I1))
      GOTO 79
202   KFLAG=1
      COUNT=20
```

```
      DECODE(COUNT,1006,ITERM(K))I1,I2
1006  FORMAT(2(9X,I1))
      GOTO 70
201   KFLAG=2
      COUNT=30
95    DECODE(COUNT,1001,ITERM(K))I1,I2,I3
1001  FORMAT(3(9X,I1))
      GOTO 80
60    I=I1
      GOTO 90
70    I=I1*10+I2
      GOTO 90
79    I=I1*1000+I2*100+I3*10+I4
      GOTO 90
80    I=I1*100+I2*10+I3
90    MTITLE=A*NDOC/I
      RATIO=-1.
      IF(I.EQ.10.OR.I.EQ.50.OR.I.EQ.100.OR.I.EQ.150) RATIO=SAVE1
      RFLAG=RERROR=.F.
      IF(MTITLE .LT. 1) MTITLE=1
      IF(RATIO .LT. 0) GOTO 810
      IF(RATIO .LE. MTIT/MTITLE) GOTO 800
      RATIO=MTIT/MTITLE
      WRITE(6,1010) I,RATIO
1010  FORMAT(1X,'GIVEN RATIO FOR TERM ',I4,' IS .GT. MTIT/MTITLE'
     1/1X,'THE NEW VALUE OF RATIO IS ',F5.2)
800   RFLAG=.T.
810   CALL RANMOD(I,RELDOC,LL1,STORE,J,RERROR)
      IP=IP+1
      TSTACK(IP)=I
      STIT(IP)=MTITLE
      GOTO 40
700   IF(OP .EQ. 1) RETURN
      EXIT=.T.
      GOTO 400
40    CONTINUE
500   RETURN
      END
C
C  *********************************************************
C
      SUBROUTINE QUPROC(DINDEX,K1,K2,OSTACK,STORE,OPP,IP,STIT,
     1FFLOP,NEG,DINDEX1,AUX)
      COMMON MTITLE,RNUM(5000),MDOC(7001)
      DIMENSION OSTACK(1),STORE(1),STIT(1),AUX(1)
      INTEGER DINDEX,OSTACK,STORE,OPP,RNUM,STIT,AUX,DINDEX1
      LOGICAL IFLAG,NFLAG,EFLAG,RCHECK,FFLOP
C
C                  THIS ROUTINE DO THE LOGICAL PROCESSING OF THE QUERY STORES THE
C                  RESULT IN ARRAY STORE AND RETURN BACK TO THE PARSER (QUPARS)
C
C                  THE INTEGERS K1,K2 ARE USED AS TERM VALUES,INCASE OF 1-ST
C                  PASS ELSE ONE OF THEM IS ZEROED(SINCE THE RESULT OF PREVIOUS
C                  LOGIC EXIST IN STORE)THE OTHER ONE IS USED AS A NEW TERM VALUE
C                  JOINED WITH A LOGICAL OPERATOR FOR THE NEXT INTERMEDIATE RESULT
C                  THIS PROCESS GOES ON UNTIL EOS(END OF STACK) IS SENSED BY THE PARSER.
C
C                  MTITLE GIVES THE NO. OF DOC'S CONTAINING THE I-TH TERM
C                  MDOC GIVES THE ACTUAL DOC. NO.'S CONTAINING THE I-TH TERM
C                  STORE IS THE ARRAY WHICH CONTAINS THE RESULT OF PREVIOUS
C                  LOGIC PROCESSING.
C                  DINDEX IS USED AS A POINTER FOR STORE GIVING THE NO. OF ELEMENTS
C                  PRESENT IN STORE.
C                  THE OR(-) LOGIC IS PROCESSED JUST BEFORE LABEL 180
C                  THE AND (+) LOGIC IS PROCESSED FROM LABEL 180 ONWARDS
C                  THE NOT (*) LOGIC IS PROCESSED FROM LABEL 270 ONWARDS
C
```

```
C
      NFLAG=EFLAG=RCHECK=.F.
      IF(NEG .EQ. -1) GOTO 500
      IF(K1 .EQ. 0 .A. K2 .EQ. 0) GOTO 510
      IF(K1 .EQ. 0 .OR. K2 .EQ. 0) GOTO 40
      J1=MTITLE
      J2=STIT(IP-1)
      IF(OSTACK(OPP) .EQ. 1R*) GOTO 5
      IF(J1 .GT. J2) GOTO 20
 5    CALL READMS(3,RNUM,J2,K2)
      DO 10 INDEX =1,J2
      DINDEX=DINDEX+1
 10   STORE(DINDEX)=RNUM(DINDEX)
      K2=0
      EFLAG=.T.
      GOTO 40
 20   CALL READMS(3,RNUM,J1,K1)
      DO 30 INDEX =1,J1
      DINDEX=DINDEX+1
 30   STORE(DINDEX)=RNUM(DINDEX)
      K1=0
      RCHECK=.T.
 40   IF(OSTACK(OPP) .EQ. 1R*) GOTO 270
      IF(K1 .EQ. 0) GOTO 55
      J1=MTITLE
      L=K1
      GOTO 57
 55   J1=STIT(IP-1)
      L=K2
C                                    THIS SYSTEM ROUTINE TRANSMIT DATA FROM MASS STORAGE TO CM.
C                                    THE CALL READMS SELECTS RECORD NO. L OF THE ARRAY FILE RNUM
C                                    AND TAKES J1 NO. OF ELEMENTS OF RNUM BELONGING TO RECORD NO. L
C                                    IN THIS CASE RECORD NO.IS CONSIDER TO BE TERM.NO.(L IS TERM NO.)
C
 57   CALL READMS(3,RNUM,J1,L)
      IF(DINDEX .NE. 0) GOTO 65
      IF(OSTACK(OPP) .EQ. 1R+) RETURN
      DO 60 I=1,J1
      DINDEX=DINDEX+1
 60   STORE(DINDEX)=RNUM(I)
      RETURN
 65   IF(DINDEX .GT. J1) GOTO 70
      JJ1=DINDEX
      JJ2=J1
      IFLAG=.F.
      GOTO 80
 70   JJ1=J1
      JJ2=DINDEX
      IFLAG=.T.
 80   IF(OSTACK(OPP) .EQ. 1R+) GOTO 180
      IF(IFLAG) 100,140
 100  DO 130 L1=1,JJ1
      DO 120 L2=1,JJ2
      IF(STORE(L2) .EQ. RNUM(L1)) GOTO 130
 120  CONTINUE
      DINDEX=DINDEX+1
      STORE(DINDEX)=RNUM(L1)
 130  CONTINUE
      RETURN
 140  DINDEX=JJ2
      DO 160 L1=1,JJ1
      DO 150 L2=1,JJ2
      IF(STORE(L1) .EQ. RNUM(L2)) GOTO 160
 150  CONTINUE
      DINDEX=DINDEX+1
      STORE(DINDEX)=STORE(L1)
 160  CONTINUE
```

```
      DO 170 L2=1,JJ2
170   STORE(L2)=RNUM(L2)
      RETURN
180   IF(IFLAG) 190,230
190   DO 210 L1=1,JJ1
      DO 200 L2=1,JJ2
      IF(STORE(L2) .EQ. RNUM(L1)) GOTO 210
200   CONTINUE
      RNUM(L1)=0
C                                THE OPTIONAL PARAMETER (HERE 5-TH) R=1 INDICATES THAT THE
C                                ARRAY FILE CAN BE REWRITTEN AT THE SAME LOCATION PROVIDED
C                                THE NEW STRING SHOULD BE .LE. THE OLD STRING SIZE.OTHERWISE
C                                A FATAL ERROR IS GIVEN.IN THIS PARTICULAR CASE EVERYTIME RNUM(L1)
C                                THE LOCATION L1(1 TO JJ1)OF ARRAY RNUM AND RECORD NUMBER L IS
C                                REWRITTEN TO 0.
      CALL WRITMS(3,RNUM,1,L,1)
210   CONTINUE
      DINDEX=0
      DO 220 L1=1,JJ1
      IF(RNUM(L1) .EQ. 0) GOTO 220
      DINDEX=DINDEX+1
      STORE(DINDEX)=RNUM(L1)
220   CONTINUE
      RETURN
230   DO 250 L1=1,JJ1
      DO 240 L2=1,JJ2
      IF(STORE(L1) .EQ. RNUM(L2)) GOTO 250
240   CONTINUE
      STORE(L1)=0
250   CONTINUE
      DINDEX=0
      DO 260 L1=1,JJ1
      IF(STORE(L1) .EQ. 0) GOTO 260
      DINDEX=DINDEX+1
      STORE(DINDEX)=STORE(L1)
260   CONTINUE
      RETURN
270   IF(EFLAG) 275,310
275   CALL READMS(3,RNUM,J1,K1)
      DO 290 I=1,J1
      DO 280 J=1,J2
      IF(STORE(J) .EQ. RNUM(I)) GOTO 285
280   CONTINUE
      GOTO 290
285   STORE(J)=0
290   CONTINUE
      DINDEX=0
      DO 300 I=1,J2
      IF(STORE(I) .EQ. 0) GOTO 300
      DINDEX=DINDEX+1
      STORE(DINDEX)=STORE(I)
300   CONTINUE
      RETURN
310   IF(K1 .NE. 0 .AND. DINDEX .EQ. 0) RETURN
      IF(K1 .EQ. 0) GOTO 410
      L=K1
      J1=DINDEX
      J2=MTITLE
      CALL READMS(3,RNUM,J2,L)
      GOTO 420
410   L=K2
      J2=MTITLE
      CALL READMS(3,RNUM,J2,L)
      IF(DINDEX .NE. 0) GOTO 415
      DO 416 I=1,J2
      DINDEX=DINDEX+1
```

```
      STORE(DINDEX)=RNUM(DINDEX)
416   CONTINUE
      RETURN
415   J1=DINDEX
      NFLAG=.T.
420   DO 440 I=1,J1
      DO 430 J=1,J2
      IF(STORE(I) .EQ. RNUM(J)) GOTO 435
430   CONTINUE
      GOTO 440
435   IF(NFLAG) 436,437
436   RNUM(J)=0
      CALL WRITMS(3,RNUM,1,L,1)
      GOTO 440
437   STORE(I)=0
440   CONTINUE
      DINDEX=0
      IF(NFLAG) 450,470
450   DO 460 I=1,J2
      IF(RNUM(I) .EQ. 0) GOTO 460
      DINDEX=DINDEX+1
      STORE(DINDEX)=RNUM(I)
460   CONTINUE
      RETURN
470   DO 480 I=1,J1
      IF(STORE(I) .EQ. 0) GOTO 480
      DINDEX=DINDEX+1
      STORE(DINDEX)=STORE(I)
480   CONTINUE
      RETURN
500   IF(K1 .EQ. 0 .OR. K2 .EQ. 0) GOTO 520
      J1=MTITLE
      J2=STIT(IP-1)
      CALL READMS(3,RNUM,J1,K1)
      DO 530 INDEX=1,J1
530   AUX(INDEX)=RNUM(INDEX)
      DINDEX1=J1
      K1=0
520   IF(K1 .EQ. 0) GOTO 537
      J1=MTITLE
      L=K1
      GOTO 540
537   J1=STIT(IP-1)
      L=K2
540   CALL READMS(3,RNUM,J1,L)
      IF(OSTACK(OPP) .EQ. 1R*) GOTO 550
      IF(OSTACK(OPP) .EQ. 1R-) GOTO 560
      IF(DINDEX1 .EQ. 0) RETURN
      DO 570 JJ1=1,DINDEX1
      DO 575 JJ2=1,J1
      IF(AUX(JJ1) .EQ. RNUM(JJ2)) GOTO 570
575   CONTINUE
      AUX(JJ1)=0
570   CONTINUE
      JJ2=0
      DO 580 JJ1=1,DINDEX1
      IF(AUX(JJ1) .EQ. 0) GOTO 580
      JJ2=JJ2+1
      AUX(JJ2)=AUX(JJ1)
580   CONTINUE
      DINDEX1=JJ2
      RETURN
550   IF(DINDEX1 .EQ. 0) GOTO 590
      DO 630 JJ1=1,J1
      DO 640 JJ2=1,DINDEX1
      IF(AUX(JJ2) .EQ. RNUM(JJ1)) GOTO 650
```

```
640     CONTINUE
        GOTO 630
650     RNUM(JJ1)=0
630     CONTINUE
        DINDEX1=0
        DO 660 JJ2=1,JJ1
        IF(RNUM(JJ2) .EQ. 0) GOTO 660
        DINDEX1=DINDEX1+1
        AUX(DINDEX1)=RNUM(JJ2)
660     CONTINUE
        RETURN
560     IF(DINDEX1 .EQ. 0) GOTO 590
        JJA=DINDEX1
        DO 600 JJ2=1,J1
        DO 610 JJ1=1,JJA
        IF(AUX(JJ1) .EQ. RNUM(JJ2)) GOTO 600
610     CONTINUE
        DINDEX1=DINDEX1+1
        AUX(DINDEX1)=RNUM(JJ2)
600     CONTINUE
        RETURN
590     DO 620 JJ1=1,J1
620     AUX(JJ1)=RNUM(JJ1)
        DINDEX=J1
        RETURN
510     FFLOP=.T.
        IF(DINDEX .EQ. 0 .A. DINDEX1 .EQ. 0) RETURN
        IF(OSTACK(OPP) .EQ. 1R*) GOTO 700
        IF(OSTACK(OPP) .EQ. 1R-) GOTO 750
        IF(DINDEX .EQ. 0 .OR. DINDEX1 .EQ. 0) RETURN
        DO 760 JJ1=1,DINDEX
        DO 770 JJ2=1,DINDEX1
        IF(STORE(JJ1) .EQ. AUX(JJ2)) GOTO 760
770     CONTINUE
        STORE(JJ1)=0
760     CONTINUE
        JJ2=0
        DO 780 JJ1=1,DINDEX
        IF(STORE(JJ1) .EQ. 0) GOTO 780
        JJ2=JJ2+1
        STORE(JJ2)=STORE(JJ1)
780     CONTINUE
        DINDEX=JJ2
        RETURN
750     IF(DINDEX1 .EQ. 0) RETURN
        IF(DINDEX .EQ.0) GOTO 790
        JJA=DINDEX
        DO 800 JJ1=1,DINDEX1
        DO 810 JJ2=1,JJA
        IF(STORE(JJ2) .EQ. AUX(JJ1)) GOTO 800
810     CONTINUE
        DINDEX=DINDEX+1
        STORE(DINDEX)=AUX(JJ1)
800     CONTINUE
        RETURN
790     DO 820 DINDEX=1,DINDEX1
820     STORE(DINDEX)=AUX(DINDEX)
        RETURN
700     IF(DINDEX .EQ. 0 .OR. DINDEX1 .EQ. 0) RETURN
        DO 830 JJ1=1,DINDEX
        DO 840 JJ2=1,DINDEX1
        IF(STORE(JJ1) .EQ. AUX(JJ2)) GOTO 835
840     CONTINUE
        GOTO 830
835     STORE(JJ1)=0
830     CONTINUE
```

```
      JJA=DINDEX
      DINDEX=0
      DO 850 JJ1=1,JJA
      IF(STORE(JJ1) .EQ. 0) GOTO 850
      DINDEX=DINDEX+1
      STORE(DINDEX)=STORE(JJ1)
850   CONTINUE
      RETURN
      END
      SUBROUTINE RANMOD(I,RELDOC,LL1,STORE,J,RERROR)
      COMMON MTITLE,RNUM(5000),MDOC(7001)
      COMMON/BLK1/RCKK,RATIO,SAVE1,MN(110),MATTERM(110,20),NPRINT(4)
      COMMON/BLK2/A,NDOC,M,C,X,PHI,PHISQ,DELTA,PC,PN,RFLAG,MASK1,MASK2
      DIMENSION RELDOC(1),STORE(1),LOCAL(100)
      INTEGER RNUM,RIMI,RICMI,RELDOC,STORE,SAVE1
      LOGICAL RFLAG,RERROR,IFLAG
      DATA LOCAL/100*0/
C
C  XX IS ASSIGNED TO BE THE SEED FOR PSEUDO RANDOM NO.
C  WHICH INTURN DETERMONE THE WEIGHT ASSIGNED TO THE I-TH TERM
C
      KLM=IJJ=M1=M2=0
      IF(RFLAG) 105,100
100   XX=(I+PC)*C*PHISQ
      CALL  RANSET(XX)
      RI=RANF(DUM)
      IF(RI .GE. 0. .AND. RI .LE. PC) GOTO 103
      IF(RI .GT. PC .AND. RI .LE. PC+PN) GOTO 104
      XX=(I+X)*PHI
      CALL RANSET(XX)
      INDEX=1
      Y=RANF(DUM)
      KK=Y*5000+1
      IF(RERROR) 111,106
111   IFLAG=.F.
      INDEX=0
      GOTO 6000
106   RNUM(1)=KK
6001  IF(MTITLE .EQ. 1) GOTO 112
      IFLAG=.T.
107   Y=RANF(DUM)
      KK=Y*5000+1
      IF(RERROR) 6000,6003
6000  KK2=KK .OR. MASK1
      IF(RNUM(KK) .LT. 0) GOTO 102
      IF(KK2 .EQ. RNUM(KK)) GOTO 6004
      KLM=KLM+1
      LOCAL(KLM)=KK
      RNUM(KK)=KK .OR. MASK2
      GOTO 6005
6004  RNUM(KK)=RNUM(KK) .OR. MASK2
      IJJ=IJJ+1
      IF(IJJ-J) 6005,6008
6005  INDEX=INDEX+1
      IF(IFLAG) 102,6001
6003  DO 119 LKL=1,INDEX
      IF(RNUM(LKL) .EQ. KK) GOTO 102
119   CONTINUE
      INDEX=INDEX+1
      RNUM(INDEX)=KK
102   IF(MTITLE .EQ. INDEX)GOTO 112
      GOTO 107
112   IF(RERROR) 109,101
101   CALL WRITMS(3,RNUM,MTITLE,I)
      RETURN
109   IF(IJJ .GT. 0 .OR. KLM .GT. 0) GOTO 6008
```

```
      RETURN
103   XX=(I+DELTA)*C*PHISQ
      CALL RANSET(XX)
      Y=RANF(DUM)
      RATIO=1+Y*DELTA
      IF(NPRINT(1) .EQ. 1) PRINT*,I,RATIO
      IF(RATIO .GT. M/MTITLE) RATIO=M/MTITLE
      GOTO 105
104   RATIO=1
105   RR1=C*RATIO*MTITLE
      RICMI=RR1
      MIMRICM=MTITLE*(1-C*RATIO)
      RRR1=RR1-RICMI
      IF(RRR1 .EQ. 0.0) GOTO 7
      IF(RRR1-.5)11,11,13
11    MIMRICM=MIMRICM+1
      GOTO 7
13    RICMI=RICMI+1
C
C
7     XX=(I+X)*PHI
      CALL RANSET(XX)
C
      INDEX=0
      JJ1=1
10      Y=RANF(DUM)
      KK=Y*5000+1
      CALL RANGET(Y)
C
C  CHECK FOR REPEATITIONS OF DOCUMENT NO'S AND IGNORE MULTIPLE
C  OCCURRENCES
C
      IF(KERROR) 22,20
22    IF(RNUM(KK) .LT. 0) GOTO 65
      GOTO 30
20    IF(INDEX .EQ. 0) GOTO 30
      DO 25 JJ2=1,INDEX
      IF(KK .EQ. RNUM(JJ2)) GOTO 65
25    CONTINUE
C  IF RC(DOC.NO.) .LE. C THE DOCUMENT IS RELEVANT ELSE NON-RELEVANT
C
30    XX=(KK+X)*C*PHI
      CALL RANSET(XX)
      RCKK=RANF(DUM)
      IF(RCKK .LE. C) GOTO 50
      CALL RANSET(Y)
      M2=M2+1
85    IF(M2 .LE. MIMRICM) GOTO 60
      GOTO 65
50    M1=M1+1
      CALL RANSET(Y)
80    IF(M1 .GT. RICMI) GOTO 65
      IF(KERROR) 81,69
81    KK2=KK .OR. MASK1
      IF(KK2 .EQ. RNUM(KK)) GOTO 7004
      KLM=KLM+1
      LOCAL(KLM)=KK
      RNUM(KK)=KK .OR. MASK2
      GOTO 7005
7004  RNUM(KK)=RNUM(KK) .OR. MASK2
      IJJ=IJJ+1
      IF(IJJ-J) 7005,6008
7005  INDEX=INDEX+1
      GOTO 65
60    IF(KERROR) 81,69
```

```
69        INDEX = INDEX+1
       RNUM(INDEX)=KK
C                                    WRITMS ROUTINE TRANSMIT DATA FROM CM TO MASS STORAGE DEVICE
C                                    AT THE SPECIFIED LOCATION.
C                                    3 IS THE UNIT DESIGNATOR(NOTE! U=3 SHOULD BE THE SAME AS OPENMS
C                                    TO OPEN THE MASS.STORAGE RANDOM FILE.
C                                    RNUM IS THE ARRAY FILE WHWRE THE RECORD IS STORED
C                                    MTITLE ARE THE NUMBE OF ELEMENTS IN THE RECORD I
C                                    I INDICATES THE RECORD NUMBER.(ANALOGOUS TO TERM NO.)
65     IF(INDEX .EQ. MTITLE) GOTO 70
       JJ1=JJ1+1
       GOTO 10
70     IF(RERROR)6008,73
73     CALL WRITMS(3,RNUM,MTITLE,I)
       RETURN
6008   DO 6015 KLM1=1,KLM
       RNUM(LOCAL(KLM1))=0
6015   LOCAL(KLM1)=0
7800   DO 7900 NNO=1,J
       IF(RNUM(RELDOC(NNO)) .GE. 0) GOTO 7900
       MN(NNO)=MN(NNO)+1
       MATTERM(NNO,MN(NNO))=I
       RNUM(RELDOC(NNO))=RNUM(RELDOC(NNO)) .AND. -MASK2
7900   CONTINUE
       RETURN
       END
       SUBROUTINE SORTING(SARRAY,LL)
       DIMENSION SARRAY(1)
       INTEGER SARRAY
25     ICOUNT=0
       DO 30 L=1,LL
       IF(SARRAY(L) .LE. SARRAY(L+1)) GOTO 30
       NUM=SARRAY(L)
       SARRAY(L)=SARRAY(L+1)
       SARRAY(L+1)=NUM
       ICOUNT=1
30     CONTINUE
       IF(ICOUNT .EQ. 1) GOTO 25
       RETURN
       END
       SUBROUTINE MEASUR(STORE,RELDOC,LL1,DINDEX)
       COMMON/BLK2/ A,NDOC,MTIT,C,X,PHI,PHISQ,DELTA,PC,PN,RFLAG,MASK1,MASK2
       DIMENSION STORE(1),RELDOC(1)
       INTEGER STORE,RELDOC,DINDEX
       IPO=0
       TOTAL=DINDEX
       CC=LL1
       DO 15 LM2=1,LL1
       DO 10 LM1=1,DINDEX
       IF(RELDOC(LM2) .EQ. STORE(LM1)) GOTO 14
10     CONTINUE
       GOTO 15
14     IPO=IPO+1
15     CONTINUE
       PR=IPO/TOTAL
       RCC=IPO/CC
       WRITE(6,20) PR,RCC
20     FORMAT(1X,'THE PRECISION PR='F4.2/1X,'THE RECALL RC=',F4.2)
       RETURN
       END
```

APPENDIX C

Results for on-line user queries

```
N=40000 M=5000 D=7000

FOR I=50,100,150  RI= 50

PC= .020 PN= .300 DELTA=100.0 X=.00001 C=.010

        PLEASE GIVE YOUR QUERY ACCORDING TO THE FOLLOWING INSTRUCTIONS
        1. USE SIMPLE INTEGERS FOR TERM VALUES
        2. FOR LOGICAL OPERATORS (AND),(OR),(NOR) USE (+),(-),(*) RESPECTIVELY
        3. USE PARENTHESIS FOR HIERARCHICAL ORDERING OF QUESTION LOGIC
        4. LEFT TO RIGHT PROCESSING IS DONE OUTSIDE PARENTHESES AND RIGHT TO LEFT INSIDE
        5. AT THE END OF EACH QUERY TYPE DOLLAR SIGN ($) AND RETURN
        6. TO EXIT FROM QUERY PROCESSING TYPE A SLASH SIGN (/) AND RETURN

                        USER PRINT OPTIONS

USERS HAVE FOUR PRINT OPTIONS TO CONTROL THE OUTPUT:
1) OPTION (1) .. PRINT TERM NUMBERS AND RELEVANCE RATING FOR CONTENT TERMS ONLY
2) OPTION (2) .. PRINT DOC.TERM LIST FOR RELEVANT DOCUMENTS IN THE DATA BASE
3) OPTION (3) .. PRINT DOC.TERM LIST FOR RETRIEVED DOCUMENTS
4) OPTION (4) .. PRINT DOC.TERM LIST FOR RETRIEVED AND RELEVANT DOCUMENTS IN THE DATA BASE

PLEASE FOLLOW THE INSTRUCTIONS TO EXECUTE ANY OF THE PRINT OPTIONS
A. AN OPTION IS EXECUTED BY INDICATING A BINARY 1 AT THE CORESPONDING OPTION LOCATION
B. IF A BINARY 0 IS FOUND THEN THE PARTICULAR OPTION WILL NOT BE EXECUTED
C. AT THE FIRST INPUT REQUEST SPECIFY ONE'S FOR OPTION TO BE EXECUTED AND ZERO FOR OPTIONS SKIPPED
NOTE: THERE SHOULD NOT BE ANY EMBEDDED BLANKS BUT A STRING OF 1'S AND OR 0'S.

PLEASE SPECIFY PRINT OPTIONS
? 1:00

USER QUERY
? 50+100+150$   (Q1)
 159 69.28916223046
 193 1.741425451153
 278 9.033792340538
 307 86.24437670938
 308 85.08208757152
 427 85.26455056412
 477 85.7660371175
 495 17.71681378481
 543 35.90240193639
 546 51.12575618926
 673 54.56247216845
 674 25.00918067419
 774 93.90166958397
 791 94.65370366475
 793 25.10041217049
 893 93.99290108027
 910 94.74493516105
 912 25.19164366679
 1012 94.08413257657
 1014 24.53084108231
 1029 94.83616665735
 1131 89.68543262553
 1133 4.908786878403
 1353 62.81682302968
 1368 63.773984639
 1397 63.56885711046
 1402 64.52601871979
 1572 31.50418441091
```

1591	62.90805452598
1606	32.2562184917
1625	63.66008860676
1640	64.61725021609
1644	63.45496107822
1810	31.59541590721
1825	32.55257751654
1829	62.99928602220
1844	63.9564476316
1863	63.75132010306
1878	64.70848171239
2067	63.09051751858
2082	64.0476791279
2086	94.49438763365
2101	63.84255159936
2135	64.59450560015
2305	63.18174901488
2320	64.1309106242
2324	94.58561912995
2339	63.93370309566
2741	1.06080017314
2760	32.57524205248
2779	32.47267820821
2798	32.37011452394
2809	33.42983909753
2817	32.26755075967
2828	33.32722613326
2836	63.773984639
2844	33.22471236899
3236	32.66647354878
3255	32.56390978451
3266	2.014637514493
3274	32.46134602024
3285	33.52107139383
3293	32.35878225597
3304	33.41850762956
3323	64.9249415089
3342	33.21338010102
3712	32.75770504508
3731	32.65514128081
3742	33.7140666544
3750	32.55257751654
3761	33.61230289013
3769	32.45001375227
3780	33.50973912586
3799	65.0161730052
3818	33.30461159732
4139	100.3853410527
4158	31.89177493206
4188	32.84093654138
4207	32.74637277711
4218	33.8060981507
4226	32.64380901284
4237	33.70353438643
4245	32.54124524857
4256	33.60097062216
4275	65.1074045015
4577	100.6017000776
4615	32.08557019263
4634	31.98300642836
4653	31.88044266409
4664	32.94016803768
4683	32.83760427341
4713	33.79476540273
4721	64.24147430848
4732	33.69220211046

```
4751 65.1986359978
5030 51.22831995353
5068 51.12575618926
5106 51.02319242499
5128 52.08291779858
5144 50.92062066072
5166 51.98035403431
5182 82.42706254006
5204 83.48678791365
5220 82.32449877579
5242 83.38422414938
5495 50.74407301488
5533 50.64150925061
5555 51.70123462421
5593 51.59067085994
5631 83.10510473927
5653 52.55583246926
5669 83.002540975
5691 52.45326870499
5729 83.95970258433
5767 83.85713882006
5982 51.31955144983
6020 51.21698768556
6058 82.72342156492
6080 52.17414929408
6096 82.62085780063
6118 52.07158553061
6134 82.51829403636
6156 83.57801940995
6194 83.47545564568
6447 50.83530451118
6485 50.73274074691
6523 82.23917462625
6545 51.68990235624
6583 83.19633623557
6621 83.0937724713
6643 84.1534978449
6659 82.99120870703
6681 84.05093408063
6719 83.94837031636
6972 51.30821918186

THE FOLLOWING ARE THE RELEVANT DOC'S:
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
  132  726 1320 1388 1972 2257 2393 3930
 THE PRECISION PR=1.00
 THE RECALL RC= .17

DOC.NO.  132 CONTAINS THE TERMS
   50   86  100  123  150  159 1902 2086 4721

DOC.NO.  162 CONTAINS THE TERMS
   50   69   99  100  159  308  427  774 1688 4275
 4825

DOC.NO.  235 CONTAINS THE TERMS
  100  307  427  513  835 1217 1810 2305 5653

DOC.NO.  520 CONTAINS THE TERMS
   50  251  307 1353 1572 3650 5495 6719 6888
```

DOC.NO. 606 CONTAINS THE TERMS
50 72 100 125 159 893 1029

DOC.NO. 692 CONTAINS THE TERMS
50 381 620 893 1387 2101

DOC.NO. 726 CONTAINS THE TERMS
50 54 100 150 159 427 893 1601 3799

DOC.NO. 898 CONTAINS THE TERMS
50 159 221 308 1829 5593 5669

DOC.NO. 984 CONTAINS THE TERMS
50 80 100 2200 4490 5261 6058

DOC.NO. 1001 CONTAINS THE TERMS
50 75 100 545 1591 1844 5182 5982

DOC.NO. 1069 CONTAINS THE TERMS
83 149 150 192 672 741 1863 5068 5220 6003
6118 6583

DOC.NO. 1148 CONTAINS THE TERMS
50 159 337 546 1131 6870

DOC.NO. 1241 CONTAINS THE TERMS
50 159 307 365 791 1825 4591 5242 5555 6194

DOC.NO. 1320 CONTAINS THE TERMS
50 69 100 150 313 427 965 2082 2320 6237

DOC.NO. 1388 CONTAINS THE TERMS
50 78 100 150 423 427 791 3790

DOC.NO. 1413 CONTAINS THE TERMS
102 159 307 477 1029 5631

DOC.NO. 1560 CONTAINS THE TERMS
50 159 308 427 546 674 841 1966 4022

DOC.NO. 1585 CONTAINS THE TERMS
50 97 121 150 278 1012 1860

DOC.NO. 1653 CONTAINS THE TERMS
50 308 427 477 672 715 1340 5144 5166 6485

DOC.NO. 1732 CONTAINS THE TERMS
50 100 159 477 526 1029 2324

DOC.NO. 1825 CONTAINS THE TERMS
50 62 1012 1014 3002 5533

DOC.NO. 1904 CONTAINS THE TERMS
50 74 150 506 672 791 910 1131 5030 6621

DOC.NO. 1972 CONTAINS THE TERMS
50 100 150 742 910 1012 1644 1878

DOC.NO. 2074 CONTAINS THE TERMS
50 191 774 2067 4224

DOC.NO. 2099 CONTAINS THE TERMS
50 100 181 672 847 5698

DOC.NO. 2257 CONTAINS THE TERMS
[illegible]

50 100 150 307 431 770 6020 6242

DOC.NO. 2393 CONTAINS THE TERMS
50 100 150 159 225 235 308 3655 4751 5106
5801 6898

DOC.NO. 2418 CONTAINS THE TERMS
50 61 94 100 308 545 6681

DOC.NO. 2737 CONTAINS THE TERMS
50 55 307 893 1625 1640 6729

DOC.NO. 2762 CONTAINS THE TERMS
50 51 138 345 477 793 1131 1353 3553 6523
6694

DOC.NO. 2787 CONTAINS THE TERMS
50 100 159 910 912 979 1012 1997 5204 6134
6156 6545

DOC.NO. 2923 CONTAINS THE TERMS
50 100 545 1402 1634 5729

DOC.NO. 3081 CONTAINS THE TERMS
50 88 191 307 605 910 5128 5556

DOC.NO. 3106 CONTAINS THE TERMS
150 159 307 308 360 427 2339

DOC.NO. 3425 CONTAINS THE TERMS
50 100 308 477 495 774 3323 6643

DOC.NO. 3561 CONTAINS THE TERMS
50 53 546 753 774 1029 1709

DOC.NO. 3586 CONTAINS THE TERMS
50 224 477 774 791 1606 3210 6080

DOC.NO. 3880 CONTAINS THE TERMS
59 68 195 214 477 672 1487 2836

DOC.NO. 3905 CONTAINS THE TERMS
80 444 1368 1387 1402 1828 4843 6447

DOC.NO. 3930 CONTAINS THE TERMS
50 53 60 100 150 159 307 594 791 1220
5839

DOC.NO. 4292 CONTAINS THE TERMS
50 100 151 308 672 706 5767

DOC.NO. 4317 CONTAINS THE TERMS
50 150 269 355 546 3022

DOC.NO. 4342 CONTAINS THE TERMS
50 307 674 1368 2086 6096 6972

DOC.NO. 4367 CONTAINS THE TERMS
50 75 159 256 5691

DOC.NO. 4392 CONTAINS THE TERMS
50 57 70 100

DOC.NO. 4639 CONTAINS THE TERMS
50 58 159 308

```
DOC.NO. 4955 CONTAINS THE TERMS
   50  150  159  307  308  523  572 2135 6663

DOC.NO. 4980 CONTAINS THE TERMS
   50  159 3520 6659

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? 50-100-150$ (Q2)

THE FOLLOWING ARE THE RELEVANT DOC'S!
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY!
   33  123  132  162  235  263  370  421  496  520
  527  605  606  638  692  726  755  898  901  984
 1001 1069 1148 1241 1308 1320 1326 1381 1388 1399
 1458 1509 1527 1560 1584 1585 1597 1653 1676 1703
 1732 1751 1777 1825 1904 1910 1917 1929 1963 1972
 2003 2006 2062 2063 2074 2080 2099 2170 2250 2257
 2315 2373 2392 2393 2418 2477 2534 2603 2737 2740
 2748 2762 2787 2826 2847 2923 2966 3055 3059 3081
 3082 3096 3106 3285 3292 3348 3425 3438 3470 3514
 3522 3561 3586 3609 3653 3670 3741 3824 3912 3930
 3955 4198 4199 4202 4292 4317 4342 4367 4392 4435
 4497 4562 4584 4630 4639 4656 4703 4709 4906 4955
 4964 4980
 THE PRECISION PR= .37
 THE RECALL RC= .94

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? 50+(100-150-159-307-308-477-774)$ (Q3)

THE FOLLOWING ARE THE RELEVANT DOC'S!
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY:
  132  162  520  606  726  898  984 1001 1148 1241
 1320 1388 1560 1585 1653 1732 1904 1972 2074 2099
 2257 2393 2418 2737 2762 2787 2923 3081 3425 3561
 3586 3930 4292 4317 4342 4367 4392 4639 4955 4980
 THE PRECISION PR=1.00
 THE RECALL RC= .83

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? (50+(100-150-159-307-308-477-774))-(307+427+(100-150))$ (Q4)

THE FOLLOWING ARE THE RELEVANT DOC'S:
  132  162  235  520  606  692  726  898  984 1001
 1069 1148 1241 1320 1388 1413 1560 1585 1653 1732
```

```
 1049 1148 1241 1320 1388 1413 1560 1585 1653 1732
 1825 1904 1972 2074 2099 2257 2393 2418 2737 2762
 2787 2923 3081 3106 3425 3561 3586 3880 3905 3930
 4292 4317 4342 4367 4392 4639 4955 4980

THE FOLLOWING DOCUMENTS ARE RETRIEVED BY THE QUERY!
  132  162  235  520  606  726  898  984 1001 1148
 1241 1320 1388 1560 1585 1653 1732 1904 1972 2074
 2099 2257 2393 2418 2737 2762 2787 2923 3081 3106
 3425 3561 3586 3930 4292 4317 4342 4367 4392 4639
 4955 4980
 THE PRECISION PR=1.00
 THE RECALL RC= .88

PLEASE SPECIFY PRINT OPTIONS
? 0000

USER QUERY
? /
    48.915 CP SECONDS EXECUTION TIME
/bye

KEMSI72   LOG OFF    15.52.04.
KEMSI72   SRU      74.073 UNTS.
```

Bibliographic references

1. Charles T. Meadow: The Analysis of Information Systems, John Wiley and Sons Inc., 1967.
2. F.W. Lancaster; Vocabulary Control for Information Retrieval, Information Resources Press, 1972.
3. Caras G.J.: Computer Simulation of Small Information System, American Documentation pp. 120-122, Vol.19,April 1968.
4. Cooper M.D.: A Simulation Model of an Information Retrieval System, Information Storage and Retrieval Vol.9, No.1, pp. 1-12 1973.
5. Jacquesson, Alain; Schieber, Coillian D.,: Term Association Analysis on a large File of Bibliographic Data, Using a Highly-Controlled Indexing Vocabulary. Information Storage and Retrieval, Vol.9, pp. 85-94 1973.
6. J.R. Pierce: Symbol Signal and Noise, Harper and Row Publishers, 1961.
7. Nona, Houston and Eugene Wall: "The Distribution of Term Usage in Manipulative Indexes", American Documentation, Vol.15, pp. 105-114, April 1964.
8. Zipf, G.K.: Human Behaviour and Principal of Least Effort, Hofner Publishing Company, New York, 1965.
9. Baldwin R.: Chemical titles - A Computer Information Retrieval Data System, Brookhaven National Laboratory Report BNL-50119 May 1968.

10. Arthur D. Little Inc.: Centralization and Documentation, Cambridge, Mass. 1963.

11. Cleverdon, et.al.: Factors Determining the Performance of Indexing Systems, Vol.1, Design. Cranfield England College of Aeronautics, ASLIB Cranfield Research Project, 1966.

12. Booth, A.D.: A Law of Occurrences for Words of Low Frequency; Information and Control, Vol.10, pp. 386-393, 1967.

13. Mandelbrot B.: On the Theory of Word Frequencies and on Related Markovian Models of Discourse, American Mathematical Society Symposium Appl. Math. Vol.7., pp. 190-219, 1960.

14. Wall R.A.: Indexing Language Structure for Automated Retrieval, Information Storage and Retrieval, Vol.9, pp. 607-619, 1973.

15. Hoyle W.G.: On the Number of Categories for Classification, Information Storage and Retrieval, Vol.5, pp. 1-6, 1969.

16. Hoyle W.G.: Automatic Indexing and Generation of Classification Systems by Algorithm, Information Storage and Retrieval, Vol.9, pp. 233-242, 1973.

17. Helander D.P.: A Feasibility Study of Automatic Indexing and Information Retrieval, IEEE Transaction on English Writing and Speech, Vol.EWS-13, No. 2, pp. 58-59, September 1970.

18. Hirschman L., Grishman R., Sager N.: Grammatically Based Automatic Word Class Formation; Information Processing and Management, Vol.11, pp. 39-57, June 1975.

19. Humphrey S.: Searching the Medlars Citation File On-line Using Eihill 2 and Stairs: A Comparison; Information Storage and Retrieval, Vol.10, pp. 321-329, 1974.

20. Jones K.O.S.: A Statistical Interpretation of Term-Specificity and Its Application in Retrieval; Journal of Documentation, Vol.28, No. 1, pp. 11-21, March 1972.

21. Stiles H.E.: The Association Factor in Information Retrieval, Journal of ACM, Vol.8, pp. 271-279, 1961.

22. Cleverdon C.W.: Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems, National Science Foundation, Washington D.C. 1962, pp. 33.

23. Brookes B.C.: The Derivation and Application of the Bradford Zipf Distribution; Journal of Documentation Vol.24, pp. 247-265, 1968.

24. G. Arthur Mihram: Simulation Statistical Foundations and Methodology, Academic Press, New York, 1972.

25. J.S. Bendat and A.G. Piersol: Random Data Analysis and Measurement Procedures, John Wiley & Sons, Inc. New York, 1971.

26. Maron M.E., Kuhns J.L.: On Relevance, Probabilistic Indexing and Information Retrieval, Journal of the ACM, Vol.9, pp. 216-244, 1960.

27. Van Rijsbergen C.J., Jones K.S.: A Test for the Separation of Relevant and Non-Relevant Documents in Experimental Retrieval Collections, Journal of the Documentation, Vol.29, No.3, pp. 251-257, Sept. 1973.

28. Oebhardt F.: A Simple Probabilistic model for the Relevance Assessment of Documents, Information Processing and Management, Vol.11, pp. 11-21, June 1975.

29. Ghosh J.S., Nenfeld M.L.: Unitedness of Articles in the Journal of the American Chemical Society, Information Storage and Retrieval, Vol.10, pp. 365-369, 1974.

30. Ghosh J.S.: Unitedness of Articles in Náture, A Multidisciplinary Scientific Journal, Information Processing and Management, Vol.11, pp. 165-169, 1975.

31. Damerau, F.J.: An Experiment in Automatic Indexing, American Documentation, Vol.16, No.4, pp. 283-289, Oct. 1965.

32. Janas J.: Resuts of an Experiment with Automatic Indexing Based on the Analysis of the Texts of Abstracts, Information Processing and Management, Vol.11, pp. 115-122, 1975.

33. Gotlieb G.C., Kumar S.: Semantic Clustering of Index Terms, Journal of the ACM, Vol.15, No.4, pp. 493-573, October 1968.

34. Heaps, H.S.: Information Retrieval, Theoretical and Computational Aspects, Academic Press (in press).

35. Huang J.C.: A note on Information Organization and Storage, CACM, Vol.16, No.7, pp. 406-410, July 1973.

36. Keen M.: Search Strategy Evaluation in Manual and Automated Systems, ASLIB Proceeding Vol.20, No.1, pp. 65-81, January 1968.

37. G. Salton: Automatic Information Organization and Retrieval, McGraw-Hill Book Company, New York, 1968.

38. J.J. Rocchio and G. Salton: Information Search Optimization and Interactive Retrieval Techniques, Proceedings Fall Joint Computer Conference, pp. 293-305, 1965.

39. G. Salton: A New Comparison Between Conventional Indexing (MEDLARS) And Automatic Text Processing (SMART). Journal of the ASIS pp. 75-84, March-April 1972.

40. Bernier, C.L., and Heumann, K.F.: Correlative Indexes III: Semantic Relations Among Semanteces - The Technical Thesaurus, American Documentation Vol.8, pp. 211-220, 1957.

41. Jones K.S.: Collection Properties Influencing Term Classification Performance, Information Storage and Retrieval, Vol.9, pp. 499-573, 1973.

42. Adamson G.W., Boreham J.: The Use of an Association Measure Based on Character Structure to Identify Semantically Related Pairs of Words and Document Titles:, Information Storage and Retrieval, Vol.10, pp. 253-260, 1974.

43. Schiminovich S.; Automatic Classification and Retrieval of Document by Means of a Bibliographic Pattern Discovery Algorithm , Information Storage and Retrieval, Vol.6, pp. 417-435, 1971.

44. Cummings L.J., Fox D.A.: Some Mathematical Properties of Cycling Strategies Using Citation Indexes , Information Storage and Retrieval, Vol.9, pp. 713-719, 1973.

45. Garfield E, and Sher I.H.: New Factors in the Evaluation of Scientific Literature Through Citation Indexing , American Documentation, Vol.14, No.3, pp. 195-201, July 1963.

46. Bichteler J., Parsons R.G.: Document Retrieval by Means of an Automatic Classification Algorithm for Citations", Information Storage and Retrieval, Vol.10, pp. 267-278, 1974.

47. Salton G.; Automatic Indexing Using Bibliographic Citations , Journal of Documentation, Vol. 27, No. 2, pp. 98-110, June 1971.

48. Salton G.: Search Strategy and the Optimization of Retrieval Effectiveness., Mechanized Information Storage Retrieval and Dissemination, North Holland pp. 73-107, 1968, IFIP Conference on Mechanized Documentation, Rome, June 1967.

49. Salton G., Yang C.S., Yu C.T.: A Theory of Term Importance in Automatic Text Analysis , Journal of the ASIS, pp. 33-44, January-February 1975.