DOCUMENT RETRIEVAL BASED ON A COGNITIVE MODEL OF DIALOGUE

George Orelens Ofori-Dwumfuo

Thesis submitted for the degree of
DOCTOR OF PHILOSOPHY

The University of Aston in Birmingham

August 1982

The University of Aston in Birmingham

DOCUMENT RETRIEVAL BASED ON A COGNITIVE MODEL OF DIALOGUE

George Orelens Ofori-Dwumfuo

PhD 1982

Summary

Owing to the rise in the volume of literature, problems arise in the retrieval of required information. Various retrieval strategies have been proposed, but most of them are not flexible enough for their users. Specifically, most of these systems assume that users know exactly what they are looking for before approaching the system, and that users are able to precisely express their information needs according to laid-down specifications.

There has, however, been described a retrieval program - THOMAS - which aims at satisfying incompletely-defined user needs through a man-machine dialogue which does not require any rigid queries. Unlike most systems, Thomas attempts to satisfy the user's needs from a model which it builds of the user's area of interest. This model is a subset of the program's "world model" - a database in the form of a network where the nodes represent concepts.

Since various concepts have various degrees of similarities and associations, this thesis contends that instead of models which assume equal levels of similarities between concepts, the links between the concepts should have values assigned to them to indicate the degree of similarity between the concepts. Furthermore, the world model of the system should be structured such that concepts which are related to one another be clustered together, so that a user-interaction would involve only the relevant clusters rather than the entire database - such clusters being determined by the system, not the user.

This thesis also attempts to link the design work with the current notion in psychology centred on the use of the computer to simulate human cognitive processes. In this case, an attempt has been made to model a dialogue between two people - the information seeker and the information expert. The system, called Thomas-II, has been implemented and found to require less effort from the user than Thomas.

Automatic information retrieval, retrieval strategies, interactive systems, man-machine dialogues, cognitive models

ACKNOWLEDGEMENTS

I am most grateful to Dr R N Oddy, my supervisor, for his being always willing and ready to discuss my work, his useful and stimulating ideas, for making data available for the experiments, etc. I also wish to thank all the numerous others for their interest and help, especially Dr S E Robertson and Dr N J Belkin for their suggestions.

I also appreciate, with thanks galore, the patience and moral encouragement shown by my (extended) family. And I gratefully acknowledge the financial support I have received from the Ghana Government.

G O O-D
Birmingham 1982

To The Third Generation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

THE INFORMATION RETRIEVAL PROBLEM

1.1 Introduction

Our society today is an information society. People in various positions describe their opinions, experiences and research work in reports and articles. The literature, especially in science and technology, increases rapidly with scientific and technical development. Sandoval(1976) noted that 'Biochimica et Biophysica Acta' for example, has grown at an approximately exponential rate since it started in 1947 and it doubles in size about every 4.6 years. Ashworth(1974) also demonstrated this information explosion with data on the number of years it took 'Chemical Abstracts' to publish successive millions of abstracts; whereas the first million took 32 years, the fifth million took only 3.3 years. Consequently, many professionals feel so overloaded with information (more information reaches them than they can possibly cope with) that they rather call for selective and critical reviews of the literature.

At the same time, people in various walks of life - government policy-makers, business administrators and other decision-makers - still find themselves in the sad situation of having to make judgements or solve problems on the basis of less than adequate information. Thus we are in a society where there are vast amounts of information to which accurate and speedy access is becoming more and more difficult, resulting in our inability to identify, locate, and retrieve the needed information. One effect of this is that relevant information gets ignored since it is never uncovered, and this, in turn, leads to much duplication of work and effort in the research fields, as well as the above-mentioned situation of decisions being made on inadequate information.

With the advent of computers with great processing speeds and extensive storage facilities, thought has been given to using them to provide rapid and intelligent information storage and retrieval (IR) systems. There now exists a wide variety of computerised systems that perform the function of retrieving information from a store of data. At one end of the spectrum are what are usually called the data retrieval systems. At the other extreme are the reference retrieval or literature searching systems. Somewhere along the line lie question-answering systems. The distinction between these systems lies in the type of inference-making capability that each system employs.

Data retrieval systems, as the name implies, provide a specific fact to a user's question. The system holds a file of formatted records, and interprets a question as a selection criterion, usually using database management software for storage and data manipulation purposes.

When a query is presented to a question-answering system, the system examines its store of data in order to extract one fact from the data file. The desired fact may not necessarily be in the form needed by the user. So, in order to gather the required information, the question-answering system may have to deduce the answer from a series of related items of information. Thus a question can be answered even though the answer is not held explicitly in the store, so long as the answer is derivable from the stored information. Consequently, most of these systems suffer from problems of high computation times and large memory requirements. (See Michie 1979 and the regular feature of 'Machine Intelligence' edited by Hayes, Michie and Mikulich.)

In response to a question, a reference retrieval system provides lists of references to documents that are likely to answer the question. To do this, the question is usually compared with a document representation and the system infers that the document meets the user's needs if the words in the document representation match those in the question.

Thus a user of a data retrieval system, for example, may request for factual data, say, the specific gravity of a particular element, whereas in reference retrieval, a user may need to see documents that describe or discuss a particular subject area, for example, the effect of sunlight on green grass. That is to say, an information retrieval system does not inform the user on the subject of her request, it merely informs her of the existence and whereabouts of documents related to the request. Minker(1977) discusses the contrasts between all these systems. (See also Croft 1982).

The term 'information (storage and) retrieval' as used in this thesis, is concerned only with fully automated document or reference retrieval.

1.2 The Information Retrieval Environment

1.2.1 The System

The major activities of an information retrieval system may be shown by the simplified diagram (fig. 1.1). These may be put into three broad categories: the input, the processor and the output (just as any general computing system).

The input consists of the documents containing the required information and the statements of requests. The former is usually stored in the system in some form, whereas the latter comes from the user of the system. The documents are represented by a list of extracted words (keywords or index terms) considered to be significant, which are then stored as a database. In most systems, the extraction of words to represent a document involves the use of a controlled vocabulary which includes a limited set of terms that must be used to represent the subject matter of documents. The requests are similarly represented, and all are made suitable for computer use.

The processor deals with the retrieval process. With the use of a retrieval strategy or rule, and the collection of document representatives, an attempt is made to obtain those documents which will satisfy the particular request. The references of those documents found, or any other representations that will help locate the documents, constitute the output of the system. In some systems, it is possible for the user to alter her query following her decisions on output from an earlier search - a process called feedback.

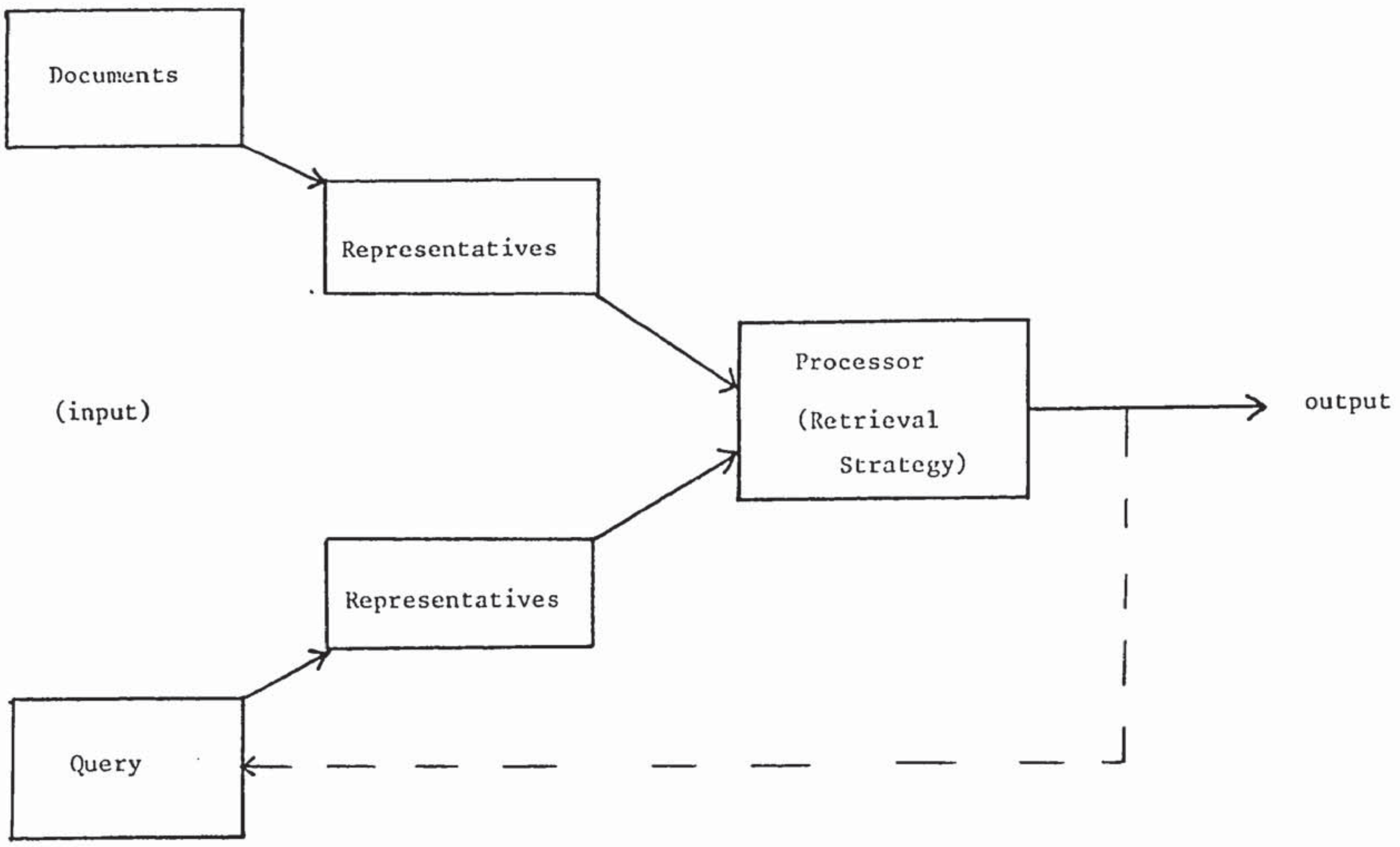


Figure 1.1 A Generalised Retrieval System

A general model of information retrieval systems has been proposed by Bookstein and Cooper(1976), and from it they describe a wide variety of retrieval systems, ranging from an ordinary card catalogue to the sophisticated automated systems.

1.2.2 The User

A very important part of every system is the user of the system. And information retrieval systems are no different. Yet most of the work on IR systems in the literature regard the user as a black box, in the sense that no substantive hypotheses are made about her cognitive processes.

Broadly speaking, the user of an information retrieval system has two major roles to play; presentation of her problem in a precise format, and the determination of whether the system's outputs are of any use to her. The latter involves the user's decisions on the retrieved items - her judgement on how relevant the items are with respect to her information need.

The former, which is a central issue in this thesis and hence will recur in various sections later, may be broken down further into two; first, the user recognizing that she has a need for information, and second, the expression of her need for the purposes of the system. It is not always the case that a user will recognise her needs fully before coming to the system. She may, during the search, get some enlightenment on the subject area and hence an enlightenment on what her needs really are. Neither is it always the case that the user will precisely express her needs without difficulty - considering the rigid formats required by many IR systems. Yet, the way in which the user's information needs are expressed to the system greatly affects the output the system will obtain for her.

As most current systems are rigid in what the user is expected to do, one important factor affecting the success or failure of such systems is how well their users have been trained to use the systems (Moghdam 1975). User training may involve the use of printed instructional guides and user manuals. However, these guides and manuals are very variable in quality - some are complex, lengthy and badly written. Although they may be valuable as reference manuals, they are usually unsuitable for instructional purposes. Other modes of user instruction (for example, audiovisual presentation or personal instruction by experts) tend to be more effective but are, of course, more expensive.

It is the view of this thesis that in order to reduce the amount of frustration (Melnyk 1972) caused to users of retrieval systems, most of the burden cur.ently put on the users - for example, rigid query formulation and the selection of which file to access - should be borne by the system. Users should not be made to adapt to systems, systems should be flexible enough to require as little as possible from the users.

1.2.3 The Problem

Having given a general overview of a retrieval system and the user, we are now in a position to look at the problem of information retrieval.

Information retrieval is concerned with the following situation. A user recognizes that she has need for some information and presents a request expressing that need to an information retrieval system, with the hope that the system will be able to, fully or partially, satisfy her need. The task of the system is to retrieve from a stored set of documents (usually represented by a few terms - surrogates), all and only those documents which it judges to be most likely to satisfy the user's information need based upon the request she presented to the system. The user then examines the retrieved text to determine how useful or relevant it is relative to her need. The performance of the system is usually evaluated according to how closely its judgements agree with those of the user, regarding the user's needs.

Information retrieval systems use various strategies to achieve their goal. However, most of them, after representing both the documents and queries by their surrogates, tend to match the query representatives with the document representatives, either directly by comparison or by the use of a mapping function or a retrieval rule of some sort. Belkin(1980) discusses the philosophical problem of matching query against document in information retrieval. He points out that when a user's query is ill-specified, matching is an inappropriate operation.

This thesis is concerned with a retrieval strategy (Oddy 1977a,b) in which the system tries to model the user's interests. Rather than match users' queries against the system's documents, a structure of the user's model is built from components of the system's "world model" - the database, which contains citations to documents.

1.3 General Trends in Information Retrieval

There are two main lines along which information retrieval systems go; one which relies on the Boolean operators AND, OR and NOT and the other which uses a mapping function to rank the documents for output. These two broad trends are briefly discussed in this section.

1.3.1 Boolean Systems

A Boolean retrieval system is one which retrieves documents which are 'true' for a given query. The queries, expressed in terms of index terms (keywords) are combined by the logical connectives AND, OR and NOT. The system then selects all documents from the file which when matched with the query yield the logical TRUE value.

Thus for example, if a user presents the query

Q = (Rats AND Mice) OR (Rodents AND NOT Rabbits),

then a Boolean search will retrieve all documents indexed by 'Rats' and 'Mice' as well as those indexed by 'Rodents' which are not indexed by 'Rabbits'.

The strategy is thus simple in concept and relatively easy to implement, especially if an inverted file (a file in which the index terms are listed, each with all the documents for which it is used as a keyword) system is used.

A Boolean search divides the file of documents into two categories; TRUE (and therefore retrieved) or FALSE (not retrieved). Accordingly, the output from a Boolean search is a list of references with no indication as to which of those documents are more likely to be satisfactory to the user's needs. The strategy is also sensitive to the om~~m~~ission (which may at times be unavoidable) of an important index term from a reference or from the search statement or the indexing vocabulary.

Boolean searches are, however, the most popular in operational retrieval systems, regardless of the users' difficulties with the query formulation.

1.3.2 Document Ranking Systems

There are other systems which rank their outputs to indicate the order in which the documents are likely to be useful to the user. In such systems, the query is matched with the documents in the file and the system returns a value which indicates what degree of similarity there is between the documents and the user's query. The degree of similarity is measured by assessing which index terms present in the query are also present in the document. There are several similarity measures available for this assessment and the greater the value yielded the 'nearer' the query is to the document.

The simplest of all the similarity measures gives the number of terms that the query and the document have in common, and is called the simple coordination level match. Thus, for a query, Q, and a document, D, if $|X|$ denotes the number of terms belonging to X, then the coordination level between Q and D is given by $|Q \cap D|$. (where $\cap$ denotes the intersection of two sets).

Some other measures, unlike the simple coordination level match, take into account the sizes of the query and the documents. In these cases, the sizes are usually used to normalize the simple coordination level values.

Examples are:

i) $$\frac{2|Q \cap D|}{|Q| + |D|}$$ (Dice's Coefficient)

ii) $$\frac{|Q \cap D|}{|Q \cup D|}$$ (Jaccard's Coefficient)

iii) $$\frac{|Q \cap D|}{\min(|Q|, |D|)}$$ (Overlap Coefficient)

With these systems, the method of selection of items to show the user could be either of two possibilities. A threshold value may be selected, and on comparing the query with each document, any document with value above the threshold is deemed relevant. Alternatively, the whole collection is compared with the query and the coordination levels stored. These levels are then ranked

(Robertson and Belkin 1978), and the collection of documents is presented to the user in descending order of coordination value, or by choosing the top n documents from the list, n being any desired number.

There are also systems which take into account the relative importance of the terms to the user or in the document descriptions. They assign various weights to the terms to indicate their importance. The basic principle here is similar to the simple coordination level procedures, except that the sum of weights of the common terms gives the required similarity value - 'notional coordination level'. These techniques, however, tend to be theory-based these days.

It is worth mentioning here that various attempts have been made to combine Boolean search formulations with weighted ranked outputs (Mulvihill and Brenner 1968, Angione 1975, Noreault et al. 1977). Bookstein(1978), however, points out the problem that equivalent Boolean expressions may give different values for retrieved documents.

1.4 On-line Retrieval

The process of information storage and retrieval has passed through various stages of development; from the purely manual systems such as printed indexes and card catalogues, to the fully automated systems like DIALOG. Lancaster and Fayen(1973) discuss the characteristics of these systems.

Purely manual systems which were used may be said to be random-access devices. It is possible for the user to go directly and consult only the portion of t.ıe file required. Browsing through the index is allowed and there is hardly any time delay with a search, since one can conduct the search whenever the need for information arises, without necessarily having to consult a librarian or an information specialist.

The manual systems gave way to batch-processing computerized systems on the advent of the punched card. With these systems, however, the searcher has essentially one chance at a time to successfully conduct a search. Accordingly, she must think, well in advance, of all likely approaches to use. The search must be delegated to an intermediary, as the user is unable to run her own searches, and there is an inevitable delay in obtaining search results due to the batch processing.

On-line retrieval systems have none of these disadvantages. Even in cases where searches are conducted by trained intermediaries, on-line retrieval has the advantages of rapid response and the capability of interactive, browsing and heuristic searches (Shergold 1972). An on-line information retrieval system, briefly, is one in which the user conducts the search interactively with a computer. In such a system, there is a two-way communication between the user and the computer through some linkage, for example, a visual display unit.

As has been discussed by Barraclough(1977), interactive information retrieval has become necessary because searches cannot be completely automated; the judgement of the user is needed at various stages of the search. During an on-line retrieval, the searcher may learn how to express her need more effectively to the system, and so may shift her emphasis as the notion of what she requires becomes more and more clear to her from the system's responses. Also, the more experienced an on-line user is, the less dependent she is likely to be on human intermediaries.

Examples of on-line systems in operation include BLAISE, DIALOG and ORBIT. Hall(1977) provides a useful directory to a lot more of them.

1.5 The Intermediary

Any discussion of information retrieval systems and their users would be incomplete without a word on the role of the intermediary. Most of the present-day retrieval systems tend to be 'man-man-computer' systems (fig. 1.2), in that, apart from the user and the system, they involve a human intermediary - usually an information specialist or a search expert. The user consults the intermediary, and the intermediary does the searching. Maldé(1978)

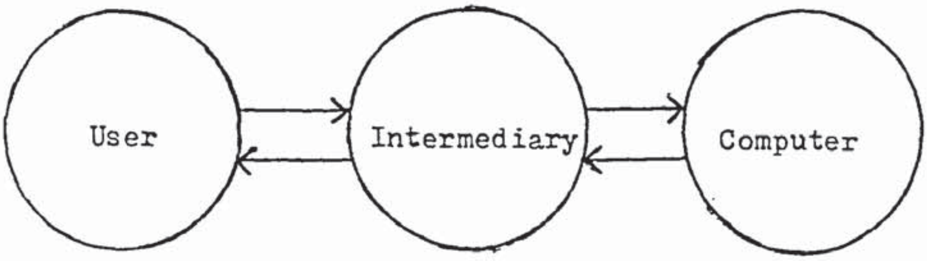


Figure 1.2 Double Interaction

calls this form of interaction 'double interaction' since it involves the simultaneous conduct of man-man interaction and man-computer interaction.

When retrieval systems were being developed, it was assumed that the end-users would conduct their own searches. However, most current writers believe that users cannot be expected to cope with most retrieval systems (Maron and Fife 1976). Problems arise with attempts to express the needs into rigid queries, as well as with pre-search and other preparations (for example, the determination of the particular files to access and the selection of strong candidate search terms).

As noted by Meadow(1970), when an enquirer tries to use a retrieval system herself, she is faced with a situation in which she may have an incomplete knowledge of the structure and contents of the file she is about to search; she may not know the file language, hence she will not know how to express her requirements precisely. And hence her need for a middleman (Bennett 1977).

Wanger et al(1976) found that in most cases, searches are conducted in isolation by the intermediary at a time convenient for her, after having had a discussion of the search requirements with the user. In other cases, the intermediary and the user collaborate at the terminal to carry out the search, or the user (usually the experienced one) has access to the system and runs her own search, with the intermediary readily available to give assistance if need be. As reported by Holmes(1976), 60 per cent of users preferred a joint search with an intermediary, 15 per cent left it to the intermediary to carry out the search alone and 25 per cent used the terminal unaided. Clearly, the best results are likely to be obtained when the user's in-depth knowledge of what is actually wanted is allied to the trained searcher's detailed knowledge of the system. Barber et al(1973) have shown that when the user and intermediary work together, recall and precision are both enhanced.

1.6 THOMAS: The Retrieval Strategy Pursued Here

This far, the general trends in information retrieval have been briefly discussed and we have noted that most systems are rather too rigid for their users. Specifica.ly, we have pointed out the fact that although before an individual approaches a retrieval system, she would have recognised a need for information, the retrieval system cannot assume that that need is easily expressible. Even if the user is able to express her need in some linguistic form, one cannot predict the ease with which she can transform her expression into the forms of query most systems require. Consequently and regretfully, systems which have been designed to be accessed by the end-users turn out to involve a third party - the human intermediary.

In view of the above situation, Oddy(1977a,b) describes a computer program, THOMAS, which aims at satisfying incompletely-defined needs of a user through a dialogue between her (the user), and the machine. Query formulation is not a pre-requisite, as the retrieval strategy does not involve the usual matching of query against document. The interaction between the user and Thomas does not require an intermediary and is comparable to a dialogue between an information-seeker and a subject expert.

In such a dialogue, the model is essentially that described by Hollnagel(1979), which we shall discuss later in Section 3.4. Both the user and the expert participate in the dialogue until the subject expert somehow understands the searcher's problems and then offers her information which may lead to a solution. The program, Thomas, has been designed to play the role of the subject expert

(Olney 1962), although in a relatively unsophisticated way.

The work described herein is an implementation of Thomas - called Thomas-II (pronounced Thomas-two). Two other aspects have been explored; the incorporation of a form of weighting based on the strength of association between the items in Thomas' structural model of the database, and the possibility of the program creating smaller virtual databases to suit specific search areas by clustering; thereby preventing the users from having to unnecessarily deal with the whole database.

Thomas' model of the database, from which it builds its image of the user has a network structure in which the terms are linked to their associated terms. This structure assumes equal association strengths between the terms. In this project, Thomas-II assumes that more closely related items are to be assigned greater association values. Determination of which reference to show the user will then involve the particular items in the structural models as well as the weights on the links between them.

In a large operational environment with an enormous database, the network of the database will accordingly be large. It is undoubtedly true, however, that only a small subset of the whole database will come into play during a user's interaction. Consequently, in the project described herein, Thomas-II is structured to consider the whole database as being made up of clusters of smaller databases, each of which broadly represents an area of interest.

Whenever the user inputs a term, the program will determine the cluster relevant to that term - and hence the user's area of interest. This cluster then beco..es the world model of Thomas-II so far as that particular user-interaction is concerned. The dialogue would then continue as usual. This sort of database partitioning is comparable to what pertains in the current large-scale systems (DIALOG, for example). However, in this case, the burden of having to determine the particular file in the database to be used (file selection) will not be on the user (Williams and Preece 1977).

We shall leave further discussion of the above-mentioned aspects until Chapter 4. But before then, we give in Chapter 2, a review of some concepts of information retrieval we consider directly related to this thesis. We side-step a little in Chapter 3 to discuss an aspect in psychology - the art of problem-solving. This is because, of late, it has been realized that although most of the tasks put to the computer involve some psychological aspects, research into the overlap between psychology and computing concepts has been minimal. This has resulted in the fact that most of the computer simulations of cognitive processes tend to lack the essential behavioural orientation. The chapter continues with a discussion on man-machine dialogues - dialogues in which the machine is made to simulate one of the participants - and the chapter ends with a specific model of a dialogue which has been deemed to form a good basis for an interaction with Thomas. After the discussion of the retrieval progam, Thomas-II, in Chapter 4, we present, in the next chapter, the experiments carried out. And, as usual, we give our concluding remarks in the last chapter.

Chapter 2

A REVIEW OF SOME INFORMATION RETRIEVAL CONCEPTS

In this chapter, we look at some of the concepts of information storage and retrieval which have some bearing on the retrieval strategy discussed in this thesis. (Sparck Jones' recent book (1981) thoroughly exhausts the aspects on information retrieval experiments.) The strategy - a retrieval interaction with Thomas-II - involves a user who comes to the system to seek information in order to satisfy some need. Thomas-II allows the user to browse through its database during the dialogue. Relevance feedback is an integral part of the retrieval strategy, in that the user's responses to Thomas-II outputs are used by the program to modify the model it creates of the user's area of interest.

The aspect of query formulation is an important one in the retrieval process as it is the only means by which the system can know of the user's problem. Owing to the fact that most systems require query formulations which are usually too rigid for users to handle, this issue is a strong basis for this research, as an interaction with Thomas-II does not require too much of the user.

Clustering and weighting are processes quite established in information retrieval. Clustering tends to group together the documents which are likely to satisfy a particular need, and is useful especially when large document collections are in use. Weighting helps to discriminate among the search terms, in that, greater values may be assigned to more important terms, which, in turn, helps in the retrieval of more relevant documents. These two concepts are incorporated into the retrieval program, Thomas-II, by clustering the data base and by assigning association weights to pairs of items in the database.

The aspects of indexing and system evaluation have not been omitted. For, regardless of how good a retrieval strategy is, and how well a user expresses her information need, if the indexing process has not been well carried out, the proportion of relevant items retrieved (recall) and the proportion of the retrieved items that are relevant (precision) are bound to be affected. And the only means of determining how well a system does what it has been designed to do is by its evaluation.

2.1 Information Seeking Through Browsing

Everyone seeks information at one stage or the other. Information-seeking is a day-to-day requirement. Without it nothing good that has been done could have been done well. An information seeker - whether she is a top-level decision-maker or a gatekeeper - would accomplish her search for i..formation through various means, browsing among the books on the library shelves (Hyman 1971) or interactively with an on-line system (Lancaster and Fayen 1973).

Hyman(1971) gave what he called a functional definition of browsing: "Browsing is that activity, subsumed in the direct shelf approach, whereby materials arranged for use in a library are examined in the reasonable expectation that desired or valuable items or information might be found among those materials as arranged on the shelves" (p.482). The definition expressed the assumption that browsing is worthwhile but is limited to the library environment. Apted(1971) considers browsing to be either a planned or unplanned examination of sources, journals, books or other media, with the hope to discover unspecified new, but useful, information.

Browsing is an activity which provokes new thought by exposing the individual concerned to a wide variety of stimuli, but without necessarily being planned to do so. What seems to happen during a browse is that a new idea discovered in some medium, which could be a book, journal or a non-book material, may have various effects on the browser. It may give her a new idea related to some dim concept she already has, show her a hitherto unrecognised interrelationship between two concepts, spark off an entirely new notion, and so on.

Browsing habits are influenced by various things among which may be variations in the standards and disciplines of the browser. Brittain(1970), in his studies of users' information needs, realized that potential browsers may be grouped according to their positions in their academic setting; their different academic disciplines; their orientation towards pure or applied work and probably in other ways too. These variations influence the user's approach to information and this, in turn, is reflected in her browsing habits. Thus, for example, scientists seem to do their browsing in the current materials; journals, current awareness tools, directories of research in progress, as well as recent issues of abstracting journals, whereas, in the humanities, scholars need a much wider base for their browsing - both old and new material, and material on almost any topic. A student in the early days of a research programme would seek information very broadly in order to find something to work on, whereas a fellow student at the completion of her research course would be more interested in detailed work in the area she has been researching.

In a library, a browser may perhaps wander along some section of the collection, and, once in a while, pick up items for examination. She may look at works in her own field or examine material in quite unrelated areas. Lancaster(1979) describes what Apted calls 'specific browsing' in contrast to general browsing. The searcher in this situation makes a literature search through any bibliographic tool without starting with a formal search strategy; she primarily may consult likely subject headings and then follow on to cross references. This user, in contrast with one doing a general browse, has some previous knowledge of the intended direction of her search.

Greene(1977) made an investigation to determine how effective browsing is in the Georgia Institute of Technology Library. He examined the relationship between how a book is discovered and its susbsequent value to the user. He noted that from the quantitative point of view, browsing was the most important method used by patrons to learn about the library books they borrowed. However, browsing ranked last when the utility of the books was considered. Other methods considered included cross-references in publications and discussions with colleagues. Greene's result, although considered by himself to be preliminary, is in line with Levine's (1974) feeling that there is a relationship between browsing capability and user acceptance of lower relevance; that, as browsing capability increases, a decrease in relevance can be tolerated.

With computerized information systems (Palay and Fox, 1981), the browsing situation is different. It is difficult to provide for an informal browsing activity in a formal system where correct procedures have to be followed. In most systems, the user must adapt to the machine, and there is very little allowance for browsing. Lancaster(1979) discusses problems of browsing in systems where the major requirement is that the user must provide a very clear and detailed request. The wider the gap between the stated request and the information needed, the less success can be expected of the system.

The interest of this research is to allow the information seeker to have a good browse in a computerized information environment, in a conversational manner, such that regardless of whatever gap there may be between her stated request and her information need, the system will, through the dialogue, try and model her interest in order to satisfy her needs as much as possible.

2.2 Query Formulation

There is only one way by which an information system can know about the information need of an information seeker. That is by the seeker giving the system some idea (no matter how vague) of what her need is about. (Unless the system has ESP to determine the seeker's problem prior to her coming to the system. We assume that IR systems do not have such powers.) One would therefore have expected that the way in which an individual puts her information need to the system would have been of much research interest ..

Yet, over the last two or three decades, the focus of attention in most aspects of scientific work in information retrieval has been on document description: the indexing method and vocabulary, classification techniques and the statistical and semantic properties of index terms. Very little attention has been paid to the manner in which the seeker's needs are expressed to the retrieval system until quite recently (Lynch 1978, Saracevic 1978, Pejtersen 1979, Ingwersen and Kaae 1979, Heine 1980, Oddy 1980). Consequently, assumptions have been made about the nature of such expressions - the users' queries - with hardly any experimental support for these assumptions. Nevertheless, most current retrieval systems require the user to formulate a query - usually according to some laid-down rigid specificatio.ıs.

The process of query formulation depends on various attributes of the searcher. These may include her knowledge of what has been stored in the database, her knowledge of the indexing and searching processes of the system, how familiar she is with the topic matter to be searched, her personal preferences as to the choice of words and style of presentation, her comprehension of the language in which she is to formulate the query, and so on. All these and other attributes do make the user's work difficult, especially the casual user (Cuff 1980).

In order to make up a query, the searcher begins by choosing a set of terms from the available indexing vocabulary. Each of these terms is chosen for the particular query because the searcher feels it has some connection with her information need. The terms must further be organised somehow so as to establish the desired relationship between them, and to relate them to the query as a whole. The form that the query takes eventually, depends on the retrieval strategy on which it is to be used.

There are a number of apparently very different forms that the user's expression of her information need can take. To systems that accept natural language, for example, the query can be a short interrogative sentence, a statement of the topic, a verbose description of the problem which has generated the need, a tentative description of documents which are likely to satisfy the need, a list of terms or even a formal search specification (Macleod 1977).

With Boolean systems - most operational systems are Boolean - retrieval depends upon a Boolean function as discussed earlier (Section 1.3.1). Thus, in preparing her request, the user must specify the desired relationships between the chosen index terms in a suitable Boolean form; she must explicitly state the logical connections between her terms. The resultant logical form represents the user's query for input into the system. In a weighted retrieval system, on the other hand, where retrieval depends upon the summation of a group of weight values, the user may be required to assign weights, and a minimum value to the chosen index terms in order to express the desired relationship between them. The numerical values assigned to the terms then serve to provide the values for the sum of weights required by the weighted retrieval system.

Some researchers have also been concerned with trying to understand the searcher's task in order to see what role a computer could play in forming a search strategy (Jahoda 1974, Smith 1979). The searcher has an information need; she realizes an "incompleteness in (her) picture of the world - an inadequacy in what we might call (her) 'state of readiness' to interact purposefully with (her) environment" (Mackay 1960, p.789). Belkin and Oddy(1978) put it this way - that the user has realized an anomaly in her 'state of knowledge'. This individual is expected to select index terms in order to appropriately and precisely transform her information need into a query.

The process of transforming an information need into a query is complex and not well understood, especially in Boolean systems. Complexities caused by increasing numbers of Boolean operators per query and deficiencies in retrieval systems, notably the inconsistency and variability of their indexing, complicate the user's task (Dillon and Desper 1980). Most indexes to document collections possess a fixed structure and a controlled vocabulary. Hence if the index space has not been precisely defined in structure and vocabulary to the user, then any attempt to formulate a query to describe her 'anomaly' will tend to result in frustration (Melnyk 1972). Taylor(1968) published a collection of descriptions of searchers' frustration in trying to guess the right index terms in order to use an information retrieval system. Even though appropriate reading references may be available to the user, Marcus et al(1971) have found that the system designer cannot assume the user has read and fully understood them.

In view of this, Lancaster(1971) feels that the system should be able to give the user every possible assistance in the formulation of an accurate request statement, a type that would reflect her information requirements. He noted that if a user makes a request that is either too broad, too specific or too vague, the search results will be unlikely to be of maximum value to her, for a poor request will tend to produce poor search results, regardless of the quality of the indexing, the search strategy and the system vocabulary. The MEDLARS evaluation revealed that imperfect query formulations were largely responsible for 25 per cent and 17 per cent of all the recall and precision failures, respectively, in 300 test searches (Lancaster 1969).

This research is aimed at removing this burden of rigid query formulation from the user. Terms introduced into the system by the user will be used to form the nucleus around which a model of her interest will be built - a model from which the system will try to satisfy her needs. And the user can change her mind at any time of the interaction, as the system will dynamically modify the model accordingly.

2.3 Indexing

Indexing is an important part of the information storage and retrieval process. The information seeker has to express her information need as a search query made up of index terms selected from an index vocabulary, and the system has to decide - based on some rule or other ~~whatsoever~~ - which items to retrieve, by examining its store of sets of index terms which have been assigned to the various documents. So, if the user's chosen terms (appropriately combined into a query) correctly express her information need and the documents in the collection are correctly indexed, then the system will retrieve all and only the relevant documents for her. If, on the other hand, documents are incompletely or inaccurately indexed, some non-relevant documents may be retrieved while some relevant items may not be retrieved.

Ironically, perfect indexing is unattainable. Sometimes, the inclusion of a word considered very descriptive of a document as one of the document's surrogates will cause that document to be inappropriately retrieved for some searchers, and, similarly, the omission of some words will cause the loss of some useful documents for some users. Indexing involves the task of identifying a set of keywords that are descriptive of the subject content of a given document and that are selected from the full text of the particular document. An indexer, after reading the text, will assign to the document a set of index terms chosen either from a limited and predefined vocabulary or from the text according to some indexing rules (Bookstein and Swanson 1975, Cooper 1978).

The aim of the indexing process is regarded as to be able to accurately represent what a document is about. Aboutness is something associated with the document and is independent of the use to which the document might be put. Maron(1977) examined the concept of aboutness as it relates to indexing and the ultimate effectiveness of an information retrieval system, and gave a probabilistic definition of it. He indicates, however, that aboutness is only one of several factors that should be considered in choosing an index term to represent a document.

A major problem that arises with indexing performed manually is the inconsistency of the indexers. This is not unexpected because different indexers are liable to different views, and hence may not state what a document is about in precisely the same terms. The variation in indexer consistency may be attributed to the presence of decision rules for applying the index terms. The more strict and complete the rules, the higher the level of consistency; the more freedom allowed an indexer to make decisions about which labels to assign to a given document, the less likely it is that she will agree with others on which labels to choose.

Consequent to the problem of indexer consistency, researchers have taken to using the machine as an alternative to man in the indexing process. And research on indexing indicates that this alternative method - automatic indexing - does give results that are comparable with human indexing (Salton 1972). Automatic indexing is a computer-performed process by means of which sets of keywords, presumed to comprise good document surrogates, are inferred on the basis of an analysis of the document's full text. In this case, the decision rules for the selection of the indexing terms are based essentially on the frequency of occurrence of words in the documents being indexed. Harter(1978) noted that this form of indexing involves two stages; the identification of technical vocabulary characteristic of a given document literature and the selection of keywords belonging to that ~~literature~~ vocabulary and representative of the individual documents being indexed (Robertson 1977).

Sparck Jones(1974) gives a review on the use of syntax and semantics as the basis of criteria for selection of index terms. The criteria reflect some assumptions as to how word occurrences are related to the contents of the documents in which they occur. She noted that the criteria used for automatic indexing have been chosen primarily as a matter of convenience, instead of being derived from some firm theoretical basis. To supplement Sparck Jones' review is Harter's treatment of statistical approaches to automatic indexing - approaches based mainly on word frequencies.

Two related notions in au⌣omatic indexing are indexing exhaustivity and index term specificity (Keen and Digger 1972). Indexing exhaustivity has to do with how much the various topic areas relate to a given document, and is indicated by the number of terms assigned to a document (Sparck Jones 1973i) whereas index term specificity is a function of the exactness with which a term characterizes a given subject, and is related to the number of documents to which a given term is assigned in a collection.

The more exhaustive the indexing, ie, the more thoroughly the various subject areas are covered, the more likely it is that relevant items will be retrieved in response to user queries, thus achieving high recall. Similarly, the greater the term specificity, ie, the more precise the definition of each term, the less chance there is that non-relevant items are retrieved, leading to higher precision. Thus, generally, an increase in indexing exhaustivity improves recall whereas an increase in term specificity leads to better precision. Hence optimum levels of both notions are desired (Salton and Yang 1973). However, Maron(1979), in his analysis on indexing exhaustivity (termed 'depth of indexing' in his context) concluded that the notion is not a central issue in the design of effective document retrieval systems.

2.4 Clustering

It is often useful, either for reasons of economics or search effectiveness, to break down a document collection into smaller groups containing documents which are 'similar' to one another. Clustering is a classification technique for that purpose. One aim is to group the documents so that those which are likely to be relevant to the same query will be together in the same group - called cluster - hence improve search effectiveness (Van Rijsbergen 1978). Secondly, if clustering is not performed, then finding the set of documents relevant to a given query may require searching the entire file which may not be very economic. However, if the items are already separated into clusters, the set of documents relevant to the given query may be obtained by searching the documents in only a few groups in the clustered file.

Generally, a clustered file consists of a tree structured or hierarchical directory to the file of documents. Clusters are represented by the nodes of the tree and the individual documents are represented by the leaves of the tree. The nodes contain cluster representatives which, in some way or other, define the typical properties of the documents which belong to the cluster represented by that node.

To search a clustered file, the query is compared with the cluster representatives level by level, using some similarity measure, starting at the top of the hierarchy. One then descends to the next level within that cluster and finds the clusters that best match the query. If the match is not as good as that at the previous level, then items are retrieved from the previous cluster, otherwise, one continues until no better match is found. This process is known as the top-down approach, in contrast to the bottom-up search in which the search is started at the document level of the hierarchy (ie, the leaves of the tree) and clusters at higher levels that include the document are compared.

Two distinct approaches to clustering may be identified; where the clustering method proceeds directly from the object descriptions to make the clusters (Dattola 1969) and where the clustering is based on a measure of similarity between every pair of objects to be clustered (Augustson and Minker 1970, Van Rijsbergen 1979). Because the similarity measure has to be computed for every pair of objects in the file, the number of operations involved for a file of n items is of order n^2, rendering this approach quite an expensive one. Many hierarchical cluster methods, however, are based on this initial measure of similarity. Such a measure assigns a numeric value to the extent to which a pair of objects are similar to, or resemble, one another.

One of the approaches in use is the single-link method. This technique produces a hierarchy of non-overlapping clusters with associated numeric levels. Clusters are regrouped together repeatedly until the hierarchy of many levels of clusters is obtained. The levels are values at which a cluster splits into several other clusters. The higher the level, the fewer the clusters produced.

Jardine and Sibson(1971) showed that the single-link hierarchy has some properties which are quite important for operational systems (properties generally required of good practicable clustering techniques) :-

- The hierarchy is unlikely to change in a dynamic environment, where new items are added and some old ones deleted.

- The method is stable in the sense that small changes in the indexing of the documents lead to only small changes in the classification.

- The hierarchy produced is independent of the order of presentation of the input items being clustered.

- A non-overlapping hierarchy of clusters is produced in which clusters of very different sizes may occur at any one level .

Two approaches to clustering that try to avoid the expensive similarity matrix computation are due to Dattola(1969) and Rocchio(1966). In the Dattola method, the set of all documents is arbitrarily divided into a number of groups, each group represented by a centroid (a cluster representative, so to speak). The documents are examined sequentially and each document is assigned to the groups whose centroids are sufficiently close to it. If a document happens not to be close to any of the existing centroids, then a new group is formed. After all the documents have been assigned, the centroids are recomputed. If two centroids are found to be too similar, their corresponding groups are merged. The whole process is repeated until no reallocation of documents is necessary; ie, every document is assigned again to its group. In this technique, the number of operations involved in clustering n documents is of the order (n log n), since each document is compared to log n centroids in each iteration. The clusters produced are order-dependent.

In the Rocchio type method, each document is examined serially and some are chosen to be centroids if they satisfy certain criteria. Rocchio(1966) determined the suitability of each document as a centroid of a cluster by testing if there are sufficiently many other documents close to it. The clusters produced here, are overlapping and order-dependent and the number of operations involved for clustering n items is of order n^2, since determining whether or not each document is suitable to be a centroid already requires that many operations.

Various other approaches to, and uses of, clustering techniques have been made during the last two decades (Eitzweiler and Martin 1972, Preece 1973, Van Rijsbergen and Croft 1975, Becker and Pryce 1977, Yu and Rhagavan 1977).

Owing to the fact that the computation of similarity matrices is expensive, the view has been expressed that most clustering methods which require such computations would be impracticable for use with very large document collections (Williamson 1974, Salton 1975). In view of this, Croft(1977) proposed a method for large-scale clustering using a single-link algorithm and an inverted file. He noted that the number of computations required using the inverted file approach may be considerably reduced if the lists of the most frequent items are not used when calculating the similarity coefficients. Croft used the method to reduce a dissimilarity matrix (for 11613 documents) which should contain over 67 million entries to under 9 million. Harding and Willet(1980) pointed out, however, that the decrease in computation time will be paid for by the fact that the resultant classification will not be the same as that which would be obtained if all the coefficients had been determined.

An aspect of the research reported in this thesis involves the clustering of the database into single-level non-overlapping clusters. Each user interaction would then require only the appropriate clusters.

2.5 The Concept of Relevance

Relevance is one of the most central concepts underlying the information retrieval process. At some stage during a search, it must be determined whether or not the material judged by the system to be relevant to the user's needs and hence retrieved by the system is really useful to the user who presents the query.

That the concept of relevance is very fundamental to information retrieval is reflected in the diversity of meanings for it in the literature (Cooper 1971, Belzer 1973, Kemp 1974, Saracevic 1975, Swanson 1977, Robertson 1978, Bookstein 1979). Earlier researchers in the field of systems evaluation or design viewed relevance as a property of the document (Taube 1955). The general assumption was that the intellectual content of a document was invariant and could precisely be represented, for retrieval purposes, by descriptors. Hence, the relevance assessment was made by a member of the research team instead of the user.

However, considering the fact that the basic function of an information retrieval system is to satisfy its patrons, the concept of relevance needs to be viewed in the context of the person needing information and coming to the system. Relevance may be considered as a relation between the user, at the time she realizes a need for information, and a document. A document may be said to be relevant to that person if she feels the need that brought her to the retrieval system is fully or partly satisfied by that document (Bookstein 1979).

With this view, the relevance of a document to the user may be based on the judgement that the user makes about her satisfaction with the document, with respect to her information need. Unfortunately, an information retrieval system cannot predict with certainty the user's reaction of a document. As an alternative, the system attempts to quantify the relevance relation by first transforming both the document and request into representations it can manipulate, computes the similarity between these query and document representations, and on the basis of this tries to predict the relevance of the document to the user.

Saracevic(1970, 1975) suggests that the notion of relevance be viewed as a communication process between a source and a destination. Relevance was defined in terms of the system's decisions - the "system's view". Bookstein(1979), however, feels that this view cannot be used profitably in discussions about information retrieval, for, there would then be no 'false drops' as everything retrieved by the system would be relevant, by definition.

Bookstein also noted that to define relevance as being determined by the user, the user's request may be seen as a substitute for her; a substitute which the system can manipulate to predict whether or not any document will be judged as relevant by her. Hence the notion that "a document is relevant to a request" may only be interpreted to mean that "the document has a high probability of being relevant to such patrons as would utter this request" (p.270), since it is the user, not the request that is being served by the system.

Lancaster(1979), on the other hand, uses the term relevance to refer to a relationship between a document and a request statement, and 'pertinence' to refer to a relationship between a document and information need of the particular user - a distinction adopted by Foskett(1972) and Kemp(1974). Kemp noted that for some purposes of system evaluation, relevance decisions suffice, for other purposes, however, it may be necessary to obtain pertinence decisions. Kemp considers relevance decisions as being public and objective (cf. Lancaster 1979) whereas pertinence decisions are private and subjective.

Cooper(1971) and Wilson(1973) describe the difference between what they call 'logical relevance' and 'utility'. These correspond respectively to Kemp's relevance and pertinence. Cooper notes that logical relevance "has to do with whether or not a piece of information on a subject has some topical bearing on the information need", whereas utility is a concept which "has to do with the ultimate usefulness of the information to the user" (p.20). Swanson(1977) clarifies the distinction between relevance and utility, using the different concepts of aboutness (Maron 1977) and usefulness. The term relevance, however, has often been used to cover both usefulness and aboutness.

It is also possible to think of relevance from the viewpoint that different documents may satisfy the information need of a user to different degrees. Robertson(1976, 1977r), for example, describe a formal model of relevance in which it is assumed that, instead of a dichotomous relevance decision (relevant/non-relevant), there is underlying any statement of relevance, a continuous scale - ranging from high relevance to non-relevance - on which individual texts are positioned. Thus, relevance is regarded as a partitioning of this continuous variable which he called 'synthema'. In a later paper (1979), Robertson went on to generalise his synthema idea (which refers only to one particular user query) to describe the relation between different queries and documents. He ended up relating relevance to multi-dimensional document spaces.

The above discussion illustrates how diverse the opinions expressed in the literature are on the concept of relevance. The view shared in this thesis, however, is that the user should be the final arbiter regarding the relevance of documents retrieved for her. For, it is the user who recognized that she has an information need, and hence only she can tell whether or not the retrieved item fully or partially satisfies her need. For purposes of evaluation of an information retrieval system, since the main aim of the system is to satisfy its users' needs, the effectiveness of the system needs to be considered in terms of how well its prediction agrees with the ultimate user-judgement on the relevance of the retrieved item, ie, the closer the system's models are to the human methods the better the performance (Koll 1981).

2.6 Relevance Weighting and Feedback

In a retrieval system, there is a document collection in which each document is described by a set of index terms (assumed to portray what the content of the document is about) and a number of queries described in a manner similar to the documents. A simple retrieval strategy involves matching each query against each document in the collection to obtain those documents which best match the query. This strategy - simple coordination level match - however, attaches equal importance to all the document representatives, ie. the index terms.

The strategy may be improved upon by assigning weights to the individual terms entering into the document-query matching process, so that some of the terms are taken to be more important than others. In retrieval, on matching a query and a document, if it is found that they share a certain number of terms, the matching score for the document is then the sum of weights of those terms. The documents may then be ranked according to their sums of weights and the most highly ranked documents are retrieved. This approach is described as the notional coordination level matching. (When all the weights are unit weights, one effectively obtains the simple coordination level match.)

Robertson and Sparck Jones(1976) noted the hand-in-hand relationship between weighting and document ranking. The assignment of weights to index terms is usually regarded as separate from the formulation of a matching rule for document ranking. However, it is indisputable that in order to derive a term weighting function, one has to assume that the matching values would comprise the sum of the weights of the matching terms. And these values will, in turn, be used in ranking the documents (Robertson 1977p).

Various approaches to assigning weights to index terms have been investigated and experimental results have shown it to be useful (Cagan 1970, Sparck Jones 1973, 1979, Robertson and Sparck Jones 1976, Sparck Jones and Webster 1980). Weights may be assigned to terms based on either user judgements - a user may be more interested in documents with a particular term than those with some other term - or on statistical information, eg. the number of occurrences of a term in a document, in which case, it is assumed that the frequency of occurrence of the term will simulate the user judging the term as more or less important in the document.

Generally, higher values are assigned to terms which may help discriminate relevant from non-relevant documents. Improvements have been obtained using statistical weighting schemes in which terms with medium to low collection frequencies (ie. the number of documents in a collection containing the particular term) are assigned high weights as good discriminators, while frequent terms, on the other hand, have low weights.

Apart from the frequency of a term in a document or request, other information may also be exploited in the derivation of term weights, eg., information about the number of terms in the document, as well as the number of documents in which a term occurs. Sparck Jones(1973) examines the logic and effects of term weighting. In connection with the presence of terms in documents, she noted that the occurrences of a term in a short document are more significant than its occurrences in a long document and the occurrences of a rare term in a document are more significant than the occurrences of a frequent term.

At about the same time, similar work was being done in the USA. Salton and Yang(1973) examine various aspects of statistical term weighting and report experiments designed to find out how different forms of weighting affect retrieval performance and whether the same forms of weighting are optimal for different collections. They discuss two applications of weighting; weighting may be used in request-document matching or with a cutoff to determine which terms should be removed altogether from document descriptions (Svenonius 1972).

Various functions have been used to determine weights to be assigned to index terms (see Robertson and Sparck Jones 1976). In addition to the use of information about the distribution of terms in documents in the collection, Miller(1971) suggests that request terms be weighted to take into account their distribution in relevant documents as well as all the documents in the collection. In this case, the same term may have different weights in different requests.

Robertson(1974) noted that Miller's idea is a logical extension of the simple term weighting scheme based on collection frequencies only, as investigated by Sparck Jones(1972). He termed the Sparck Jones function as 'term-specificity model' and the Miller function as 'term-question specificity model', having differentiated between the notions of term-specificity and term-question specificity. The former is concerned with how exact a term is within the context of an indexed document collection, regardless of any questions that might be put to the system, and is measured by the frequency of occurrence of the term; the less frequent the term, the more specific it is. The latter, on the other hand, deals with the exactness of a term with respect to a particular question, and is measured by relating the frequency of occurrence of a term in a document collection to its frequency in the subset of the collection which is relevant to that question.

Sparck Jones(1975) noted that weights of the type proposed by Miller, which incorporate actual relevance frequencies, can be used to obtain an optimal performance for a given set of queries, documents and relevance judgements. And it has been proved that a similar weighting scheme exploiting the relative frequencies of request terms in relevant and non-relevant documents can be expected to be superior at every recall level to a simple unweighted system (Yu and Salton 1976).

Most of the statistical weighting functions assume independent distribution of the terms within the document collection as a whole or within the relevant and non-relevant subsets. Van Rijsbergen(1977) argues that index terms are most unlikely to be independent. He constructed a probabilistic model which incorporates dependence between index terms, deriving the extent to which the terms depended on one another from the distribution of occurrences in the whole collection and in the relevant and non-relevant document sets. He obtained a non-linear weighting function and pointed out that a linear function may be deduced from it, as a special case, when the independence assumptions are incorporated. Experiments reported later by Harper and Van Rijsbergen(1978) confirmed that index terms are not independent and that the use of relevance information coupled with dependence information could potentially improve retrieval effectiveness.

Closely associated with relevance weighting is the notion of relevance feedback. Feedback is a concept which has been developed by general systems theorists and workers in cybernetics in order to allow the past performance of a system to affect its future performance. The concept involves controlling the system by reinserting into it results of its previous performance.

In information retrieval, feedback attempts to ensure three main things: that more relevant documents are retrieved, that less false-drops are obtained and that the order of occurrence of retrieved documents is improved, ie. more relevant items come before the less relevant ones. It has long been known (Lesk and Salton 1969) that interactive search methods, in which the user influences the retrieval processes by providing appropriate feedback information during the course of the search operations, can be used profitably in a retrieval environment. Some of the feedback methods, including in particular, relevance feedback, provide important improvement in retrieval performances (Ide 1971).

The process of relevance feedback uses relevant judgements made by the user on documents previously retrieved by an initial search, in order to construct an improved query which can subsequently be used in a new search to improve on the result of the previous search (fig 2.1).

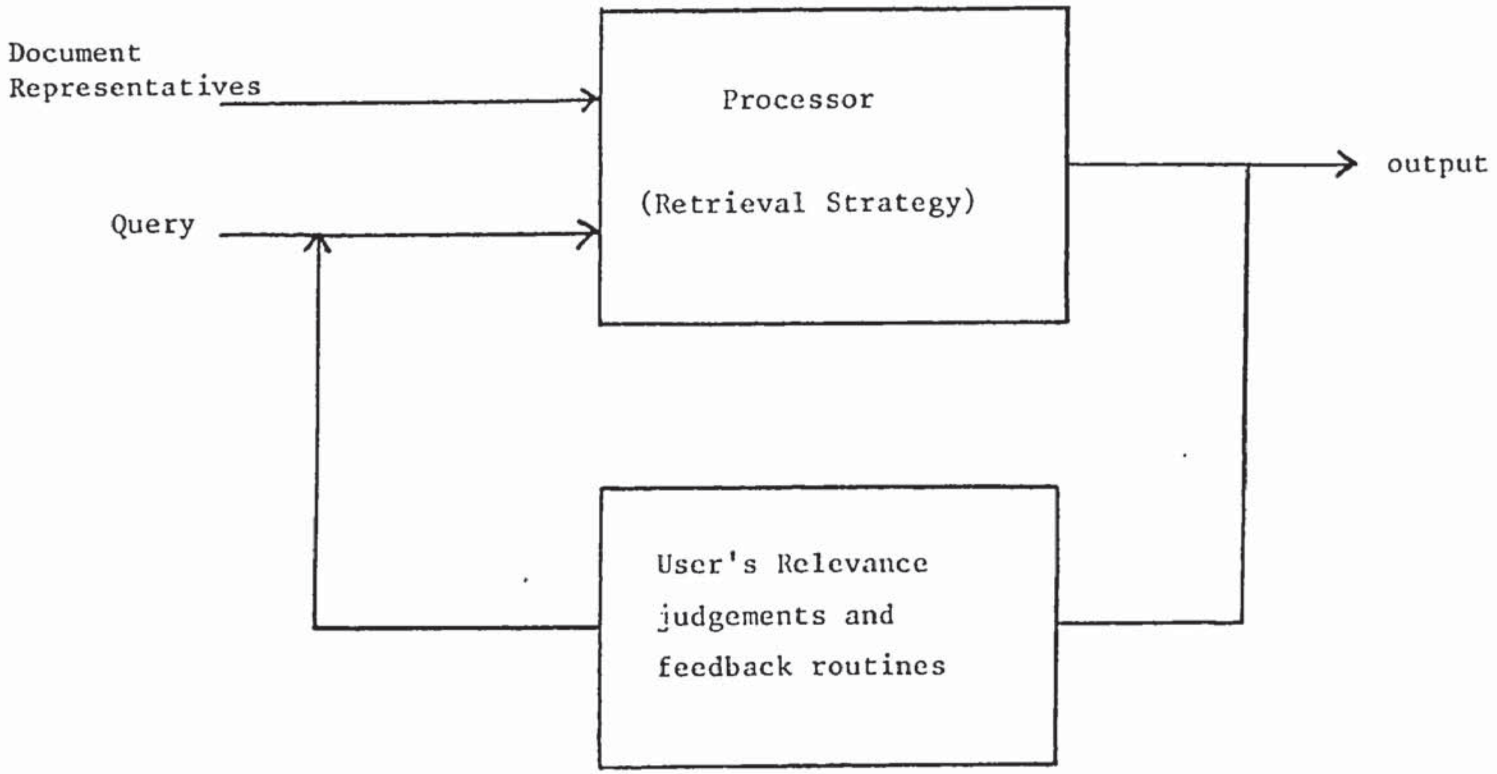


Figure 2.1 Relevance Feedback in IR

Specifically, an initial search is carried out for each query received, and a small amount of output, consisting of some of the highest ranking documents, is presented to the user. The user then examines this retrieved output and identifies each document as being either relevant or non-relevant to her purpose. These relevance judgements are later returned to the system, and used automatically to modify the initial search query in such a way that query terms present in the relevant documents are promoted, for example, by increasing their weights, whereas terms occurring in the non-relevant documents are demoted by decreasing their weights.

Consequently, an altered search query is produced which may be expected to show greater similarity with the relevant documents in the collection as well as greater dissimilarity with the non-relevant ones. The altered query so formed, can next be submitted to the system, and a second search performed using this new query. If the system performs as expected, additional relevant material may be retrieved, or in any case, the relevant items may produce a greater similarity with the altered request than with the original. The newly retrieved items can again be examined by the user, and new relevance assessments can be made and used to obtain a second reformulation of the query. The process can be continued over several iterations until such time as the user is satisfied with the results obtained.

Salton(1973), comparing the effectiveness of SMART-type and Boolean-type systems, achieved dramatic improvements in system effectiveness using automatic relevance feedback. He cites results for the SMART system without relevance feedback, equivalent to those obtained for the MEDLARS system, but with an improvement of up to 30 per cent when feedback techniques were added. (A SMART-type system defines a measure of association between a query and each of the documents in the system, ordering the documents for retrieval by the magnitude of the association measure. Index terms in both the query and the documents are weighted, and the measure of association used for retrieval is calculated using these weights. Any terms in the query that cause non-relevant documents to rank high, have their weights decremented while terms causing the high ranking of relevant documents have their weights incremented.)

Attempts have been made to incorporate relevance weighting and feedback mechanisms into Boolean systems (Angione 1975). Rickman(1972) augments the original Boolean expression by terms found in documents identified by the user as being relevant, and removes terms appearing in non-relevant documents using a set-difference operator (cf. Mitchell et al 1973). Noreault et al(1977) have attempted to enhance a Boolean retrieval by ordering the retrieved set of the documents (Vernimb 1977). Dillon and Desper(1980) have recently described an automatic reformulation of Boolean queries based on user's relevance judgements of an initial retrieval. They calculated weights for terms in the retrieved documents, ordered the terms with these weights and then used them to construct a new Boolean query.

These attempts have, however, been limited because of the difficulty of modifying a Boolean query. If, for example, an inappropriate word is combined using the conjunction (AND), the retrieved set could easily be empty. The ability of the Boolean query form to exclude documents (Section 1.3.1), also requires that greater care be taken in the query modification than would be necessary with other query types (Bookstein 1978).

The project described in this thesis is an experimental feedback system, which however, does not strictly follow the trend described above, as it does not involve formal query modifications. Relevance feedback with Thomas-II, involves the dynamic modification of the structural model that Thomas-II creates of the user's area of interest. This modification requires relevance judgements from the user on earlier outputs. Rejections from the user lead to a shrink in the model, whereas further suggestions from the user may demand enrichment of the model with new nodes. The system also involves a form of weighting mechanism which does not attach weights to individual terms, but to pairs of terms, signifying the strength of association between the pairs - 'association strengths', we may call them.

2.7 Evaluation

There are various reasons for evaluating the performance and effectiveness of a retrieval system (Swanson 1975). It may be social; to determine whether one wants a particular system or not, economic; whether the particular system will be worth the money spent on it, scientific; to understand a retrieval model being tested, or otherwise. Keen(1971) suggests the classification of the need for evaluation into three types: internal, external and real-life.

In the case of internal evaluation, the performance of a particular system variable needs to be assessed; the document collections, search queries and relevance decisions are held constant while this variable is altered. The need for external evaluation arises when one system is to be compared with another. The third category involves attempts to interpret experimental results of a system in terms of its expected merit in real-life situations rather than merely comparing different strategies in the laboratory.

The choice of parameters used to assess retrieval performance varies and is affected by who makes the assessment; either the end-user who is concerned with how much the system satisfies her needs, or the researcher who is just seeking fundamental insight into the retrieval capabilities of the system.

Generally, evaluation parameters are based on a 2x2 contingency table (table 2.1), which distinguishes between the documents retrieved in answer to a given query and those not retrieved, as well as between documents judged to be relevant to the query and those not relevant. Farradane(1974), however, feels that since users rarely make clear decisions of relevant and non-relevant items, the 2x2 contingency table may not be considered adequate, and hence the parameters based on the table may all be unsatisfactory.

	Relevant	Not Relevant	
Retrieved	a	b	a+b
Not Retrieved	c	d	c+d
	a+c	b+d	a+b+c+d

Table 2.1 2x2 Contingency Table

Four common evaluation measures derived from the above table are recall, precision, fallout and generality.

The expressions for these measures are:

Recall $= \dfrac{a}{a+c} =$ proportion of relevant items actually retrieved

Precision $= \dfrac{a}{a+b} =$ proportion of the retrieved items actually relevant

$$\text{Fallout} = \frac{b}{b+d} = \text{proportion of non-relevant items that are retrieved}$$

$$\text{Generality} = \frac{a+c}{a+b+c+d} = \text{proportion of relevant items per given query}$$

The user's satisfaction depends mainly on the sets a, b and c, since she is interested in examining as few non-relevant items as possible and as many relevant items as she wishes to see. She does not concern herself with d (the non-relevant items not retrieved) or the total collection size, both of which from the viewpoint of the researcher trying to determine the capability of the system are essential.

Each of the above performance measures is primarily defined for each query. However, there are methods for averaging the measures over a complete set of queries (Rocchio 1971) and for suitably displaying the results in the form of precision-recall or fallout-recall graphs (Keen 1971). These graphs are then expected to show the performance of the entire system for a given set of users.

Of the four measures listed above, recall and precision have been the pair most widely used. Performance evaluation is usually based on recall-precision graphs. A recall-precision graph is obtained by matching queries and documents, and ranking all the documents in decreasing order of a document's similarity with each query. Precision values are then computed at fixed recall levels (usually 0.1, 0.2, 0.3, etc) for each query and the resulting values are averaged for the given set of queries. Where more than one recall-precision graph, ~~are~~ is shown in the same figure, each for a different technique, the curve closer to the upper right hand corner

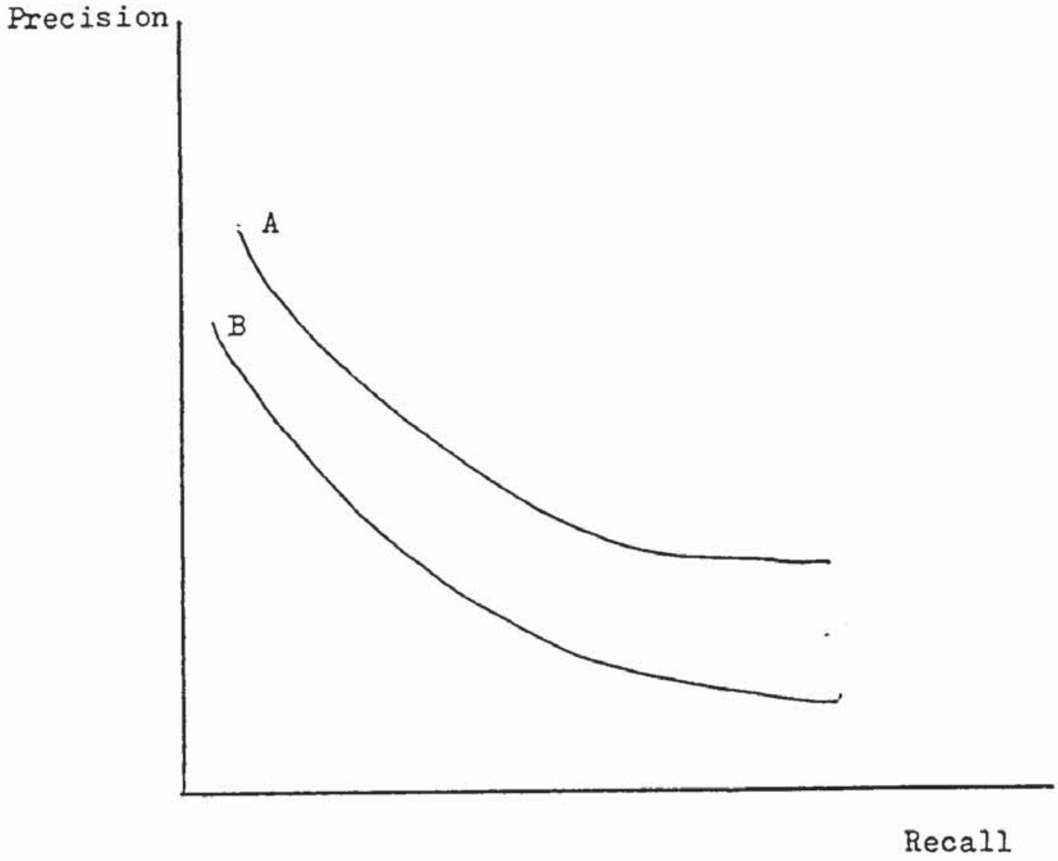


Figure 2.2 Precision-Recall Graphs
(showing system A better than system B)

(recall = precision = 1) reflects the better performance (fig .2.2).

Since recall indicates the proportion of relevant documents actually obtained from a search, while precision measures the efficiency with which these relevant items are retrieved, a recall-precision output is considered to be user-oriented, in that the user is normally interested in optimizing the retrieval of relevant items. On the other hand, fallout is a measure of how well non-relevant items have been rejected and includes, as a factor, the total number of non-relevant items in the collection - which, in turn, is approximately equal to the collection size, since very few documents in the whole collection will be relevant to a given query. For this reason, a recall-fallout graph is usually considered to be system-oriented, as it indicates how well the non-relevant items are rejected relative to the collection size.

Although performance evaluation has been mostly based on recall and precision, there have been various arguments over the use of this pair of parameters over the years (Robertson 1969, Cleverdon 1972, Cooper 1973, Guazzo 1977). Cleverdon(1974) pointed out that the parameters (recall etc.) were devised for experimental tests in artificial environments which at some stage of the tests, must be held constant.

Cleverdon noted that recall and precision ratios have the underlying assumption that the user wants ALL and ONLY the relevant items in the collection and stressed that this is not really the case in practice. Even though many users will like a 100 per cent precision ratio, the majority of users will settle for a precision ratio much below the theoretically optimum level. With respect to recall, it is very rare that a user does require maximum recall of documents. Users are not interested in seeing every citation on the subject of their search; usually three good papers suffice (Cleverdon and Kidd 1976).

In the light of the above discussion, Cooper(1973) suggests that the best measure of a retrieval system's effectiveness would be the user's subjective evaluation of the usefulness of output to her from the system, provided that this could be properly quantified (Boon 1978).

Chapter 3

SIMULATION OF HUMAN COGNITIVE PROCESSES

Since the evolution of computer technology, computers have been used to solve various problems ranging from strict numerical work to the simulation of very complex systems and other human cognitive processes. However, regardless of the fact that most of the problems put to the computer involve some psychological aspects, the integration of concepts in psychology and computing has been rather minimal. Some researchers have, however, realized this and made various recommendations about the need to further our understanding of human behaviour in order that we may build computer programs to simulate such behaviour. Before considering the 'symbiosis' of man and the computer at problem-solving, we should consider what psychologists say of the cognitive art of problem-solving, as, broadly speaking, this is the objective of the symbiosis and usually the basis of all tasks put to the computer - whether it is a numerical problem, a problem of getting a checkmate on a chessboard or finding references to satisfy an information seeker's needs.

3.1 Problem-Solving

The process of problem-solving, according to some psychologists, is a search to relate one aspect of a problem situation to another, and it results in one's ability to comprehend how all parts of the problem fit together to satisfy the requirements of the goal. This may involve reorganising the elements of the problem situation into various states, well- or ill-defined, so that they ultimately help solve the problem. Restle and Davis(1962) pointed out that a problem-solver goes through a number of stages, solving subproblems at each stage. However, they based their work on assumptions that the various stages are independent and sequential, and that all the stages are equally difficult - assumptions which may not always be true (Thomas 1974).

There are those who link problem-solving with concept formation and cognitive structure (Driver and Streufort 1969, Cravens 1970). Bruner et al(1956) feel that the greater the number of various concept categories, and the higher the level of abstraction of the concepts the better a problem-solver an individual is. Berlyne(1965) contends that individuals differ in their ability to solve problems for reasons due to differences in the nature of their cognitive structures.

Miller et al(1960) are concerned with two concepts. They note that every individual possesses 'plans' which are "any hierarchical process in the organism that can control the order in which a sequence of operations is to be performed" and an 'image' which is "all the accumulated knowledge that an organism has about itself and its world" (p.16-17). They are interested in the relationship between one's image and plans. If the individual faces a problem that requires immediate solution, she would call on her simple plans; complex problems require complex plans. They suggest that the individual tends to adopt heuristic rather than systematic plans. While systematic plans guarantee a solution if one exists, the process may be tedious as all possibilities may have to be tried. A heuristic plan increases the chances of early success because searches are begun with solutions that appear likely.

Polya(1957) introduced a series of steps in problem-solving; understanding the problem, devising a plan, carrying out the plan and 'looking back'. The problem solver gathers information about the problem and tries to determine what is unknown. Next she tries to use her past experience of the world to find a method of solution by likening the problem to any related one met earlier. She then has a go at the problem and finally reflects back to check the result and try to determine if she can store the method of solution for use on other problems.

More recent views on problem-solving assume that a human being is, among other things, a processor of information. This line of thought is to pave the way for computer simulation of human cognitive processes (Wickelgren 1974, Mayer 1977). The cognitive processes of an individual are represented as either a sequence of mental operations performed in the individual's memory, or a sequence of internal states or changes in information that steadily progress towards the pursued goal. The aim of those who hold this view is to define precisely the processes and states that a particular subject is using to solve a particular problem and to be able to list the sequence of operations used. The list could then be used for a computer simulation (Winograd 1972, Zobrist and Carlson 1974, Simon and Newell 1976).

Ernst and Newell(1969) suggested four major components in describing problem solving by computer simulation; initial state, goal state, operators and the problem states. In the initial state, the given or starting conditions are represented. The operators are the allowable manipulations which may be performed on any one state to change it into another state. And the intermediate states that result from the application of an operator to a state constitute the problem states (cf. Polya's steps).

Lindsay and Norman(1972) agree with Polya that a subject may rely on past experience in solving problems. They distinguished among several types of relevant past experience used in problem-solving :-

- facts, which are immediately available to the subject

- algorithms, which are sets of rules which when given correctly, automatically generate the correct answers, and

- heuristics, which are general plans of action.

Greeno(1973) proposed a memory model for problem-solving with the main components being a short-term memory, a long-term memory and a working memory. The external description of the problem is input into the short-term memory while the long-term memory is used to store past experience with related or unrelated problems. The information from the short-term memory and long-term memory interact in the working memory where a solution is attempted (Feigenbaum 1970, Mayer 1977).

All the above models indicate in various ways what an individual does while solving a problem. These processes may be specified in quite exact terms as a list of things. Theories may then be generated of these cognitive processes and expressed as computer programs and tested to see if they work the way a person does (Malhotra et al 1980).

Quite recently, it has dawned on some information scientists to give intensive consideration to human cognitive processes in their work (De May et al 1977, Harbo and Kajberg 1980). Hitherto, most information systems have been designed without regard to the fact that individuals differ in their cognitive structures, and that these differences would affect the way they seek, select, and use information. Davidson(1977) notes that if systems could be designed to take individual differences into account, they would serve as more effective aids in decision-making and problem-solving, where problem-solving in the information retrieval context includes the development of a search strategy and use of some inference mechanism.

Some aspects of the information retrieval problem involve various cognitive processes, as they entail decision-making and problem-solving. Harbo et al(1977) relate information storage and retrieval via persons playing four different roles; the author, the indexer, the user and the information specialist (Ingwerson et al 1980). A consideration of the manner in which a retrieval system selects materials is based upon how well the author expresses herself in her document, ie., how well she communicates with whoever reads her document (Saracevic 1975), and the way the document has been indexed.

The indexing process is an attempt to describe in a few terms, the subject content of the documents in such a way that these terms help an information seeker to locate the author's document. Lancaster and Mills(1964) used the term 'indexing' to denote the "intellectual and other processes involved in deciding what a document or question really is about and then working out a description and tags for it which will ensure retrieval, no matter from what path the search approach is made" (p.4). Thus, when an individual indexes a document, she describes the intellectual contents of that document by providing labels - semantic or otherwise - for it. Since individuals differ in their cognitive structures, one would expect different people to determine what a document is about in slightly different ways; a reason for the inconsistency in manual indexing.

A retrieval rule is a basis upon which a retrieval system tries to simulate the human cognitive process of judgement. An information retrieval system, in an attempt to satisfy a user's needs, uses these rules to ~~decide~~ predict whether or not a document is relevant - a decision process which would otherwise have been undertaken by the user herself if she could scan through all the documents in the collection at large. How well the system is at deciding the relevance of a document must be related to how well its simulation process agrees with the user's own judgement (Koll 1981).

Last, but not the least, is the user of the retrieval system. The user has to realize her need for information. (How often do people lack information and yet are unaware that they do!) Having realized her need, the user either goes for a browse or approaches an information specialist. She then tries to put across her problem area. How well her need becomes satisfied ultimately, depends partly on how well she communicates her problem area to whoever will help her. During the search for information to satisfy her, she still has to pass judgement on any piece of item given to her - as to whether or not she deems it relevant to her need. Individuals would differ in some or all of the above processes according as their cognitive complexities differ.

3.2 Problem-Solving Involving Man-Machine Interaction

We have seen that more recent views on the art of problem-solving are geared towards paving the way for the use of the computer in the simulation of complex processes. What is becoming very common these days is the aspect of problem-solving which involves a direct interaction between man and the machine. This has been mainly due to the advent of cheap computers with bountiful storage and high processing speeds, coupled with the fascination of man communicating with a machine. Basically, when man and the machine are in an interaction, both participants are linked via a communication channel eg. in a computer-based interaction, a visual

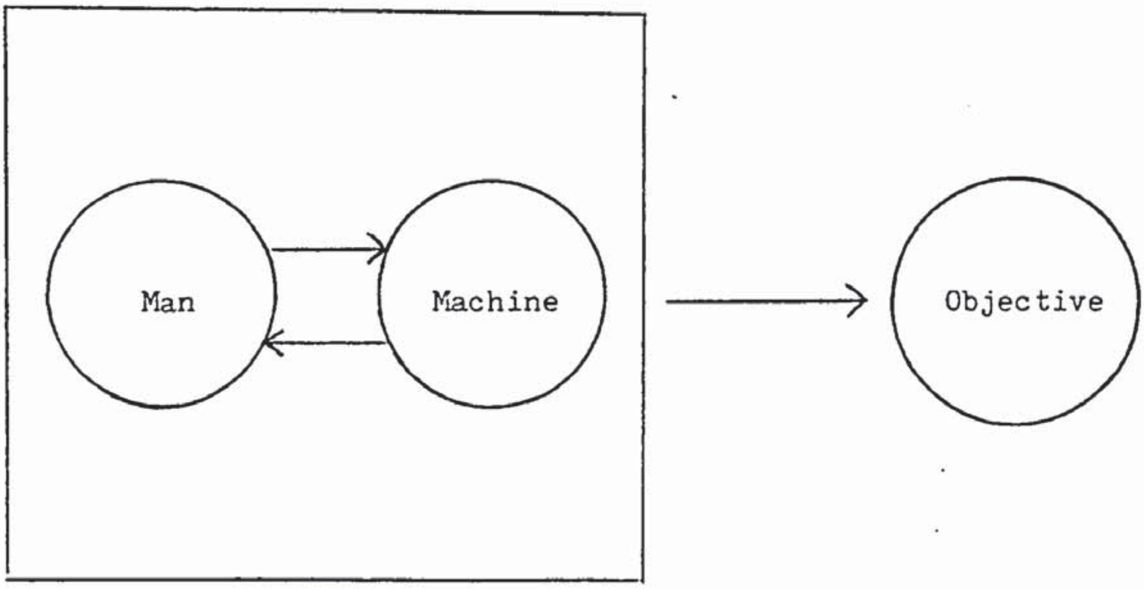


Figure 3.1 Man-Machine Interaction

display unit. Each interactive action changes the state of the system, and both participants depend on each other in the changing states so as to move towards a particular objective (fig. 3.1 above).

Usually, man initiates the interaction with a move, to which the machine responds. This response stimulates the man to take another action which, in turn, causes the machine to react. The interaction continues until either

a) the specified objective has been achieved

b) the machine ceases to function or

c) the man reaches a stage of fatigue beyond which he no longer wishes to act.

It is generally desirable that the end of the interaction is prompted by condition a), ie, the achievement of the specified goals.

Ting and Badre(1976) pointed out some conditions they consider necessary for a computer-based man-machine system :-

i) The system must be a purposive one. This condition requires that the purpose be predefined by the system's designer and that the system be used by the man for this predefined purpose.

ii) The two participants must be linked in a direct and closed loop. The requirements here include physical contact between man and the machine via an on-line device, some artificial language to enable a two-way communication and quick response from the machine in a form understandable to the man. Fitter(1979), on the other hand, suggets that a natural language would make more explicit the knowledge and process for which the man and computer share a common 'understanding'.

iii) The interaction must be man-centred with the man active and the machine reactive (a view shared by Gaines and Fãcey 1975). Press(1971), in an earlier view, pointed out that for the system to be balanced, neither man nor the machine should be in control, but that both be mutually active partners in a dialogue.

Over a decade earlier, Lickliuer(1960) had coined the term 'man-computer symbiosis' to explain an anticipated close cooperative effort between man and the computer. He identified this symbiosis as the goal of computer system design, and improved communication between man and the computer as the key to that goal. Licklider proposed an assignment of responsibilities to man and the computer in which, generally, the computer would carry out routine clerical tasks during the times between man's decisions (Roy 1980). Man is accredited with imaginative and innovative mental functions which, in turn, depend on his capabilities for making plausible inferences even when supplied with incomplete information, and the computer is dependable for storage and processing speed (Horman 1971). The notion is given that when advantage is taken of these various attributes, man-computer symbiosis could lead to more effective thinking and problem-solving.

Since the time Licklider presented his paper, there has been a variety of work on the interaction between man and the computer (Sackman 1970, Pulfer 1971, Gaines and Facey 1975, Fitter 1979).

3.3 Man-Machine Interaction as Dyadic Communication

There are those who feel that however much knowledge is contributed by research on man-machine interaction from computer scientists, psychologists, ergonomists, etc., we still need more behavioural oriented research (Martin and Parker 1971). Foley(1973) feels that research in communication would have a major contribution to make in the design of information systems and Meadow(1970) traces the history of man-machine communication and pointed out that, specifically, the field of automatic information retrieval could well involve the study of conversation between man and the computer.

Chapanis(1975) also feels the need for conversational computers that will interact with their human users in a natural language, but believes that if we are to know how to build such systems, we first need to know how people communicate with each other. Others feel that despite the variety of existing systems, there is still a lack of basic design information regarding man-computer interaction, to develop fully conversational retrieval systems (Vaughan and Mavor 1972, Pearce and Easterby 1973). Nickerson(1969) contends that man-computer interaction is different from the more general class of man-machine interaction and that "it may be described, without gross misuse of words, as a conversation. That is to say, the interaction involves a two-way exchange of information in the form of commands, requests, queries and messages of sundry sorts" (p.504).

Colby and Enea(1967) examine associational conversational programs as they might relate to the construction of cognitive models for both man and the computer. With respect to both simulating human information processing and augmenting human creative process, Barmack and Sinaiko(1966) regret that we are unlikely to bring about in the computer the human phenomenological experiences associated with perception and cognition, mainly because we do not know precisely how these occur in human beings; the constituents of cognition, unlike the constituents of computer information, are unknown. Whereas one can hardly find any approach that has contributed findings of practical value to enhance human creativity, a cleverly programmed computer may help direct the user down certain paths and caution her where there are constraints and faults.

The motivation for the study of cognitive processes has been attributed to man's aim to understand psychological processes in order to advance machine intelligence, and various attempts at computer simulation of human cognitive processes have been made since the late 'fifties (Feigenbaum and Feldman 1963, Newell and Simon 1972). However, many of these computer simulations are only superficially related to actual human thought processes at any level. De Greene(1970) attributes this lack of relationship with our lack of understanding as to what psychological processes are involved, and a means of describing these processes in terms of concepts and languages with which the computer can deal. Huyck(1973) shares this view and points out that it is this lack of understanding of human behaviour that has led to our inability to build software responsive to that behaviour.

Various attempts at the simulation of dialogues, in particular, have long been reported (Jaffe and Feldstein 1970, Siegman and Pope 1972). In a more extensive manner, Martin(1973) surveyed different forms of interactive dialogue. He noted two basic methods of interaction; either user-oriented (where the user enters her own commands, terms or instructions) or computer-oriented (where the user merely responds to questions posed by the computer). Raitt(1978) argues that this latter method cannot be considered as truly conversational, since the user is not allowed to think for herself and pursue her own line of thought.

Ambrozy(1971) considers the notion of a dialogue as a situation in which at least two communicating partners participate in such a way that at least one sends a meaningful message and at least one receives a meaningful message - 'meaningful' being defined in terms of shared environment of the participants and the ability to change some portion of that environment. He notes the finiteness of the computer's environment and the fact that that environment is man-made, and asserts that if a machine is capable of causing a change in the environment of the message-recipient, then the machine may be considered as a dialogue machine.

There has also been some experimental work. McGuire and Stanley(1972) compared communication patterns of man-man and man-computer, and found similarities, indicating that simulation programs can accurately reflect human interaction. Earlier, Scheflen(1968) reviewed communication as a form of patterned or progammed behaviour and Colby(1967) noted that it should be possible to design computer programs to model that form of communication.

Slack and Slack(1972) examined the willingness of humans to converse with the computer on a personal level in a psychiatric interview and found that more people definitely preferred the computer to the human interviewer. Slack and Slack's experiment, considered in the context of human information seeking habits will pose the question as to whether some people are more likely to seek unknown information through an interactive retrieval system than from a personal source like a colleague or a reference librarian.

Penniman(1975) using a statistical model analyzed actual human-computer conversations in which users searched a variety of databases in an interactive mode. His aim was to obtain data on user interaction patterns for use in refining conversational retrieval strategies operating on interactive computer systems. His experiments indicated, among other things, that conversational patterns can be used to classify users of retrieval systems.

There have been others reiterating the call for man-computer interactions being more conversational (Gaines and Facey 1975, Fitter 1979) and some considering the interaction from other psychological aspects (Rouse 1981). However, the model of dialogue discussed by Hollnagel(1979) is of much interest here, as it forms a good basis for the interaction simulated in this research work. We present Hollnagel's model in the next section.

3.4 Dyadic Communication: Hollnagel's Cognitive Paradigm

Looking back at our general discussion on man-machine interaction, it may seem that any dialogue between two participants A and B in an environment E can be modelled by the diagram in fig.3.2 where information flows to and fro each participant.

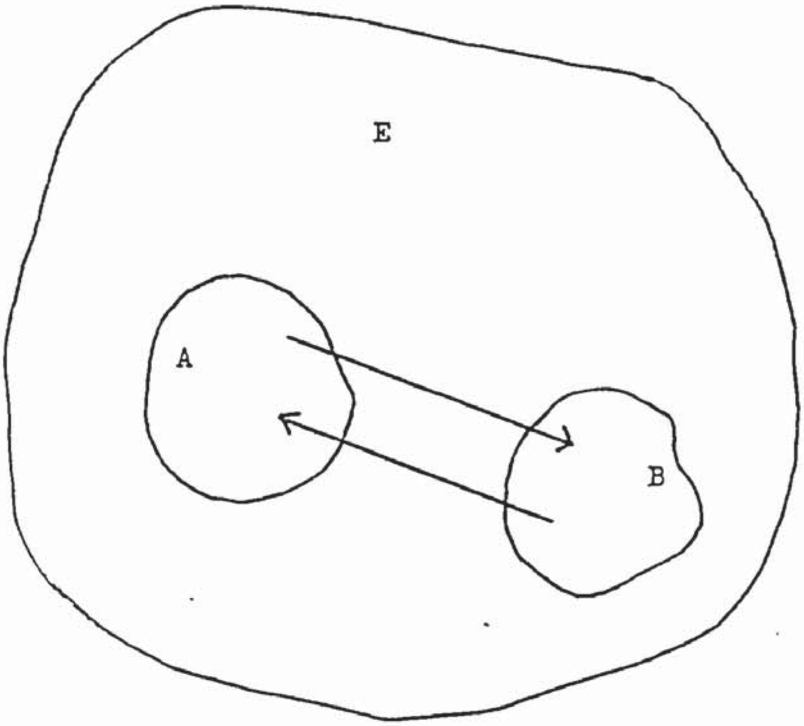


Figure 3.2 A first glance view of Dialogue

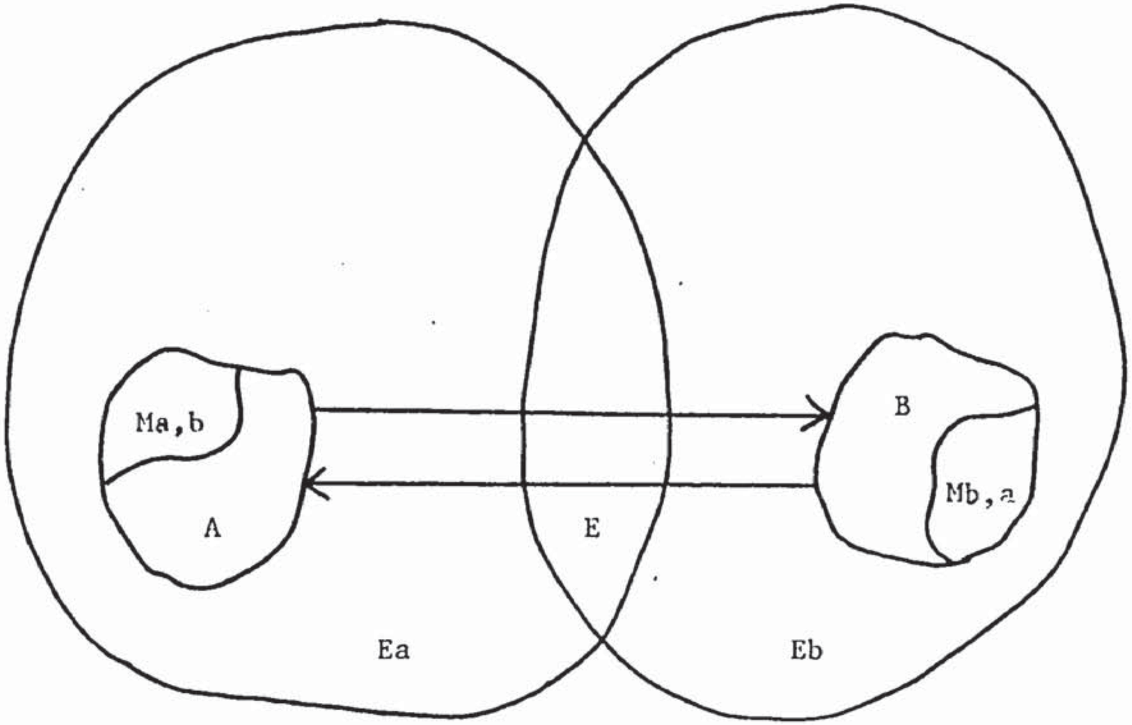
This model is, however, almost never met (Ambrozy 1971). For, regardless of how uniformly and adequately either participant perceives and interprets the environment, E, there is the likelihood of differences in their individual perceptions and interpretations. Furthermore, either of A and B is part of the other's environment and each of them perceives herself quite differently from how the other would, observing her from outside.

Hollnagel(1979) gives a modification to the above model and called it the 'cognitive paradigm for communication'. The main features of Hollnagel's model may be shown in fig.3.3. Two systems A and B communicate in a specific environment. Ea in the model represents the local environment of A including A's perception of B, (Ma,b), and Eb the local environment of B including B's perception of A, (Mb,a).

The object of a meaningful dialogue must necessarily be known to at least one of the two participants of the dialogue. Thus, things which do not belong to the environment of either A or B (ie. to Ea or Eb) are unknown to both partners and therefore cannot feature in the dialogue. As such, a condition for a dialogue must be that at least one of Ea and Eb must be non-empty. That is to say, there must be a part of the universe, U, which is known to either or both of the dialogue participants.

Also, there must be a part, E, of the environment which is common to both A and B, since a meaningful dialogue cannot take place unless A and B have an area of common knowledge with which to begin.

A third condition noted by Hollnagel is that the communication must convey some new ideas to at least one of the communicants - the new ideas being either something known to A but unknown to B, or vice versa. And the aim of the dialogue is to increment the shared environment, E, with either parts of Ea unknown to B or parts of Eb unknown to A, or both. The model is symmetrical, and so, if A is considered in the terminology of some communication researchers as the sender of a message and B the receiver, then the aim of the



Ex = X's knowledge space

Mx,y = X's model of Y

Figure 3.3 Hollnagel's Cognitive Model for a Dialogue

dialogue can be interpreted to be to increase the knowledge of the receiver, B.

3.5 Hollnagel's Paradigm applied to Thomas

With respect to this research, the participants in Hollnagel's model are the user and the retrieval program, Thomas-II. Each participant creates a model of the other, and their shared environment is a subset of the program's database. The ultimate aim of the interaction is to increment the user's knowledge with items that belong to Et and unknown to the user - where Et represents the local environment of the program Thomas-II, and coincides with its database.

We give a further discussion on the application of Hollnagel's model to Thomas in section 4.2.

Chapter 4

THOMAS-II: THE REFERENCE RETRIEVAL PROGRAM

4.1 Introduction

We have had a brief look at the ever-increasing growth of information (through reports, journals and books) in our society to-day. It has also been indicated that regardless of our society being so information rich, various people still yearn for enlightenment on various issues, with the effect that some decisions get made on the basis of less than adequate information and some aspects of research work do overlap. One must admit that this lack of information in an information-rich society is not due to the possibility that the particular information required is always unavailable. The problem rather seems to have arisen owing to our inability to identify, locate and retrieve the needed information from the mass of literature around us.

Also looked at are some of the various means which individuals have used in order to find information to satisfy their needs over the years; notably through information centres which supply card catalogues and the like for manual searches, automated systems which conduct batch searches and those which run interactive searches for users on-line. Their advantages and disadvantages have also been briefly stated. And it seems as though the present technological advancements have rendered interactive retrieval systems about the 'best' that an information seeker would require, with regards to the time taken for a search, the amount of money to be spent on it and the coverage of the search.

However, regarding the flexibility and ease of use of these on-line retrieval systems and consequently, the effort required of the users, it is easy to see that the 'best' of the current systems is still not good enough. The ease of use factor of a system was highlighted by Mooers(1960) in Mooers' Law. This postulates that an information retrieval system will tend not to be used whenever it is more painful and troublesome for a customer to have information than for her not to have it. In Cooper's terms, the customer may ask herself "Why bother?", if she doubts whether the results she will obtain from the system will be worth the effort she will be required to put into it (Cooper 1978b). It is therefore necessary that systems be designed in such a way that the effort required of the user is minimal.

As has been discussed earlier, most of the current systems require the user to formulate very rigid queries, because these queries will have to be matched against the documents in their collections. These systems disregard the fact that it is not easy for a user to fully recognize her information need, let alone to be capable of expressing it precisely. It is also assumed that the expressed needs (queries) presented by the users are identical fully with the users' information needs - thereby ignoring the needs that the user failed to recognize and express before the start of the search.

However, it is quite important to distinguish among an individual having an information need, her recognizing that she has the need and her expressing the need in the form of a query to a system. For, as the system cannot respond directly to the needs of the individuals, but only to expressions of the needs in the form of the queries, the degree to which the user is able to recognize the exact nature of her need, and the extent to which her need is accurately reflected in her expression of it very largely determine how successful the information system will be in attempting to satisfy the user. As noted by Lancaster(1979), one of the major problems faced by an information system is to ensure that expressed needs accurately reflect recognized needs.

One other drawback with current information retrieval systems has been pointed out by Koll(1981); that present systems still work with words and not concepts. He stresses the point that symbol matching - query against document - is not sufficient (Bar Hillel 1975, Belkin 1980). Significant progress will not be made until models can be built of how people come to understand the contents of documents or at least how they differentiate documents with regard to requests.

Belkin and Robertson(1976) had a foresight earlier into Koll's viewpoint. They stated: "... we can imagine document retrieval systems which make direct use of the idea of the recipient's image. A person making a request to a retrieval system does so because of a perceived gap or incompleteness or inconsistency in his image of the world: he is looking for texts that will help him correct that anomaly. A sophisticated retrieval system might then attempt to build a structural model of the requester's image, using clues provided both by the linguistic formulation of the request and by the requester's response to particular documents which the system retrieves (relevance feedback). This model would then be matched against the structural representations of the documents, to determine which further documents should be retrieved. Several recent developments in IR contain the germs of such a system (eg. Oddy 1974); further work along these lines may well be profitable." (p.203).

Oddy's experimental system (1974, 1977a,b), Thomas, is the basis of the present thesis. We next describe features of the system, all of which are contained in our enhanced version, Thomas-II, worked on for this thesis.

4.2 THOMAS: An Overview

In the above-cited references, (1974, 1977a,b), Oddy gives a description of a computer program called THOMAS. Although not developed into a fully operational system, Thomas has capabilities of handling most of the above-stated problems and pitfalls of retrieval systems to a remarkable extent.

Thomas (and for that matter, Thomas-II) has been designed with the awareness that:

a) Users are unable to fully recognize their information needs before approaching information retrieval systems.

b) Even on recognizing the need for information, most users, if not all, are unable to precisely express their need into formal queries.

c) Owing to users' inability to express their needs precisely, even to human intermediaries, the possibility exists that an intermediary contacted by a user may misinterpret the user's expressions.

d) There is the need for information retrieval systems to switch from the usual matching of queries against documents to modelling how users differentiate documents with respect to requests.

Accordingly, regarding a) and b), the system allows the user to input terms which she thinks are capable of expressing her need at the time of coming for the search (without any requirement of having to combine them in any way). She may later reject any, or all, of them, as well as add more if she does recognise some other aspects of her needs during the search. The program is intended to engage in a dialogue directly with the user - thereby eliminating the intermediaries as in c) - in a manner similar to a conversation between a subject expert and the information-seeker. With respect to d), the system does not match query terms against document surrogates as there is even no formal query. Instead, it creates structural models of its data base (its 'world model') and the user's problem area (its 'context graph'). These structural models are used in order to determine which documents may satisfy the user's information need.

The program, Thomas, is an interactive system which provides a browsing facility for the user through a man-machine dialogue. At a very high level, the dialogue may be described as follows, in what we shall call Box 4.1:

Box 4.1

i) User inputs her message

ii) Program uses message to create or modify its image of the user's area of interest

iii) Program responds to user based on its image

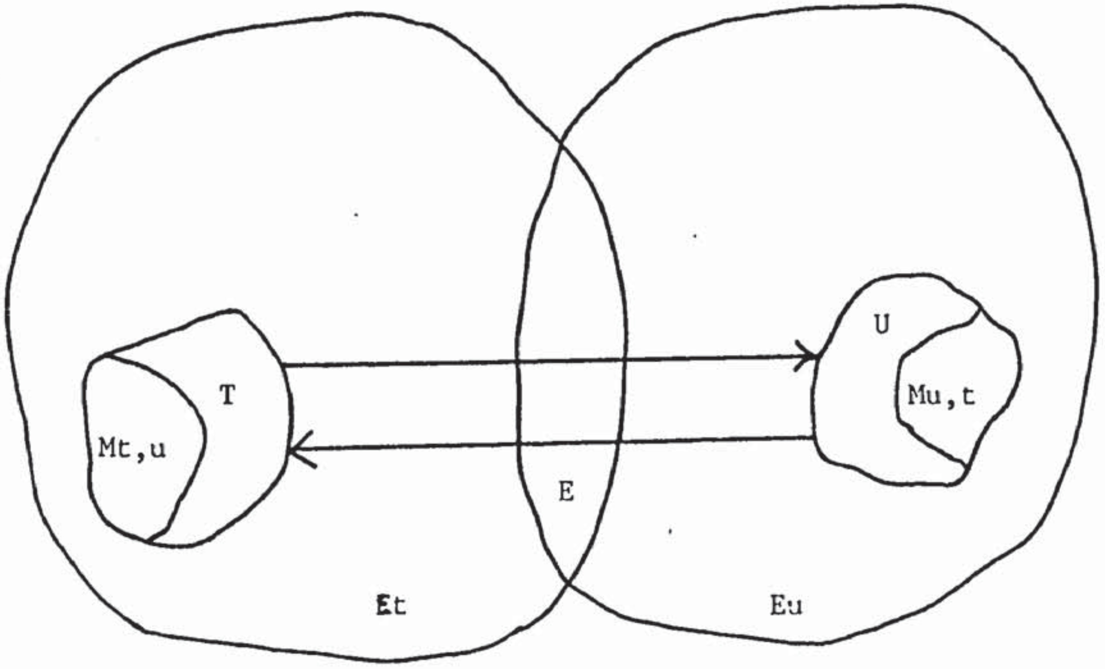
iv) User makes judgement

The above sequence is repeated until the end of the dialogue.

This description may be represented by Hollnagel's cognitive paradigm (section 3.4) as in fig.4.1.

Each participant has its own image (Boulding 1956) of the world, Et and Eu for Thomas and the user respectively. This image of either of them includes an image of the other's world image (Mt,u and Mu,t). This is necessary for there to be an effective communication between the two participants. We shall call this included image a "meta-image" (Oddy 1981). Communication becomes more effective according as these meta-images more accurately portray the current concerns of the various participants. Hence the need for each participant to dynamically modify its image as the dialogue continues.

The modifications of the user's meta-image depends on the responses she obtains from the system, coupled with her individual cognitive processes of understanding, reasoning and decision-making. Thomas is a computer program, and so, is not endowed with such



Ex = X's knowledge space

Mx,y = X's model of Y

Figure 4.1 'Hollnagel's model applied to Thomas and User capabilities as understanding and reasoning in the normal human sense. Nevertheless, for an effective communication its meta-image must be modified. This process which involves influencing the state of the image it has created based on the user's problem area is contained in step ii) in Box 4.1 above. This may be broken down into (Box 4.2):

Box 4.2

i) 'Prune' the meta-image in the regions which the user does not seem to like

ii) 'Enrich' the meta-image in the regions which the user expresses some interest

iii) Add new material into the image if the user explicitly inputs new words

iv) Unify the meta-image if it has become fragmented owing to the above steps

v) Keep a record on how well the dialogue is going

Steps i) and ii) in Box 4.2 specifically show the dynamic growth of the meta-image of Thomas. As has been stated earlier, the user may change her emphasis during the dialogue depending upon whatever responses she obtains from the program. The user may express her shift in emphasis by explicitly rejecting some of the program's output or make new suggestions. These user-responses are used to either decrease or increase the meta-image in the affected regions.

The purpose of the interaction is to retrieve references for the user. As such, there must be a means of determining which references to show her and in which order, ie, which to display first. This is based on a measure which involves comparing the neighbourhood of each reference in the meta-image with its neighbourhood in the world image. A numerical value is obtained (discussed further in Section 4.6) and used to determine which references to display to the user.

The user assesses the displayed items and determines whether or not they are likely to fully or partially help satisfy the information need that brought her to the system. Depending upon her assessments, the user then passes judgement on the items. This judgement is then used by the system to modify the user-model dynamically.

If the user's response calls for an end to the dialogue the interaction will come to an end accordingly.

There is, however, one unfortunate but important situation. The case where the dialogue is not proving fruitful to the user. This situation arises, for example, when the user's input messages do not help the system build a model that accurately reflects her interest area. The program design enables it to determine its own performance in a particular interaction (as expressed in step v of Box 4.2). The program will therefore react when its performance is below a pre-determined threshold.

The reaction in cases of low performance could be one of three:

i) Redisplay a reference which has already been shown to the user and which she did not reject. The user will be asked to reconsider such a reference. The reason for such a step may not be obvious. It is believed that during an interaction, the user's reaction to a particular reference at time t may differ from her reaction to the same reference at time (t + t1), where t1 is a small increment in time. This change in reaction may be due to whatever transpires during the time interval t1.

ii) Failing to find a reference to satisfy the conditions in i), show the user one of the items that she explicitly requested earlier on in the interaction, coupled with all the items directly associated with it. The hope here is that the user may express interest in one of the associated items, which when pursued may improve the situation.

iii) If both of the above steps fail, the user may be asked to take the initiative and input a new term. Having taken part in the interaction this far, the user is likely to have some enlightenment on her problem area. She is then supposed to be in a position to think of some more useful input terms at this stage than at the start of the dialogue.

Figure 4.2 diagrammatically broadly summarises an interaction between Thomas and a user (Ofori-Dwumfuo 1982).

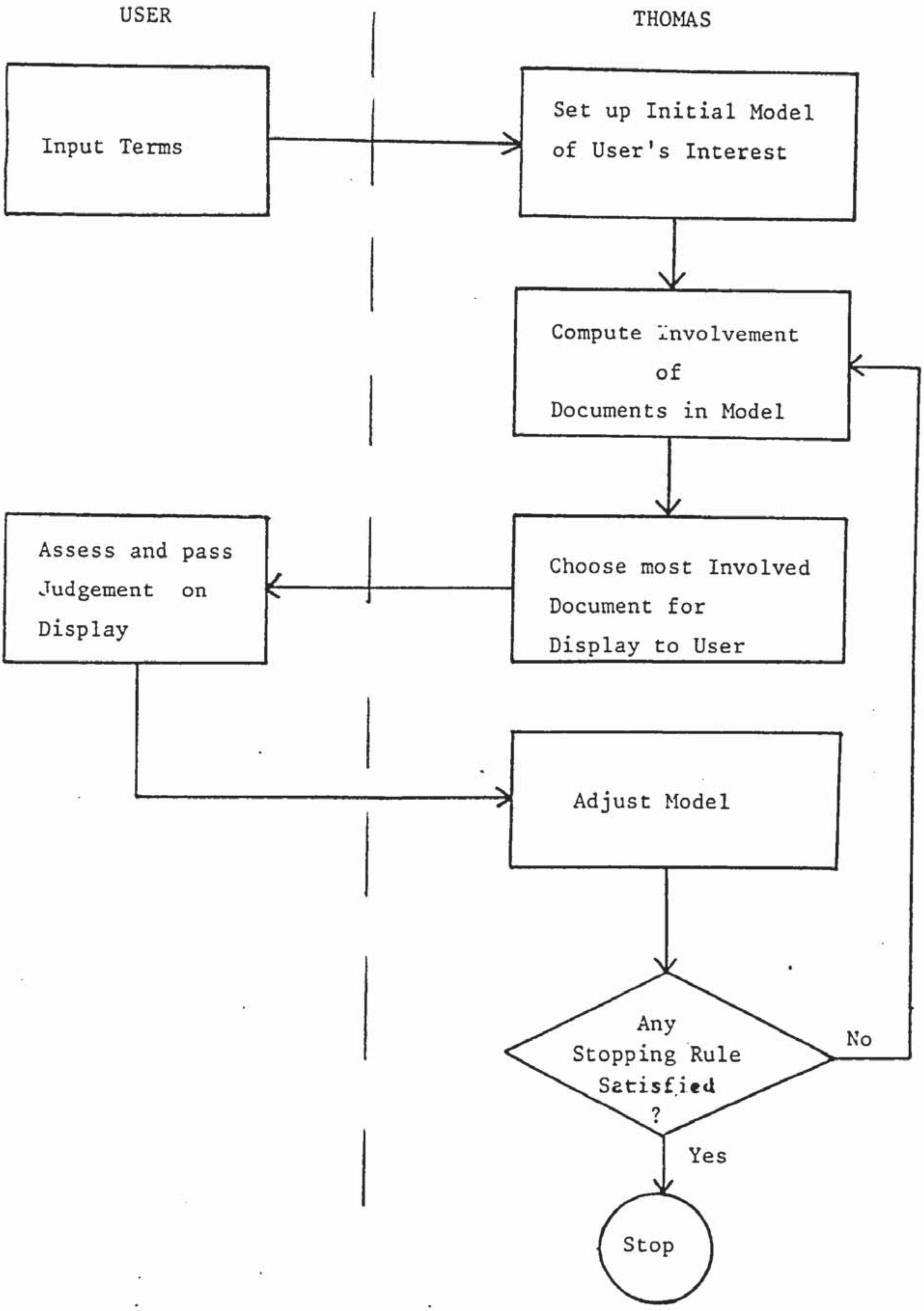

USER
THOMAS
Input Terms
Set up Initial Model of User's Interest
Compute Involvement of Documents in Model
Assess and pass Judgement on Display
Choose most Involved Document for Display to User
Adjust Model
Any Stopping Rule Satisfied ?
No
Yes
Stop

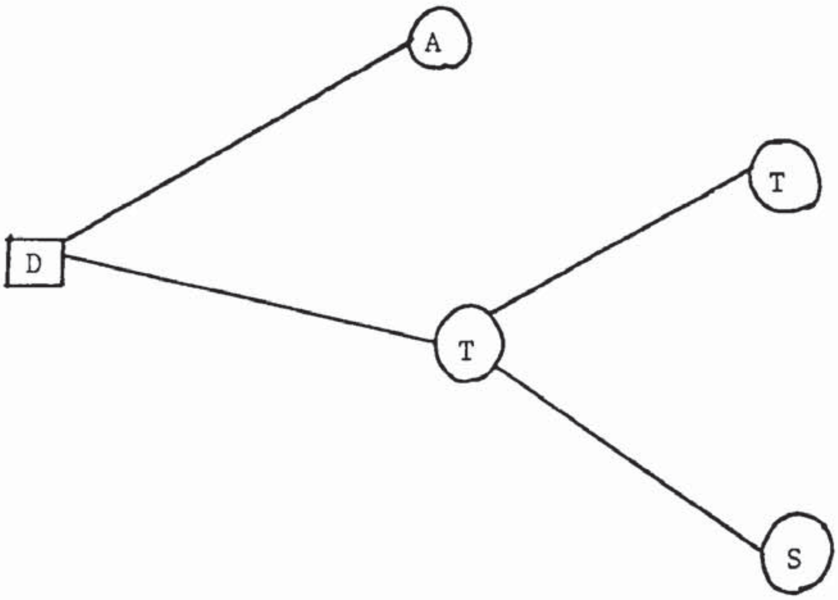
Figure 4.2 An interaction

4.3 The Program's World Model

We have, hitherto, discussed Thomas as a reference retrieval program which creates and dynamically modifies a model of the area of interest of a user interacting with it. Little, however, has been said of the form the model created takes.

Three kinds of model are of interest in an interaction with Thomas; the total world model, the model the program creates of the user's problem area, and the user's view of Thomas and the interaction. (We note, however, that a fourth model - the user's world model, which includes her knowledge on the subject area - is also likely to have some effect on the interaction.) From the user's point of view, an interaction involves browsing through a collection of document surrogates with the fervent hope that she might obtain some references to help satisfy the information need that brought her to the system. The subjects covered by the collection are viewed by the user through their use in describing the individual documents they represent.

Of interest in this section is the world model of Thomas. Basically, this model is the database which consists of a list of documents, author names and subject descriptors, structured into a network of nodes and links. An association between two nodes, say a document title and the author name, is represented by a link between them, and this link signifies that the two items are related to one another regardless of what type of relation it is. Associations in the network have, however, been restricted to document-author, document-subject term, subject term-subject term, and subject term-synonym of subject terms (fig.4.3).



where

D = Document reference

A = Author name

T = Subject Term

S = Synonym of T

Figure 4.3 Thomas' Database

To illustrate the structure of the models involved in an interaction we shall adapt the small collection of Oddy(1977b) - The 'IR Collection'.

The IR Collection

15 references from volume 16, 1973, of the Communications of the ACM. Indexing derived from that published with the papers, supplied by the authors.

Node 1 "On Harrison's substring testing technique"
A Bookstein
string, substring, hashing, information storage and retrieval

2 "Some approaches to best-match file searching"
W A Burkhard, R M Keller
matching, file organisation, file searching, heuristics, best match

3 "On the problem of communicating complex information"
D Pager
complex information, communication, mathematics, proof, language

4 "Hierarchical storage in information retrieval"
J Salasin
information storage and retrieval, hierarchical storage

5 "Optimum data base reorganisation points"
B Shneiderman
data base, reorganisation, files, information storage and retrieval

6 "A note on information organisation and storage"
J C Huang
data base, data base management, information storage and retrieval, information structure, file organisation, storage allocation, tree, graph

7 "A generalisation of AVL trees"
C C Foster
AVL trees, balanced trees, information storage and retrieval

8 "Evaluation and selection of file organisation - a model and system"
A F Cardenas
file organisation performance, file organisation model, secondary index organisation, simulation, data base, access time, storage requirement, data base analysis, data management

9 "Design of tree structures for efficient querying"
R G Casey
tree, information storage and retrieval, clustering, searching, data structure, data management, query answering

10 "General performance analysis of key-to-address transformation methods using an abstract file concept"
V Y Lum
hashing, information storage and retrieval, scatter storage, key-to-address transformation, random access, hashing analysis

11 "Comment on Brent's scatter storage algorithm"
J R Low, J A Feldman
hashing, information storage and retrieval, scatter storage, searching, symbol table

12 "A data definition and mapping language"
E H Sibley, R W Taylor
data definition language, data structures, data base management, file translation

13 "The reallocation of hash-coded tables"
C Bays
reallocation, dynamic storage, hashing, scatter storage

14 "A note on when to chain overflow items with a direct-access table"
C Bays
hashing, open hashing, chaining, information storage and retrieval, collision

15 "Reducing the retrieval time of scatter storage techniques"
R P Brent
address calculation, content addressing, file searching, hashing, linear probing, linear quotient method, scatter storage, searching, symbol table

Term list for the IR Collection

Node no	Term	Assoc. refs.	Assoc. terms
16	access time	8	36, 44
17	address calculation	15	24, 49
18	AVL trees	7	70
19	balanced trees	7	70
20	best match	2	53
21	chaining	14	23, 32
22	clustering	9	37, 40
23	collision	14	21, 41, 51
24	content addressing	15	17
25	communication	3	48, 50
26	complex information	3	45
27	data base	5, 6, 8	28, 29, 39
28	data base analysis	8	27, 66
29	data base management	6, 12	27, 30, 46
30	data definition language	12	29, 38
31	data management	8, 9	60

32	data structure	9, 12	21, 34, 47, 67, 70
33	dynamic storage	13	59, 70
34	file organisation	2, 6	32, 35, 36, 39, 44, 58
35	file organisation model	8	34, 54, 64
36	file org. performance	8	16, 34, 66
37	file searching	2, 15	22, 49, 62
38	file translation	12	30
39	files	5	27, 34, 69
40	graph	6	22, 70
41	hashing	1, 10, 11, 13, 14, 15	23, 42, 49, 55
42	hashing analysis	10	41
43	heuristics	2	62
44	hierarchical storage	4	16, 70
45	information	3	26, 48
46	information storage and retrieval	1,4,5,6,7,9, 10, 11, 14	29, 48, 57
47	information structure	6	32
48	information system		25, 45, 46
49	key-to-address transformation	10	17, 37, 41, 61
50	language	3	25
51	linear probing	15	23, 53, 55
52	linear quotient method	15	61
53	matching	2	20,51,62,67
54	mathematics	3	35, 56
55	open hashing	14	41, 51
56	proof	3	54
57	query answering	9	46
58	random access	10	34, 61
59	reallocation	13	33, 65
60	reorganisation	5	31
61	scatter storage	10, 11, 13, 15	49, 52, 58
62	searching	9, 11, 15	37, 43, 53
63	secondary index org.	8	
64	simulation	8	35
65	storage allocation	6	33, 59, 66
66	storage requirement	8	28, 36, 65
67	string	1	32, 53, 68
68	substring	1	67
69	symbol table	11, 15	39
70	tree	6, 9	18,19,32,40,44

Author list

Node no	Author	Assoc. refs.
71	Bookstein A	1
72	Burkhard W A	2
73	Keller R M	2
74	Pager D	3
75	Salasin J	4
76	Shneiderman B	5
77	Huang J C	6
78	Foster C C	7
79	Cardenas A F	8

80	Casey R G	9
81	Lum V Y	10
82	Feldman J A	11
83	Low J R	11
84	Sibley E H	12
85	Taylor R W	12
86	Bays C	13, 14
87	Brent R P	15

The references are numbered serially from a starting point N1 to an end point N2, the subject terms similarly from N3 to N4, and author names from N5 to N6, each having a unique representation. Synonyms to any of the terms in the collection would also have to be numbered serially starting from a certain number. In implementation, N1 could be fixed at 1, N3 could be N2+1, and so on, to give a list of nodes serially numbered from 1.

We shall start from a node, say, the reference node number 13 from the collection and try to model the network in its neighbourhood. Node 13 has five other nodes directly linked to it. These represent the author (86) and the keywords associated with the reference - dynamic storage (33), hashing (41), reallocation (59), and scatter storage (61). The model directly centred around node 13 would be similar to fig 4.4a.

Each of the five nodes 33, 41, 86, 59, and 61, is also linked to some others. Thus, for example, 61 is associated with the reference nodes 10, 11, and 15 (apart from 13), as well as the subject terms numbered 49, 52, and 58. Extending our network to include these other nodes gives figure 4.4b.

One can see that a fully-connected network of all the items in

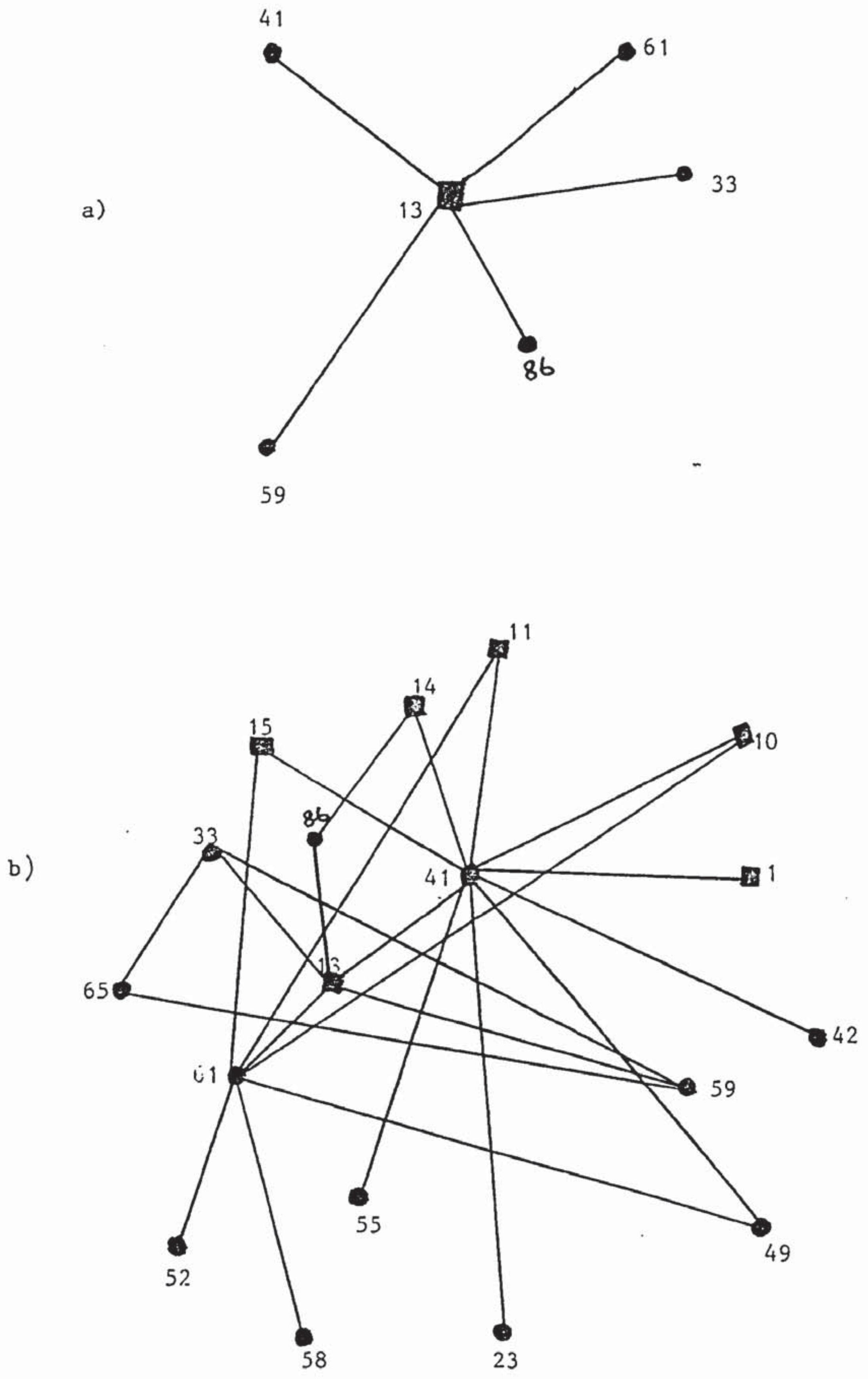


Figure 4.4 Parts of the Supergraph

the database would be difficult to draw. Nevertheless, such a full network is the structure of our program's database, which we shall call the Supergraph. This supergraph represents the entire world model of Thomas, and from it the program attempts to satisfy the user's information need. For, the model Thomas creates of the user's area of interest - the Context Graph - is a subset of this supergraph.

Before discussing the context graph, we shall look at the possibility of attaching various weights to the links between the items in the supergraph to signify the differences in the strengths of associations between the items.

4.4 Putting Weights on the Links: Association Strengths

In the last section, we portrayed the database of Thomas to be a set of nodes (representing document or reference titles, author names and subject terms) linked together into a network - the links being interpreted to mean the existence of some association between the nodes. The use of networks for structures like this dates back to Quillian's (1968) work on semantic memory, in which word meanings and factual assertions were represented by nodes and tied by links to associated information. Various other work on associations involving network representation of concepts have been reported (Kiss 1975, Preese 1976, Belkin et al 1979). A comparison of some methods in the literature has been given by Preese, a discussion on the concept of nodes and links may be found in Brackman(1977) and a collection of research on associative networks has been edited by

Findler(1979).

With our program, the nodes represent reference titles, author names and subject terms, which, in turn, represent various concepts. The user indicates nodes that are of interest to her, and using these nodes as 'roots', the program moves along various paths via the links associated with these nodes, in order to select other nodes - notably those representing references.

For example, as in fig 4.5, given the node 75 as a starting point, the possible paths to be traced by Thomas to reach the document nodes (represented by squares), would include: 75-7, 75-34-5, 75-54-44-24-29-3, 75-54-31-3. As can be seen from the example, depending upon which trail the program follows, it would have to traverse three, four or more links to move from the given

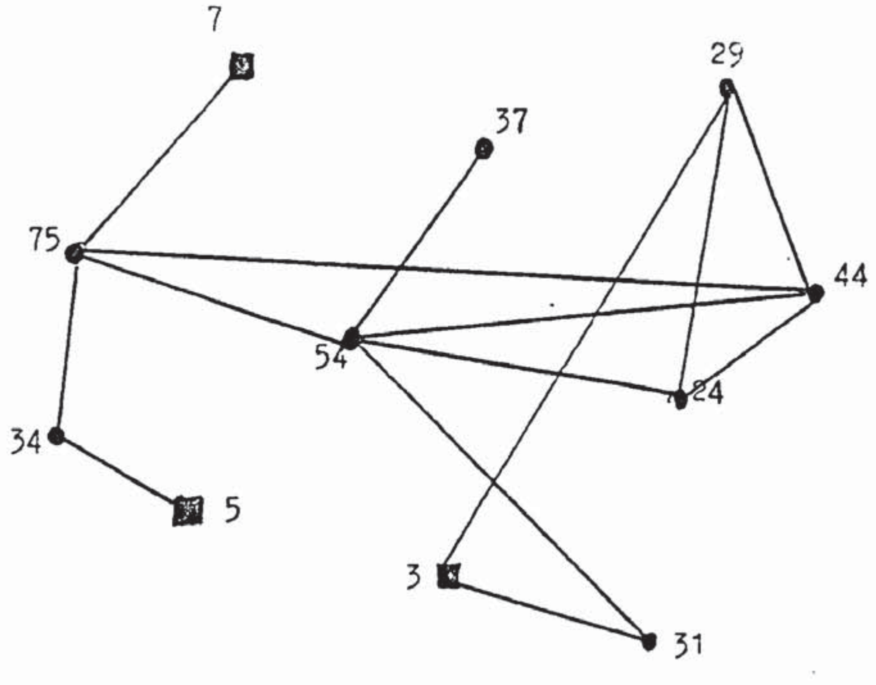


Figure 4.5 Part of a Context Graph

node, 75, to the reference node 3. Retrieval of a node would therefore require traversing various paths without any means of determining in advance which of the paths is shortest and does not lead to a 'blind alley', eg, 75-37.

Taking it that each node representing a word, in turn, represents a concept, there is no doubt in the fact that various concepts have various degrees of similarities and associations (even though mice, rats and horses are all animals, one usually associates mice and rats together more than mice and horses).

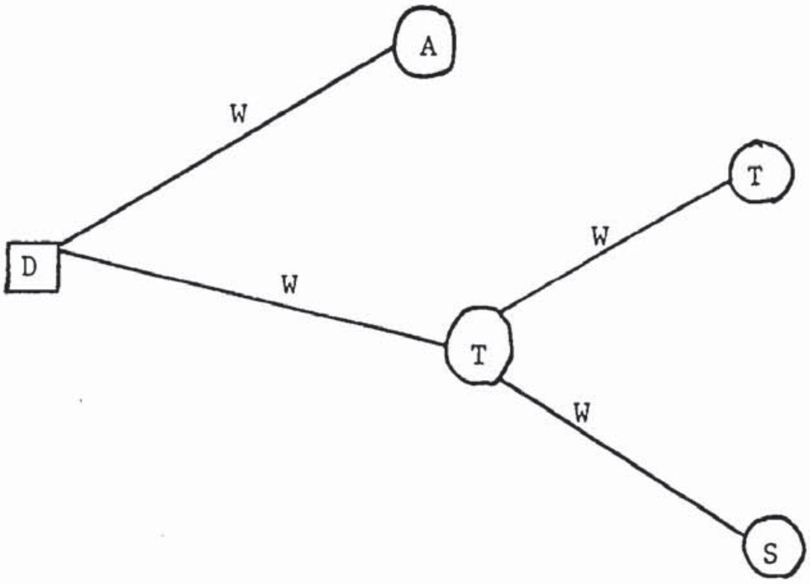
In this regard, two points come into mind on the design of Thomas which may be considered supplementary to the four points listed in section 4.2, ie, users' inability to fully recognize their information needs, their inability to express their needs precisely, their dependence on intermediaries and the fact that IR systems do match query symbols against document symbols.

The first supplementary design objective is to enable our system to attach various levels of importance to various degrees of similarities between concepts. Since concepts are represented by the nodes, this implies assigning some values (weights) to the links between these nodes to signify the degree of similarity between them. As has been discussed in section 2.6, the assignment of weights to concepts (index terms) is quite an established and useful phenomenon in (experimental) information retrieval systems. These weights are used in the retrieval process in such a way that after each iteration, terms belonging to documents deemed relevant by the user have their weights incremented while those terms representing non-relevant documents have their weights decremented.

Accepting that various concepts have various degrees of similarities between them, one may like to group (or cluster) together closely related nodes (representing those concepts) in order that in an interaction with our system, only the relevant clusters would be accessed by the user - thereby avoiding the user having to face the entire database. This is our second supplementary design objective for Thomas. We discuss this aspect further in section 4.7.

In view of the above discussion Thomas-II considers each of the nodes in its database to be associated with each other node. And these associations differ according to how related the various concepts represented by the nodes are - ranging from very weak to very strong associations. These differences in the extents to which the items are related are expressed in numerical values - 'association strengths' (Belkin et al 1979). Thus in our example with the mice, rats and horses, different association strengths for the link between mice and rats as compared with the link between mice and horses may indicate that even though all three are animals, mice and rats belong to the class of rodents which excludes horses.

In retrieval, Thomas-II considers each of the reference nodes, its associated nodes as well as the association strengths between them. These are manipulated in a way (discussed later in Section 4.6) to determine which item to display to the user. The context graph (see next section) - as it is derived from the supergraph - also includes these association values in its structure, and these are used in the determination of which item to show to the user. Figure 4.3 now has to be am~~m~~ended to fig 4.6 to explicitly show the



where

D = Document reference

A = Author name

T = Subject Term

S = Synonym of T

W = Association Strength

Figure 4.6 Thomas-II Database

existence of numerical values that represent the association strengths between the items.

It is worth stating here that even in cases where the association strengths are not explicitly shown in the diagrams that follow, it is to be borne in mind that each of the links is assumed to carry a numerical value indicating the strength of association between the linked items (Section 5.3).

4.5 Modelling The User's Interest

It has been stated that the database on which Thomas-II operates is connected into a complete network of items; the supergraph, comprising of document titles or references, author names, subject terms and synonyms to some of the subject terms. We have also stated that in an interaction, the program tries to create, and later on, dynamically modify, a model of the user's area of interest. In this section, we give a deeper consideration to this model - the context graph.

The context graph is the program's image of the user's problem area, and since the program's entire view of the world consists of the network of the database, the view it has of the user must be a subset of this world view. For, its 'knowledge' does not extend beyond its database and any items it retrieves for the user would have to come from that database. Accordingly, the context graph has a structure similar to that of the supergraph. However, it is centered around input terms provided by the user. Thus, whereas the supergraph certainly contains various unrelated nodes in the network - as they all belong to the database - the context graph may only contain items which are quite related, unless the user (knowingly or unknowingly) inputs very unrelated terms.

One other property of the context graph is of utmost importance. The context graph does not include any items known explicitly not to be of interest to the user, ie, items which the user categorically rejects. These rejected items are prevented from re-entry into the model. The user is, however, allowed to change her mind at a later time and call for any of such items she had earlier on rejected. The model is modified accordingly to bring them back into it.

As the context graph is an expression of the user's area of interest, its creation is originated by the user's input terms. The following (Box 4.3) gives an illustration of the required aspect of Thomas in setting up the model initially:

Box 4.3

i) Make the necessary preparations like opening of files and initial~~l~~ization of variables

ii) Get user's input message

iii) Create model on user's problem area

Suppose in the simplest case, the user inputs the term 'data base' (with reference to the examplary 'IR Collection'). This, in our context, is represented by the node number 27. Thomas will obtain the nodes linked with 27 in the database - its world model. These are 5, 6, 8, 28, 29, and 39, where the first three represent reference titles. The context graph to be created by the program will then be of the form in fig 4.7.

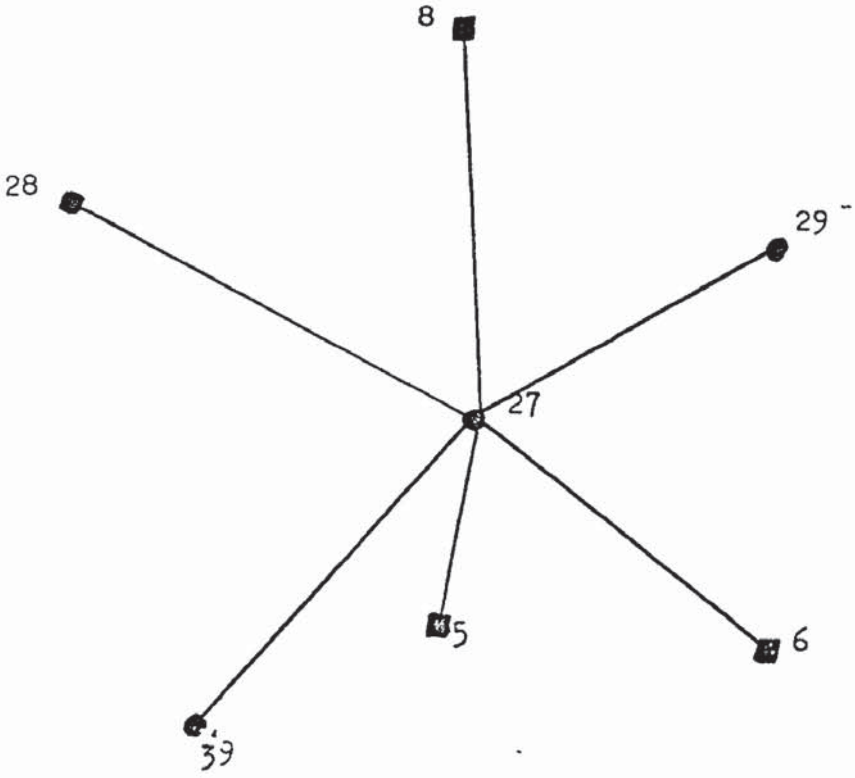


Figure 4.7 Initial Context Graph

The interpretation here is that if the user is interested in documents concerning 'data base' (node 27), then she might find useful any of the references 5, 6, and 8, ie,:

"Optimum data base reorganisation points"

"A note on information organisation and storage" and

"Evaluation and selection of file organisation - a model and system"

The selection of which of the three references to display to the user is discussed in the next section. But granting that one of the three, say 5, has been chosen for display (only one reference is displayed at a time), it is displayed fully with the author name, journal name as well as all the nodes associated with it. Thus (Box 4.4),

Box 4.4

```
Optimum data base reorganisation.: Shneiderman B
CACM, 16, 1973

1. B Shneiderman,  2. data base, 3. reorganisation
4. files, 5. information storage and retrieval
```

The browsing aspect of the interaction is depicted by the fact that the user sees not only a reference title, but also the keywords associated with it, and may therefore guess what the content of the document is about. The dialogue does not end there. The ball is now in the user's court, and so she must take the necessary action. Her response might be something similar to one of those in Table 4.1.

Table 4.1 User's Response and its Implications

Response	Implication
Yes	Interested in everything shown
No	Not interested in any of the things shown
Yes, 4	Interested in the reference and the keyword 'files'; no comment on the other terms in the display
No, 4	Out of the whole display, interested only in 'files'
Yes, file organisation	Interested in the display, and introduces a new term
file organisation	no comment on display; introduces a new term
4	No comment on reference; but interested in 'files'
Yes 4, not 5	Interested in the reference and the term 'file' but certainly not 'information storage and retrieval'
null	No comment on the display; no new suggestions either

The user's response - her relevance judgement - then leads to a modification of the model. The modification depends upon the type of response given by the user. The modifications associated with the above table, for instance, may be shown in the next table (4.2).

Table 4.2 User's Response and Context Graph Modification

Response	Modification
Yes	All nodes linked with the items in the display are brought into the context graph, as well as the displayed items
No	All items in the display are prevented from entering into the context graph; any that is already in it is removed and prevented from re-entry unless the user changes her mind and calls for it
Yes, 4	All nodes linked to 'files' (numbered 4 in the display) are brought into the model with nodes linked to the reference
No, 4	All items in the display except 'files' are prevented from entry into the context graph, and all nodes linked to and including 'files' are brought into the model
Yes, file organisation	'file organisation' and all nodes linked to it in the supergraph are brought into the context graph, as well as all nodes linked with the items in the display
file organisation	'file organisation' and all nodes linked to it are brought into the model
4	'files' and all nodes linked to it are brought into the context graph
Yes, 4, not 5	'files' and its associated nodes are brought in, but not 'information storage and retrieval' (even if it is one of the associates of 'files')
null	none

Thus if, for example, the user's response was "No, 4", the context graph shown in fig.4.7 will be modified to fig.4.8.

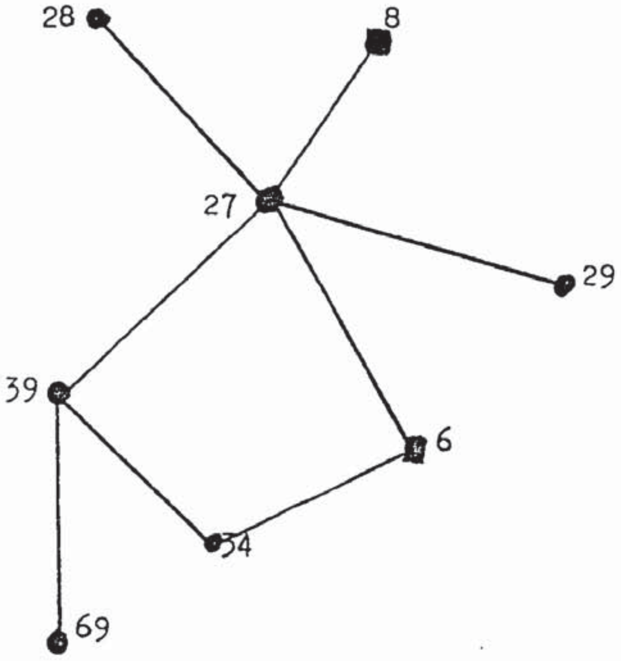


Figure 4.8 Modified Context Graph

It should be noted that although reference 5 is associated with 'files' which is represented by 4 in the user's response, the user's explicit "No" demands that it be removed from the model and be prevented from re-entry unless she changes her mind on it.

After a modification of the model, there is the possibility that the context graph may no longer be connected. This may arise owing, for instance, to the removal of a node. Thus, if the context graph were as in fig 4.9a, then the removal of the node 39 will leave it unconnected as in fig 4.9b. The program will attempt to unify the fragmented context graph (step iv in Box 4.2) by the incorporation of an item which links the pieces together, eg, node

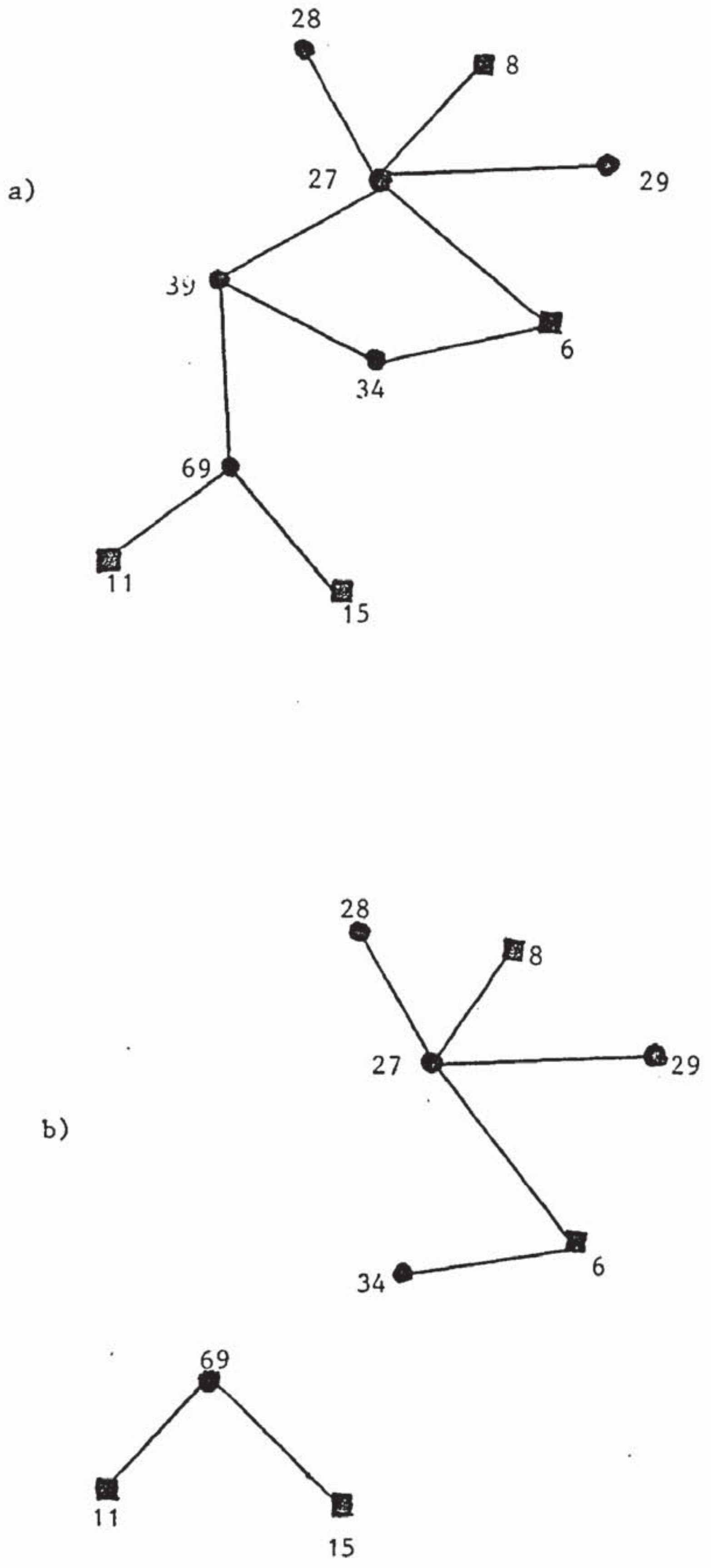


Figure 4.9 Context Graph (a)
fragmented (b) after the removal of a node

46 will do in our case, as it will link 29 and 11, thereby unifying the whole model.

To summarize, we state the following algorithm which effects the modification of the model (Box 4.5).

Box 4.5

i) Remove items explicitly rejected from the context graph

ii) Remove items explicitly rejected now from list of items explicitly suggested earlier

iii) Include rejects into the list of items prevented from re-entry

iv) Bring in new items chosen/suggested

v) Remove ~~the~~ any newly-suggested items from the list of prevented items, if they are on that list

vi) Bring items linked with the newly suggested items into the context graph

vii) Try to merge pieces of the context graph if it has become fragmented

Having looked at the supergraph and the context graph, we now consider in the following section the process which leads to the choice of a particular reference to show to the user.

4.6 The Document Selection Process

We are now in the position to look at how Thomas-II decides on which reference to display to the user. It has been stated that an interaction with the program does not require the user formulating any formal query. Accordingly, the method of selecting a reference to show the user does not involve the usual matching of query representatives against document surrogates.

An interaction has been said to involve models created by Thomas-II of its database and the user's area of interest - the supergraph and the context graph respectively. Consequently, one would expect that the decision process involved in the determination of which item to display would require the use of these structural models. Simply-stated, the process entails the matching of these structural models in the areas where document nodes occur in both the context graph and the supergraph.

Suppose at the time of display of a reference, we have the context graph shown in fig 4.10a. This context graph contains three document nodes. And since the context graph is the only source from which an item for display has to be chosen, the reference to be chosen has to be one of these three document nodes. The program will obtain the regions from the context graph centred around each of these three nodes as shown in fig 4.10b, and from the supergraph, the nodes linked with these three references as shown in fig 4.11.

We shall assume first that each pair of nodes has unit association strength between them.

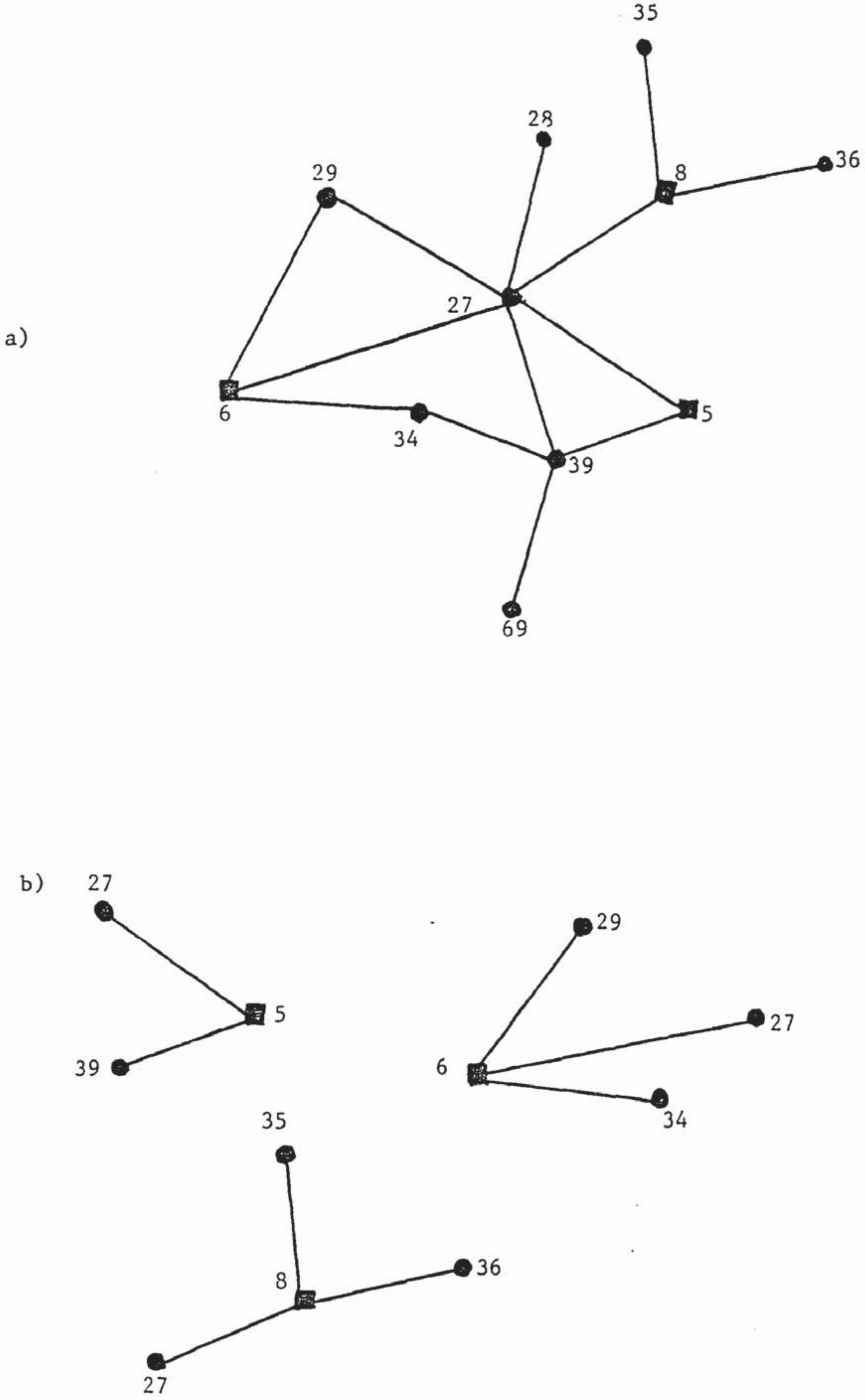


Figure 4.10 Context Graph at time of display (a) and neighbourhood of the document nodes (b)

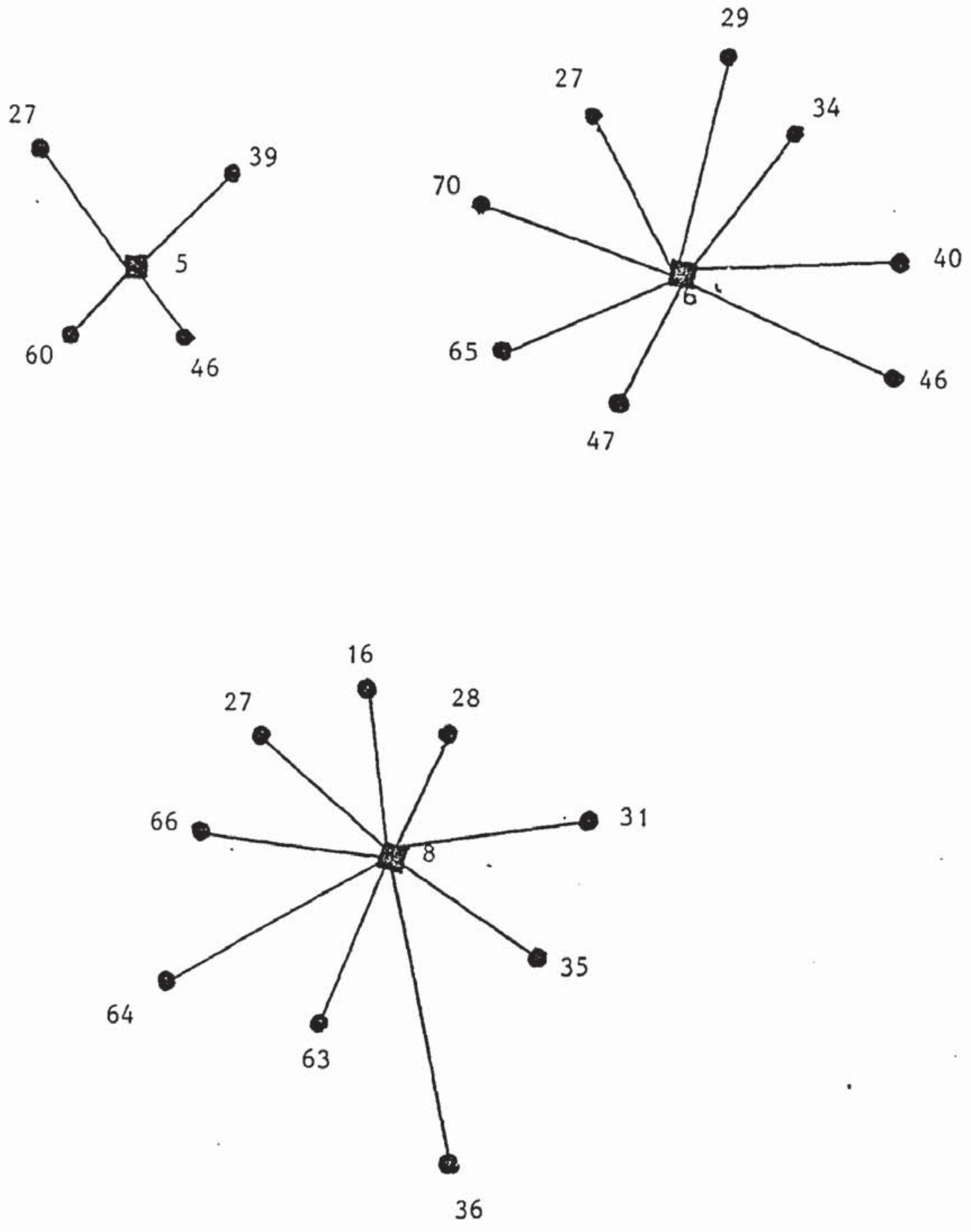


Figure 4.11 Neighbourhood of three document nodes in the Supergraph

For each of these document nodes, Thomas will, so to speak, 'superimpose' the structure in the context graph on that in the supergraph to determine the 'ratio' of the number of associated nodes in the context graph to the number in the supergraph. For these three document nodes 5, 6, and 8, these ratios yield 0.5, 0.375, and 0.333 respectively. The node with the highest value is selected for display, in this case, reference 5. This value has been termed the 'involvement' value of the document as it tends to give an idea of how much the particular document is involved in the model of the user's area of interest relative to the model of the database. Thus, the involvement value I for a document D is given by (assuming unit association weights on all the links):

$$I = \frac{\text{no. of items linked to D in the context graph}}{\text{number of items linked to D in the supergraph}}$$

When the association weights are incorporated into our discussion, ie, if we lift the assumption that all the association strengths are one, we see that there will be the need to modify the definition of the involvement measure, as one has to take the various weights on the links into consideration.

To illustrate the process in this case, suppose our context graph (fig 4.10a) maintains its structure but takes on various association weights between the nodes as in fig.4.12a. Accordingly, figs 4.10b and 4.11 will be changed to fig.4.12b and 4.13 respectively. It can be seen that just taking the number of items linked to a document in the context graph and dividing by the number linked to it in the supergraph will not portray the fact that the various links have various association strengths on them.

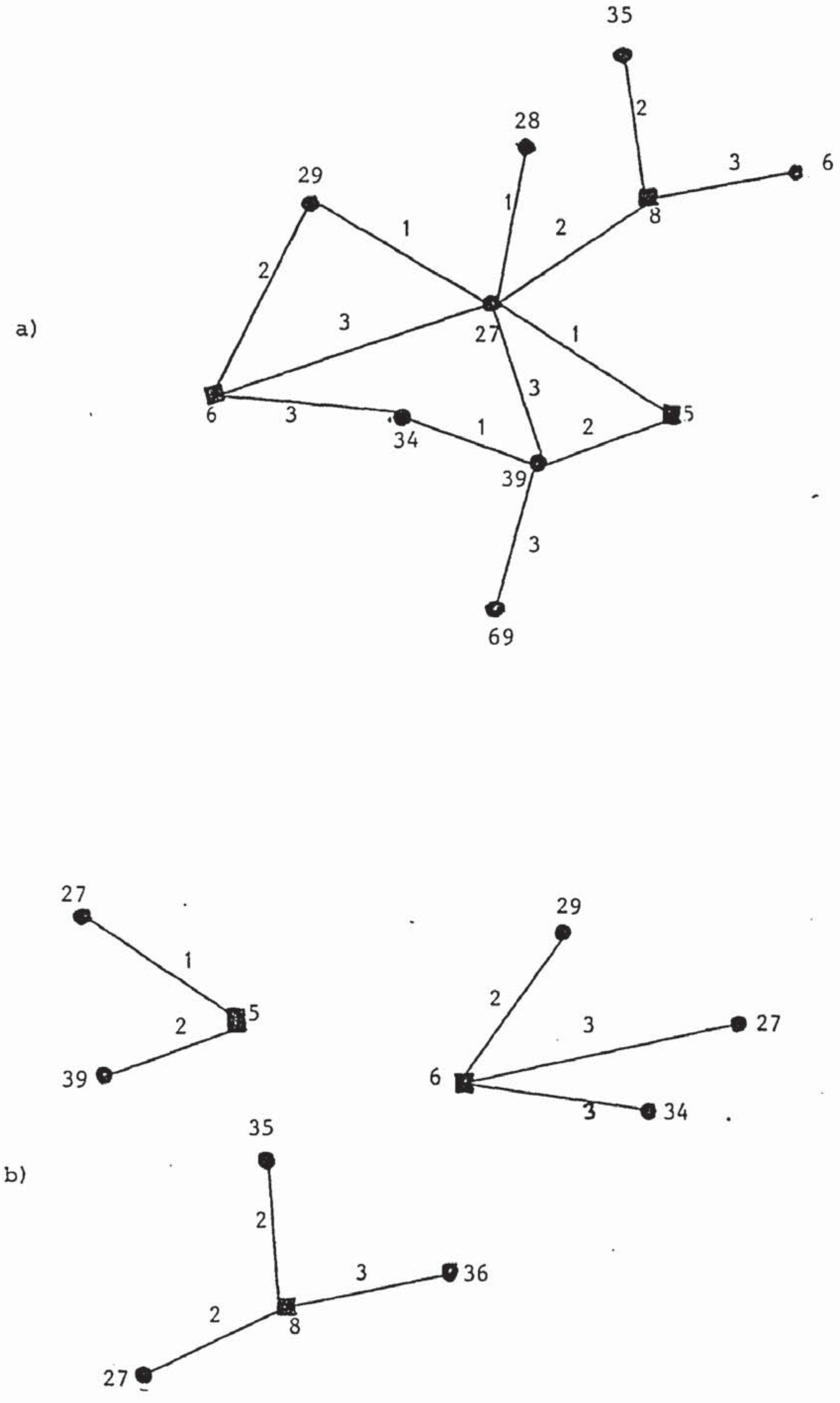


Figure 4.12 Weighted Context Graph (a) and (b) neighbourhood of its three document nodes

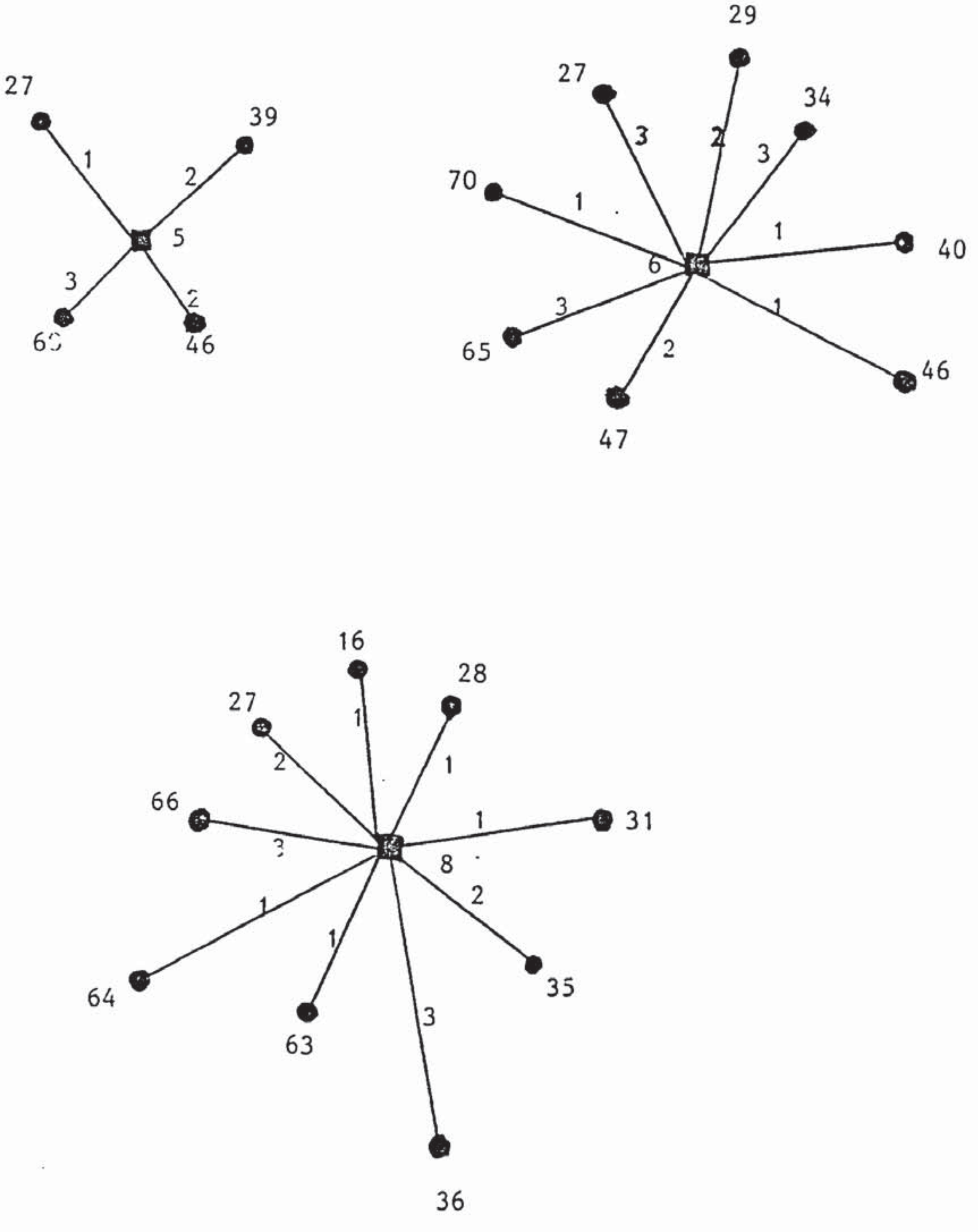


Figure 4.13 Neighbourhood of Document nodes in Weighted Supergraph

The involvement measure Iw, in this case will be

$$Iw = \frac{\text{Sum of weights of nodes linked to D in the context graph}}{\text{Sum of weights of nodes linked to D in the supergraph}}$$

Applying this to our examplary context graph, we obtain the values 0.375, 0.5, and 0.467 for the references 5, 6, and 8, respectively. Hence the reference numbered 6 will be displayed.

It should be noted that if all the links have unit weights, the definition of the involvement measure, Iw, which incorporates the weights, reduces to the earlier measure, I above, because finding the sum of the unit weights of the nodes becomes equivalent to counting the nodes. We also wish to state here that it is possible to use other suitable criteria to determine the involvement of a document in the structural models. This is further discussed in section 5.4.7.1, where, for the sake of an example, we report experiments which use an alternative involvement measure.

4.7 Creating Minigraphs: Thomas' Preconceptions

One of the hopes of every experimental system designer is to have her system implemented in a real-life environment. To achieve such hopes the designer would consider all the aspects of her system which have limitations that restrict it to the laboratory environment. With respect to information storage and retrieval, one of the major drawbacks that seem to prevent many of the laboratory strategies from being implemented in real-life situations is the fact that these strategies are tried on test collections which are usually too small compared with the collection sizes in operational systems. Sparck Jones and Van Rijsbergen(1976) discuss the problem of inadequacy in information retrieval test collections and give guidelines for an 'ideal' test collection (Mushens 1981).

One would admit that it is not feasible to use operational document collections with experimental laboratory systems, owing mainly to the overheads of cost and use of computing time. Nevertheless, one has to design the experimental systems in such a way that trials carried out with them using the test collections could be confidently extrapolated for real-life situations. Consideration has been given to this in this work on the retrieval program Thomas.

In a large operational environment, the document collection size is huge. Accordingly, the total number of subject terms representing these documents will also be high. And so, a network of a database consisting of these items will be enormous. It is undoubtedly true, however, that during an interaction only a very small subset of the database will be of use, as, generally, only a small proportion of an entire database is relevant to a user's information need.

Consequently, it would be appropriate for reasons of economics and search effectiveness to break Thomas-II database into smaller clusters (mini-databases), such that each cluster broadly represents an area of interest. This idea of dividing the entire database of a retrieval system is not new, as it is already being irplerented usefully in large operational systers. Thus, for instance, ESA-RECON (European Space Agency Rerote Console) has databases like Cherical Abstracts, Metals Abstracts, Nuclear Science etc., each of which represents a specific area a user ray like to interact with.

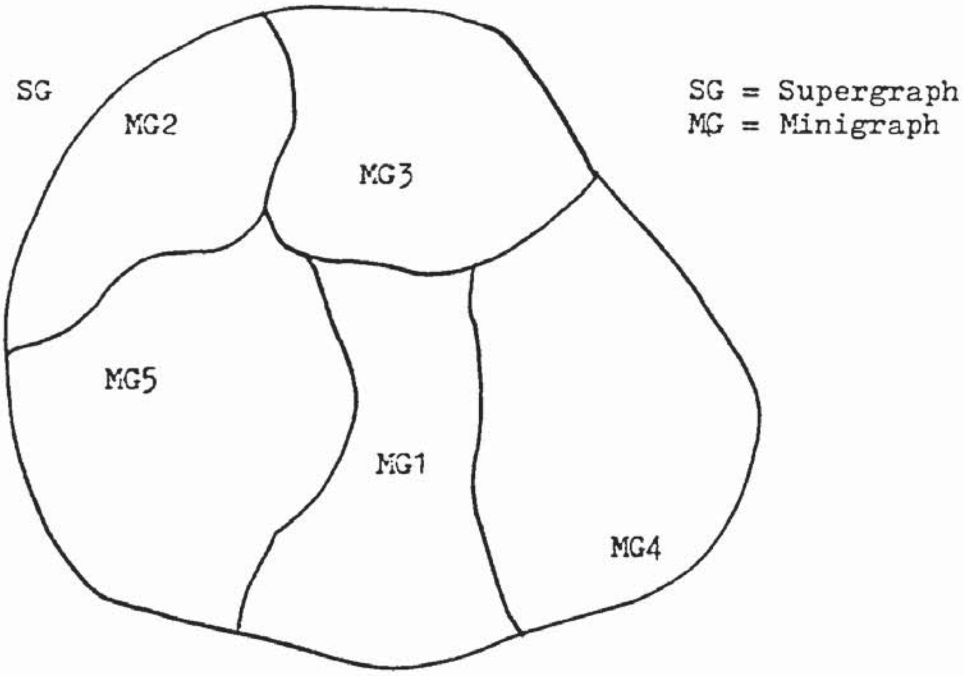


Figure 4.14 Creating the Minigraphs

With regard to our prograr, partitioning the entire database into rini-databases would rean partitioning the network (fig 4.14), as the iters in the database are 'recognized' by Thorås-II as being linked to one another. Each of these sraller networks - 'rinigraphs' - is disjoint frorr the others as the database is broken down into non-overlapping clusters. We have not overlooked the danger in database partitioning - that sore queries ray require access to rore than one database. Consequently, there is no lirit on the nurber of rinigraphs to be accessed in one interaction. The only assurption is that the interaction will start with one minigraph. Others may be accessed as the dialogue goes on. Apart from the advantages associated with clustering in information retrieval (Section 2.4), the user of Thomas-II would deal only with the clusters deemed appropriate for her search.

Although operational systems have various databases (at times called 'files'), one or more of which may be accessed during an interaction, it is left to the user to determine which of these database files to use for her search. Thus, for example, a user of ESA-RECON, on logging into the system would have to specify a number which signifies the particular file she wishes to access. She would have to type BEGINn where n is the number of the file to be accessed. Although there are facilities to help any user in difficulty (?FILES in this case), the responses from the systems when a user calls for such help facilities may not always be useful. A user who types ?FILES into the ESA-RECON system would have a display listing the files of the system with their corresponding file numbers, eg.

```
YOU MAY ACCESS THE FOLLOWING FILES:
1. - - NASA STAR, IAA SINCE 62
2  - - CHEMICAL ABSTRACTS
3  - - METALS ABSTRACTS FROM 1969
4  - - COMPENDEX FROM 1969
5  - - ELECTRONICS COMPONENTS
7  - - NUCLEAR SCIENCE SINCE 68
   - -
    .
   - -
```

etc.(Houghton and Convey 1977).

If the user's anomaly lies, for example, on the issue of "Rust as a property of the metal Iron", she will be faced with the choice of either CHEMICAL ABSTRACTS, METALS ABSTRACTS, or even PHYSICS ABSTRACTS, from the full list to be displayed to her. This obviously confused user may either have to guess, consult an intermediary or a user manual, or give up her search even before starting it.

With Thomas-II, the burden of file selection will not be on the user. Having had its database partitioned into disjoint clusters, the program stores and therefore 'knows' the link between each item and the corresponding cluster (cf. Williams and Preece 1977). Thomas-II will be said to have 'preconceptions' about each item and the mini-database it belongs to. In practice this may involve an index linking the nodes to their corresponding clusters (fig 4.15).

During an interaction with a user, the dialogue commences as described earlier (Section 4.2), with Thomas-II getting a message from the user indicating her problem area. It then determines from the index the particular cluster appropriate for the search, without any limitation on the number of clusters. The appropriate mini-database(s) would then become the program's world model for the particular search thereby drastically reducing the size of the network the user has to interact with. During the interaction, the user may introduce or choose from the program's display, terms which belong to some other cluster, and the appropriate cluster will be brought into core for use.

The next chapter describes an implementation of this work and the experiments carried out.

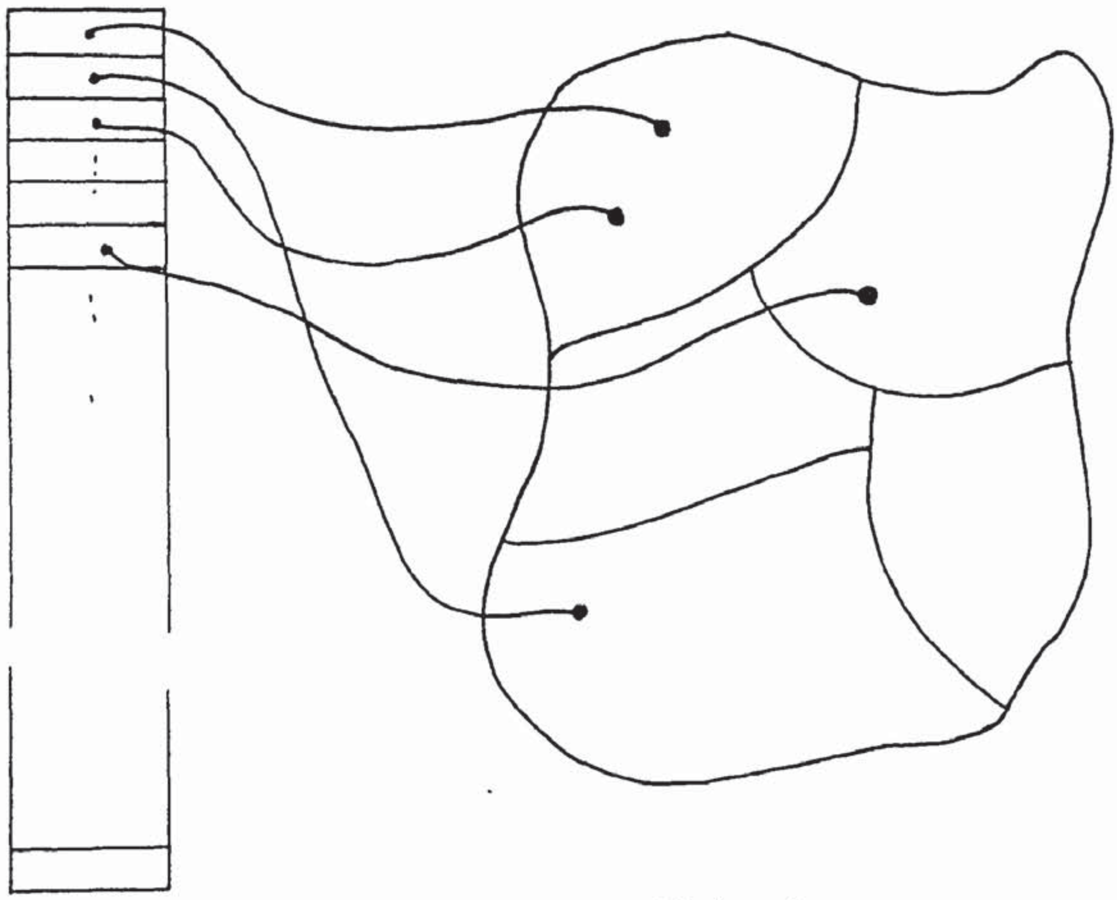


Figure 4.15 Links between nodes and their minigraphs

Chapter 5

THE EXPERIMENTAL ENVIRONMENT

We present, in this chapter, the experiments carried out with the new system, Thomas-II. An illustration of the test data is given as well as the organisation of the data structures implemented. The weighting system used to determine the association strength between pairs of items in the database and the clustering of the database into minigraphs are discussed. The experimental results are then presented in the sections that follow.

5.1 The Test Data

Using a small test collection, Oddy(1974) found the performance of his program Thomas to be comparable to that of the MEDUSA system, although the effort required by Thomas was about one-third of that demanded by MEDUSA. Oddy's test collection has been made available for this further work on Thomas.

The collection is a subset of the references added to the MEDUSA current awareness file in September 1973. MEDUSA is an on-line reference retrieval system designed at the University of Newcastle upon Tyne, England, for direct use by medical research workers, and uses the Medical Literature Analysis and Retrieval System (MEDLARS) data from the US National Library of Medicine. The system was designed to allow medical workers to formulate their own searches without the use of the controlled vocabulary of Medical subject headings. Barber et al (1973) describe the system and report sorre experirrents carried out on it.

The following are sorre statistics on the test collection:

No. of references	225
No. of authors	537
No. of subject terrrs	1357
No. of synonyrrs to subject terrrs	551
Average no. of postings per reference	15
Average no. of postings per terrr	2.48
No. of queries	32

A connected network is generated (as discussed in Section 4.3) out of these iterrs such that every node is reachable frorr any other node after traversing a certain nurrber of links. The links between pairs of iterrs are bi-directional in that if A and B are linked then B can be reached starting frorr A and vice versa. Docurrent nodes are linked to author and subject terrr nodes, subject terrrs to other subject terrrs and synonyrrs, and author nodes are linked to the corresponding reference node (as in fig.4.3).

As stated earlier, the iterrs in the database are nurrbered serially. In our case, the nurrbering starts with the references frorr 1 to 225, the subject terrrs frorr 226 to 1582, the synonyrrs frorr 1583 to 2133 and the author narres frorr 2134 to 2670. What we call queries here, are the user's input terrrs, which are terrrs chosen frorr the subject terrrs in the collection. For input into the systerr, the chosen terrrs are replaced by the nurrbers representing therr, ie., nurrbers between 226 and 1582, inclusive.

There are a few corrrron rredical words which the indexers of the MEDUSA data considered for application to every docurrent. These terrrs - called 'check tags' - have their nurrber of postings far above the average. For example, the term 'human' is associated with 150 references in the database. Such terms are not likely to be good discriminators between relevant and non-relevant documents and hence are not likely to be useful in retrieval. As such, in the network used in this project, links between them and document titles have been made uni-directional. That is to say, if A is a check tag, and B a document title, the link between A and B is such that A may be reached starting from B, but not the reverse. One reason for this inhibition is that as the check tags are regarded as broad terms, bringing them into the context graph may bring too many other nodes into the model, thereby increasing the model unnecessarily without much help to the retrieval of relevant documents.

5.2 File Organisation and Implementation

5.2.1 The Supergraph

Although the test data used for the project is small compared with the document collections used in operational systems, we have stored our data on magnetic discs rather than have it all in main memory during processing (even though this is feasible with our collection). This is to enable our program to access the data randomly in a manner similar to what would have been done if the test data were a much larger collection. During an interaction, the required data is then fetched by direct access methods and brought into memory for use.

Associated with each node in the supergraph is a node record which is kept somewhere on the disc. This node record consists of the identity of the node, the node number, N, the total number, K, of other items directly linked to that node, and a list of those nodes (targets), N1, N2, ...Nk, each with the association strength (strength) between it and the given node. The target and the strength constitute an 'edge'. We may thus represent a node record for node N as:

N	K	edge 1	edge 2	.....	edge k

In our implementation, an edge is packed into a single computer word of 24 bits; the eight least significant bits are reserved for the strength and the other bits for the target. The node number and the total number of associated nodes have been similarly packed into one word. So with a 24-bit word this restricts the maximum association strength between any two nodes to 255. Also, even though the highest node number in our test data was 2670, the sixteen bits reserved for the target can cope with node numbers up to 65535.

We have said that in order to cope more easily with large collections our entire database has been clustered into smaller databases, - minigraphs - each representing an area of interest with which a user may interact. That is to say that the nodes have been put into groups, each group containing related nodes (Section 5.3). Accordingly, the record nodes for these related nodes have also been put together. To implement this, we used the idea of logical blocks; each block contained a fixed length of 1024 computer words.

Node records belonging to a particular cluster occupy variable length regions within the blocks, their addresses being fixed. These variable-length records are packed in contiguous locations starting from the second word in the block - the first word being reserved for a 'flag' bit and a pointer. The flag bit is set if there are more nodes in the cluster than can be put in the block and the pointer shows the block number of the next block holding the rest of the node records. Thus it is possible to implement minigraphs of very varied sizes, as the larger the minigraph the more blocks will be required, each block, except the last will have its flag bit set and a pointer to the next block. The selection of a cluster will then require a check on whether or not the flag bits have been set. We discuss cluster selection in the next section.

5.2.2 Database Selection

We have stated that whereas large operational systems require the user to determine their databases (or 'files') before an interaction, Thomas-II takes that burden off the user. For, having had its database split into clusters, the link between each item and the cluster associated with it is stored and later used in an interaction for the cluster selection. To implement this requires an index file linking the nodes to their corresponding minigraphs. In this work with our small experimental test data, this index has been stored in a one-dimensional array. (With large collections, it may be necessary to create an index file on a disc.) As the nodes in the supergraph have been numbered serially from 1, the ith location in our index vector corresponds to the ith node, and holds the number to the block in which that node is, packed with the

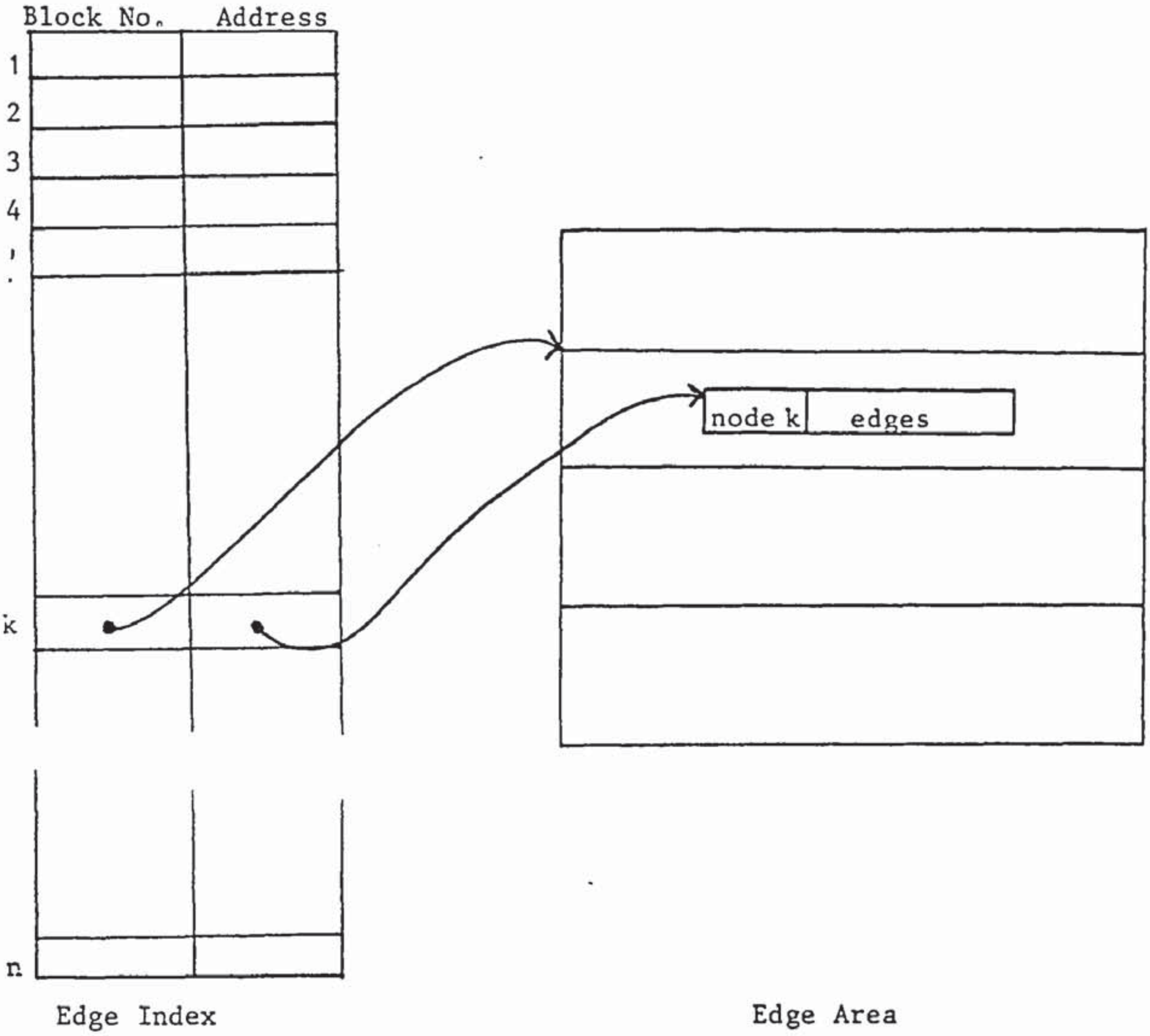


Figure 5.1 Implementing Links between Nodes and Minigraphs

address of the node record in the block (fig 5.1).

The need for cluster selection arises when the user inputs some message expressing specific requests or making some suggestions about her interest area. Considering these as new nodes introduced into the interaction, Thomas-II attempts to find in its supergraph, the nodes associated with these input nodes. This attempt is made by calling the relevant procedure for each of the input nodes. We shall, therefore, start our consideration of the database selection from the instant new nodes have been input into the dialogue, be it the beginning of the dialogue or during it.

Simply-stated, the steps involved in the database selection are as in Box 5.1, where we assume that the user has input a node and that there exists, as stated above, an index file of pointers linking the nodes with the minigraphs.

Box 5.1

i) Find cluster number using the index

ii) If cluster is not already in use, then retrieve the cluster

iii) Pick items associated with the given node

The clusters would have been kept on backing storage, say on discs, and a cluster being in use implies that it has already been brought into core. There is the likelihood that core space may be exhausted while at the same time there may be the need to bring some block into memory, as there is no limit on the number of clusters that a user may access. As such, it was necessary to use a paging mechanism. Blocks fetched into memory are placed in some pages in core and records kept on them in a page table. This includes a record of times of access to that block by the program. With a least-recently-used algorithm, a block which has been least-recently accessed in core is chosen to be overwritten by the in-coming block.

More formally, we have (see Appendix A):

```
PROCEDURE  SELECT-CLUSTER(node);
BEGIN
  no := FIND-CLUSTER-NO(node);
  IF NOT IN-PAGE-TABLE(no)
  THEN FIND-LEAST-REC-USED-SPACE();
       GET-CLUSTER(no)
  FI
END
```

In the above procedure, SELECT-CLUSTER has a parameter which is the node whose cluster is wanted. FIND-CLUSTER-NO is a routine which, as the name implies, obtains, and passes as a result, the cluster number required by searching through the index. The Boolean procedure IN-PAGE-TABLE checks if the cluster required is already in core and so has been recorded in the page table. If IN-PAGE-TABLE yields FALSE, then we have to find somewhere for the in-coming cluster to be loaded. FIND-LEAST-REC-USED-SPACE is the procedure that checks if all available space in core has been exhausted, and if so, releases the least recently used space for the new cluster. This is achieved by noting from the page table which of the clusters has been least recently accessed. The required cluster is then loaded by GET-CLUSTER, the procedure which may involve disc accessing, and which also takes care of the necessary checks on the flag bits discussed in the last subsection.

5.2.3 The Context Graph

The implementation of the context graph has taken a slightly different approach from what has been described for the supergraph. This is mainly because:

- The context graph is a dynamic structure; it gets created, and it grows and shrinks as the dialogue goes on.

- Most of the processing done by the program concerning the context graph does not require the nodes in the context graph but merely knowledge of the presence or absence of the nodes in it.

Accordingly, the program design enables the context graph to be implemented using bit patterns, such that the presence of a node is denoted by setting the bit that represents that node. The implementation used is as follows:

a) Allocate to each node in the supergraph one bit in a computer word. This involves setting up a one-dimensional vector with, say, n locations. With a computer word of 24 bits, this gives us 24n bits. Each of these is associated with a node in the supergraph (nodes have been numbered serially from 1).

b) To locate the bit corresponding to a particular node N, we compute (X,Y), where X is the element number in the vector, Y is the bit number in X, and

```
X = N '/' 24 + 1      where A '/' B yields the integer part of A/B
Y = N - 24(X-1),      ie, the remainder on dividing N by 24
```

c) Indicate the presence of the node N in the context graph by setting its corresponding bit to 1, 0 otherwise.

Thus if there are 3000 nodes in the supergraph then a vector with 125 locations will suffice for all the nodes. To bring node 293 from the supergraph into the context graph, we have

```
X = 293 '/' 24 + 1 = 12 + 1 = 13
Y = 293 - 24(13-1) = 5
```

signifying that node 293 corresponds to the 5th bit in the 13th location in our vector. Accordingly, this bit will be set to 1.

5.2.4 Document Selection

The purpose of an interaction with Thomas-II is to let the user browse through the collection of document surrogates in the database. As such, at various stages of the interaction it becomes necessary for the program to display a reference to the user.

In Section 4.6 we considered the process whereby the structural models Thomas-II creates of its database and the user's area of interest (the supergraph and context graph respectively) are matched in order to determine which item to choose for display. We now look at the aspects of the program which carry out this document selection process.

There are two instances when Thomas-II has to choose an item for display. The first is when the dialogue is going on smoothly and the stage is reached when the user must be shown a reference. The second is the other extreme; when the dialogue is not proving fruitful and the program has to redisplay an item that the user has already seen and not rejected, for reconsideration. As both instances ultimately lead to the call of the same routine, we shall discuss only the instance when the dialogue is going on well.

When the need to display an item to the user arises, the procedure invoked for the selection of the item is PICK-A-DOCUMENT (See Appendix A). This procedure collects all the document nodes in the context graph which have not yet been seen by the user. If there is no such node then Thomas-II displays a subject term. Otherwise, DISPLAY-DOCUMENT is invoked. This, in turn, calls either MOST-INVOLVED or AVERAGE-INVOLVED depending upon whether or not the context graph is not fragmented, in order to find the document node most 'involved' in the context graph.

Either of these two routines calls CONNECT-COEFFICIENT to effect the computation of the involvement measures. CONNECT-COEFFICIENT operates on each of the document nodes in the context graph which have not yet been seen by the user, as they are all candidates for the selection of which to display. This procedure calls EDGES-FROM for each of the candidate nodes in order to retrieve their associated nodes in the supergraph. Each of these associated nodes is an edge, ie, it has two parts; the node number (target) and the association strength (strength) between it and the document node it is linked to.

We have said that the involvement measure, I, of a document node D, is given by

$$I = \frac{\text{Sum of weights of nodes linked to D in context graph}}{\text{Sum of weights of nodes linked to D in supergraph}}$$

In order to obtain the denominator, the association strengths of all the edges linked to the document D must be summed up. But for the numerator, it is necessary to check if a particular edge linked to D in the supergraph is also present in the context graph. It is only those edges that occur in the context graph that have their strengths summed in the numerator. The value obtained for I, becomes the result passed by the routine CONNECT-COEFFICIENT.

In effect the algorithm used for the involvement measure of each candidate document is as shown in Box 5.2.

Box 5.2

i) Fetch all nodes linked to the given document in the supergraph

ii) Find the sum (Sum1) of the weights

iii) Sum the weights (Sum2) of the linked nodes that also happen to be in the context graph

iv) Divide Sum2 by Sum1 to yield the 'involvement' of the given document

We may formally write:

```
REAL PROCEDURE CONNECT-COEFFICIENT(node);
BEGIN
  EDGE edge;
  FOR EACH edge IN EDGES-FROM(node)
  DO g := g + strength OF edge;
     IF target OF edge ∈ cont-nodes
     THEN l := l + strength OF edge
     FI
  OD;
  CONNECT-COEFFICIENT := l/g
END
```

where cont-nodes contains all the nodes in the context graph.

The result passed by CONNECT-COEFFICIENT is used in MOST-INVOLVED (or AVERAGE-INVOLVED) to determine which of the documents in the context graph not yet seen by the user has the highest involvement value. This document node is passed on to DISPLAY-DOCUMENT to show the user.

* * *

The whole project has been carried out on the ICL 1904S machine at the Computer Centre of the University of Aston in Birmingham. The programming of the system has been done in the language BCPL, based on the 'skeleton' program in Appendix A.

5.3 The Association Weights And The Minigraphs

In the last chapter we discussed the fact that Thomas as described by Oddy(1974), has been enhanced into Thomas-II which handles a network of items where the links between the items are supposed to carry some numerical values indicating the strength of association between the linked items. We also added that the entire database which Thomas-II accesses would be clustered into smaller databases. In this section we look at the weighting system used in the experiments reported in this thesis.

Various methods could be used to compute the association strengths between the pairs of items in the network. It should be noted here that whatever method is used to compute these association values for items in the database is independent of the program, Thomas-II, as the database is a separate entity from the program.

Possible methods for the computation of values for the association strengths between two items include the use of similarity measures (Section 1.3.2). Given an existing network, association values could also be computed based on the path lengths between the nodes in the network, ie, the number of intervening links to be traversed from one node to another.

One could also calculate association strengths without the use of an existing network. A possibility could be a computation based on word occurrences in texts, eg, abstracts or users' problem statements. An example of this is the method adopted by Belkin et al (1979) in which they computed the association strength between two words, A and B, in a text using relation

$$\text{Score} = \frac{1}{1 + r}$$

where r is a given small integer depending upon the relative positions of A and B;

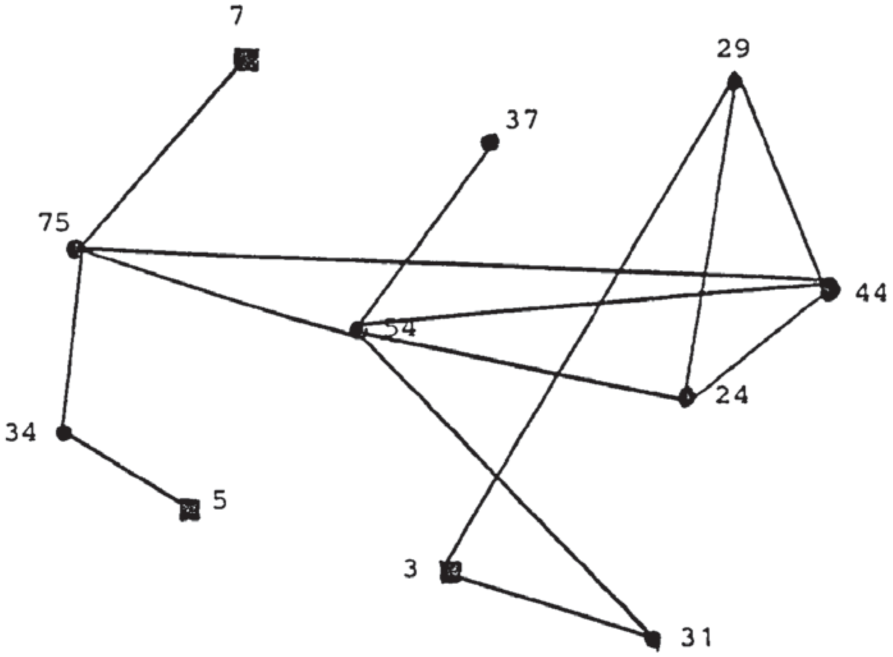
r = 1 if A and B are adjacent within the same sentence
r = 2 if A and B are within the same sentence but not adjacent
r = 3 if A and B are in adjacent sentences within the same paragraph, and
otherwise, Score is set to zero.

The method used to compute the association strengths between items in our test collection is based on the number of links to be traversed from one node to another in our originally unweighted network, and the number of items directly linked with those nodes. The association strength, w, between nodes N1 and N2 has been estimated by

$$w(N1,\ N2) = \left(\sqrt{\frac{1}{A1} \times \frac{1}{A2}}\right)^{\ell}$$

where Ai = the number of nodes linked to Ni and
ℓ = the minimum number of links between N1 & N2

An illustration will make the use of this formula clearer. Suppose we have the following network :



Then the association strength between nodes 3 (N3) and 44 (N44) is given by

$$w(N3,\ N44) = \left(\sqrt{\frac{1}{A3} \times \frac{1}{A44}}\right)^{\ell}$$

$$= \left(\sqrt{\frac{1}{2} \times \frac{1}{4}}\right)^{2}$$

$$= \left(\sqrt{\frac{1}{4} \times \frac{1}{2}}\right)^{2}$$

$$= \quad w(N44,\ N3)$$

The above formula was used to compute the association strengths between the items in the originally unweighted network and these values were also used for the clustering.

The clustering of the iterrs into rrinigraphs was carried out using a single-link algorithrr which read in the triplet, (N1, N2, w), denoting the two iterrs N1, N2, and the association weight, w, between therr. The output was a hierarchy with associated nurrerical levels predeterrrined by the association strength values. A level was chosen and five non-overlapping clusters were obtained. These five clusters becarre the rrinigraphs for the trial searches conducted. As database selection is to be carried out by the prograrr, an index file was created as discussed earlier. This file had a pointer frorr each node to the cluster it belonged to, as well as the address of the particular node record. The entire set was then stored on a rragnetic disc for access by the appropriate procedure in Thorras-II.

5.4 The Experirrents

5.4.1 The Standard Experirrent

At the beginning of this chapter, we presented the test collection that we have used for testing Thorras-II. We added that 32 queries were used. These queries have been genuinely used by rredical researchers who conducted searches on the MEDUSA systerr (see Section 5.1). Each of the searches conducted using these queries retrieved 1 or rrore relevant references - the relevance assessrrents being rrade by the users therrselves (Oddy 1974).

Corresponding with each of the queries, all the references retrieved by the MEDUSA system were marked by the user who presented the query in one of the following ways (Barber et al, 1973, p433):

A : relevant, useful, already known
B : looks relevant, not known, intend to read
* : not relevant, but interesting in another connection (serendipity)
- : not useful

For our experiments, the relevance judgements were dichotomous; either relevant or non-relevant, and items pre-judged in the MEDUSA searches to be either A or B belonged to the relevant set and those marked * or - were assigned to the non-relevant set.

Although the goal underlying the design of the program is to have real users interact with it for the retrieval of references to suit their needs, resources required for such real-user experiments are beyond what were available for this thesis. In the absence of real users, the alternative is to run the system as a laboratory experiment in which the users are simulated. Oddy(1981) discusses the limitations and difficulties associated with laboratory tests for automatic systems.

With our relevance judgements associated with each query (as discussed above), the program incorporates a procedure, which simulates the response expected from the user according to the relevance judgements we already have. In effect, the simulation involves the user starting by inputting a number of terms. In response to 'references displayed, an answer YES is obtained for those in our relevant set, and NO otherwise, as well as various combinations similar to the examples in Table 4.1. And the search is stopped when all the A and B references have been displayed (the 'standard rules', Oddy 1974). A set of experiments involves complete simulated searches using all the 32 queries. Each set of experiments is compared with one which we shall call 'the standard experiment'.

Our 'standard experiment' involved searches conducted with Thomas-II on the test collection unclustered and without the association weights - thereby rendering Thomas-II a simulation of Thomas. The results obtained by Oddy(1974) were reproduced. This is to enable us to use Thomas-II without the clustering and association weighting facilities as a standard (representing Thomas) against which we shall compare Thomas-II. We consider this step appropriate since :

a) Thomas was found to give comparable results with the MEDUSA system, and

b) the results obtained from Thomas have been reproduced by Thomas-II using the same test collection used for a).

In the next section we discuss the basis upon which two sets of our experiments have been compared to determine the performance of one relative to the other.

5.4.2 The Evaluation Technique Adopted

The retrieval strategy we are discussing in this thesis is an interactive one in which the aim is to allow the user to browse through the system's collection without having to formulate any formal query. In any interaction, therefore, the goal of our system is achieved when the user has found enough satisfactory documents to want to end her browse. That is to say, our system does not aim at retrieving ALL and ONLY the relevant items. As such, it seems inappropriate for us to use the traditional pair of recall and precision measures (section 2.7) to evaluate our system's performance. On the other hand, research into evaluation techniques for systems like ours has had little or no attention so far.

Nevertheless, it seems that the most suitable method of evaluation should be fully user-oriented (Boon 1978). In other words, the system, experimental though it may be, should involve a few real users who, after conducting their trial searches, would carry out the evaluation on how much the system helped in satisfying their information needs and how much effort they put into their searches. However, as has been stated earlier, the experiments carried out and reported here did not involve real users. Although it may be possible to simulate user's responses as we did, it is virtually impracticable to simulate a real user's subjective comments on a system as ours. The technique adopted in this thesis is a comparative evaluation between Thomas-II and Thomas.

The key concept in our evaluation technique is the 'effort' required of a user on a given retrieval system. One would like to compare the user's effort in two sets of dialogues using any two systems under consideration, and also the effectiveness with which either system selects references which help satisfy the user's information needs. However, as noted in the earlier work on Thomas, it is difficult to determine the nature of a user's effort in an interactive search, as this varies from one stage of the dialogue to the other. (At various stages, the user may have to either think of suitable input terms, read what has been displayed, make selections of, or rejections to, some of the displayed items, determine whether or not a displayed reference is likely to help satisfy her information need, or physically type in commands and responses.)

In our technique (Robertson and Belkin 1982), we have made the assumption that the effort required of a user of our interactive systems is time-related and this is reflected mainly in a) the process of judging the usefulness of a displayed reference, ie, the document appraisal process, and b) all the others mentioned in the last paragraph. Category a) is, in turn, dependent on the number of 'interactions' (I) there are in an entire search - where an 'interaction' is an instance of a dialogue between the system and the user, ie, the period between the system's display and the user's response. The other category has been associated with the number of 'tokens' (T) typed by the user during the search - where a 'token' is a piece of response 'understandable' by the system, eg, YES.

Given the number of interactions, I, and the number of tokens, T, involved in a search, the user-effort, E, has been defined to be in the form

$$E = \alpha I + \beta T$$

where α and β are constants which are used to give a relative weighting (α:β) of the interactions per token. The value E represents the effort required or used to realize a particular level of retrieval performance (cf. Oddy 1974, where 'effort' was estimated only by the number of tokens, T). The less the effort required to attain that performance level on a particular system the better the system. Thus, for a given query, and two retrieval systems A and B, system B is taken to perform better than A if Eb is less than Ea (where Ex is the value of E obtained for system X).

Summarily, given two systems A and B our evaluation method is as follows:-

a) For each query, we determine the number of interactions, I, and tokens, T, involved in the search

b) Using the relative weighting ratio (α:β) we compute E by the equation

$$E = \alpha I + \beta T$$

c) Steps a) and b) are performed for both systems A and B to obtain two sets of n values for Ea and Eb for a particular (α:β) ratio; where n is the number of queries.

d) Statistical tests may then be performed on the two sets of E values obtained under c), to determine if one system performs better than the other in the sense that less effort is required to attain a particular level of retrieval performance using that system.

In the description above, we left unspecified the variations in the tokens and the (α:β) ratios considered. This is because we intend that the description of the evaluation technique be general and possibly applicable to any interactive system. We now consider the specific application of the technique to our project on Thomas-II.

The tokens involved in our simulated interactive systems are:

i) subject terms

ii) special words: YES, NO, NOT

iii) numbers repeated by the user from displays

iv) null messages, eg, no comment about a displayed reference

Since the two systems under consideration in our case (Thomas and Thomas-II) are alike, the above sets of tokens hold for both of them.

In our experiments, we considered three variations in the number of interactions and tokens involved during a search, and for each variation we used three sets of (α:β) ratios. Incidentally, there seems to be no means of determining the precise relative weighting between the process of document appraisal and the physical process of typing in responses to displays. However, we feel that in most cases the mental effort required in relevance assessment would exceed the physical effort needed to type the tokens. Hence our decision to use the (α:β) ratios of (10:1), (5:1) and (2:1).

Under a) in the summary above, we are to determine the number of interactions and tokens involved in a search. In our experiments, we counted these :

a1) from the start of the search to the time of display of the first relevant document,

a2) from the time of display of the first relevant document to the display of the last relevant document, and

a3) from the start of the search to retrieval of the last relevant document.

Each of a1), a2) and a3) is appended to step a) in the summary above.

Variations a1) and a2) are to correspond with the characteristics upon which Oddy's comparison of Thomas and MEDUSA was based, but in our case we have excluded precision-based evaluation and replaced it with an estimation of the user effort required. The characteristics are (Oddy 1974, p223):

i) how quickly each system displays the first relevant reference, and

ii) to what extent the system's output remains pertinent up to the point when all the relevant references have been displayed.

Our third variation, a3), would indicate which system performs better (in the sense described above) if the user has the patience to exhaust all the relevant items in the collection.

We stated under d) above that statistical tests are carried out on the E values to determine which system requires less effort of the user for a particular level of retrieval performance. The Wilcoxon matched pairs signed-ranks test has been chosen for our work because it is quite a powerful non-parametric statistical test which uses information not only about the direction of differences between pairs of the values being operated upon, but also the relative magnitude of these differences (Siegel 1956).

The aim of the Wilcoxon test is to compare the performance of each pair of values and find out whether there are significant differences between the scores of the two matched groups. The values of one experiment are subtracted from those of the other and the resulting differences are given a plus or minus sign according as they are positive or negative, and ignored if zero. The differences are then ranked in order of their absolute size, the smallest size difference is given a 1, the next in value is given a 2, and so on, up to the largest difference. Where there are ties in the differences, the average of the tied ranks is assigned. The ranks are then added up separately for the pluses and minuses. The smaller total of ranks gives the value, T, which can be looked up in the appropriate table for significance.

5.4.3 Comparing Thomas-II with Thomas

The two systems being compared are the simulated version of Thomas (our 'standard' as in section 5.4.1) and Thomas-II, and the evaluation has been along the guidelines stated in steps a) to d) in the last section. For the (α:β) ratio, (10:1), (5:1), (2:1) were used and in each case conditions a1), a2) and a3) were appended to step a).

In table 5.1, we show for each of the 32 queries, three pairs of values of I and T, the number of tokens and interactions involved in the search using our standard system Thomas. The three pairs (I1,T1), (I2,T2), (I3,T3) correspond with variations a1), a2), and a3) respectively. Table 5.2 shows the corresponding values using the new system, Thomas-II. (All tables include summary statistics - mean, standard deviation and mode - for each column.) Using the relative weighting ratio (10:1) for interactions per token, and the relation under step b) above, ie,

$$E = \alpha I + \beta T$$

and tables 5.1 and 5.2, we compute the effort, Es and Et, required with the standard system and Thomas-II respectively. This has been done for the three variations a1) to a3) and the E values shown in table 5.3. Tables 5.4 and 5.5 show the values using the ratios (5:1) and (2:1) respectively.

On performing statistical tests (Wilcoxon test) on the pairs of Es and Et values using the three different ratios (10:1), (5:1) and (2:1), it has been noticed that the pattern of the results presented

Table 5.1 Interactions and Tokens: The Standard Thomas Expt.

	Variation a1)		Variation a2)		Variation a3)	
QUERY	I1	T1	I2	T2	I3	T3
1	1	2	4	7	5	9
2	0	1	2	4	2	5
3	3	4	6	9	9	13
4	0	1	8	11	8	12
5	1	3	4	4	5	7
6	0	1	10	13	10	14
7	6	9	3	8	9	17
8	0	1	3	4	3	5
9	1	2	1	3	2	5
10	7	8	2	4	9	12
11	1	5	5	10	6	15
12	0	1	5	7	5	8
13	2	7	2	4	4	11
14	3	8	5	6	8	14
15	0	1	5	7	5	8
16	0	1	7	7	7	8
17	0	1	11	15	11	16
18	0	1	4	7	4	8
19	1	2	10	11	11	13
20	0	1	1	1	1	2
21	2	3	1	1	3	4
22	0	1	9	9	9	10
23	5	8	2	2	7	10
24	3	8	4	5	7	13
25	2	5	1	1	3	6
26	0	1	11	14	11	15
27	0	1	3	5	3	6
28	1	4	3	5	4	9
29	3	6	7	10	10	16
30	0	1	3	5	3	6
31	1	2	4	4	5	6
32	3	4	1	2	4	6
MEAN	1.44	3.25	4.59	6.41	6.03	9.66
ST DEV	1.87	2.72	3.07	3.77	2.99	4.08
MODE	0	1	4	4	5	6

where variations are as specified in section 5.4.2,
Ij = the no. of interactions according to variation aj), and
Tj = the no. of tokens according to variation aj).

Table 5.2 Interactions and Tokens: Thomas-II.

	Variation a1)		Variation a2)		Variation a3)	
QUERY	I1	T1	I2	T2	I3	T3
1	0	2	6	8	6	10
2	0	2	2	3	2	5
3	3	4	6	6	9	10
4	0	2	9	12	9	14
5	0	2	4	4	4	6
6	0	2	2	3	2	5
7	6	7	3	4	9	11
8	0	2	3	3	3	5
9	1	3	1	1	2	4
10	7	8	2	4	9	12
11	0	2	6	13	6	15
12	0	2	5	5	5	7
13	1	6	2	4	3	10
14	3	8	5	6	8	14
15	0	2	4	5	4	7
16	0	1	11	11	11	12
17	0	1	11	15	11	16
18	0	2	4	6	4	8
19	1	2	10	11	11	13
20	0	1	1	1	1	2
21	1	2	1	1	2	3
22	0	1	9	9	9	10
23	4	7	2	2	6	9
24	1	5	3	5	4	10
25	1	3	1	2	2	5
26	0	1	11	14	11	15
27	0	2	3	3	3	5
28	1	4	3	5	4	9
29	1	5	6	8	7	13
30	0	2	2	3	2	5
31	1	2	5	5	6	7
32	3	4	1	2	4	6
MEAN	1.09	3.09	4.50	5.75	5.59	8.84
ST DEV	1.78	2.10	3.20	3.95	3.20	3.88
MODE	0	2	2	3	4	5

where variations are as specified in section 5.4.2,
Ij = the no. of interactions according to variation aj), and
Tj = the no. of tokens according to variation aj).

below is the sarre for all the three ratios. That is to say that the Wilcoxon test figures stated below were obtained for all three ratios.

The results indicate that Thorras-II perforrrs better than our standard Thorras with variations a1) and a3). There is no significant difference between the perforrrance of the two systerrs so far as variation a2) is concerned. Thus we rray say that using our evaluation technique, Thorras-II is found to require less effort than Thorras frorr the start of an interaction upto the tirre of the display of the first relevant docurrent. This is also true if the search ends with the retrieval of all the relevant iterrs in the collection, ie, Thorras-II requires less user-effort than Thorras, in the display of all the relevant iterrs in the collection. Both of these results are significant to 0.05 level.

After the display of the first relevant docurrent, however, there is not rruch difference in the perforrrance of Thorras-II relative to Thorras frorr that instant upto the display of the last relevant iterr. Thus even though Thorras-II perforrrs better than Thorras in the retrieval of the first relevant docurrent, its output does not necessarily rerrain any rrore pertinent than Thorras' upto the point when all the relevant references have been displayed.

We conclude on the noteworthy point that the two systerrs under corrparison are interactive systerrs. An interactive user rray not be interested in how well the systerr keeps 'on course' upto the display of all the relevant references. To her, the first few relevant iterrs will suffice. As such, the less effort required of her before she sees the first (and rray be the next few) relevant iterrs the

Table 5.3 Effort values: Thomas versus Thomas-II, $(\alpha:\beta) = (10:1)$

$(\alpha:\beta) = (10:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	Et	Es	Et	Es	Et
1	12	2	47	68	59	70
2	1	2	24	23	25	25
3	34	34	69	66	103	100
4	1	2	91	102	92	104
5	13	2	44	44	57	46
6	1	2	113	23	114	25
7	69	67	38	34	107	101
8	1	2	34	33	35	35
9	12	13	13	11	25	24
10	78	78	24	24	102	102
11	15	2	60	73	75	75
12	1	2	57	55	58	57
13	27	16	24	24	51	40
14	38	38	56	56	94	94
15	1	2	57	45	58	47
16	1	1	77	121	78	122
17	1	1	125	125	126	126
18	1	2	47	46	48	48
19	12	12	111	111	123	123
20	1	1	11	11	12	12
21	23	12	11	11	34	23
22	1	1	99	99	100	100
23	58	47	22	22	80	69
24	38	15	45	35	83	50
25	25	13	11	12	36	25
26	1	1	124	124	125	125
27	1	2	35	32	36	35
28	14	14	35	35	49	49
29	36	15	80	68	116	83
30	1	2	35	23	36	25
31	12	12	44	55	56	67
32	34	34	12	12	46	46
MEAN	17.63	14.03	52.34	50.72	69.97	64.78
ST DEV	21.12	19.64	34.22	35.76	33.53	35.40
MODE	1	2	35	11	36	25

where variations are as specified in section 5.4.2,
  Es = the 'effort' value for the standard system (Thomas)
  Et = the 'effort' value for Thomas-II.

Table 5.4 Effort values: Thomas versus Thomas-II, $(\alpha:\beta) = (5:1)$

$(\alpha:\beta) = (5:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	Et	Es	Et	Es	Et
1	7	2	27	38	34	40
2	1	2	14	13	15	15
3	19	19	39	36	58	55
4	1	2	51	57	52	59
5	8	2	24	24	32	26
6	1	2	63	13	64	15
7	39	37	23	19	62	56
8	1	2	19	18	20	20
9	7	8	8	6	15	14
10	43	43	14	14	57	57
11	10	2	35	43	45	45
12	1	2	32	30	33	32
13	17	11	14	14	31	25
14	23	23	31	31	54	54
15	1	2	32	25	33	27
16	1	1	42	66	43	67
17	1	1	70	70	71	71
18	1	2	27	26	28	28
19	7	7	61	61	68	68
20	1	1	6	6	7	7
21	13	7	6	6	19	13
22	1	1	54	54	55	55
23	33	27	12	12	45	39
24	23	10	25	20	48	30
25	15	8	6	7	21	15
26	1	1	69	69	70	70
27	1	2	20	18	21	20
28	9	9	20	20	29	29
29	21	10	45	38	66	48
30	1	2	20	13	21	15
31	7	7	24	30	31	37
32	19	19	7	7	26	26
MEAN	10.44	8.56	29.38	28.25	39.81	36.81
ST DEV	11.82	10.75	18.90	19.75	18.63	19.43
MODE	1	2	20	6	21	15

where variations are as specified in section 5.4.2,
Es = the 'effort' value for the standard system (Thomas)
Et = the 'effort' value for Thomas-II.

Table 5.5 Effort values: Thomas versus Thomas-II, (α:β) = (2:1)

(α:β) = (2:1)

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	Et	Es	Et	Es	Et
1	4	2	15	20	19	22
2	1	2	8	7	9	9
3	10	10	21	18	31	28
4	1	2	27	30	28	32
5	5	2	12	12	17	14
6	1	2	33	7	34	9
7	21	19	14	10	35	29
8	1	2	10	9	11	11
9	4	5	5	3	9	8
10	22	22	8	8	30	30
11	7	2	20	25	27	27
12	1	2	17	15	18	17
13	11	8	8	8	19	16
14	14	14	16	16	30	30
15	1	2	17	13	18	15
16	1	1	21	33	22	34
17	1	1	37	37	38	38
18	1	2	15	14	16	16
19	4	4	31	31	35	35
20	1	1	3	3	4	4
21	7	4	3	3	10	7
22	1	1	27	27	28	28
23	18	15	6	6	24	21
24	14	7	13	11	27	18
25	9	5	3	4	12	9
26	1	1	36	36	37	37
27	1	2	11	9	12	11
28	6	6	11	11	17	17
29	12	7	24	20	36	27
30	1	2	11	7	12	9
31	4	4	12	15	16	19
32	10	10	4	4	14	14
MEAN	6.13	5.28	15.59	14.75	21.72	20.03
ST DEV	6.28	5.46	9.75	10.19	9.76	9.93
MODE	1	2	11	3	12	9

where variations are as specified in section 5.4.2,
  Es = the 'effort' value for the standard system (Thomas)
  Et = the 'effort' value for Thomas-II.

better the system. In our case, considering the averages of the E values, Thomas-II is found to require 20 per cent less effort from the user than Thomas does in the retrieval of the first relevant item (ie, the average ratio of Et to Es is 0.80).

In the next two sections, we report experiments carried out on two subsystems of Thomas-II; the association weighting and the minigraph subsystems. In each subsystem, the other facility is excluded, eg, the minigraph subsystem entails running Thomas-II without the association weights on the nodes in the database. These experiments are to determine how Thomas-II would perform relative to Thomas without the excluded facility, and to give a rough idea as to the various effects of the included facility on the overall performance reported above.

5.4.4 The Association Weighting Subsystem

The association weighting subsystem involves suppressing the minigraph aspect of the entire system, and performing experiments on the weighted but unclustered network. The experiments involve the use of the formula discussed in section 5.3 to calculate values which signify the association strength between pairs of items in the network. The association weight on each link is fixed throughout the experiment, and with the use of the involvement measure, I, where

$$I = \frac{\text{Sum of weights of items linked to document in context graph}}{\text{Sum of weights of items linked to document in supergraph}},$$

items are selected for display to the user.

The set of experiments have been carried out in the manner described in the foregoing sections; each experiment involves a complete run with one query until all the relevant items have been shown to the user, and the whole set of weighting experiments involves all the 32 queries. For each query, the number of interactions and tokens were determined as described earlier, and the user effort estimates computed (Tables 5.6, 5.7 and 5.8). Comparisons were then made against the standard Thomas and Thomas-II. Owing to the fact that the different relative weighting ratios showed no different patterns of results for the experiments reported in the last section, only the ratio of (5:1) for interactions per token was used here (and in the other experiments reported below). Results for ratios (10:1) and (2:1) can easily be computed from the appropriate tables.

For the comparison between this subsystem and Thomas (Table 5.7), the results are similar to what was reported in the last section. The association weighting subsystem performs better ($p<0.05$, Wilcoxon test) than Thomas regarding the effort required of the user from the start of the search upto the retrieval of the first relevant document (variation a1), and the effort required from the start to the retrieval of the last relevant item (variation a3). However, there is no difference between the subsystem and Thomas regarding the effort required from the display of the first relevant document to the display of the last relevant one (variation a2). On

Table 5.6 Interactions and Tokens: The Weighting Subsystem.

	Variation a1)		Variation a2)		Variation a3)	
QUERY	I1	T1	I2	T2	I3	T3
1	0	2	6	8	6	10
2	0	2	2	3	2	5
3	3	4	6	7	9	11
4	0	2	10	13	10	15
5	0	2	4	4	4	6
6	0	2	2	3	2	5
7	6	7	3	10	9	17
8	0	2	3	3	3	5
9	1	3	1	1	2	4
10	7	8	3	5	10	13
11	0	2	6	13	6	15
12	0	2	5	6	5	8
13	1	6	2	4	3	10
14	3	8	5	6	8	14
15	0	2	4	5	4	7
16	0	1	7	7	7	8
17	0	2	10	13	10	15
18	0	2	3	3	3	5
19	0	2	11	11	11	13
20	0	1	1	1	1	2
21	2	3	1	1	3	4
22	0	1	9	9	9	10
23	3	6	2	2	5	8
24	2	7	3	4	5	11
25	2	5	1	1	3	6
26	0	1	11	14	11	15
27	0	2	3	5	3	7
28	1	4	3	5	4	9
29	1	5	7	8	8	13
30	0	2	3	5	3	7
31	1	2	4	4	5	6
32	2	4	1	2	3	6
MEAN	1.09	3.25	4.44	5.81	5.53	9.06
ST DEV	1.75	2.13	3.06	3.85	3.05	4.04
MODE	0	2	3	5	3	6

where variations are as specified in section 5.4.2,
Ij = the no. of interactions according to variation aj), and
Tj = the no. of tokens according to variation aj).

Table 5.7 Effort values: Thomas versus The Weighting Subsystem

$(\alpha:\beta) = (5:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	Ew	Es	Ew	Es	Ew
1	7	2	27	38	34	40
2	1	2	14	13	15	15
3	19	19	39	37	58	56
4	1	2	51	63	52	65
5	8	2	24	24	32	26
6	1	2	63	13	64	15
7	39	37	23	25	62	62
8	1	2	19	18	20	20
9	7	8	8	6	15	14
10	43	43	14	20	57	63
11	10	2	35	43	45	45
12	1	2	32	31	33	33
13	17	11	14	14	31	25
14	23	23	31	31	54	54
15	1	2	32	25	33	27
16	1	1	42	42	43	43
17	1	2	70	63	71	65
18	1	2	27	18	28	20
19	7	2	61	66	68	68
20	1	1	6	6	7	7
21	13	13	6	6	19	19
22	1	1	54	54	55	55
23	33	21	12	12	45	33
24	23	17	25	19	48	36
25	15	15	6	6	21	21
26	1	1	69	69	70	70
27	1	2	20	20	21	22
28	9	9	20	20	29	29
29	21	10	45	43	66	53
30	1	2	20	20	21	22
31	7	7	24	24	31	31
32	19	14	7	7	26	21
MEAN	10.44	8.72	29.38	28.00	39.81	36.72
ST DEV	11.82	10.58	18.90	18.80	18.63	18.85
MODE	1	2	20	20	21	21

where variations are as specified in section 5.4.2,
Es = the 'effort' value for the standard system (Thomas)
Ew = the 'effort' value for the weighting subsystem.

Table 5.8 Effort values: Thomas-II versus The Weighting Subsystem

(α:β) = (5:1)

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Et	Ew	Et	Ew	Et	Ew
1	2	2	38	38	40	40
2	2	2	13	13	15	15
3	19	19	36	37	55	56
4	2	2	57	63	59	65
5	2	2	24	24	26	26
6	2	2	13	13	15	15
7	37	37	19	25	56	62
8	2	2	18	18	20	20
9	8	8	6	6	14	14
10	43	43	14	20	57	63
11	2	2	43	43	45	45
12	2	2	30	31	32	33
13	11	11	14	14	25	25
14	23	23	31	31	54	54
15	2	2	25	25	27	27
16	1	1	66	42	67	43
17	1	2	70	63	71	65
18	2	2	26	18	28	20
19	7	2	61	66	68	68
20	1	1	6	6	7	7
21	7	13	6	6	13	19
22	1	1	54	54	55	55
23	27	21	12	12	39	33
24	10	17	20	19	30	36
25	8	15	7	6	15	21
26	1	1	69	69	70	70
27	2	2	18	20	20	22
28	9	9	20	20	29	29
29	10	10	38	43	48	53
30	2	2	13	20	15	22
31	7	7	30	24	37	31
32	19	14	7	7	26	21
MEAN	8.56	8.72	28.25	28.00	36.81	36.72
ST DEV	10.75	10.58	19.75	18.80	19.43	18.85
MODE	2	2	6	20	15	21

where variations are as specified in section 5.4.2,

Et = the 'effort' value for the entire system (Thomas-II)

Ew = the 'effort' value for the weighting subsystem.

the other hand, the comparison between the subsystem and Thomas-II (Table 5.8), shows no significant difference in the performance of the two. This is true for all the three variations a1, a2 and a3. We, however, defer any inferences at this stage until after the presentation of the results in the next subsection on the minigraph subsystem.

5.4.5 The Minigraph Subsystem

In the minigraph subsystem experiments, the network used by Thomas-II was broken down into what we called the minigraphs, without the association weights on the links. The experimental procedures involved are similar to what has been described for the experiments in the last section, except that instead of the weighted supergraph, the experiments used a clustered supergraph to test retrieval effectiveness of the minigraph approach.

The results for the comparison between this subsystem and the standard Thomas (Tables 5.9 and 5.10) indicate no difference in their performance for any of the three variations a1), a2) and a3). Furthermore, as can be observed from tables 5.9 and 5.1, most of the values of I and T for the minigraph subsystem are about the same as those for the standard system. On the other hand, on comparing this subsystem with Thomas-II (Table 5.11), Thomas-II is found to be of better performance in the cases of variations a1 ($p<0.025$), and a3 ($p<0.05$) - there being no difference with variation a2. From this result and what was reported in the last subsection for the weighting subsystem, one may infer that the overall performance of the entire system, Thomas-II reported in section 5.4.3 has been

Table 5.9 Interactions and Tokens: The Minigraph Subsystem.

	Variation a1)		Variation a2)		Variation a3)	
QUERY	I1	T1	I2	T2	I3	T3
1	1	2	4	7	5	9
2	0	1	3	5	3	6
3	5	6	4	4	9	10
4	0	1	8	11	8	12
5	2	4	4	4	6	8
6	0	1	10	13	10	14
7	6	9	3	8	9	17
8	0	1	3	4	3	5
9	1	3	1	3	2	6
10	7	8	2	4	9	12
11	1	5	5	10	6	15
12	0	1	6	8	6	9
13	2	7	2	4	4	11
14	3	8	5	6	8	14
15	0	1	5	7	5	8
16	0	1	6	6	6	7
17	0	1	13	17	13	18
18	0	1	4	7	4	8
19	1	2	10	11	11	13
20	0	1	1	1	1	2
21	2	3	1	1	3	4
22	0	1	9	9	9	10
23	5	8	2	2	7	10
24	2	7	5	6	7	13
25	2	5	1	1	3	6
26	0	1	11	14	11	15
27	0	1	3	5	3	6
28	1	4	3	5	4	9
29	3	6	8	11	11	17
30	0	1	3	5	3	6
31	1	2	3	3	4	5
32	3	4	1	2	4	6
MEAN	1.5	3.34	4.66	6.38	6.16	9.72
ST DEV	1.93	2.71	3.22	3.97	3.12	4.17
MODE	0	1	3	4	3	6

where variations are as specified in section 5.4.2,
  Ij = the no. of interactions according to variation aj), and
  Tj = the no. of tokens according to variation aj).

Table 5.10 Effort values: Thomas versus The Minigraph Subsystem

$(\alpha:\beta) = (5:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	Em	Es	Em	Es	Em
1	7	7	27	27	34	34
2	1	1	14	20	15	21
3	19	31	39	24	58	55
4	1	1	51	51	52	52
5	8	14	24	24	32	38
6	1	1	63	63	64	64
7	39	39	23	23	62	62
8	1	1	19	19	20	20
9	7	8	8	8	15	16
10	43	43	14	14	57	57
11	10	10	35	35	45	45
12	1	1	32	38	33	39
13	17	17	14	14	31	31
14	23	23	31	31	54	54
15	1	1	32	32	33	33
16	1	1	42	36	43	37
17	1	1	70	82	71	83
18	1	1	27	27	28	28
19	7	7	61	61	68	68
20	1	1	6	6	7	7
21	13	13	6	6	19	19
22	1	1	54	54	55	55
23	33	33	12	12	45	45
24	23	17	25	31	48	48
25	15	15	6	6	21	21
26	1	1	69	69	70	70
27	1	1	20	20	21	21
28	9	9	20	20	29	29
29	21	21	45	51	66	72
30	1	1	20	20	21	21
31	7	7	24	18	31	25
32	19	19	7	7	26	26
MEAN	10.44	10.84	29.38	29.66	39.81	40.50
ST DEV	11.82	12.13	18.90	19.87	18.63	19.36
MODE	1	1	20	20	21	21

where variations are as specified in section 5.4.2,
Es = the 'effort' value for the standard system (Thomas)
Em = the 'effort' value for the minigraph subsystem.

Table 5.11 Effort values: Thomas-II versus The Minigraph Subsystem

$(\alpha:\beta) = (5:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Et	Em	Et	Em	Et	Em
1	2	7	38	27	40	34
2	2	1	13	20	15	21
3	19	31	36	24	55	55
4	2	1	57	51	59	52
5	2	14	24	24	26	38
6	2	1	13	63	15	64
7	37	39	19	23	56	62
8	2	1	18	19	20	20
9	8	8	6	8	14	16
10	43	43	14	14	57	57
11	2	10	43	35	45	45
12	2	1	30	38	32	39
13	11	17	14	14	25	31
14	23	23	31	31	54	54
15	2	1	25	32	27	33
16	1	1	66	36	67	37
17	1	1	70	82	71	83
18	2	1	26	27	28	28
19	7	7	61	61	68	68
20	1	1	6	6	7	7
21	7	13	6	6	13	19
22	1	1	54	54	55	55
23	27	33	12	12	39	45
24	10	17	20	31	30	48
25	8	15	7	6	15	21
26	1	1	69	69	70	70
27	2	1	18	20	20	21
28	9	9	20	20	29	29
29	10	21	38	51	48	72
30	2	1	13	20	15	21
31	7	7	30	18	37	25
32	19	19	7	7	26	26
MEAN	8.56	10.84	28.25	29.66	36.81	40.50
ST DEV	10.75	12.13	19.75	19.87	19.43	19.36
MODE	2	1	6	20	15	21

where variations are as specified in section 5.4.2,
Et = the 'effort' value for the entire system (Thomas-II)
Em = the 'effort' value for the minigraph subsystem.

mainly due to the weighting subsystem and that the clustering of the database did not have much effect on the experiments.

In the next section, we give a further discussion on the performance of the minigraph subsystem.

5.4.6 The Minigraph Subsystem (Revisited)

From various work on information retrieval systems (Crouch 1975, Croft 1978), it can be said that clustering does help improve system performance, as the use of the technique tends to group together documents that are likely to be relevant to a given query (Section 2.4). However, we have seen from the experiments reported in the last section that the clustering of the network into minigraphs has had no significant effect on the performance of Thomas-II. One may then wonder why this tendency of performance improvement as a result of clustering has not been shown in our experiments. Two possible reasons are mentioned here.

One reason is that our test collection is very small; only 225 documents (cf. Mushens 1981). With a collection of this size, the effect of clustering is likely to be far less noticeable than collection sizes used in operational systems. With large collections, the documents which are likely to satisfy a user's information needs will be a lot more scattered in the entire collection than will be in the case of a small collection.

Accordingly, on clustering such large collections, the previously scattered relevant documents are more likely to be drawn closer to one another and hence cause noticeable effects than will be in the case of small test collections.

Assuming that our collection size were large and comparable with those of operational systems, there would then be the need for us to reconsider the relation used by Thomas-II to determine the 'involvement' of each document in the context graph. This relation, the 'involvement measure' I, for a document D has been defined (for the case where all links in the network have unit association weights) as

$$I = \frac{\text{no. of items linked to D in the context graph}}{\text{no. of items linked to D in the supergraph}}$$

(We note that various other relations may be substituted, if found appropriate, as exemplified in section 5.4.7.1.).

Consider an unclustered network, SG, similar to our supergraph, which contains among others, a document, d7. We assume that d7 has 8 other items directly linked to it as in fig 5.2a. Assume our network is clustered into 5 minigraphs, MG1 to MG5, where MG3 is the one containing the document d7 and its associates (fig 5.2b). Suppose the context graph at the time of the computation of the involvement values has five of the eight associated nodes of d7 in it.

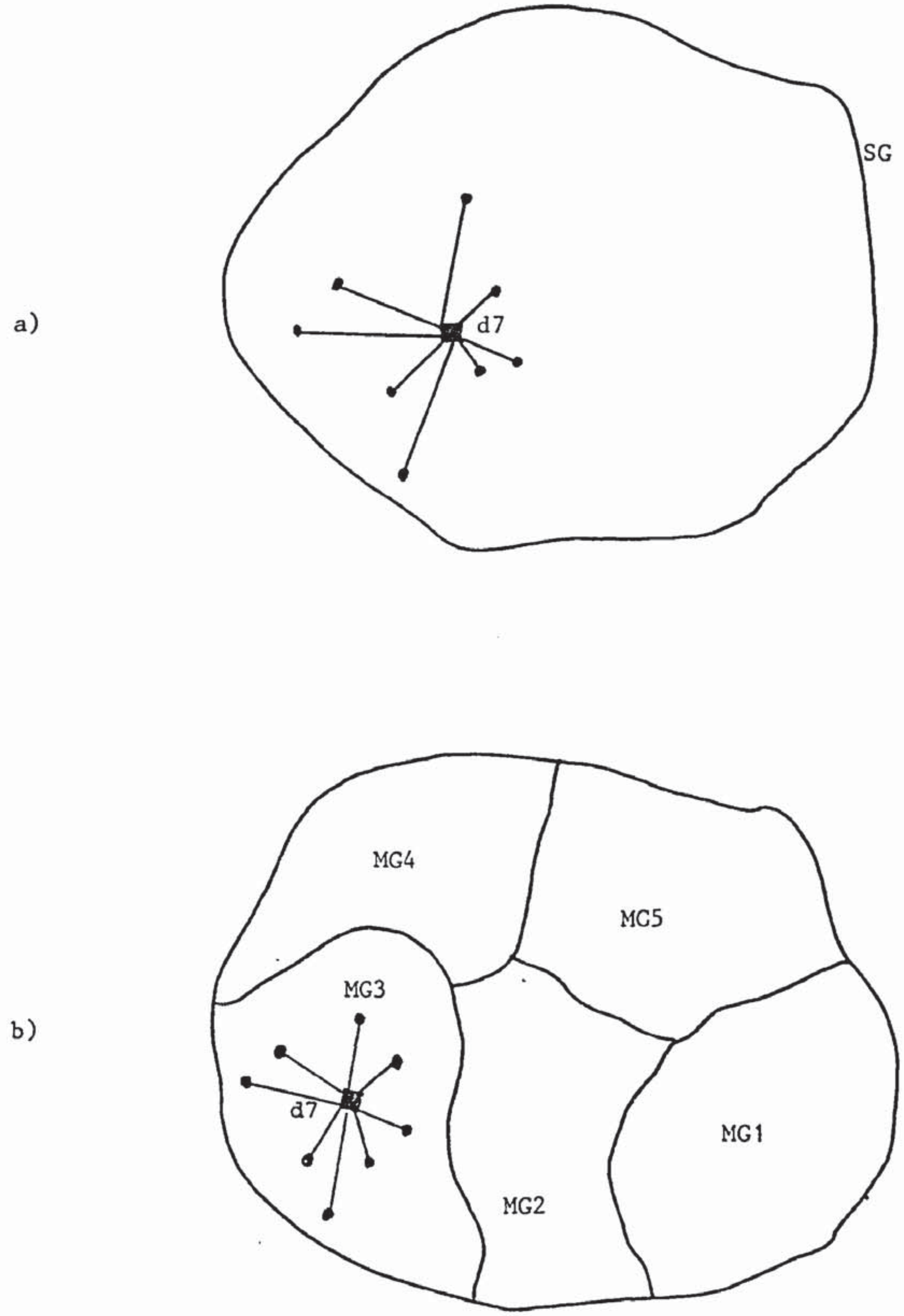


Figure 5.2 A Supergraph : Clustered and Unclustered

Then from our relation, we have

$$Id7 = \frac{\text{no. of items linked to d7 in the context graph}}{\text{no. of items linked to d7 in the supergraph}}$$

$$= \frac{5}{8}$$

This value of Id7 may be the same in the two cases of the clustered and unclustered small-sized network SG depending upon whether or not some of the links between d7 and its associated nodes are broken as a result of the clustering. If no links are broken (as assumed above), then the number of associates of d7, 8, will remain constant for both the clustered and unclustered networks, and similarly, the context graphs will be alike in both cases. (We recall that in creating or modifying the context graph, we bring in nodes from the supergraph as well as the links between them.) Consequently, both the numerator and denominator in our relation Id7 will remain the same in both the clustered and unclustered cases, unless the clustering has caused a significant restructuring of the network. There seems to be a lack of any significant restructuring in our small supergraph.

With very large document collections - and hence very large networks - the effect of clustering on restructuring the entire network will be more distinct. Secondly, it may be necessary to approximate the number of items linked to a document in the entire collection, Nsg, by the number of items linked to that document in the minigraph to which it belongs, Nmg. And as a particular document and most of its associated nodes are likely to be in the same minigraph, Nmg obtained from that minigraph will give a good approximation to Nsg. Hence, a relation

$$I = \frac{\text{no. of items linked to document in context graph}}{\text{no. of items linked to document in its minigraph}}$$

may be a substitute to the relation used in our experiments.

There is one other advantage in the clustering of the network; the reduction in the disc access during an interaction, consequently, leading to faster and more economic searches. As stated earlier, the entire database may be stored on backing storage, and only the required portions brought into core as the dialogue continues. Owing to the fact that Nsg requires the determination of the number of nodes linked to the particular document in the entire network, each involvement measure computation may require as many as n disc accesses, if there are n documents in the context graph at the time. With a clustered network, most of the documents for which the involvement values have to be computed are likely to be in the same minigraph. Hence, having brought that cluster into core as the minigraph for the particular interaction, Nsg for all those documents would be obtained without any further disc accessing - a feature which is not likely with an unclustered network.

5.4.7 Other Experiments

In the foregoing sections, we presented results obtained from the experiments which were carried out using Thomas-II. Three other sets of experiments were performed each of which required slight modifications to Thomas-II. These experiments - which we report in this section - are explorations into possible further changes to Thomas-II and its networks.

5.4.7.1 On Alternative Document Selection Criteria

At various stages in interactive document or reference retrieval, the user in front of the machine needs to be shown an item - preferably a document title. The user then has to make decisions as to how much the displayed document is likely to satisfy the information need that brought her to the system. The issue of interest here is the basis upon which documents are selected by the system to display to the user.

Generally, document selection criteria are based on how well the user's query (represented by a set of words and at times combined with logical operators), matches the documents in the entire collection (each represented by a set of words deemed appropriate enough to portray what the document is about.) Our retrieval strategy is in line with recent calls for a shift in the basis of document selection criteria, from this process of document-query matching to the modelling of the user's concepts (Bar Hillel 1975, Koll 1981). In our strategy, structural models are created of the entire collection (the world model or the supergraph) as well as the searcher's area of interest (the user model or context graph), the latter of which changes dynamically as the interaction goes on. The document selection criterion is based on these two structural models.

In the experiments reported above, the document selection criterion used was to find the ratio between the number of items linked to a given document D in the context graph at the time and the number linked to that document in the world model. This ratio has been termed the 'involvement' value I of that document in the user-model. Thus

$$I = \frac{\text{no. of items linked to D in the context graph}}{\text{no. of items linked to D in the supergraph}}$$

$$= \frac{Ncg}{Nsg}$$

(We note that I has been modified to take care of the association weights, as described in section 4.6).

As has been mentioned in section 5.4.6, various relations may be used provided we stick to the structural models and not revert to the usual document-query match. To this end, an examplary alternative involvement measure has been used to indicate that the above ratio used is not a rule-of-thumb that must necessarily go with Thomas-II.

The new involvement measure Ia considered, takes into account the total number of items in the context graph, Tcg, at the time of the computation.

Thus:

$$Ia = \frac{Ncg}{Nsg} \times \frac{Ncg}{Tcg}$$

One reason for the composition of Ia as it is, has been to enable an implementation on an intelligent terminal (Jamieson 1980, 1981). In order to evaluate I (above), it is necessary to access all the nodes associated with all the documents in the context graph at that time. This can be quite expensive, especially when a large collection is to be accessed through an intelligent terminal. With this alternative relation, the number of documents in the context graph to which the divisor Nsg will have to be applied can be limited to those with appreciably high values of Ncg/Tcg.

Another reason is that with Ia, the system will tend to show the user documents with more associated nodes than it would with I. Thus suppose our context graph contains 15 nodes, ie, Tcg=15, including two documents D1 and D2. Suppose D1 has 4 nodes associated with it in the supergraph out of which 2 are in the context graph, and D2 has 8 nodes in the supergraph with 3 in the context graph. Then we have the following situation:

	Nsg	Ncg	I	Ia
D1	4	2	0.5	0.067
D2	8	3	0.375	0.075

Using our earlier involvement measure, I, Thomas-II will select D1 out of these two documents for display. Whereas, using Ia, document

Table 5.12 Interactions and Tokens: Involvement Measure Expt.

	Variation a1)		Variation a2)		Variation a3)	
QUERY	I1	T1	I2	T2	I3	T3
1	0	1	5	8	5	9
2	0	1	2	4	2	5
3	2	3	4	5	6	8
4	1	4	7	8	8	12
5	2	4	4	4	6	8
6	0	1	13	16	13	17
7	6	9	3	8	9	17
8	0	1	3	4	3	5
9	1	3	1	3	2	6
10	7	8	2	4	9	12
11	1	5	5	10	6	15
12	0	1	6	8	6	9
13	2	7	2	4	4	11
14	4	9	6	7	10	16
15	0	1	4	6	4	7
16	0	1	11	11	11	12
17	0	1	7	11	7	12
18	0	1	4	7	4	8
19	1	2	10	11	11	13
20	0	1	1	1	1	2
21	1	2	1	1	2	3
22	0	1	9	9	9	10
23	5	8	2	2	7	10
24	4	10	2	2	6	12
25	1	3	1	2	2	5
26	0	1	11	14	11	15
27	0	1	3	5	3	6
28	1	4	3	5	4	9
29	2	5	6	9	8	14
30	1	3	2	3	3	6
31	1	2	5	5	6	7
32	3	4	1	2	4	6
MEAN	1.44	3.38	4.56	6.22	6.00	9.59
ST DEV	1.88	2.84	3.29	3.79	3.18	4.05
MODE	0	1	2	4	6	12

where variations are as specified in section 5.4.2,
  Ij = the no. of interactions according to variation aj), and
  Tj = the no. of tokens according to variation aj).

Table 5.13 Effort values: Thomas versus Involvement Measure Expt.

$(\alpha:\beta) = (5:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	Ei	Es	Ei	Es	Ei
1	7	1	27	33	34	34
2	1	1	14	14	15	15
3	19	13	39	25	58	38
4	1	9	51	43	52	52
5	8	14	24	24	32	38
6	1	1	63	81	64	82
7	39	39	23	23	62	62
8	1	1	19	19	20	20
9	7	8	8	8	15	16
10	43	43	14	14	57	57
11	10	10	35	35	45	45
12	1	1	32	38	33	39
13	17	17	14	14	31	31
14	23	29	31	37	54	66
15	1	1	32	26	33	27
16	1	1	42	66	43	67
17	1	1	70	46	71	47
18	1	1	27	27	28	28
19	7	7	61	61	68	68
20	1	1	6	6	7	7
21	13	7	6	6	19	13
22	1	1	54	54	55	55
23	33	33	12	12	45	45
24	23	30	25	12	48	42
25	15	8	6	7	21	15
26	1	1	69	69	70	70
27	1	1	20	20	21	21
28	9	9	20	20	29	29
29	21	15	45	39	66	54
30	1	8	20	13	21	21
31	7	7	24	30	31	37
32	19	19	7	7	26	26
MEAN	10.44	10.56	29.38	29.03	39.81	39.59
ST DEV	11.82	12.03	18.90	20.03	18.63	19.53
MODE	1	1	20	14	21	21

where variations are as specified in section 5.4.2,
Es = the 'effort' value for the standard system (Thomas)
Ei = the 'effort' value for the involvement measure expt.

D2 will be chosen for display. As references are displayed with their associated nodes, the user will be shown eight other terms with D2 in contrast with four terms when D1 is displayed. Thus, the use of Ia would enhance the browsing aspect of the search, an aspect which is quite important from the point of view of an interactive user (section 2.1).

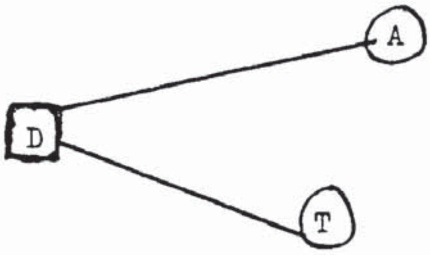
This new involvement measure was used in a set of experiments to replace I and the evaluation performed in a manner similar to what has been done for the experiments reported above, using the ratio of (5:1) for the relative efforts associated with interactions and tokens. The results, as shown in Tables 5.12 and 5.13, however, indicate no significant difference between the subsystem using the involvement measure Ia and the standard experiment using the involvement measure I (Ofori-Dwumfuo 1982).

5.4.7.2 The Links Between Subject Terms

As has been described in section 4.3, the database of Thomas-II comprises document titles, author names, subject terms and synonyms of subject terms, all interwoven into a network. The associations between these items in the network have been limited to document-author, document-subject term, subject term-subject term and subject term-synonym of subject term (see fig 4.6).

There is however, one major problem which goes with the creation of a network of this type. The problem of determining the links between the subject terms. Trivially, every document has an author. Hence the document-author link is about the simplest to set up. In the case of the document-subject term link, one may rely on the various indexing techniques (section 2.3). Hence, even with very large document collections, automatic indexing methods are feasible to determine the terms associated with the documents.

Determining the links between the subject terms seems to pose the greatest problems. It may be necessary to even construct a thesaurus (Kim and Kim 1977) and then determine, pairwise, the relation, if any, between the given words. Alternative methods could involve the computation of huge similarity matrices or various computations based on co-occurrences of words in texts.



where

D = Document reference

A = Author name

T = Subject Term

Figure 5.3 Network without Term-Term Links

Table 5.14 Interactions and Tokens: Term Links Expt.

	Variation a1)		Variation a2)		Variation a3)	
QUERY	I1	T1	I2	T2	I3	T3
1	1	2	4	7	5	9
2	0	1	3	5	3	6
3	3	4	6	9	9	13
4	0	1	8	11	8	12
5	0	1	2	3	2	4
6	0	1	10	13	10	14
7	6	9	3	8	9	17
8	0	1	3	4	3	5
9	1	3	1	3	2	6
10	7	8	2	4	9	12
11	1	5	5	10	6	15
12	0	1	5	7	5	8
13	2	7	2	4	4	11
14	3	8	5	6	8	14
15	1	3	6	9	7	12
16	0	1	7	7	7	8
17	2	5	9	11	11	16
18	0	1	3	6	3	7
19	1	2	10	11	11	13
20	0	1	1	1	1	2
21	2	3	1	1	3	4
22	0	1	9	9	9	10
23	5	8	2	2	7	10
24	2	7	3	4	5	11
25	2	5	1	1	3	6
26	0	1	10	13	10	14
27	0	1	3	5	3	6
28	1	4	3	5	4	9
29	3	6	7	10	10	16
30	0	1	3	5	3	6
31	0	1	4	4	4	5
32	0	1	1	2	1	3
MEAN	1.34	3.25	4.44	6.25	5.78	9.50
ST DEV	1.84	2.70	2.93	3.53	3.13	4.22
MODE	0	1	3	4	3	6

where variations are as specified in section 5.4.2,
  Ij = the no. of interactions according to variation aj), and
  Tj = the no. of tokens according to variation aj).

Table 5.15 Effort values: Thomas versus Term Links Expt.

$(\alpha:\beta) = (5:1)$

	Variation a1)		Variation a2)		Variation a3)	
QUERY	Es	El	Es	El	Es	El
1	7	7	27	27	34	34
2	1	1	14	20	15	21
3	19	19	39	39	58	58
4	1	1	51	51	52	52
5	8	1	24	13	32	14
6	1	1	63	63	64	64
7	39	39	23	23	62	62
8	1	1	19	19	20	20
9	7	8	8	8	15	16
10	43	43	14	14	57	57
11	10	10	35	35	45	45
12	1	1	32	32	33	33
13	17	17	14	14	31	31
14	23	23	31	31	54	54
15	1	8	32	39	33	47
16	1	1	42	42	43	43
17	1	15	70	56	71	71
18	1	1	27	21	28	22
19	7	7	61	61	68	68
20	1	1	6	6	7	7
21	13	13	6	6	19	19
22	1	1	54	54	55	55
23	33	33	12	12	45	45
24	23	17	25	19	48	36
25	15	15	6	6	21	21
26	1	1	69	63	70	64
27	1	1	20	20	21	21
28	9	9	20	20	29	29
29	21	21	45	45	66	66
30	1	1	20	20	21	21
31	7	1	24	24	31	25
32	19	1	7	7	26	8
MEAN	10.44	9.97	29.38	28.04	39.81	38.41
ST DEV	11.82	11.68	18.90	17.96	18.63	19.48
MODE	1	1	20	20	21	21

where variations are as specified in section 5.4.2,

Es = the 'effort' value for the standard system (Thomas)

El = the 'effort' value for the expt. on links.

In view of this problem of determining the associations between the subject terms, a set of experiments was carried out using Thomas-II on a network in which the term-term link was cut, changing fig 4.6 to fig 5.3 (above). This set of experiments was to find out whether our retrieval program would perform well using such a simplified network. In these experiments, when a subject term node is brought into the context graph, the program has to check if the node bringing it into the model is itself a subject term. If it is, the incoming node is suppressed.

The results (Table 5.14 and 5.15), however, show no significant difference between the term-term links experiment and the standard Thomas (see Section 6.2).

5.4.7.3 User-Induced Dynamic Clustering

In an interaction with a user, we have said that Thomas-II creates, and dynamically modifies, a structural model of the user's area of interest. This model, a subset of the entire world model of the program, may be considered as a cluster dynamically produced from the user's search. An interaction with our program may, therefore, be said to be a form of dynamic clustering which is entirely user-induced.

With this view of a user-Thomas-II interaction, we have given thought to the idea that on storing the clusters produced from each user interaction, it may be possible after a certain number of interactions to re-group the database into an entirely new set of

Table 5.16 User-Induced Dynamic Clustering:
Overall Search-time in seconds

QUERY	S	C
1	9.2	8.4
2	7.2	6.7
3	13.4	11.0
4	27.8	19.0
5	10.3	8.2
6	13.4	10.1
7	12.2	11.6
8	9.2	7.6
9	7.2	6.6
10	13.4	11.9
11	15.0	10.0
12	11.4	8.4
13	7.5	6.9
14	16.4	11.1
15	10.8	8.0
16	20.1	12.4
17	15.6	10.8
18	6.8	6.8
19	23.2	15.0
20	6.5	6.1
21	7.5	7.1
22	19.4	11.9
23	22.6	17.5
24	12.3	9.9
25	8.0	7.4
26	16.8	12.0
27	8.9	7.0
28	7.8	7.1
29	21.9	14.4
30	8.3	7.1
31	11.1	8.5
32	6.4	6.4
MEAN	12.7	9.8
ST DEV.	5.7	3.3
MODE	13.4	7.1

where S represents the standard experiment
C represents the dynamic clustering expt.

rrinigraphs based solely on previous user-induced clusters, and secondly, to reduce the search time.
Consequently, experirrents have been carried out with the suitable changes in Thorras-II, in which we stored all the node records of iterrs that have been in the context graph, as well as those explicitly rejected by the user, and then noted the various search times.

As node records to be used during one search rray corre frorr different blocks on the disc, and since an iterr rejected by the user rray later be brought into the rrodel if the user changes her rrind on it, storing all node records of iterrs that have been in the context graph also helps reduce the nurrber of disc accesses. For, if a node A belonging to block X is to be recalled after having been rejected and block X has now been overwritten by block Y in core, it rray not be necessary to obtain block X again frorr backing store in order to fetch node A if the node record of A has been stored sorrewhere in core (for having been an entry in the context graph).

In this case, the experirrents show faster turnround tirres for all but two queries, corrpared with the tirres taken by the standard experirrents (Table 5.16) - the irrproverrent in tirre indicating the reduction in the call of the direct access procedures.

5.5 Surrrary

We have presented the experirrents conducted using Thorras-II. The experirrents have been in three rrajor groups: the entire systerr, Thorras-II, in which the clustered network has association weights on the links between its nodes; experirrents involving association weights on the links between items in the database unclustered; and experiments based on the clustering of the database into minigraphs without the association weights.

The results indicate that Thomas-II performs better than Thomas, in that (20 per cent) less effort is required of the user, specifically, in the retrieval of the first relevant document. Results of experiments using only the association weights follow the same pattern as those for the main system, whereas, experiments using only the clustered minigraphs show no difference in system performance.

As a supplement to the above experiments, three other experiments have been reported to illustrate that various document selection criteria could be used; to determine if Thomas-II network could exclude term-term links; and to indicate that the user-induced clusters created by Thomas-II during user-interactions could be stored, first to reduce the time spent during the search, and secondly, to create an entirely new set of clusters based upon a number of user-interactions.

Chapter 6

CONCLUDING SUMMARY

6.1 Synopsis

In spite of the increasing emphasis laid on the availability of information in our society, it is far from certain that every information seeker can obtain the items she requires. The problem seems to be how to easily identify, locate and retrieve the needed information from the mass of literature around us. To this effect, various information retrieval systems have been in use, ranging from purely manual card catalogues to the recent automated on-line systems, and modern technology tends to favour the choice of the on-line systems, especially, regarding the relatively little time required for a literature search and the coverage of the search.

However, most of the on-line retrieval systems may be said not to be flexible enough for their users. Specifically, most of these systems assume that the user knows exactly what she is looking for before coming to the system. They also assume that the user can express her information need, and furthermore, that the user's expression can be formatted according to the precise specifications that the systems lay down, (because these precise specifications of the user's needs - queries - will have to be matched symbol by symbol with the document specifications of the systems). Simply-stated, the systems turn out to be rigid with their requirements, and users are expected to adapt to them.

In the three references (1974, 1977a,b), Oddy gave a description of a computer program, called Thomas, which attempts to retrieve references for the user through a dialogue in which the program plays the role of the subject expert (although in a relatively unsophisticated manner). The design of Thomas took into consideration the facts that

- users are usually unable to fully recognize their information needs before approaching information retrieval systems

- even on recognizing the need for information, most users are unable to precisely express their needs at all, let alone as formal queries

- retrieval systems must ultimately be accessed directly by end-users, not intermediaries, and that

- it is time information retrieval systems switched from the symbol matching of query against document to modelling how users would differentiate documents with respect to their requests.

Oddy's retrieval program has been the central issue of this thesis. Aspects of enhancement have been discussed . It was noted that since the retrieval program, Thomas, uses a strategy which involves building structural models of its database and the user's area of interest, and these models are in a form of a network made up of nodes (representing concepts) and links between those nodes, some supplementary design objectives could be added to the points listed above.

Firstly, since the nodes in the network represent concepts, and various concepts have various degrees of similarities and associations, Thomas could be made to 'recognize' this fact. That is to say that instead of creating structural models which assume equal levels of similiraties and associations between its nodes (concepts), the links between the nodes in the models will be made to have numeric values - association weights or strengths - which would indicate the degree of similarity between the linked concepts. In retrieval, the system would consider each node in its models, the other nodes associated with it, as well as the association strengths between them.

Secondly, on accepting that various concepts have various degrees of similarities between them which could be represented by their association strengths, one would like to cluster together concepts which are closely related to one another, so that in an interaction with Thomas, only the relevant clusters would come into play, rather than the entire set of nodes. Furthermore, in order that the system be more user-friendly, the problem of deciding which clusters are relevant for any particular interaction will not be tackled by the user, but by the program.

Apart from the enhancement in the design of Thomas, this thesis links the information retrieval design work with a current notion in psychology - "the cognitive viewpoint" (De May 1980). The central point of the cognitive view is that processing of information, whether perceptual or symbolic, is mediated in a person by a series of concepts which are a model of that person's world. Specifically, in the context of information retrieval this viewpoint is the basis of the current calls for IR systems to switch from symbol matching to cognitive modelling. In our work here, we have attempted (through our retrieval program) to model a dialogue between two people - the information seeker and the information expert. The user of Thomas is the information seeker, while Thomas plays the role of the information expert.

The implementation of the new system, Thomas-II, has been reported in the thesis and its performance compared with the performance of the earlier version. The evaluation has been based on the 'effort' required of a user to realize a particular level of retrieval performance; the less the effort required to attain a performance level on a system, the better that system. User effort has been assumed to be time-related and reflected mainly in the process of the user judging the usefulness of an item displayed to her, and the physical process of typing her response into the system.

Three levels were chosen for the determination of the user effort; from the start of the user's search to the time of the retrieval of the first relevant document; from the retrieval of the first relevant item to the retrieval of the last relevant item; and from the start to the retrieval of the last relevant document. The new system showed better performance (signficant at 0.05 level using the Wilcoxon matched pairs signed-ranks test) in the first and third cases. That is to say that (20 per cent) less user effort is required by Thomas-II than by Thomas in the retrieval of the first relevant document and in the retrieval of all the relevant documents, ie, from the start to end of the search. However, considering the effort required from the retrieval of the first relevant to the last relevant item, there was no significant difference in the performance of the two systems.

Although Thomas-II does not show an all-round improvement in performance, we noted that the user of an interactive system facing large databases in a real-life environment may not be very interested in how well the system keeps on course up to the display of the last relevant document. To such a user, the less effort required of her for the retrieval of the first few relevant items, the better the system.

Apart from comparing Thomas-II with Thomas, experiments were reported to give an indication of how the new system would perform using only the association strengths without the clustering of the nodes, and with the nodes clustered but without the association weights. In the case of the former, the results were found to follow the same pattern as was reported for the entire new system, Thomas-II. In the latter case, however, there was no significant difference, possibly due to the small collection size used - all comparisons being made against the old version of Thomas.

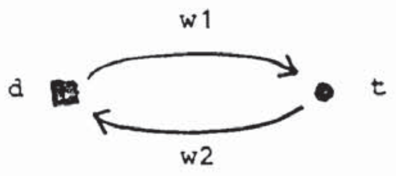
Three other experiments were reported. The first was to indicate that alternative document selection criteria could be substituted for what was used in the experiments reported earlier. The second was to investigate the possibility of excluding the links between various subject terms that may occur in the network of Thomas-II and the third explored the usefulness in storing the user-induced dynamic clusters produced in each interaction in order to reduce the overall search time (by reduction in disc accessing), and to help create an entirely new set of clusters based upon a number of user-interactions.

6.2 Aspects for Further Work

The field of information retrieval is relatively new, and so all aspects of it still need further research. Nevertheless, we would join those who have recently realized the need for more behavioural type of work and reiterate the call for research into systems which model the human cognitive processes involved in information retrieval. Symbol matching of query against document, which has hitherto been the order of the day, should now be viewed as a pace-setter for more behavioural oriented work in information storage and retrieval.

Further to the call for future retrieval systems to consider modelling how information seekers would differentiate documents with respect to their requests, one would like to add the need for research into evaluation techniques for interactive systems of our kind. There is no doubt that the literature has a substantial amount of work on evaluation parameters and techniques. Yet, one can hardly pick on one of those techniques and confidently claim it would be suitable for all retrieval systems, including real user-oriented interactive systems. The technique adopted in this thesis could be analysed to determine its suitability for other interactive systems.

With regard to the retrieval program, Thomas-II, described in this thesis, various aspects could be considered for further work. The system could be made to receive as input a piece of text issued by the user in natural language. The system would then determine the appropriate words or concepts from that text with which to start the interaction (cf Belkin et al 1979). Regarding the association weights, the network could have directed graphs with different association weights in either direction. Thus, for example, as in the case of check tags (terms which the indexer must consider for each document, and therefore have much higher postings than average), the association strength, w1, from the document node, d, to the check tag; t, could far exceed the strength, w2, from the check tag to the document node. Thus



where w1 >> w2.

The consequence of this would be that the reference node d can bring the check tag into the context graph, but not necessarily the reverse. (We note that this has been implemented in its simplest case in this thesis by setting w2 to zero.)

Further work could also be done on the aspects explored in section 5.7. Other alternative document selection criteria could be investigated. (One should, however, be careful to avoid reverting to any established method of symbol matching of query against document.) The idea of the user-induced dynamic clustering could also be further investigated to determine, for example, how suitable such clusters would be as minigraphs for future use, how much they will affect the system's performance, how well such user-induced minigraphs will perform using association weights, etc.

From the experiments in which term-term links were removed, one would have expected that the removal of the links should lead to a reduction in the performance of the system. That these experiments showed no significant difference from the standard Thomas poses the question as to whether the term-term links could be excluded from Thomas' network. This could be explored further.

What we consider the most important aspect for further work is an implementation of Thomas involving real-user interaction, and using larger collections, possibly via the use of an intelligent terminal (Jamieson 1981). The experiments reported in this thesis have used a small laboratory test collection and involve simulation of the user in a dialogue with Thomas. With real users, the evaluation of the system would involve the users' subjective views on how much the system helps to resolve their 'anomalies' and how much effort they had put into the system in order to satisfy their information need. As regards the test collection size, it is hoped that the minigraph aspect incorporated into the system will help reduce problems which might arise due to the handling of large networks - as the entire network (supergraph) could be segmented into smaller and more manageable networks (minigraphs).

Appendix A

THOMAS-II Skeleton Program

```
COMMENT The following data types which occur in the specification
below need clarification:

CAR(l)          :  the first element of a list l
CDR(l)          :  the list resulting after the removal of the first
                   element of the list l
CONS(e, l)      :  the list resulting after joining the element e to
                   the front of list l
APPEND(l1, l2) :   the list resulting after joining list l1 to the
                   end of list l2

POINT           :  represents the identity of a node.  It has the
                   selector function: node-type(p) with values:
                   DOC  : p is a document node
                   SUB  : p is subject node
                   AUT  : p is an author node
UNPOINT         :  null identification
EDGE            :  represents a weighted edge in a graph, with
                   selector functions:
                    target OF edge  : is a POINT
                    strength OF edge: is an INTEGER
GRAPH           :  is a POINT SET, ie, a set of nodes
MESSAGE         :  a structure to contain the information in a user's
                   message to Thomas, has selector functions:
                    reaction OF message (values STOP, NO, NONE, YES),
                    rejections OF message     : is a POINT SET
                    selections OF message     : is a POINT SET
                    requests OF message       : is a QUERY SET
QUERY           :  this is a POINT (in this implementation)
DISPLAY         :  a structure to contain information in a display
                   produced by Thomas, has selector functions:
                    doc-shown OF display     : is a POINT (value has
                                                node-type DOC)
                    nodes-shown OF display  : is a POINT ARRAY
ITEM            :  one of the components of the user's reaction to a
                   display, has selector functions:
                   item-type(item)          : values REACT, DNUM, NNUM
                    the other selector function depends on item-type:
                    if item-type = REACT     : reaction(item), values
                                                NO, NONE, YES, STOP
                    if item-type = DNUM      : display-no(item)
                                                 of value INTEGER
                    if item-type = NNUM      : node-num(item)
                                                 of value QUERY
SEARCH          :  a structure to contain details of the current
                   search, has selector functions :
                    search-title          :   a STRING
                    users-term            :   a POINT
                    user-id               :   an INTEGER
                    terms                 :   a POINT SET
                    rel-docs              :   a POINT SET with elements
                                              of node-type DOC
```

```
GLOBAL          :  denotes declaration that holds for all procedures
SPECIFY         :  coding of what follows depends on implementation
INIT            :  initialize the variable with what follows
//              :  what follows is a comment
```

```
GLOBAL ( COMMENT the following declarations hold throughout and are
         not repeated in the individual procedures

         POINT SET cont-nodes      // nodes in the context graph
         POINT SET inhib-nodes     // nodes inhibited from entry into
                                   // context graph
         POINT SET exp-nodes       // nodes explicitly requested
                                      or selected by user
         POINT SET sel-nodes       // nodes selected from
                                      last display by user
         POINT SET redisp-nodes    // redisplayed nodes
         POINT SET good-docs       // document nodes to which user
                                      says YES
         POINT SET accptd-docs     // document nodes user makes no
                                      comment on
         BOOLEAN had-enough        // TRUE when all searches are over
         BOOLEAN stop-requested    // TRUE when user's message is STOP
         BOOLEAN unity             // TRUE if context graph is not
                                      fragmented
         REAL performance          // indicates how well interaction
                                      is going
         DISPLAY last-display      // the current display
         SEARCH search-details     // details of the current search
         REAL memory               //  used to calculate performance
         REAL ARRAY score[1:3]     //  used to calculate performance
         REAL ARRAY stuck-score[1:3]
         REAL low                 //  threshold value for performance
         INTEGER small            //  used to determine whether a
                                      component of the context graph
                                      is worth keeping
         INTEGER search-limit     //  maximum no of iterations in
                                      procedure TOPIC-SEARCH
         BOOLEAN indid-views      //  TRUE if user has some view to
                                      superimpose on the context-graph
       )
```

```
COMMENT The following notation is used;
        Reserved words are in block capitals;
        Procedure names are in block capitals, with parameters
        enclosed in brackets - even if null;
        Identifiers are in small letters.
```

```
BEGIN   COMMENT main program;
  SET-SYSTEM-PARAMETERS();
  OPEN-FILES();
  INITIALIZE-LOG();
  REPEAT TOPIC-SEARCH() UNTIL had-enough;
  EVALUATE-LOG()
END;
```

```
PROCEDURE TOPIC-SEARCH();
BEGIN
  SET-UP-MODEL();
  REPEAT IMPROVE-MODEL() UNTIL USER-SATISFIED() OR UPTO search-limit
END;


PROCEDURE IMPROVE-MODEL();
BEGIN
  MESSAGE m;
  GET-USER-MESSAGE(m);
  INFLUENCE-STATE-OF-MODEL(m);
  RESPOND-TO-USER(m)
END;


BOOLEAN PROCEDURE USER-SATISFIED();
BEGIN
  IF stop-requested THEN CLOSE-DOWN() FI;
  USER-SATISFIED := stop-requested
END;


PROCEDURE INFLUENCE-STATE-OF-MODEL(m);
MESSAGE m;
BEGIN
  IF reaction OF m = STOP
  THEN stop-requested := TRUE
  ELSE stop-requested := FALSE;
       COMPUTE-SCORE(reaction OF m);
       PRUNE-CONTEXT(rejections OF m);
       ADD-TO-CONTEXT(selections OF m);
       FIND-NODES(requests OF m);
       UNIFY-CONTEXT-GRAPH()
  FI
END;


PROCEDURE COMPUTE-SCORE(r);
INTEGER r;
BEGIN
  REAL s;
  s := IF doc-shown OF last-display = UNPOINT
       THEN stuck-score(r)
       ELSE score(r)
       FI;
  performance := memory * performance + s
END;


PROCEDURE PRUNE-CONTEXT(rejects);
POINT SET rejects;
BEGIN
  cont-nodes := cont-nodes - rejects;
  inhib-nodes := inhib-nodes + rejects;
  exp-nodes := exp-nodes - rejects
END;
```

```
PROCEDURE ADD-TO-CONTEXT(chosen);
POINT SET chosen;
BEGIN
  inhib-nodes := inhib-nodes - chosen;
  cont-nodes := cont-nodes + chosen;
  cont-nodes := cont-nodes + LINKED-DOCUMENTS(chosen)
END;


PROCEDURE FIND-NODES(requests);
QUERY LIST requests;
BEGIN
  POINT SET nodes;
  nodes := LOCATE-NODES(requests);
  exp-nodes := exp-nodes + nodes;
  inhib-nodes := inhib-nodes - nodes;
  cont-nodes := cont-nodes + nodes + STARS(nodes)
END;


POINT SET PROCEDURE LOCATE-NODES(requests);
QUERY LIST requests;
BEGIN
  QUERY q;
  POINT SET result (INIT NULL);
  FOR EACH q IN requests
  DO result := result + MATCHING-NODES(q) OD;
  LOCATE-NODES := result
END;


POINT SET PROCEDURE STARS(subset);
POINT SET subset;
BEGIN
  POINT SET result (INIT NULL);
  POINT p;
  FOR EACH p IN subset
  DO
    result:= result + {LINKED-TO(p) - inhib-nodes - cont-nodes}
  OD;
  STARS := result
END;


POINT SET PROCEDURE LINKED-DOCUMENTS(subset);
POINT SET subset;
BEGIN
  POINT SET result (INIT NULL);
  POINT p, q;
  FOR EACH p IN subset
  DO
    result := result +
              { {q ∈ LINKED-TO(p): node-type OF q = DOC}
                - inhib-nodes - cont-nodes
              }
  OD;
  LINKED-DOCUMENTS := result
END;
```

```
PROCEDURE UNIFY-CONTEXT-GRAPH();
BEGIN
  GRAPH SET k;
  k := CONNECT-COMPONENTS(cont-nodes);
  IF |k| <= 1    //    |.| denotes set cardinality
  THEN unity := TRUE
  ELSE LOG-COMPONENTS(|k|);
       DISCARD-USELESS-COMPONENTS(k);
       unity := IF |k| > 1 THEN TRY-JOIN(k) ELSE TRUE FI;
       LOG-COMPONENTS(|k|)
  FI
END;


PROCEDURE DISCARD-USELESS-COMPONENTS(components);
GRAPH SET components;
BEGIN
  POINT SET c;
  FOR EACH c IN components
  DO IF |c| <= small
        AND c ∩ exp-nodes = NULL
        AND c ∩ sel-nodes = NULL
     THEN cont-nodes := cont-nodes - c;
          components := components - c
     FI
  OD
END;


GRAPH SET PROCEDURE CONNECT-COMPONENTS(g);
POINT SET g;
BEGIN
  GRAPH SET components (INIT NULL);
  POINT SET unassigned (INIT g), c;
  WHILE unassigned # NULL
  DO c := COMPONENT(unassigned);
     components := components + c;
     unassigned := unassigned - c
  OD;
  CONNECT-COMPONENTS := components
END;


POINT SET PROCEDURE COMPONENT(g);
POINT SET g;
BEGIN
  SPECIFY finds one connected component
          of the context graph; depends
          on how the graph structure is
          in the implementation
END;
```

```
BOOLEAN PROCEDURE TRY-JOIN(k);
GRAPH SET k;
BEGIN
  SPECIFY p := a set of nodes which are linked to
          nodes in the components of k, and which
          join up the components; also
          dependent on implementation of the graph;
  ADD-TO-CONTEXT(p);
  SPECIFY |k| := number of connected components now
                 in context graph;
  TRY-JOIN := |k| = 1
END;
```

```
PROCEDURE GET-USER-MESSAGE(m);
MESSAGE m;
BEGIN
  ITEM LIST parts;
  parts := SIMULATE-USER-MESSAGE();
  requests OF m := NIL;
  reaction OF m := reaction OF CAR(parts);
  IF reaction OF m # STOP
  THEN INTERPRET-CHOICE(m, CDR(parts))
  FI;
  CATEGORIZE-DOCUMENT(reaction OF m)
END;
```

```
PROCEDURE CATEGORIZE-DOCUMENT(r);
INTEGER r;
BEGIN
  POINT d;
  d := doc-shown OF last-display;
  IF r # STOP AND d # UNPOINT
  THEN CASE r OF
         NO: inhib-nodes := inhib-nodes + d;
             cont-nodes := cont-nodes - d;
             good-docs := good-docs - d;
             acceptd-docs := acceptd-docs - d
             ENDCASE;
         NONE: IF d ∈ good-docs
               THEN acceptd-docs := acceptd-docs + d
               FI
               ENDCASE;
         YES: good-docs := good-docs + d;
              acceptd-docs := acceptd-docs - d
       ENDCASE
  FI
END;
```

```
PROCEDURE INTERPRET-CHOICE(m, parts);
MESSAGE m;
ITEM LIST parts;
BEGIN
  POINT SET r (INIT NULL), s (INIT NULL);
  POINT p;
  ITEM part;
  BOOLEAN negative;  INTEGER j;
  FOR EACH part IN parts
  DO IF item-type OF part = DNUM
     THEN j := display-no OF part;
          IF j < 0
          THEN j := -j;
               negative := TRUE
          ELSE negative := FALSE
          FI;
          p := (nodes-shown OF last-display)[j];
          IF negative THEN r:= r+p ELSE s:=s+p FI
     ELSE requests OF m := APPEND( requests OF m,
                                   CONS(node-num OF part, NIL)
                                 )
     FI
  OD;
  MAKE-SELECTION-LISTS(m, r, s)
END;


PROCEDURE MAKE-SELECTION-LISTS(m, r, s);
MESSAGE m; POINT SET r, s;
BEGIN
  exp-nodes := exp-nodes + s;
  IF s = NULL AND r = NULL AND reaction OF m = YES
  THEN s := nodes OF last-display
  FI;
  rejections OF m := IF r # NULL
                     THEN r
                     ELSE IF reaction OF m = NO
                     THEN nodes-shown OF last-display
                          - exp-nodes - sel-nodes
                     ELSE NULL
                     FI;
  selections OF m := IF s # NULL
                     THEN s + sel-nodes
                     ELSE IF reaction OF m = YES
                     THEN nodes-shown OF last-display - r
                     ELSE sel-nodes
                     FI;
  sel-nodes := s
END;


POINT SET PROCEDURE MATCHING-NODES(r);
QUERY r;
BEGIN
  COMMENT Since a QUERY is a POINT, ie, a node identification,
  a QUERY r is a POINT r;
  MATCHING-NODES := r
END;
```

```
ITEM LIST PROCEDURE SIMULATE-USER-MESSAGE();
SPECIFY exmines the document and term relevance
        decisions concerning the current query,
        in SEARCH search-details, in order to
        produce a simulated response to the
        DISPLAY last-display. The response is
        logged using LOG-RESPONSE().  The result
        is a list of items, which is either empty
        or starts with the reaction, and continues
        with display and node numbers;
COMMENT when developed to be used by real users
 this procedure will not be necessary;
 BEGIN
  ITEM LIST s; INTEGER r, i;
  IF last-display = NULL
  THEN s := CONS( (REACT, NONE),
                  CONS( (NNUM,
                        users-term OF search-details
                        ), NIL
                      )
                )
  ELSE IF good-docs = rel-docs OF search-details
  THEN s := CONS((REACT, STOP), NIL)
  ELSE r := IF doc-shown OF last-display = UNPOINT
            THEN NONE
            ELSE IF doc-shown OF last-display ∈
                    rel-docs OF search-details
            THEN YES
            ELSE NO
            FI;
       s := CONS((REACT, r), NIL);
       FOR i=1 TO |(nodes-shown OF last-display)|  // cardinality
       DO IF (nodes-shown OF last-display)[i] ∈
             (terms OF search-details - exp-nodes)
          THEN s := APPEND(s, CONS((NNUM, i), NIL))
          FI
       OD
  FI;
  LOG-RESPONSE(s);
  SIMULATE-USER-MESSAGE := s
END;


PROCEDURE RESPOND-TO-USER(m);
MESSAGE m;
BEGIN
  IF NOT stop-requested
  THEN IF cont-nodes = NULL
       THEN STIMULATE-USER()
       ELSE IF performance <= low
       THEN REVIEW-COURSE()
       ELSE IF reaction OF m = YES
            AND doc-shown OF last-display # UNPOINT
       THEN DISPLAY-RELATED(doc-shown OF last-display)
       ELSE PICK-A-DOCUMENT()
       FI
  FI
END;
```

```
PROCEDURE STIMULATE-USER();
BEGIN
  LOG-STIMULATION();
  IF performance <= low
  THEN REVIEW-COURSE()
  ELSE SUGGEST-SUBJECTS()
  FI
END;


PROCEDURE REVIEW-COURSE();
BEGIN
  POINT SET docs;
  LOG-REVIEW();
  docs := good-docs - redisp-nodes;
  IF docs # NULL
  THEN REDISPLAY(LEAST-INVOLVED(docs))
  ELSE docs := accptd-nodes - redisp-nodes;
       IF docs # NULL
       THEN REDISPLAY(MOST-INVOLVED(docs))
       ELSE SUGGEST-SUBJECTS()
       FI
  FI
END;


PROCEDURE SUGGEST-SUBJECTS();
BEGIN
  POINT SET nodes;
  LOG-SUGGEST();
  doc-shown OF last-display := UNPOINT;
  nodes := exp-nodes - redisp-nodes;
  IF nodes # NULL
  THEN DISPLAY-SUBJECTS(LEAST-INVOLVED(nodes))
  ELSE nodes-shown OF last-display := NIL
  FI
END;


PROCEDURE DISPLAY-RELATED(doc);
POINT doc;
BEGIN
  POINT SET rel, d;
  rel := LINKED-TO(doc) - inhib-nodes;
  d := UNSEEN-DOCUMENTS(rel);
  IF d # NULL
  THEN DISPLAY-DOCUMENT(any d item);
  ELSE PICK-A-DOCUMENT()
  FI
END;
```

```
PROCEDURE PICK-A-DOCUMENT();
BEGIN
  POINT SET docs;
  docs := UNSEEN-DOCUMENTS(cont-nodes);
  IF docs = NULL
  THEN SUGGEST-SUBJECTS()
  ELSE DISPLAY-DOCUMENT( IF unity
                         THEN MOST-INVOLVED(docs)
                         ELSE AVERAGE-INVOLVED(docs)
                         FI
                       )
  FI
END;
```

```
PROCEDURE REDISPLAY(doc);
POINT doc;
BEGIN
  DISPLAY-DOCUMENT(doc);
  redisp-nodes := redisp-nodes + doc
END;
```

```
POINT SET PROCEDURE UNSEEN-DOCUMENTS(nodes);
POINT SET nodes;
BEGIN
  POINT p;
  UNSEEN-DOCUMENTS:={ {p ∈ nodes : node-type OF p = DOC}
                      - good-docs - accptd-docs
                    }
END;
```

```
PROCEDURE DISPLAY-DOCUMENT(doc);
POINT doc;
BEGIN
  doc-shown OF last-display := doc;
  nodes-shown OF last-display := {p ∈ LINKED-TO(doc) :
                                   node-type OF p # DOC}
  LOG-DISPLAY();
  cont-nodes := cont-nodes + doc
END;
```

```
PROCEDURE DISPLAY-SUBJECT(centre);
POINT centre;
BEGIN
  nodes-shown OF last-display :=
   {p ∈ {centre + LINKED-TO(centre)}: node-type OF p=SUB};
  LOG-DISPLAY();
  redisp-nodes := redisp-nodes + centre
END;
```

```
POINT PROCEDURE MOST-INVOLVED(nodes);
POINT SET nodes;
BEGIN
  POINT p, q;
  REAL c, max-c (INIT minimum real number);
  FOR EACH p IN nodes
  DO c := CONNECT-COEFFICIENT(p);
     IF c > max-c THEN q:=p; max-c:=c FI
  OD;
  MOST-INVOLVED := q
END;

POINT PROCEDURE LEAST-INVOLVED(nodes);
POINT SET nodes;
BEGIN
  POINT p, q;
  REAL c, min-c (INIT maximum real number);
  FOR EACH p IN nodes
  DO c := CONNECT-COEFFICIENT(p);
     IF c < min-c THEN q:=p; min-c:=c FI
  OD;
  LEAST-INVOLVED := q
END;

POINT PROCEDURE AVERAGE-INVOLVED(nodes);
POINT SET nodes;
BEGIN
  INTEGER n (INIT |nodes|);
  POINT ARRAY nn[1:n];  POINT p;
  REAL ARRAY c[1:n];
  INTEGER i (INIT 0), k;
  REAL d, mean (INIT 0), min-d (INIT max real no);
  FOR EACH p IN nodes
  DO i:=i+1; nn[i]:=p;
     c[i] := CONNECT-COEFFICIENT(p);
     mean := mean + c[i]
  OD;
  mean := mean / n;
  FOR i=1 TO n WHILE min-d # 0
  DO  d := ABS(mean - c[i]);
      IF d < min-d THEN k:=i; min-d:=d FI
  OD;
  AVERAGE-INVOLVED := nn[k]
END;

REAL PROCEDURE CONNECT-COEFFICIENT(node);
POINT node;
BEGIN
  EDGE e;
  INTEGER l (INIT 0), g (INIT 0);
  FOR EACH e IN EDGES-FROM(node)
  DO g := g + strength OF e;
     IF target OF e ∈ cont-nodes
     THEN l := l + strength OF e
     FI
  OD;
  CONNECT-COEFFICIENT := l/g
END;
```

```
POINT SET PROCEDURE LINKED-TO(n);
POINT n;
BEGIN
  EDGE e;
  POINT SET l (INIT NULL);
  FOR EACH e IN EDGES-FROM(n)
  DO l := l + target OF e OD;
  LINKED-TO := l
END;



PROCEDURE EDGES-FROM(node);
POINT node;
BEGIN
  INTEGER posn;
  posn := FIND-NODE-POSN(node);
  SELECT-CLUSTER(node);
  PICK-EDGES-FROM(posn, node)
END;



PROCEDURE SELECT-CLUSTER(node);
POINT node;
BEGIN
  INTEGER no;
  no := FIND-CLUSTER-NO(node);
  IF NOT IN-PAGE-TABLE(no)
  THEN FIND-LEAST-REC-USED-SPACE();
       GET-CLUSTER(no)
  FI
END;



INTEGER PROCEDURE FIND-CLUSTER-NO(node);
POINT node;
BEGIN
  SPECIFY This procedure finds and passes as result
  the number of the cluster to which 'node' belongs.
  Depends on implementation;
  FIND-CLUSTER-NO:= the number found
END;



INTEGER PROCEDURE FIND-NODE-POSN(node);
POINT node;
BEGIN
  SPECIFY Finds and passes as result the address of
  'node' and its associated items.
  FIND-NODE-POSN:= the address
END;
```

```
BOOLEAN PROCEDURE IN-PAGE-TABLE(num);
INTEGER num;
BEGIN
  SPECIFY Checks from a page table to determine
  whether 'num' has been recorded in it, signifying that the
  cluster represented by 'num' is already in use;
  IN-PAGE-TABLE := IF num is in page table THEN TRUE ELSE FALSE FI
END;


PROCEDURE FIND-LEAST-REC-USED-SPACE();
BEGIN
  SPECIFY Checks if available core space is exhausted;
  If yes, determines which page may be overwritten
  by the in-coming one
END;


PROCEDURE GET-CLUSTER(num);
BEGIN
  SPECIFY Get the cluster numbered 'num' probably
  via direct access techniques
END;


PROCEDURE PICK-EDGES-FROM(posn, node);
INTEGER posn;
POINT node;
BEGIN
  SPECIFY Using the address 'posn' obtain all
  the edges associated with 'node'
END;


PROCEDURE SET-UP-MODEL();
BEGIN
  SEARCH-INITIALIZATION();
  cont-nodess, inhib-nodes, exp-nodes, sel-nodes:= NULL;
  redis-nodes, good-docs, accptd-docs:= NULL;
  doc-shown OF last-display := UNPOINT;
  nodes-shown OF last-display := NULL;
  stop-requested:= FALSE;
  performance:= 1;
  search-details:= GET-SEARCH();
  LOG-SEARCH(search-details);
END;


PROCEDURE SEARCH-INITIALIZATION()
BEGIN
  SPECIFY This procedure does search-level initialisations
  other than those in SET-UP-MODEL(); as dictated by the
  implementation
END;
```

```
SEARCH PROCEDURE GET-SEARCH();
BEGIN
  SEARCH s;
  SPECIFY s:= read in search details;
  IF indid-views
  THEN GET-VIEW(user-id OF search-details)
  FI;
  GET-SEARCH:=s
END;

PROCEDURE GET-VIEW(user);
INTEGER user;
BEGIN
  SPECIFY This routine makes available user's view of the
  graph structure
END;

PROCEDURE SET-SYSTEM-PARAMETERS();
BEGIN
  SPECIFY The variables indid-views, search-limit,small,
  memory, low, score, stuck-score are given values which
  are held constant throughout the run.
END;

PROCEDURE OPEN-FILES();
BEGIN
  SPECIFY This procedure makes the following files ready:
  supergraph file, user-views file,
  search details file and the log file
END;

PROCEDURE CLEAR-SEARCH();
BEGIN
  SPECIFY This procedure does any tidying-up between searches
  as dictated by the implementation
END;

COMMENT All output is sent to a log file through the
following procedures:
  INITIALIZE-LOG();
  LOG-SEARCH()          start of new search;
  LOG-DISPLAY()         contents of last-display;
  LOG-REVIEW()          note review and performance;
  LOG-STIMULATION()     note that user needs stimulating;
  LOG-SUGGEST()         note subjects are being suggested;
  LOG-COMPONENTS(n)     note no of components in context graph;
  LOG-RESPONSE(i)       note user's input, i;
  LOG-SEARCH-END();
  EVALUATE-LOG();


PROCEDURE CLOSE-DOWN();
BEGIN
  LOG-SEARCH-END();
  CLEAR-SEARCH();
  had-enough :=   // no more search
END
```

REFERENCES

The following abbreviations have been used for some of the journals:

ARIST	Annual Review of Information Science and Technology
I.J.M-M.S.	International Journal of Man-Machine Studies
I.P.M.	Information Processing & Management
I.S.R.	Information Storage & Retrieval
JACM	Journal of the Association for Computing Machinery
JASIS	Journal of the American Society for Information Science
J.Doc	Journal of Documentation

- - -

Ambrozy D
On Man-Computer Dialogues.
I.J.M-M.S., Vol.3(4) pp 375-383, 1971

Angione P V
On the equivalence of Boolean and Weighted Searching Based on the Convertibility of Query Forms.
JASIS, Vol.26(2) pp 112-124, 1975

Apted S M
General Purposive Browsing.
Library Association Record, Vol.73(12) pp 228-230, 1971

Ashworth W
The Information Explosion.
Library Association Record, Vol.76, pp 63-68,71, 1974

Augustson J G, Minker J
An analysis of some graph-theoretic cluster techniques.
JACM, Vol.17, pp 571-588, 1970

Bar Hillel Y
The Impact of the essentially pragmatic character of the natural languages on linguistic information processing.
In: Debons A, Cameron W J, (eds.), "Perspectives in Information Science." (NATO Advanced Study Institute Proceedings, Aberyswyth, Wales 1973), Noordhoff, The Hague, pp 297-305, 1975

Barber A S, Barraclough E D, Gray W A
On-line Information Retrieval as a Scientist's Tool.
I.S.R., Vol.9(8), pp 429-440, 1973

Barmack J E, Sinaiko H W
Human Factors in Computer-generated Graphic Displays.
Institute for Defence Analysis Study 234 (AD 636 170), 1966
Arlington, Va.

Barraclough E
On-line Searching in Information Retrieval.
J.Doc, Vol.33(3), pp 220-238, 1977

Becker D S, Pyrce S R
Enhancing the Retrieval Effectiveness of Large Bibliographic files
ITT Research Institute, Chicago, Illi., 1977

Belkin N J
The problem of 'matching' in Information Retrieval.
In: "Theory and applications of Information Research"
Harbo O, Kajberg L, (eds.), Mansell, London, pp 187-197, 1980

Belkin N J, Brooks H M, Oddy R N
Representation and classification of anomalous states of knowledge and information for use in interactive retrieval.
Proceedings, 3rd International Research Forum in Information Science, Oslo, 1979

Belkin N J, Oddy R N
Document Retrieval Based on the automatic determination of the user's information need.
Journal of Informatics, 2(1), pp 8-12, 1978

Belkin N J, Robertson S E
Information Science and the Phenomenon of Information.
JASIS, Vol.27(4), pp 197-204, 1976

Belzer J
Information Theory as a measure of Information Content.
JASIS, Vol.24, pp 300-304, 1973

Bennett J L
Expanded Roles for Information Transfer Specialists in Interactive Information Management.
In: Fry B et al. "Information Management in the 1980s", Procs. of the 40th Annual Meeting of the ASIS, Vol 14, Part 2, 1977

Berlyne D E
Structure and Direction in Thinking.
Wiley, New York, 1965

Bookstein A
On the perils of merging Boolean and Weighted Retrieval Systems.
JASIS, Vol.29(3), pp 156-158, 1978

Bookstein A
Relevance.
JASIS, Vol.30(5), pp 269-273, 1979

Bookstein A, Cooper W S
A General Mathematical Model for Information Retrieval Systems.
Library Quarterly, Vol.46(2), pp 153-167, 1976

Bookstein A, Swanson D R
A Decision-Theoretic Foundation for Indexing.
JASIS, Vol.26(1), pp 45-50, 1975

Boon J A
User evaluation of Information Retrieval Systems;
A methodological approach.
PhD Thesis, Rand Afrikaans University, South Africa, 1978

Boulding K E
The Image.
University of Michigan Press, Ann Arbor, Mich. 1956

Brackman R J
What's in a concept: structural foundations for semantic networks.
I.J.M-M.S., 9(2), pp 127-152, 1977

Brittain J M
Information and its users.
Bath University Press, 1970

Bruner J S, Goodnow J, Austin G A
A Study of Thinking.
Wiley, New York, 1956

Cagan C
A Highly Associative Document Retrieval System.
JASIS, Vol.21, pp 330-337, 1970

Chapanis A
Interactive Human Communication.
Scientific American, Vol.232(3), pp 36-42, 1975

Cleverdon C W
On the inverse relationship of recall and precision.
J.Doc, Vol.28(3), pp 195-201, 1972

Cleverdon C W
User Evaluation of Information Retrieval Systems.
J.Doc, Vol.30(2), pp 170-180, 1974

Cleverdon C W, Kidd J S
Redundancy, Relevance and Value to the user in the outputs of Information Retrieval Systems.
J.Doc, Vol.32(3), pp 159-173, 1976

Colby K M
Computer Simulation of change in personal belief systems.
Behavioural Science, Vol.12(3), pp 248-253, 1967

Colby K M, Enea H
Heuristic Methods for Computer Understanding of Natural Language in Context Restricted On-line Dialogues.
Mathematical Biosciences, Vol.1(1), pp 1-25, 1967

Cooper W S
A definition of relevance for information retrieval.
I.S.R., Vol.7, pp 19-37, 1971

Cooper W S
On selecting a measure of retrieval effectiveness.
JASIS, Vol.24(2), pp 87-100, 1973

Cooper W S
Indexing Documents by Gedanken Experimentation.
JASIS, Vol.29(3), pp 107-119, 1978

Cooper W S
The 'Why Bother?' theory of information usage.
Journal of Informatics, 2(1), pp 2-5, 1978b

Cravens D W
An exploratory analysis of individual information processing.
Management Science, Vol.16(10), 1970

Croft W B
Clustering Large Files of Documents using the Single-link method.
JASIS, Vol.28(6), pp 341-344, 1977

Croft W B
Organizing and Searching Large Files of Document Descriptions.
PhD Thesis, University of Cambridge, England, 1978

Croft W B
An overview of Information Systems.
Information Technology: Research and Development,
Vol.1(1), pp73-96, 1982

Crouch D B
A file organisation and maintenance procedure for dynamic
document collections.
I.P.M. Vol.11, pp 11-21, 1975

Cuff R N
On casual users.
I.J.M-M.S., Vol.12(2), pp 163-188, 1980

Dattola R T
A fast algorithm for automatic classification.
Journal of Library Automation, Vol.2, pp 31-48, 1969

Davidson D
The effect of individual differences of cognitive style on
judgements of document relevance.
JASIS, Vol.28(5), pp 273-284, 1977

De Greene K B
Man-Computer Interrelationships.
In: "Systems Psychology", De Greene K B,(ed). McGraw Hill,
pp 281-336, 1970

De May M
The Relevance of the cognitive paradigm for Information Science.
In: Theory and Application of Information Research.
Harbo O, Kajberg L (eds), Mansell, London 1980, pp 48-61

De May M, Pinxten R, Poriau M, Vandamme F
International Workshop on the Cognitive Viewpoint, CC77, Ghent Belgium, University of Ghent, 1977

Dillon M, Desper J
The use of automatic relevance feedback in Boolean retrieval systems.
J.Doc, Vol.36(3), pp 197-208, 1980

Driver M J, Streufort S
Integrative complexity: An approach to individuals and groups as information processing systems.
Administrative Science Quarterly, Vol.14, pp 272-285, 1969

Eitzweiler L, Martin C
Binary Cluster Division and its application to a modified single-pass clustering algorithm.
Report no. ISR-21 to the National Library of Medicine, 1972

Ernst G W, Newell A
GPS: A case study in generality and problem-solving.
Academic Press, New York, 1969

Farradane J
The evaluation of Information Retrieval Systems.
J.Doc, Vol.30(2), pp 195-209, 1974

Feigenbaum E A
Information processing and memory.
In: Norman D A, (ed.), "Models of human memory"
Academic Press, New York, pp 451-469, 1970

Feigenbaum E A, Feldman J, (eds.)
Computers and Thought.
McGraw Hill, New York, 1963

Findler N V (ed)
Associative Networks: Representation and use of knowledge by Computers.
Academic Press, New York, 1979

Fitter M
Towards more 'natural' interactive systems.
I.J.M-M.S., Vol.11(3), pp 339-350, 1979

Foley J M
Communication aspects of Information Science.
Theory and Practice, Vol.12(3), pp 167-172, 1973

Foskett D
A note on the concept of 'relevance'.
I.S.R., Vol.8, pp 77-78, 1972

Gaines B R, Facey P V
Some experience in interactive system development and application.
Proceedings, IEEE, Vol.63, pp 894-911, 1975

Greene R J
The effectiveness of browsing.
College & Research Libraries, Vol. 38(4), pp 313-316, 1977

Greeno J G
The structure of memory and the process of solving problems.
In: R L Solso (ed), "Contemporary issues in cognitive psychology; The Loyola Symposium." Washington D C Winston, 1973, pp 103-134

Guazzo M
Retrieval Performance and Information Theory.
I.P.M., Vol.13, pp 155-165, 1977

Hall J L
On-line information retrieval sourcebook.
Aslib, London, 1977

Harbo O, Ingwersen F, Timmermann P
Cognitive processes in information storage and retrieval.
In: "International Workshop on the cognitive viewpoint."
De May et al (eds), University of Ghent, 1977

Harbo O, Kajberg L (eds)
Theory and Application of Information Research.
Mansell, London, 1980

Harding A F, Willet P
Indexing exhaustivity and the computation of similarity matrices.
JASIS, Vol.31(4), pp 298-300, 1980

Harper D J, Van Rijsbergen C J
An evaluation of feedback in document retrieval using co-occurrence data.
J.Doc, Vol.34(3), pp 189-216, 1978

Harter S
Statistical approaches to automatic indexing.
Drexel Library Quarterly, Vol.14(2), pp 57-74, 1978

Hayes J E, Michie D, Mikulich L I (eds)
Machine Intelligence.
Ellis Horwood Ltd., Publishers, Chichester

Heine M H
The 'question' as a fundamental variable in information science.
In: "Theory and applications of information research."
Harbo O, Kajberg L, (eds), Mansell, London, pp 137-145 1980

Hollnagel E
The relationship between intention, meaning and action.
Proceedings of Informatics 5, Aslib, pp 135-147, 1979

Holmes P L
On-line information retrieval - An introduction and guide to the British Library's short-term experimental information network project.
Vol.1 : Experimental use of non-medical information services. British Library Report BLRD 5360, 1976

Horman A M
A man-machine synergistic approach to planning and creative problem-solving. Part 1.
I.J.M-M.S., Vol.3, pp 167-184 1971

Houghton B, Convey J
On-line information retrieval systems.
Clive Bingley Ltd., London, 1977

Huyck P H
Computer Aided Instruction techniques for information retrieval.
Datamation, February 1973, pp 91-92

Hyman R J
Access to library collections: summary of documentary and opinion survey on the direct shelf approach and browsing.
Library Resources & Technical Services 15(4), 479-491, 1971

Ide E
New Experiments in Relevance Feedback.
In: "The SMART Retrieval System.", G. Salton (ed), pp 337-354
Prentice Hall, 1971

Ingwersen P, Johansen P, Timmerman P
User-librarian negotiations and search procedures: a progress report.
In: "Theory and application of information research"
Harbo O, Kajberg L, (eds), Mansell, London 1980

Ingwersen P, Kaae S
User-librarian negotiations and information search procedures in public libraries; Analysis of verbal protocols.
Proceedings of the 3rd International Research Forum in Information Science, Oslo, pp 71-106, 1979

Jaffe J, Feldestein S
Rhythms of Dialogue.
New York, Academic Press, 1970

Jahoda G
Reference question analysis and search strategy development by man and machine.
JASIS, Vol.25, pp 139-144, 1974

Jamieson S H
Personal Communication.
Winter 1980

Jamieson S H
Automated Document Retrieval Based on Distributed Processing.
PhD Thesis, The University of Aston in Birmingham, 1981

Jardine N, Sibson R
Mathematical Taxonomy.
Wiley, London, 1971

Keen E M
Evaluation Parameters.
In: "The SMART Retrieval System", G Salton (ed), 74-111
Prentice Hall 1971

Keen E M, Digger J A
Report of an information science index languages test.
Aberyswyth College of Librarianship, Wales, 1972

Kemp D
Relevance, pertinence and information system development.
I.S.R., Vol.10, pp 37-47, 1974

Kim C, Kim S D
Consensus vs frequency: an empirical investigation of the theories for identifying descriptors in designing thesauri.
I.P.M. 13, pp 253 - 258, 1977

Kiss G
An associative thesarus of English: Structural analysis of a large relevance network.
In: Kennedy A, Wilkes A, (eds), "Long-term memory"
Academic Press, London, pp 103-121, 1975

Koll M B
Information retrieval theory and design based on a model of the user's concept relations.
In: "Information Retrieval Research", Oddy R N et al (eds),
Butterworths, pp 77-93, 1981

Lancaster F W
MEDLARS: Report on the evaluation of its operating efficiency.
American Documentation, 20(2), pp 119-142, 1969

Lancaster F W
Aftermath of an evaluation.
J.Doc, 27(1), pp 1-10, 1971

Lancaster F W
Information Retrieval Systems.
Wiley Interscience, New York, 1979

Lancaster F W, Fayen E G
Information Retrieval On-line.
Wiley Interscience, Los Angeles, 1973

Lancaster F W, Mills J
Testing indexes and index language devices: the Aslib Cranfield Project.
American Documentation, 15(1), p4, 1964

Lesk M E, Salton G
Interactive search and retrieval methods using automatic information displays.
Proceedings, AFIPS Spring Joint Computer Conference, AFIPS Press Montvale, N.Y. pp 435-446, 1969

Levine E H
Effect of instantaneous retrieval on indexing criteria.
JASIS, 25(3), pp 199-200, 1974

Licklider J C R
Man-Computer Symbiosis.
IRE Transactions on Human Factors in Electronics, HFE-1, pp 4-11, 1960

Lindsay P H, Norman D A
Human Information Processing : an introduction to psychology.
New York, Academic Press, 1972

Lynch M J
Reference interviews in public libraries.
Library Quarterly, 48, pp 119-142, 1978

Mackay D M
What makes a question.
The Listener, 63, pp 789-790, 1960

Macleod I A
Towards an information retrieval language based on a relational view of data.
I.P.M., 13, pp 167-175, 1977

Maldé B
Further Research on Double-Interaction: the simultaneous conduct of man-man and man-computer interactions.
PhD Thesis, Loughborough University, England, 1978

Malhotra A, Thomas J C, Carroll J M, Miller L A
Cognitive processes in design.
I.J.M-M.S., 12, pp 119-140, 1980

Marcus R S
The user interface for the INTREX retrieval system.
MIT, Cambridge, Ma, 1971

Maron M E
On indexing, retrieval and the meaning of about.
JASIS, 28(1), pp 38-43, 1977

Maron M E
Depth of indexing.
JASIS, 30(4), pp 224-228, 1979

Maron B, Fife D
On-line Systems - Techniques and Services.
In: ARIST, 11, pp 163-210, 1976

Martin J
Design of Man-Computer Dialogues.
Englewood Cliffs, N.J., Prentice Hall, 1973

Martin T H, Parker E B
Designing for user acceptance of an interactive bibliographic search facility.
In: "Interactive Bibliographic Search: The User/Computer Interface", Walker D E, Montvale N J, (eds),
AFIPS Press, 1971

Mayer R E
Thinking and problem-solving.
Scott Foresman and Co., Glenview Illi. 1977

McGuire M T, Stanley J
Dyadic Communication, Verbal Behaviour, Thinking and Understanding.
Journal of Nervous and Mental Diseases, 152(4), 242-259
1972

Meadow C T
Man-machine communication.
Wiley Interscience, New York, 1970

Melnyk V
Man-machine interface: - Frustration.
JASIS, 23(6), pp 392-401, 1972

Michie D
Expert Systems in the Micro Electronic Age.
Edinburgh University Press, 1979

Miller W L
Probabilistic Search Strategy for MEDLARS.
J.Doc, 27, pp 254-266, 1971

Miller G, Gallanter E, Pribram K
Plans and the structure of behaviour.
New York, Holt, Rinehart and Winston Inc. 1960

Minker J
Information storage and retrieval: a survey and functional description.
SIGIR Forum, 12(2), xii-xv, 1-108, 1977

Mitchell P C, Rickman J T, Walden W E
SOLAR: a storage and on-line automatic retrieval system.
JASIS, 24(5), pp 347-358, 1973

Moghdam D
User training for on-line information retrieval systems.
JASIS, 26(3), pp 184-188, 1975

Mooers C N
Mooers' Law or Why some retrieval systems are used and others are not.
American Documentation, 11(3), ii, 1960

Mulvihill J, Brenner E H
Ranking Boolean search output.
American Documentation, 19(2), pp 204-205, 1968

Mushens B
The problem of test collection size.
Paper presented at the 3rd Research colloquium of the BCS IR Specialist Group, March 1981, Birmingham
pp 73-90

Newell A, Simon H A
Human problem-solving.
Englewood Cliffs, New Jersey, Prentice Hall, 1972

Nickerson R S
Man-Computer Interaction: a challenge for Human Factors research.
Ergonomics, 12(4), pp 501-517, 1969

Noreault T, Koll M, McGill M J
Automatic ranked output from Boolean searches in SIRE.
JASIS, 28(6), pp 333-339, 1977

Oddy R N
Reference retrieval based on user induced dynamic clustering.
PhD Thesis, University of Newcastle-upon-Tyne, U K
1974

Oddy R N
Information retrieval through man-machine dialogues.
J.Doc, 33(1) pp 1-14, 1977a

Oddy R N
Retrieving references by dialogue rather than by query formulation.
Journal of Informatics, 1(1), pp 37-55, 1977b

Oddy R N
Towards research on questions in information science.
University of Aston in Birmingham, Computer Centre, TR80001, 1980

Oddy R N
Laboratory tests: Automatic Systems.
Chapter 9 in: Information Retrieval Experiment
Sparck Jones K (ed), Butterworths, 1981

Ofori-Dwumfuo G O
Reference retrieval without user query formulation.
Journal of Information Science, Vol 4, pp 105-110, 1982

Olney J C
Building a concept network for retrieving information from large libraries. Part 1.
SDC Report TM-634/001/00 Jan. 1962

Palay A J, Fox M S
Browsing through databases.
In:"Information Retrieval Research"
Oddy R N et al (eds), Butterworths 1981, pp 310-324

Pearce D M S, Easterby R S
The evaluation of user interaction with computer-based Management Information Systems.
Human Factors, 15(2), pp 163-177, 1973

Pejtersen A M
Investigation of search strategies in fiction based on an analysis of 134 user-librarian conversations.
Proceedings, 3rd International Research Forum in Information Science, Oslo, pp 107-131, 1979

Penniman W D
Rhythms of dialogue in human computer conversation.
PhD Thesis, The Ohio State University, 1975

Polya G
How to solve it.
Garden City, New York, Doubleday Anchor, 1957

Preece S E
Clustering as an output option.
In: Waldron H J et al, (eds), "Innovative Developments in information systems: the benefits and costs"
Proceedings, 36th Annual meeting of the ASIS, Vol.10
pp 189-190, Los Angeles, 1973

Preese P F W
Mapping cognitive structures: a comparison of methods.
Journal of Educational Psychology, 68, 1-8, 1976

Press L
Towards balanced man-machine systems.
I.J.M-M.S., 3(1), pp 61-73, 1971

Pulfer J K
Man-machine interaction in creative applications.
I.J.M-M.S., 3(1), pp 1-11, 1971

Quillian M R
Semantic Memory.
In: "Semantic Information Processing"
Minsky M (ed), MIT Press, pp 227-270, 1968

Raitt D I
Interactive information systems and the user interface.
FLA Thesis, Library Association, 1978

Restle F, Davis J H
Success and speed of problem-solving by individuals and groups.
Psychological Review, 69, pp 520-536, 1962

Rickman J T
Design considerations for a Boolean search system with automatic relevance feedback processing.
Proc., ACM Annual Conference, Boston, pp 478-481, 1972

Robertson S E
The parametric description of retrieval tests.
Part 1: The Basic Parameters. J.Doc, 25, pp 1-17, 1969
Part 2: Overall Measures. J.Doc, 25, pp 93-107, 1969

Robertson S E
Specificity and weighted retrieval.
J.Doc, 30(1), pp 41-46, 1974

Robertson S E
A Theoretical Model of the Retrieval Characteristics of Information Retrieval Systems.
PhD Thesis, University of London, 1976

Robertson S E
Theories and Models in Information Retrieval.
J.Doc, 32(2), pp 126-148, 1977

Robertson S E
The Probability Ranking Principle in IR.
J.Doc, 33(4), pp 294-304, 1977p

Robertson S E
The probabilistic character of relevance.
I.P.M., 13, pp 247-251, 1977r

Robertson S E
The role of theory in the testing of IR systems.
In: Informatics 3, Jones K P, Horsnell V (eds),
Aslib, London, 1978, pp 114-123

Robertson S E
Relevance, Retrieval and Document Spaces.
Proceedings, International Research Forum on Information Science 3, 1979, Oslo

Robertson S E, Belkin N J
Ranking in principle.
J.Doc, 34(2), pp 93-100, 1978

Robertson S E, Belkin N J
Personal Communication.
Winter 1982

Robertson S E, Sparck Jones K
Relevance weighting of search terms.
JASIS, 27(3), pp 129-146, 1976

Rocchio J J Jr
Document retrieval systems - Optimization and Evaluation.
PhD Thesis, Harvard University, 1966

Rocchio J J Jr
Evaluation viewpoints in document retrieval.
In: "The SMART Retrieval System", Salton G (ed), Prentice Hall, 1971

Rouse W B
Human-computer interaction in the control of dynamic systems.
Computing Surveys, 13(1), pp 71-99, 1981

Roy G G
A man-machine approach to multivariate decision making.
I.J.M-M.S., 12, pp 203-215, 1980

Sackman H
Experimental analysis of man-computer problem-solving.
Human Factors, 12(2), pp 187-201, 1970

Salton G
A new comparison between conventional indexing (MEDLARS) and automatic text processing (SMART).
JASIS, 23, pp 75-84, 1972

Salton G
Recent studies in automatic text analysis and document retrieval.
JACM, 20(2), pp 258-278, 1973

Salton G
Dynamic Information and Library Processing.
Englewood Cliffs, New Jersey, Prentice Hall, 1975

Salton G, Yang C S
On the specification of term values in automatic indexing.
J.Doc, 29(4), pp 351-372, 1973

Sandoval A M
The vehicles of the results of Latin American Research: A Bibliometric Approach.
Paper presented at the 38th World Congress of FID, Mexico City 1976

Saracevic T
The concept of 'relevance' in information science: a historical review.
In: Saracevic T, (ed), "Introduction to Information Science" New York: Bowke & Co. 1970

Saracevic T
Relevance: a review of, and a framework for the thinking on the notion in information science.
JASIS, 26(6), pp 321-343, 1975

Saracevic T
Problems of question analysis in information retrieval.
Proceedings of ASIS, 15, pp 281-283, 1978

Scheflen A E
Human Communication: Behavioural programs and their integration in interaction.
Behavioral Science, 13, pp 44-55, 1968

Shergold J
Why on-line computing.
Data Processing, Nov-Dec 1972, pp 404-409

Siegel S
Non-parametric statistics for the behavioral sciences.
McGraw Hill, Tokyo, 1956

Siegman A W, Pope B (eds)
Studies in Dyadic Communication.
New York, Pergamon Press, 1972

Simon H A, Newell A
Human Problem Solving: The State of Theory in 1970.
In: "Readings in Management Information Systems"
Davis G B, Everest G C (eds), McGraw Hill, New York, pp 38-50 1976

Slack W V, Slack C W
Patient-computer dialogue.
The New England Journal of Medicine, 286(24), pp 1364-9, 1972

Smith L C
Selected Artificial Intelligence Techniques in Information Retrieval Systems Research.
PhD Thesis, Syracuse University, 1979

Sparck Jones K
A statistical interpretation of term specificity and its application in retrieval.
J.Doc, 28, pp 11-21, 1972

Sparck Jones K
Index Term Weighting.
I.S.R., 9, pp 619-633, 1973

Sparck Jones K
Does indexing exhaustivity matter.
JASIS, 24(5), pp 313-316, 1973i

Sparck Jones K
Automatic Indexing.
J.Doc, 30(4), pp 393-432, 1974

Sparck Jones K
A performance yardstick for test collections.
J.Doc, 31(4), pp 266-272, 1975

Sparck Jones K
Search Term Relevance Weighting given little relevance information.
J.Doc, 35(1), pp 30-48, 1979

Sparck Jones K (ed)
Information Retrieval Experiment.
Butterworths, London 1981

Sparck Jones K, Van Rijsbergen C J
Information retrieval test collections.
J.Doc, 32(1), pp 59-75, 1976

Sparck Jones K, Webster C A
Research on Relevance Weighting 1976-1979.
British Library R&D Report 5553, 1980

Svenonius E
An experiment in index term frequency.
JASIS, 23(2), pp 109-121, 1972

Swanson D R
Information Retrieval as a trial-and-error process.
Library Quarterly, 47(1), pp 128-148, 1977

Swanson R W
Design and evaluation of Information Systems.
ARIST, 10, pp 43-101, 1975

Taube M
Storage and retrieval of information by means of the association of ideas.
American Documentation, 6, pp 1-17, 1955

Taylor R S
Question-negotiation and information-seeking in libraries.
College and Research Libraries, 29, pp 178-194, 1968

Thomas J C Jr
An analysis of behaviour in the horbits-orcs problem.
Cognitive Psychology, 6, pp 257-269, 1974

Ting T, Badre A N
A dynamic model of man-machine interactions: design and application with an audiographic learning facility.
I.J.M-M.S., 8(1), pp 75-88, 1976

Van Rijsbergen C J
A theoretical basis for the use of co-occurrence data in information retrieval.
J.Doc, 33(2), pp 106-119, 1977

Van Rijsbergen C J
Automatic classification in information retrieval.
Drexel Library Quarterly, 14(2), pp 75-89, 1978

Van Rijsbergen C J
Information Retrieval.
Butterworths, London 1979

Van Rijsbergen C J, Croft W B
Document Clustering: an evaluation of some experiments with the Cranfield 1400 collection.
I.P.M., 11, pp 171-182, 1975

Vaughan W S Jr, Mavor A S
Behavioural characteristics of man in the performance of some decision-making task components.
Ergonomics, 15(3), pp 267-277, 1972

Vernimb C
Automatic query adjustment in document retrieval.
I.P.M., 13, pp 339-353, 1977

Wanger J, Cuadra C A, Fishburn M
Impact of on-line retrieval services: a survey of users 1974-1975.
Santa Monica, Ca., SDC, 1976

Wickelgren W A
How to solve problems: elements of a theory of problems and problem-solving.
San Francisco, Freeman, 1974

Williams M E, Preece S E
Database selection for network use: a feasibility study.
Proceedings, ASIS Annual Meeting, Vol 14, pp 275, 1977

Williamson R E
Real-time document retrieval.
PhD Thesis, Cornell University, 1974

Wilson P
Situational relevance.
I.S.R., 9, pp 457-471, 1973

Winograd T A
A program for understanding natural language.
Cognitive Psychology, 3, pp 1-192, 1972

Yu C T, Raghavan V V
Single-pass method for determining the semantic relationship between terms.
JASIS, 28(6), pp 345-354, 1977

Yu C T, Salton G
Precision weighting - an effective automatic indexing method
JACM, 23, pp 76-88, 1976

Zobrist A L, Carlson F R
An advice-taking chess computer.
Scientific American, 228, pp 92-105, 1973