Interactive Query Expansion
and
Relevance Feedback
for
Document Retrieval Systems

Efthimis Nikolaos Efthimiadis

Thesis submitted for the degree of Doctor of Philosophy

Department of Information Science
The City University
London

July 1992

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank all those who have contributed to this work and helped make it possible. In particular I would like to thank:

Steve Robertson, whose ideas initiated this research, for his guidance, support, advice and patient supervision that saw me through this thesis.

The academic, administrative and research staff of the Department of Information Science.

My colleagues, at the Graduate School of Library and Information Science at the University of California, Los Angeles, for their encouragement to complete this work.

Data-Star, ESA/IRS and INSPEC for the free online time. P.G. Marchetti, Roy Kitley and Mike Everest of ESA were particularly helpful.

All the friends who variously discussed and commented.

Jean and Stathis for their support and more.

My parents, who have never failed to encourage, support and express enthusiasm for all my work,

..., and Jagoda, for her encouragement, support, patience and understanding.

Declaration of Copyright

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

This thesis is aimed at investigating interactive query expansion within the context of a relevance feedback system that uses term weighting and ranking in searching online databases that are available through online vendors. Previous evaluations of relevance feedback systems have been made in laboratory conditions and not in a real operational environment.

The research presented in this thesis followed the idea of testing probabilistic retrieval techniques in an operational environment. The overall aim of this research was to investigate the process of interactive query expansion (IQE) from various points of view including effectiveness.

The INSPEC database, on both Data-Star and ESA-IRS, was searched online using CIRT, a front-end system that allows probabilistic term weighting, ranking and relevance feedback.

The thesis is divided into three parts.

Part I of the thesis covers background information and appropriate literature reviews with special emphasis on the relevance weighting theory (Binary Independence Model), the approaches to automatic and semi-automatic query expansion, the ZOOM facility of ESA/IRS and the CIRT front-end.

Part II is comprised of three Pilot case studies. It introduces the idea of interactive query expansion and places it within the context of the weighted environment of CIRT. Each Pilot study looked at different aspects of the query expansion process by using a front-end. The Pilot studies were used to answer methodological questions and also research questions about the query expansion terms. The knowledge and experience that was gained from the Pilots was then applied to the methodology of the study proper (Part III).

Part III discusses the Experiment and the evaluation of the six ranking algorithms. The Experiment was conducted under real operational conditions using a real system, real requests, and real interaction. Emphasis was placed on the characteristics of the interaction, especially on the selection of terms for query expansion.

Data were collected from 25 searches. The data collection mechanisms included questionnaires, transaction logs, and relevance evaluations.

The results of the Experiment are presented according to their treatment of query expansion as main results and other findings in Chapter 10. The main results discuss issues that relate directly to query expansion, retrieval effectiveness, the correspondence of the online-to-offline relevance judgements, and the performance of the $w(p-q)$ ranking algorithm.

Finally, a comparative evaluation of six ranking algorithms was performed. The yardstick for the evaluation was provided by the user relevance judgements on the lists of the candidate terms for query expansion. The evaluation focused on whether there are any similarities in the performance of the algorithms and how those algorithms with similar performance treat terms.

This abstract refers only to the main conclusions drawn from the results of the Experiment:

(1) One third of the terms presented in the list of candidate terms was on average identified by the users as potentially useful for query expansion;

(2) These terms were mainly judged as either variant expression (synonyms) or alternative (related) terms to the initial query terms. However, a substantial portion of the selected terms were identified as representing new ideas.

(3) The relationship of the 5 best terms chosen by the users for query expansion to the initial query terms was: (a) 34% have no relationship or other type of correspondence with a query term;

(b) 66% of the query expansion terms have a relationship which makes the term: (b1) narrower term (70%), (b2) broader term (5%), (b3) related term (25%).

(4) The results provide some evidence for the effectiveness of interactive query expansion. The initial search produced on average 3 highly relevant documents at a precision of 34%; the query expansion search produced on average 9 further highly relevant documents at slightly higher precision.

(5) The results demonstrated the effectiveness of the $w(p-q)$ algorithm, for the ranking of terms for query expansion, within the context of the Experiment.

(6) The main results of the comparative evaluation of the six ranking algorithms, i.e. $w(p-q)$, EMIM, F4, F4modifed, Porter and ZOOM, are that: (a) $w(p-q)$ and EMIM performed best; and (b) the performance between $w(p-q)$ and EMIM and between F4 and F4modified is very similar;

(7) A new ranking algorithm is proposed as the result of the evaluation of the six algorithms.

Finally, an investigation is by definition an exploratory study which generates hypotheses for future research. Recommendations and proposals for future research are given. The conclusions highlight the need for more research on weighted systems in operational environments, for a comparative evaluation of automatic vs interactive query expansion, and for user studies in searching weighted systems.

Chapter 1

Introduction

To date most information retrieval research experimentation on relevance feedback systems has been conducted in the laboratory. However, Sparck Jones (1988) and other researchers have called for more testing of probabilistic retrieval techniques in operational environments. There is therefore an apparent need for carrying out real, rather than simulated, interactive searching, in order to investigate the behaviour of relevance weighting under the constraints imposed by real users.

The research presented in this thesis follows this line of research thinking and is an investigation of interactive query expansion.[1] The overall aim of this research is to investigate the process of interactive query expansion (IQE) from various points of view including effectiveness. In other words, the aim was broader than just effectiveness. In order to investigate the process of query expansion as well as its effectiveness one needs to have a real system. I made use, therefore, of real users with their real requests in an operational environment, as opposed to searching a static test collection with fixed (artificial) queries, in order to study query expansion in a dynamic user centred environment.

For the research reported the INSPEC database, on both Data-Star and ESA-IRS, was searched online using CIRT, a front-end system that allows weighting, ranking and relevance feedback.

The thesis is divided into three parts.

Part I: background information

Part I of the thesis covers background information and appropriate literature reviews.

An overview of document retrieval systems highlights the major deficiencies of Boolean retrieval systems and discusses the probabilistic approaches to IR with special emphasis on the relevance weighting theory (Binary Independence Model) (Robertson & Sparck Jones, 1976). Chapter 2 also introduces what is experiment and what is investigation in the

[1] The terms interactive query expansion and semi-automatic query expansion are used interchangeably in the text.

context of information retrieval testing. This provides the reasons and the explanation for the adoption of the investigation approach in this research.

The chapter on 'interaction in IR' covers the interactional aspects between the user, the intermediary and the retrieval system. There is a discussion on search strategy, query formulation, query reformulation, simple relevance feedback, and the idea of 'interactive' query definition and expansion.

A more technical discussion on query expansion is given in chapter 4 which starts with the issue of the curse of dimensionality. Approaches to automatic and semi-automatic query expansion are reviewed and examples of how these have been implemented in various systems are given.

A detailed treatment of the ZOOM and SuperZOOM facilities of ESA/IRS is also presented here because of its importance for my experimental investigation. ZOOM was used for the analysis of the relevant document set that provided the terms for query expansion.

The last chapter of Part I introduces the idea of front-ends for document retrieval systems (DRS) and discusses in detail CIRT. CIRT, the front-end used for conducting this research, allows weighting ranking and relevance feedback while searching a Boolean vendor system. Chapter 5 provides a step-by-step discussion of the underlying theory of CIRT and its development. A technical description and a detailed discussion of how a weighted search is conducted through CIRT are given in order to provide the necessary background information to facilitate the discussion in the subsequent chapters.

Part II: Interactive query expansion: the pilot case studies

Part II starts by introducing the idea of interactive query expansion and how it relates to searching. It discusses why there is a need for a module for interactive query expansion. The emphasis is placed on a weighted environment especially one implemented in a front-end like CIRT.

A general description of the three Pilot case studies and the Experiment is given together with the overall methodology. Each Pilot study looked at different aspects of the query expansion process by using a front-end. The knowledge and experience that was gained from the Pilots was then applied to the methodology of the study proper (i.e., in the operational situation with real users) that is described in Part III.

Pilot 1

The aim of Pilot 1 (chapter 7) was to look at the process of query expansion as a whole and see what can be learned from it. It investigated the different sources for selecting query expansion terms for the front-end, e.g., using the relevant document set and ZOOM on certain fields of the record, such as descriptors, titles, abstract; selecting terms from the INSPEC thesaurus, etc.

There is a discussion on the ranking of terms, on the formulae used to rank the initial query terms, on the need for a different formula for the ranking of the query expansion terms, and on the six algorithms used in the experiments. The reasons for the selection of $w(p - q)$ as the ranking formula for the query expansion terms are given as well as the reasons why this algorithm is good for term discrimination.

Pilot 2

Pilot 2 looked at the process of adding new terms in the search. Its aim was to identify how searchers of a weighted system do query expansion without getting any help from the system.

CIRT did not offer any means of query expansion and the task was left entirely on the user. Searches from the CIRT evaluation project were analysed in order to see whether the users expanded their queries. If they did, then I investigated what was the source of the query expansion terms.

Pilot 3

Pilot 3 looked into the questions of 'what evidence is there to indicate that terms taken from relevance judgements of the first iteration search might subsequently be useful?', 'How predictive is the ranking of the first iteration (set) in retrieving the documents of the second set?' and 'Are the terms high-up on both lists?'

In summation, the Pilot studies presented in Part II were used to answer methodological questions as well as research questions about the query expansion terms. The main questions were: 'How useful are the query expansion terms?', 'Where to get them from for the proper experiment?', 'How to rank them?', 'Which ranking method to use in ranking query expansion terms?'.

Part III: Interactive query expansion: the experiment

Part III discusses the final full scale real-life experiment of this research and the evaluation of the six ranking algorithms.

The experiment

Having investigated query expansion under the controlled experimental conditions of the Pilot studies and having gained experience and learned from it I then proceeded with the Experiment under real conditions. I used an operational system, CIRT, to search a commercially available database, INSPEC on Data-Star, in order to study query expansion in a dynamic user centred environment.

When looking at a real system, real requests, and real interaction, it is in the characteristics of the interaction where a lot of the emphasis lies. The most obvious characteristic, for example, is the selection of terms at a particular stage in the process. The results, therefore, have a quantitative as well as a qualitative component.

Data were collected from 25 searches. The data collection mechanisms included questionnaires, transaction logs, and relevance evaluations. The variables examined were divided into seven categories which include: retrieval effectiveness, user effort, subjective user reactions, user characteristics, request characteristics, search process characteristics and term selection characteristics.

Studies of operational systems produce a wide range of results and this study is no exception. However, the most important results with respect to query expansion are reported from the term selection characteristics and from the evaluation of the six ranking algorithms, which were the focus of this research.

The results of the Experiment are presented in two sections in Chapter 10, i.e. main results and other findings according to the treatment of query expansion. The main results relate directly to query expansion, e.g. what type of term relationships users identify between the initial query terms and the query expansion terms. In addition the main results discuss retrieval effectiveness, the correspondence of the online-to-offline relevance judgements, and the performance of the $w(p - q)$ ranking algorithm. The other findings from the analysis of the results of the questionnaires and the searches that do not relate directly to query expansion cover the user characteristics, ˉsubjective user reactions and search process characteristics.

Evaluation of the six ranking algorithms

Chapter 11 presents a comparative evaluation of the six algorithms that were introduced in Pilot 1 and were considered for the ranking of the query expansion terms. These, i.e. $w(p - q)$, EMIM, F4, F4modified, Porter and ZOOM were put to test. The yardstick for the evaluation was provided by the user relevance judgements on the lists of the candidate terms for query expansion. The evaluation focused on whether there are any similarities in the performance of the algorithms and how those algorithms with similar performance treat terms.

The general conclusions drawn from this investigation of interactive query expansion in a weighted environment are presented in Chapter 12. An investigation is by definition an exploratory study which generates hypotheses for future research. Recommendations and proposals for future research follow the presentation of the general conclusions.

The conclusions highlight the need for more research on weighted systems in operational environments and for a comparative evaluation of automatic vs interactive query expansion. An additional recommendation is made for user studies in searching weighted systems.

Part I

Background information and literature reviews

Chapter 2

Document Retrieval Systems

2.1 A brief overview of DRS

As information systems, document retrieval systems (DRS), database management systems (DBMS) and expert systems try to satisfy one main function, i.e., to retrieve information from a store or database in response to a user's query (Croft, 1982). In addition the system should retrieve all and only that information which the enquirer wants or would want. In this sense *information retrieval* (IR) is studied by a wide range of disciplines in computer-based non-numeric processing. Topics such as automatic natural language processing, algorithms for searching, compression techniques, multi-media information systems and novel computer hardware are relevant to IR. There is therefore more to an information system than just retrieval. Historically IR has been taken to refer to techniques for the storage and retrieval of textual information. The discussion below is concentrated on DRS, leaving aside both DBMS and expert systems. (For an overview of information systems the reader is referred to Croft (1982)).

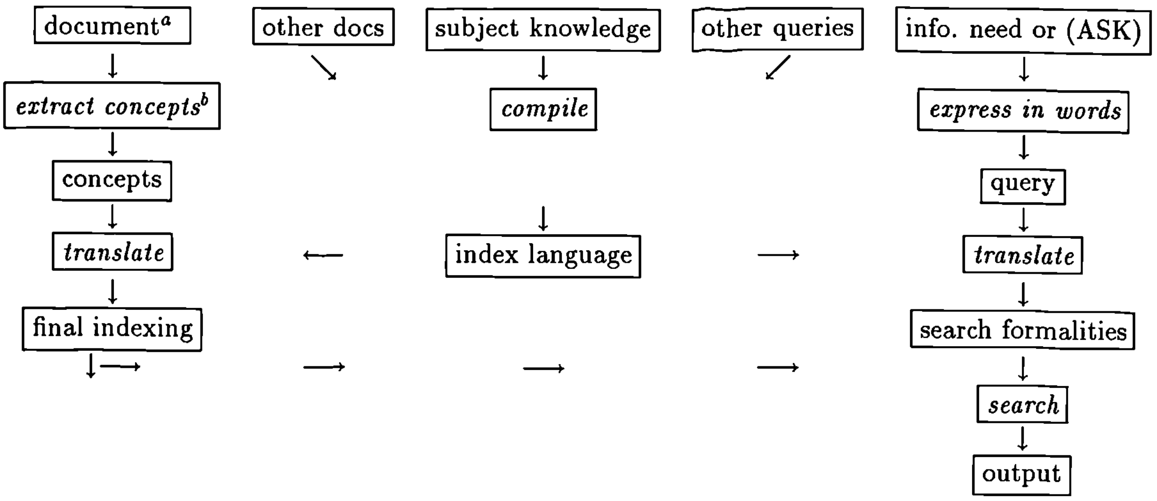
The difference of DRS to the other information systems lies mainly in the nature of the information it stores and retrieves. The goal of a DRS is to retrieve information in response to human articulation of information needs by informing the enquirer which documents contain the needed information, i.e., reference retrieval[1] [2]. A wide range of issues have to be addressed in IR. These include, especially recently, user modelling, implementation of text databases for efficient access and the user interface. A core operation of a DRS is the comparison of user queries to document representations, and it is on this operation that my research is partly focused.

Figure 2.1 presents the traditional model of a DRS system in a simplified form (Robertson, 1978). The diagram contains the two most important elements of DRS, i.e., indexing and searching. At the left of the diagram is the 'indexing side' and at the right is the 'searching or query side'. For simplification both sides are presented as linear, but both involve feedback. This model is presented here, firstly, because it provides the 'standard view' of DR, and secondly, as a reminder of the co-existence of the indexing and

[1] Information retrieval, document retrieval and reference retrieval will be referred to as IR from now on.

[2] The terms document retrieval systems (DRS) and information retrieval systems (IRS) will also be used interchangeably from now on.

searching sides. The model contains **entities** and *activities* which may take different forms in different systems, e.g., online search, menu selection process (as determined by form of index language). Activities may be shifted from one part of the diagram to another. For example, an 'index language' may be highly constrained, therefore, 'translation' is very important to it; or, 'translation' may be trivial and consequently the'translation' of the 'information need' may be more important.



[a]normal text = entities
[b]*italicised text = activities*

Figure 2.1: A model of a document retrieval system

In principle, a DRS has all the parts shown in the diagram. This is of particular importance to those interacting with a DRS because for most of the time they might have control over one part of the system only. For example, searchers do not have control of the indexing side but their searching effectiveness is determined of how well they have understood the indexing of the database they search. It is also useful to think of the different roles of people within the DRS, e.g., how indexers may have put information in, etc.

The term *document*, mentioned above, is being very loosely defined here, covering almost any type of representation of information across many possible format. This includes abstracts, journal articles, newspaper articles, technical papers, reports, descriptions of audiovisual material (films, records) and magnetic media, and so on. The text of documents is stored by the DRS as a structure that could be provided in some form of *text representation.* The most commonly used method is to represent the document by an unordered subset of words that appeared in it (uncontrolled keywords/terms — free-text indexing) or that appear in the indexing language used by the system (controlled keywords/terms). Index terms can be individual words or phrases and both controlled and uncontrolled terms have some semantic meaning. The retrieval process could be modelled as having the following form:

- The DRS derives text representation of documents, as mentioned above.

- The user approaches the system and submits an *ad hoc* query.

- The DRS processes the query and for each document in the store compares the representation of the user query to the document representation it holds.
- This results in the retrieval of some documents from the store which are predicted by the system to be relevant to the users query and which are presented to the user for further consideration.

There may be several iterations of all but the first of the above steps during which the user and/or the system may modify the initial query.

2.1.1 User-approach to the system

On the query side of Figure 2.1, one model of the user as part of the information system assumes that users approach the system because they recognise the existence of some anomaly in their state of knowledge (Belkin, Oddy & Brooks, 1982). This anomaly or information need which may take the form of a written statement of interest or in some cases is merely a vague idea has then to be described by the users.

This description, which will be referred to as the *query*, must be translated into a language that is acceptable to the system. For example, the query may be reduced to a number of search terms (keywords) that appropriately represent the contents of the query.

In order to satisfactorily achieve this transformation (query formulation) the user needs to have good knowledge of the database to be searched, the subject representation (controlled vocabulary), Boolean logic and so on. The query formulation can be performed by the user himself or it can be delegated to a search intermediary.

2.1.2 Query-document comparison

2.1.2.1 System comparison

The process of comparing the query to the documents is said to be done by a *matching function* (van Rijsbergen, 1979, p.97).

Simple matching or *co-ordination level matching* is an example of a matching function. It assigns a score to a document equal to the number of terms that the document representation has in common with the query representation. Since simple matching assigns a numeric score to each document it is a kind of *weighted matching.* A score, like the above, obtained by some matching function can be used either for ranking a set of documents (ranked retrieval) or for retrieving a single set of documents scoring above a predefined threshold (set retrieval). In a Boolean system the binary-valued functions used (i.e., strict yes-no matching function) predetermines the use of set retrieval.

Nevertheless, as Robertson and Belkin (1978) argue, the manner in which the DRS presents the user with the document(s) is of interest in retrieval.

> "...the system may present only one text, or an unordered set, or may rank the texts in some order. But any of these processes (or indeed, any act of retrieval) *is* a ranking process (see Cooper (1968)). ... An explicit ranking beyond simple set retrieval, is normally based on a matching process: it reflects the *degree of match*, as measured by a *matching function*, between the texts and the request as put to the system..."

Term-matching, i.e., the process of comparing the query to the documents is an inherent problem in traditional IR (this description does not apply to other types of IR systems, like menu-driven or hypertext systems). Whatever the retrieval mechanism and the retrieval technique used by the system (see above, Figure 2.2 and Belkin and Croft (1987)) terms from the user's query are matched at a symbolic, text-matching level against document representations. This form of matching is some distance removed from any semantic and contextual information in the document. Effective use of existing Boolean IR systems depends heavily on the users' perception of the data structure that their queries are addressing. The data structure (database view) is reflected in the query language, and vice versa. In other words, present query languages reflect the model of the logical structure of the databases. The IR situation is confined to set operations (creation and manipulation). Successful handling of these operations will result in a final set which to some degree will satisfy the user's expressed information need.

2.1.2.2 User comparison

The process of deciding upon the relationship that exists between a retrieved document and the query is called *relevance judgement* and plays a very important role in IR. What constitutes relevance and the distinction between relevance and usefulness has been extensively studied by many IR researchers and in many contexts. Saracevic (1975) and more recently Schamber, Eisenberg & Nilan (1990) provide an extensive review of the notion of relevance and its treatment in information science.

In general, relevance is defined as the extent to which the subject matter of the document is *about* the query. Therefore, associated with each query is a variable whose value is the relevance of the document to the query. Robertson (1977a) describes a model for relevance in which relevance is assumed to be a continuous variable. Degrees of relevance, which relate to the process of judging relevance, are defined by dividing the continuous scale over which the variable ranges into as many divisions as there are degrees.

In order to facilitate the discussion below I will make use of a rather simple definition of relevance, adopting a binary view of relevance. This assumes that relevance (or usefulness, or user satisfaction) is a basic dichotomous criterion variable, defined outside the retrieval system itself, so that a document is either relevant or not and there are no in-between states (Robertson, 1977b).

2.2 Retrieval mechanisms

Belkin and Croft (1987) have developed a classification of retrieval mechanisms, summarised in Figure 2.2. In their definition, a Retrieval Technique (RT) is a technique for comparing the query with the document representations. RTs are further classified according to the characteristics of the retrieved set of documents and the representations that are used. Some RTs fall into more than one category, and others are a mixture of techniques from different categories.

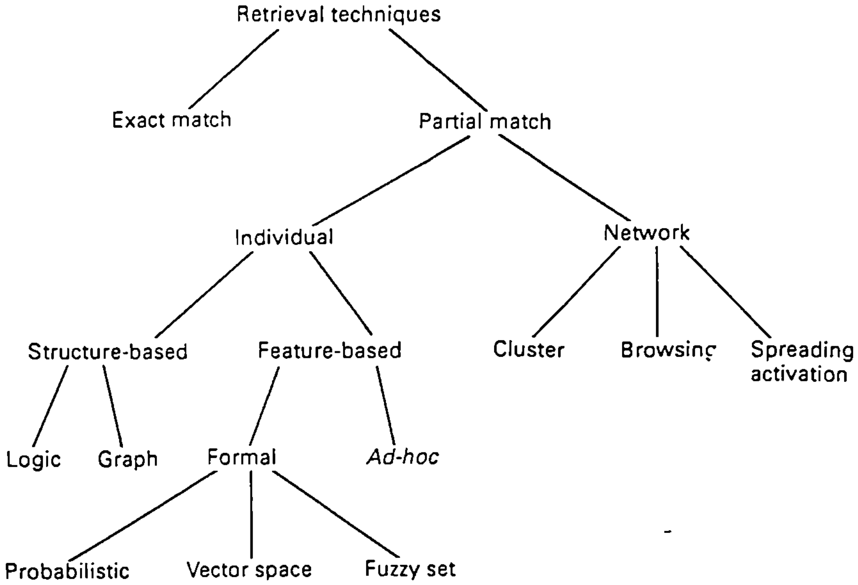


Figure 2.2: A classification of retrieval techniques.

The first distinction is between exact and partial match. In the former category are RTs that require any retrieved item to match the query exactly, which includes Boolean systems, those using the pseudo-Boolean free text operators, and string-searching systems. Partial match RTs, by which items may be ranked according to degree of match, are further divided into individual and network RTs. The former are based on matching queries against document representatives treated individually, whereas the latter make direct use of inter-document links of some kind.

Following this brief introduction to the RT I will concentrate in my discussion only on two RTs, Boolean from exact match and probabilistic from the partial match techniques.

2.2.1 Boolean searching

Most present day large operational DRS rely heavily on exact match techniques, i.e., Boolean logic. A variety of aids such as thesauri are needed to achieve reasonable performance. In the IR literature, therefore, the traditional Boolean searching has been the subject of many textbooks, including Henry *et al* (1980), Meadow and Cochrane (1981), Harter (1986), Hartley *et al* (1990), etc., and articles in journals such as *Online Review, Online, Database*, and *Database Searcher.*

The figure describing the 'steps in the pre-search interview and the online search' (Figure 2.3) has been adopted from the Meadow and Cochrane (1981) book on online searching.

Steps

1. Clarifying and Negotiating the Information-Need and Search Objectives
    - Interviewing the information requester clarifies the narrative form of the request and determines search objectives:
        (a) retrieve all relevant items (high recall);
        (b) retrieve only relevant items (high precision);
        (c) retrieve some relevant items.
    - Identify constraints (e.g., books only as output or only in English, or only if published after 75)
2. Identifying Relevant Online System and Databases
    - Determining which online system and data base to use first, which next, etc.
3. Formulating Basic Search Logic and Planning Search Strategies
    - Analysing the search topic into parts called facets or concept groups. Planning approaches to search strategy for combining concepts of the topic.
4. Compiling the Search Terms
    - Choosing indexing terms from the data base's thesaurus or other printed word lists.
    - Selecting terms for free text searching of the subject-conveying fields (title, abstract, etc.).
    - Deciding to use thesaurus and alphabetic word lists online.
5. Making Output Choices
    - Choosing limits on, and printing of, output.
    - Selecting an approach to search strategy that best satisfies the search objectives expressed by the requester.
6. Conceptualising the Search as Input to the Retrieval System
    - Arranging the search terms into concepts or facets for search strategies using features of the retrieval system, for example, word proximity.
    - Noting most important and less important concepts groups and deciding on sequence of input to access these concept groups efficiently.
    - Restricting or limiting output based on search objectives.
7. Evaluating Preliminary Results
    - Reviewing search results, step by step.
    - Considering alternative search strategies to meet search objectives (recycling Steps 1-6).
8. Evaluating Final Results
    - Determining requester's satisfaction with search results.

Figure 2.3: Steps in the pre-search interview and the online search.

These steps will be examined and discussed. It is worth noting here that even if the end-user searches alone, i.e., without the help of an intermediary, there will still be a need to clarify and negotiate the request and to know enough about the language of the database being searched to match the request with the basic index provided. In other words the user would have to follow all the steps on her or his own.

In Figure 2.3 steps 1, 2 and 4 could be said to belong in the pre-search procedures. Steps 3, 5, 6 and 7 are activities used in both pre-search procedures and during the search and step 8 is a post-search action. The user-intermediary/intermediary-system interactions would determine the order of the steps and any possible repetitions of them. The steps, thus, are not in any fixed order, apart from the first one, which involves the user efforts to express and delegate her/his information need to the system (i.e., here the intermediary). The pre-search reference interview (Eichman, 1978; Knapp, 1978; Markey, 1981; Taylor, 1968) is the user's first interaction and the feedback from it creates and links the conceptual analysis of the request (query) and the manipulation of concepts to develop a search strategy.

Search intermediaries have their own stereotypes both about how to deal with users and the online services, and also of how to conduct searches. At this pre-search stage the intermediary and the user are involved in a cognitive exchange. The stages of this interaction (e.g., steps 1-6 but especially step 3) involve feedback that may change the searcher's (human intermediary's and/or end-user's) view of what is wanted and consequently affect query formulation. A Boolean search system has nothing that could be described as a dynamic cognitive structure (Ingwersen, 1984). Hence any evolution in the interaction has to involve changes in the cognitive structures of the human beings.

Having dealt with steps 1-6 the searcher then goes online. At this interactive session the search formulation is being submitted to the database and feedback is involved as soon as the searcher receives messages that would alter her/his search strategy. As a general rule, it would be fair to assume that any online search will involve feedback and modifications to the search formulation - except perhaps in the case of a one concept one statement search for, say, an exhaustive literature search or for a known item search. Thus, the searcher would have to iterate some if not all of the steps 1-6. The work by Bates (1979a; 1979b; 1981; 1986; 1987), Fidel (1985), Harter and Peters (1985), etc., deals with this problem and suggests various tactics (or moves) to be made during the search. All these tactics in effect are suggestions for feedback to the database as responses to its messages (e.g., postings, error messages, etc.).

Boolean systems have disadvantages which are both well known and well documented (Belkin & Croft, 1987; Bookstein, 1985; Willett, 1988):

- they may miss many relevant records whose representations match the query only partially.
- there is complete lack of control over the size of the output produced by a particular query. A searcher is unable to predict *a priori* how many records will be retrieved. There may be none at all or too many. Only by query reformulation a more useful number of records could be hopefully retrieved.

- they do not rank retrieved records. This is a consequence of the retrieval operations which result in a simple partition of the database into discrete sub-sets, i.e., those records that satisfy the query and are being retrieved and those which do not. All records within the retrieved set are presumed of equal usefulness to the user and therefore cannot be ranked in order of decreasing probability of relevance.
- they cannot take into account the relative importance of concepts either within the query or within the document. There are no obvious means by which one can reflect the relative importance of different components of the query, since Boolean searching implicitly assumes that all terms have weights of either 1 or 0, depending upon whether they happen to be present or absent in the query.
- retrieval depends on the two representations being compared having been drawn from the same vocabulary.
- they require complicated query logic formulation. The logic associated with the Boolean operators AND, OR, NOT has poorly understood consequences. Most users are usually unable to make good query formulations and require the assistance of trained intermediaries.

2.2.2 The probabilistic approach

IR researchers in their attempt to respond to the limitations of Boolean systems have developed a number of alternative RT as seen in Figure 2.2 under *partial match* techniques which are also known as *best match* techniques. These grew out of a number of different theoretic models which, although they have been tested mostly under artificial laboratory conditions, are promoted as having the potential for transforming the way searches are implemented and for significantly improving system performance. The most successful of these models are the probabilistic model (Maron & Kuhns, 1960; Robertson & Sparck Jones, 1976; Croft & Harper, 1979; van Rijsbergen, 1979) and the less formal vector space model (Salton, 1971).

Before discussing the probabilistic approach it is worth commenting on some general issues applicable to most best match systems. An essential feature of the partial match techniques is term weighting and the weighting matching function used. Term weights are precision devices and distinguish the better or more important terms from the less important ones. Such discrimination helps to rank the output in decreasing order of presumed importance, most relevant documents at the top and least relevant at the bottom. This comes in contrast to Boolean set retrieval where a set is retrieved and all documents in that set are being treated as having equal importance. A weighting function therefore assigns measures of relative importance to the terms which have been selected to describe a document or a query. In other words, weighting is a feature that can be associated with any of the partial match techniques, but the importance lies on the way one arrives at a weighting function, i.e., theoretically or empirically.

The work of Maron and Kuhns (1960) provides a good starting point to overview the theory of probabilistic retrieval. They were primarily interested in the probabilistic indexing of documents, although the indexing decision was related through probabilities to its effect on the retrieval of documents. A key concept in their theory was the notion of relevance and their work on this subject made substantial contributions in the field of IR.

In their 'probability of indexing' model the probability of relevance is computed relative to evidence consisting of the type of query(-ies) that the user(s) has(have) submitted. The probability is then interpreted in its frequency sense. For example, if a query consists of a single term Q_{t_i} the probability that a given document D_n^R will be judged relevant, by the user who submitted the query term Q_{t_i} is simply the ratio of the number of users who submit term Q_{t_i} as their search term and judge document D_n^R relevant, to the number of users who submit term Q_{t_i} as their search term. Because of lack of actual statistics, which can only be derived from user feedback, on which to base the estimation of the values of these probabilities, an indexer can only guesstimate the values of the probabilities. Hence, in probabilistic indexing the task of the indexer is precisely defined as that of the estimation of the values of the probabilities that a term Q_{t_i} will be used in a query where the document D_n^R will be judged as relevant by that user, i.e., $P(Q_{t_i}|A, D_n^R)$, and then to assign those terms Q_{t_i} to the corresponding documents D_n with the values of those estimates.

In my discussion on the theories of probabilistic retrieval I will assume that document indexing is of the conventional non-probabilistic kind, i.e., it is binary subject indexing. This is because the ideas of probabilistic indexing have different assumptions to those of probabilistic retrieval as already explained. Nevertheless, it is worth mentioning here that some of the important research work on probabilistic indexing has been done by Bookstein and Swanson (1974; 1975), Harter (1975a; 1975b) and by Salton *et al.* (1981). A recent theoretical advance is the effort to combine these theories of indexing and retrieval into a unified theory of information retrieval (Robertson, Maron & Cooper, 1982; Robertson, Maron & Cooper, 1983).

The probabilistic theory of retrieval explicitly recognises the element of uncertainty involved in the retrieval process, i.e., that given a request (query) the retrieval mechanism would have to decide which documents to retrieve and in doing so there would be some inappropriately retrieved documents while some other more desirable documents would not be retrieved.

In probabilistic retrieval index terms form the basis of the retrieval decision. Thus, a probabilistic system may begin a search by assigning probabilities, i.e., numerical measures of uncertainty, to index terms. This indicates how likely it is that those terms will appear in a relevant or non-relevant document. These probabilities are further manipulated to derive the probability that a document is relevant to a query. Let's assume that in a collection of N documents each document is described by the presence or absence of a number of index terms n which correspond to the terms found in the collection's main index. A document in the collection can be represented as:

$$D = (t_1, t_2, \ldots, t_n)$$

where $t_i = 0$ indicates the absence of the term, and
$t_i = 1$ indicates the presence of the term

We have also assumed that relevance is a dichotomous variable so that a document is either relevant or non-relevant to a query. The system needs first to estimate the probability of relevance $P(relevant|D)$ or non-relevance $P(nonrelevant|D)$ of the documents. The probability of relevance is computed relative to the set of document properties, i.e., index terms assigned to it. Then the documents are ranked in descending order of their estimated probability of relevance $P(relevant|D)$ and by using some cutoff point or threshold a top-ranked portion of the documents is presented to the user.

The formal basis about ranking in probabilistic retrieval is given by the Probability Ranking Principle (PRP) formulated by Cooper (1977) and quoted from Robertson (1977b):

> "... If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data... "

2.2.2.1 Parameter estimation

One of the most difficult parts of the probabilistic approach is that it depends on parameters, not all of whose values are known. Thus parameter estimation has always been a stumbling block in the development of probabilistic models.

A weighting scheme is needed to start the search. For example, quorum searching (or co-ordination level matching) (Cleverdon, 1984) or inverse document frequency weighting (Sparck Jones, 1972) could be used for this purpose. The IDF, an empirical weighting rule as introduced by Sparck Jones, got a theoretical justification as being a limiting case of probabilistic relevance weights (Croft & Harper, 1979; Robertson, 1986) and is discussed on page 18.

After the initial search has been carried out a set of documents is retrieved and presented for evaluation to the user. The relevance feedback provided by the user at this stage forms the basis for estimating the parameters for subsequent retrievals. The theory of relevance weights (Robertson & Sparck Jones, 1976) considered the use of relevance information as the basis for the weighting of query terms. It makes use of two kinds of assumptions: term independence assumptions and document ordering assumptions. These are:

I2 *Independence assumption*: The distribution of terms in relevant documents is independent and their distribution in non-relevant documents is independent.

O2 *Ordering principle*: That probable relevance is based on both the presence of search terms in documents and the absence from documents.

By using the above assumptions they were able to provide a theoretical framework for term weighting. Because of these assumptions, especially of **I2**, the relevance weighting theory is also known as the binary independence retrieval model (BIM). The basic formula is:

$$w_t = \log \frac{p_t(1-q_t)}{q_t(1-p_t)} \tag{2.1}$$

where p_t is the probability of term t occurring in a relevant document
q_t is the probability of term t occurring in a non-relevant document

The application of this formula requires some knowledge as to the relative occurrence of the term in relevant or non-relevant documents. The probability p and q may be estimated from relevance feedback information. So, given a sample of some, but not all, of the relevant documents probability estimates can be made. In practice it is convenient to replace the probabilities by frequencies. Let us consider the estimate for a single term t and query q by assuming that:

N is the total number of documents in the collection

R is the sample of relevant documents as defined by the user's feedback

n is the number of documents indexed by term t

r is the number of relevant documents (from the sample R) assigned to term t

For each such term t we can construct a 2 × 2 'contingency' table as follows:

		Document Relevance		
		Relevant	Non-relevant	
Document	$t_i = 1$	r	$n - r$	n
Indexing	$t_i = 0$	$R - r$	$N - n - R + r$	$N - n$
		R	$N - R$	N

From the above table four relevance weighting formulae can be derived. All four formulae use the same probability estimates, i.e. $p_t = \frac{r}{R}$ and $q_t = \frac{n-r}{N-R}$, in different ways. However, only one satisfies the assumptions as mentioned above. This has become known as the F4 formula.

$$w_t = \log \frac{\frac{r}{R-r}}{\frac{n-r}{N-n-R+r}} = \log \frac{r(N-n-R+r)}{(n-r)(R-r)} \tag{2.2}$$

The relevance weighting theory is based on the idea that both the presence and absence of query terms in a document are important. In other words, the user wants to accept documents with good terms, i.e., ones correlated with relevance, and reject documents with bad terms, correspondingly rejecting all those without good terms and accepting the ones without bad terms[3]. For a request and document with many terms, the final result would be the net balance for all good and bad terms (Sparck Jones, 1979a, p.38). The theory provides therefore an ordering mechanism which is quantified as the *'simple sum-of-weights'* over all

[3] The basic problem is of course that the same term is accepted in one document but rejected in others because the meaning of any term is largely its context in a particular document. Strictly, the same term can be both good and bad in different document contexts. The user is deciding on meaning not absence or presence of terms. But, the sheer complexity of this situation makes it rather difficult to model it successfully that is why the relevance weighting theory makes all the simplified independence assumptions.

of the terms in the query. If t_i indicates the presence ($t_i = 1$) or absence ($t_i = 0$) of a term in a document D (see page 14) the matching function is:

$$w_D = \sum t_i \log \frac{p_t(1-q_t)}{q_t(1-p_t)} \tag{2.3}$$

There are a number of problems with the estimation of the F4 parameters. At first this information is not available. After the relevance feedback of the first iteration some relevant documents may be known and the p_t, q_t and w_t may be estimated. Robertson and Sparck Jones (1976) have shown that p_t should be estimated from the known relevant documents, but the base for estimating q_t is not quite so obvious. Harper and van Rijsbergen (1978, p.204) argued that the *complement method* should be used, i.e., instead of using the (very few) known non-relevant documents for the estimation of q_t, the remainder of the collection (i.e., all those not known to be relevant) should be used.

The second problem concerns the validity of the simple proportion estimates used in F4. If any of the four cells in F4 (equation 2.2) is zero then it yields infinite weights. This possibility arises when the sample of known relevant documents is very small or zero. To overcome this problem Robertson and Sparck Jones modified the formula by adding 0.5 to each of the four quantities in the F4. The result is known as the point-5 formula:

$$w_t = \log \frac{(r+.5)(N-n-R+r+.5)}{(n-r+.5)(R-r+.5)} \tag{2.4}$$

This correction minimises bias and does not yield infinite weights. An account of the structure of relevance weights, the presence-absence components and the approaches to estimation are given by Sparck Jones (1979a, pp.38-41).

Harper and van Rijsbergen (1978) have suggested a different version of the F4 matching function (equation 2.4) which is thought to overcome the parameter estimation problems. This function is a modification of the expected mutual information measure (EMIM) (van Rijsbergen, 1977), and because it has no theoretical basis will not be discussed any further in this chapter. A detailed account of EMIM will be given in section 7.4 on page 111.

The relevance feedback techniques mentioned so far have assumed the use of some relevance information which in a retrieval situation are available on the second or subsequent iterations of the search. In the case where no relevance information is available, i.e., in the first iteration, it has been suggested by Croft and Harper (1979) that we could assume that all the query terms have equal probabilities of occurring in the relevant documents. We could also assume that the occurrence of a term in a non-relevant document may be estimated by its occurrence in the entire collection. The two assumptions correspond to setting all p_t from equation 2.1 equal to a constant (k), where $C = \log \frac{p_t}{1-p_t} = \log \frac{k}{1-k}$, and $q_t = \frac{n}{N}$. So, they arrive at a weighting function:

$$C \sum t_i + \sum t_i \log \frac{N - n_t}{n_t} \tag{2.5}$$

where n_t is the number of documents indexed by term t
N is the total number of documents in the collection
$i = 0$ or 1 indicates the absence or presence of the ith term

This expression is called the *combination match* because it is a weighted combination of a simple *co-ordination level match* (the first part of the expression) and the *IDF weighting* (the second part).

The *inverse document frequency* (IDF) weighting or the *collection frequency* weighting as introduced by Sparck Jones (1972) is determined by the function:

$$w_t = -\log \frac{n}{N} = \log N - \log n \tag{2.6}$$

where w_t is the weight for term t
n is the number of documents indexed by term t
N is the total number of documents in the collection

The rationale behind this approach is that of the discriminating power of a term, i.e., a frequently occurring term would occur in many irrelevant documents whereas infrequent terms have a greater probability of occurring in relevant documents. Low frequency terms are thereby given the highest weight and are most influential in determining which documents will be retrieved. In practice this weight is implemented as (Sparck Jones, 1972; Robertson, 1974)

$$w_t = \log \frac{N}{n} + 1$$

The IDF weight that is derived from the Croft and Harper weighting function (equation 2.5) is marginally different from Sparck Jones'.

Two special cases could occur with the combination match depending on the values of the constant C. If $C = 0$, i.e., all $p_t = 50\%$, then the weighting defaults to an IDF weighting. If C approaches infinity ($C \to \infty$) the weighting is approximately equivalent to ranking the documents by IDF within co-ordination level matching. This means that if more than one document have the same co-ordination level weight (i.e., they are tied on the same rank) these will be further ranked by their IDF weight.

There are two final points to be made here. Firstly, that an analysis similar to the above that demonstrates the close relationship between IDF and the relevance weight theory is given by Robertson (1986). The second point also derives from the first and challenges the Croft and Harper claim that the relevance weight theory cannot be used in the initial search. This claim holds for F4 (equation 2.2) because when $R = r = 0$ it becomes undefined, but not for the point-5 formula (equation 2.4), which for an initial search, where $R = r = 0$, it becomes:

$$w_t = \log \frac{N - n + .5}{n + .5} \tag{2.7}$$

Thus, according to Robertson the point-5 formula behaves like a Bayesian estimator. It provides an estimate when there is no evidence available (i.e., initial search) and modifies the initial estimate as the evidence is obtained (i.e., second and subsequent iterations). This behaviour of the relevance weight function is useful because it demonstrates some learning properties and also because it opens itself to further modification as it will be discussed later 7.4.

2.2.3 Term Dependencies

The probabilistic theory for retrieval discussed so far is based on the independence assumptions given on page 15. The term independence models assume that the probability of a term occurring in one of the sets of documents (relevant or non-relevant) is the product of the probabilities of all the individual terms of that document occurring in that set, thus

$$P(D|relevant) = P(t_1|relevant)P(t_2|relevant)\ldots P(t_n|relevant)$$

This assumption, as mentioned in the previous section, simplifies the mathematical analysis and the modelling of the retrieval process and facilitates the development of any matching function based on it. Another area of research attempted to relax this assumption, by assuming that within a single relevance class of documents terms are not distributed independently of one another. If this assumption is true then the incorporation of dependency information into the models should yield better retrieval effectiveness in the retrieval based on such models.

Two ways have been identified so far that introduce term dependence. One is using general distribution function approximation techniques and the other is attempting to model the cause of term dependence. The latter has been suggested by Bookstein and Kraft (1977) but it is still untested and therefore it will not be discussed any further.

Van Rijsbergen (1977) applied the work of Chow and Liu (1968) on tree dependences to the IR situation. By using conditional probability distributions he developed a retrieval model which incorporates first order tree dependence between terms. This could be explained as ordering the terms so that:

(a) term t_i is dependent on one of its preceding terms only, and

(b) in every such tree there is only one independent term (i.e., the root of the tree).

Of all the possible dependence trees van Rijsbergen defined the best dependence tree for the Chow expansion as being the Maximum Spanning Tree (MST). This theoretical tree dependence model has been developed into an algorithm (EMIM) (Harper & van Rijsbergen, 1978) which is discussed in section 7.4.

The second approximation technique is based on the Bahadur-Lazarfeld expansion (BLE) as described by Duda and Hart (1973) and it was introduced in IR by Clement Yu and co-workers (see Yu *et al* (1979; 1983). The BLE expansion involves the sum of a constant term, a set of terms with a single variable, another set with pairs of variables, then another set with triplets of variables and so on. The term dependence is calculated by correlation type coefficients which measure the times that terms t_i co-occur in documents with a document collection. In general, parameter estimation is problematic as it is difficult to interpret the results in a satisfying way (Bookstein, 1985, p.132).

2.3 Performance Evaluation

IR systems are complex. They contain a number of components which interact with each other to determine the overall performance. In addition, these components may be related to each other in complicated and unobvious ways. The problem of evaluating IR systems in general and of measuring retrieval performance in particular, has been very widely discussed (Lancaster (1979), van Rijsbergen (1979), Salton & McGill (1983)). There appears to be no simple solution to this problem and it is possible to object to any of the specific measures which have been put forward. Lancaster (1979, p.109), identifies 3 levels of important criteria for the evaluation of an IR service, of which an adapted version is given below.

Level 1 Evaluation of effectiveness (user satisfaction)

1. Cost criteria
   Monetary costs. Also, costs in terms of effort involved on the part of the user in learning how to use the system, using the system, reviewing the search output, and in obtaining documents (through, for example, document delivery).
2. Time criteria
   i.e., response time of host computer from submission of request to retrieval of citations, or from submission of request to retrieval of documents, etc.
3. Quality considerations
   (a) the coverage of the database
   (b) the completeness and accuracy of the data
   (c) the timeliness of the output
   (d) the recall of the system
   (e) the precision of the system

Level 2 Evaluation of cost effectiveness
i.e., user satisfaction related to internal system efficiency and cost considerations.

Level 3 Cost-benefit evaluation
i.e., value of the system balanced against costs of operating it.

Our main concern lies with the criteria found under quality considerations and especially recall and precision, which attempt to measure the effectiveness of the system. In choosing measures for retrieval effectiveness there are three points that have to be considered (Sparck Jones, 1971, p.94):

1. the factors to be taken into account as constituents of the measure;
2. the way in which these are related to one another and incorporated in a formula; and
3. the way in which a series of measurements describing different systems are to be evaluated and compared.

The specific retrieval factors which should be taken into account in measuring performance can be drawn from the well known 2×2 contingency table, and include (a) the relevant documents retrieved, (b) the non-relevant documents retrieved, (c) the relevant documents not retrieved and (d) the non-relevant documents not retrieved. The layout of the table can be briefly explained as follows. We assume that the retrieval system makes binary decisions in response to a request, so that a document is either retrieved or not retrieved.

Thus the result of a request, after processing all the documents held in the system, is to divide the collection into two parts, one of which is being retrieved and presented to the user. The user now goes through the retrieved documents making also binary decisions, so that a document is either relevant of non-relevant. If the user goes through the whole collection making relevance judgements for the documents that the system has decided not to retrieve[4], then we can establish the quantities for each of the four cells of the 2 × 2 table.

	Relevant	Non-relevant	
Retrieved	a Hits	b Noise	$a+b$
Not Retrieved	c Misses	d Rejected	$c+d$
	$a+c$	$b+d$	$N = a+b+c+d$ Total collection

In partial match systems either the entire collection or a subset is retrieved in descending weight order as the result of a search. The user may stop searching at any point in this ranked list. Consequently, the 2×2 table is not a complete description of the system output. Various alternative approaches have been proposed for the measurement of effectiveness irrespective of cutoff, with the main ones being 'rank recall' with 'log precision' and 'normalised recall' with 'normalised precision' (Keen, 1971). Nevertheless, the 2 × 2 table is valid in partial match systems at a specified cutoff point, e.g. the point at which a search is terminated.

2.3.1 Recall and Precision

A large number of effectiveness measures can be derived from the 2 × 2 table with the most widely used being recall (R), precision (P), fallout (F) and generality(G).'Recall' refers to the ability of the system to present all relevant documents in response to a request. 'Precision' is the system's ability to present only the relevant documents. These are defined as:

$$Recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents\ in\ the\ collection} \tag{2.8}$$

$$Precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved} \tag{2.9}$$

The degree of recall as well as that of precision achieved in a search may be expressed as ratios, which in terms of the 2 × 2 table are $\frac{a}{a+c}$ for recall and $\frac{a}{a+b}$ for precision. Neither on its own can give a complete picture of the effectiveness of a search. Recall and precision tend to be inversely related, i.e., as recall increases the precision levels decrease. In partial match retrieval the overall effectiveness for a single request can be expressed as a set of pairs of precision-recall figures and a recall-precision curve can be computed (variable cutoff). This curve shows for any desired level of recall what level of precision will have to be accepted. For statistical validity of the overall retrieval effectiveness recall-precision curves are usually averaged (micro- and macro-evaluation) over a large number of queries. *Micro-evaluation* or the 'average of numbers' is an averaging technique used when co-ordination level matching

[4] Rather unrealistic for operational conditions, but used here for the argument's sake.

is employed as a RT (van Rijsbergen, 1979, p.150). The number of documents is totalled over the requests and then averaged[5]. An alternative approach which can be independent of any parameter, such as co-ordination level, is *macro-evaluation* or 'the average of ratios'. Here, the average recall-precision curve is computed as the arithmetic mean of the precision-recall pairs for each request. Neither of these averaging techniques is really satisfactory and Sparck Jones (1978) has thoroughly discussed the problems relating to the recall-precision graph.

The recall-precision measures are undeniably important aspects of retrieval, widely used, extremely easy to use and there are no compatibility problems involved (Sparck Jones, 1971, p.97). Nevertheless, these measures have been heavily criticised for a number of reasons, the most persistent being estimation problems. In a test collection environment, for example, recall may not be defined if no relevant documents exist in the collection ($R = \frac{0}{0}$), whereas in an operational system the exact estimation of recall is impossible. Similarly, precision is undefined when no items are retrieved. Nevertheless, although it is easy to criticise the recall-precision measure, no one has proposed any obviously better alternative (Sparck Jones, 1971, p.97).

The third most commonly used measure is fallout (F) which is the number of non-relevant documents retrieved over the total number of the non-relevant documents in the system ($F = \frac{b}{b+d}$). Finally, in an ordinary retrieval environment there is also a functional relationship between recall, precision and fallout involving a parameter called generality (G) (van Rijsbergen, 1979, p.149). This is a measure of the density of the relevant documents in the system ($G = \frac{a+b}{a+b+c+d}$), so that, for example, precision may be determined in terms of recall, fallout and generality as

$$P = \frac{R \times G}{(R \times G) + F(1 - G)}$$

The higher the level of generality, the greater the density of relevant documents in the system and consequently the 'easier' the search will tend to be.

A number of alternative measures have been proposed which can be derived from the 2 × 2 table. All these measures, including the ones mentioned earlier, when used on their own are called 'single measures' of effectiveness. When combined in pairs, as in the recall-precision graph, are known as 'twin variable measures'. Finally, 'composite' measures are also being derived from the table but combine parts of it, i.e., two separate measures into a 'single-valued' measure.

A number of researchers have rejected the 2 × 2 table as the basis for the construction of parameters capable of reflecting retrieval effectiveness because these are empirically formed. Instead, they turned to formulate evaluation models based on statistical theory, so that formal models would allow performance to be predicted. The classic work on this subject was the Swets model (Swets, 1963) which tries to measure performance over the entire curve. He defined recall, precision and fallout in probabilistic terms and proposed a single-valued measure of retrieval performance known as the Swets' E measure of effectiveness (Swets, 1969). Subsequently, there were a number of modifications and extensions to the Swets model suggesting alternate single measures of system performance, such as Brookes'

[5] Robertson (1977b, p.299) suggested that micro-averaging techniques are in general of dubious value and are inadequate for probabilistic retrieval and especially so for the Probability Ranking Principle.

S measure (Brookes, 1968) and Heine's D measure (Heine, 1973). Similar measures are *normalised recall* and *normalised precision* (Rocchio, 1966), and *rank recall* and *log precision* (Salton, 1968).

Other single-valued measures include Cooper's *expected search length* (1968), Pollock's *sliding ratio* (1968) and van Rijsbergen's E measure (1979, pp.174-175). Extensive, rather complete, discussions of evaluation measures, of averaging techniques and of methods for presenting results of retrieval tests as well as an analysis of the pros and cons of the various measures that have been proposed or used till 1969, are given in Robertson (1969ab) and Keen (1971). A wide variety of measures has also been surveyed by Kraft and Bookstein (1978) and discussions on retrieval evaluation are found in van Rijsbergen (1979, pp.144-183) and Salton and McGill (1983, pp.157-198).

This section on performance evaluation has been discussed here in order to complete the overview of document retrieval systems that has been presented so far. The research reported in this thesis has been conducted in an operational system and this has constrained the choice of measures for the evaluation of performance. This is discussed in detail in section 10.2.3.1.

2.4 Information Retrieval Tests

Thus far we have seen the two main rival RT, Boolean and probabilistic. We have also seen the performance evaluation criteria used to judge the effectiveness of a retrieval system. We need to consider the environment in which these RT have been evaluated. Have these evaluations taken place in real operating system conditions or in strict scientific experimental conditions?

What are experimental conditions? In an experiment, one investigates the relationship between two variables by deliberately producing a change in one of them and looking at, observing, the change in the other. The variable which the experimenter directly manipulates is the independent variable, whereas the variable in which one is looking for any consequent changes is the dependent variable. The experiment proper is a situation of controlled observation where a variable is deliberately manipulated while controlling all other variables so that they do not affect the outcome.

In pure sciences such procedure is often easier to accomplish, because the variables, the various components of a system, may easily be identified and controlled. Repetition of experiments under exactly the same conditions is also fairly easy to accomplish. But, when moving from pure sciences to social sciences, variables become progressively more difficult to control. A new variable is also being introduced, the user, whose unpredictable behaviour makes experimentation even more difficult. What makes the difference between pure and soft scientific experiment greater is repeatability. The greater the user involvement the less the chances of repeating the results of the experiment. We arrive therefore in the IR environment where, despite the progress made and the lessons learned from IR testing in the last 20-30 years, the statement:

> "...the fact that the whole process of document description and retrieval is ill understood means that it is not easy to make this analysis of a system, and to

> list the various factors which are involved in its operation..." (Sparck Jones, 1971, p.85)

unfortunately still holds the same way it did then.

2.4.1 Experiment and Investigation

Information retrieval testing can be divided into experiment and investigation (Sparck Jones, 1981). The questions an experiment is designed to answer are of the form of 'what would happen if I do X?' or 'What would happen if I do X rather than Y?'. The questions an investigation is trying to answer are of a rather more general nature, such as 'What happens in this specific system?' or 'what happens in general?' or 'what might be happening in this system or in general?'

From that it might be easy to draw a line of what constitutes an experiment and what an investigation in an IR environment. An experiment aims at explanation, seeking to answer questions about what happens if 'this and that' is done, by showing why it happens. An experiment focuses on individual variables and it is in principle hypothesis-driven. Control over both primary and secondary test variables is a key requirement for the IR experiment, which is, consequently, concerned with measurement. Explicit comparative measurements are then required for different values of the test variables so that the function of retrieval systems (i.e., retrieval effectiveness) can be assessed. On the other hand, an investigation aims only at description and indicates only what happens if 'this and that' is done. An investigation exhibits system behaviour as a whole and it is hypothesis-generating. It may produce measurements which may be merely descriptive and therefore only implicit comparisons may be possible.

The distinction between experiment and investigation, as has been highlighted above, is according to Sparck Jones *(1981, p.214) an ideal which is very difficult to maintain when* real system tests are discussed. In her comprehensive review of IR system tests from 1958-1978 (Sparck Jones, 1981, pp.213-255) and in a more recent discussion paper on IR research covering the period after 1978 (Sparck Jones, 1988, pp.13-29), she states that work done so far cannot be described as unequivocally experimental or investigative, especially where real system studies are concerned. The essential difference between experiment and investigation is in the application of control in the former. This introduces to the discussion the kind of the environment in which a test could be conducted, the laboratory or the operational environments. In theory both tests (experiment and investigation) can be conducted in any type of environment (laboratory or operational), in practice, though, experimentation has been restricted to laboratory testing and investigation to operational system testing.

The IR system could be taken as having a core, which is the character of the indexing data and retrieval mechanisms available, and a periphery, which is the user and literature characteristics, the administrative concerns of the system and so on. Most IR testing has been confined to laboratory environments of the study of indexing and retrieval mechanisms. Operational system investigations have been relatively few in comparison to laboratory tests, for example, the Medlars, Miller's and UKCIS evaluations. Problems of the early IR tests included the identification of primary and secondary variables in retrieval systems and the validity and clarity of the hypotheses underlying the IR tests. One major concern throughout was the standard of the experiments, in terms of collection size. Most early IR tests were done by using very small collections and very few requests which led to the idea of developing specifications for the 'Ideal Test Collection' (Sparck Jones, 1975; Sparck Jones & van Rijsbergen, 1976; Sparck Jones & Bates, 1977).

During the 70s one of the main research aims was to compare partial match (statistically-based) techniques. Research on relevance weighting (Robertson & Sparck Jones, 1976), especially that by Sparck Jones in the late 70s and early 80s, demonstrated that it is a realistic measure of optimal performance. All this work was done on test collections, of which only some were rather large in size, and however well the simulation of a search might have been it was never like the real situation. Spark Jones felt at that time (circa 1980) '...that the required next step in this line of work was to carry out real, rather than simulated, interactive searching, to investigate the behaviour of relevance weighting under the constraints imposed by real users...'[6] (Sparck Jones, 1988, p.17).

Moving towards the operational-environment end, of testing relevance weights, is not an easy task. Carrying out non-matched paired experiments to compare relevance weights with Boolean searching, like the CIRT experiments, can be very costly, would require very large samples (Robertson, 1990a), and there are difficulties in evaluating different strategies (e.g., ranked-unranked output).

Considering interactive searching with real users in an IR test would introduce all these problems that laboratory tests were trying to avoid. There would be problems in the experimental design, e.g., increasing the emphasis on the user-system interaction reduces repeatability. Taking real users throughout means we can have only one retrieval strategy per information request which consequently would require large samples for achieving any statistical significance. Nevertheless, as Sparck Jones and many other researchers have advocated, this seems to be the right way to move and the research described in this thesis follows along these lines. Since laboratory tests on relevance weighting have come a long way and have proven its worth under such conditions, it can also be argued that this is the right time to take up such investigations.

The research presented in this thesis is an investigation of interactive query expansion. It made use of real users with their real requests in an operational environment. The INSPEC database, on Data-Star and ESA/IRS, was searched online using the CIRT front-end system.

[6] However, she did not pursue this line of research any further. A brief explanation of the reasons that led her to this change of research interests is given in the cited reference.

Chapter 3

Interaction in information retrieval

The overview of DRS presented in the preceding chapter covered issues, such as retrieval techniques, that form what might be seen as the core of a DRS. Another very important aspect in IR is the user-system interaction which is the focus of this chapter.

Online retrieval of whatever sort allows the searcher to make a stab at a search, and then try again if the results are not satisfactory. Nobody wants to go back to the situation of batch searching, where the user had only one attempt in a reasonable timescale. This paradigm is made explicit and extensively analysed in Belkin and Vickery (1985). Nevertheless, as seen in the review by Belkin and Croft (1987) much of the theoretical work on IR contrives to ignore the interaction process.

User-system interaction is discussed below in relation to the search formulation and search reformulation stages of the online search. Query expansion, including some specific approaches, is also taken into account and intertwined in a non-technical manner in the discussion.

3.1 Intermediary mechanisms

The interaction in IR is complicated by the existence of various kinds of intermediary mechanism (human or machine). Many IR situations can be conceptualised in something like the form indicated in Figure 3.1. A simple two stage model describing it includes the end-user, the intermediary mechanism, and the 'raw' database (Efthimiadis & Robertson, 1989). This intermediary mechanism may be a human being or some software, for example, a user interface, front-end system or expert system. There may also be more than one intermediate stage.

If we consider the interaction possibilities in the diagram, the picture becomes very complex. There may be interaction between the end-user and the intermediary, without reference to the database; interaction between the intermediary and the database without reference to the end-user; or interaction that spans all three parties. In this last case,

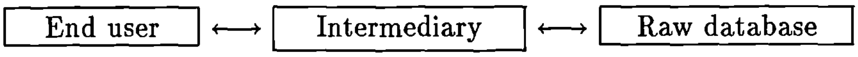


Figure 3.1: The two stage model.

messages from the end-user may or may not be interpreted by the intermediary before being transmitted to the database; and similarly with return messages. 'Interpretation' may mean anything from a slight re-arrangement to a complete change of form, syntactically and/or semantically.

Much recent research is informed by the perception that some (human or machine) intermediary is desirable, specifically to encourage interaction and feedback. It is usually suggested that this desirability is simply the result of the inadequacy of the 'raw' database, though the conceptualisation of IR as involving an intermediary mechanism seems to represent some deeper recognition of the centrality of feedback.

If we take a narrow view of what constitutes the IR system in a given IR situation, it would probably be seen as the 'raw' database in the terminology of Figure 3.1, or the retrieval mechanism in the terminology of Belkin and Croft (see page 10). But a general systems theory approach would suggest taking at least the whole of Figure 3.1 as the 'system' to be examined. In that view, the user would not be seen as an independent entity, but as part of the system, to be accounted for in any model of the system.

But looking at the situation from the user's point of view, it is clear that the user sees the IR system as something outside her/himself, which s/he approaches and from which s/he expects something. In this view, the system must be everything in Figure 3.1 except the user (but including the interaction between the intermediary and the user).

For the IR theorist, it is essential to distinguish his own model of a system from the user's model of the system or the systems design model of user activity. If one takes into consideration the model of communication described in the Belkin & Vickery review, in which each participant has a model of the other, then we can perhaps take the user's model of the system to be itself part of the system. Thus the boundary of the system (that is, the system of concern to the IR theorist or designer) would occur somewhere within the mind of the user.

In this chapter *feedback* refers mainly to information crossing the boundary of the system (in either direction). But within the total system under consideration there are subsystems, as indicated in Figure 3.1; feedback between the subsystems may also be of concern. This definition of feedback is a broad one which encompasses the very stylised form of feedback as expressed by 'simple relevance feedback' which was briefly discussed in chapter 2 and will be examined in more detail below.

3.2 User-system interaction

The interactions which may take place between a human user and a machine system can now be considered. In terms of the two-stage model, any mechanical intermediary, is part of the machine system. However, a similar analysis would apply to the interaction between a human intermediary and a raw database.

In this interaction several levels of complexity might be identified. Examining first the information going from the machine system to the user, systems may provide information about their own facilities. Examples include error messages, command menus and help screens. They may provide subject-related information, such as menus of subjects or thesaurus or dictionary extracts. Finally they may provide information deriving from an actual search. This last category includes partial or full records, and statistical information such as number of postings. Some types of information may come into more than one category: e.g., an analysis of term occurrences in a retrieved set comes into the second and third groups.

Users, on the other hand, may provide commands summoning particular system facilities. They may provide subject descriptions (terms, phrases, natural language sentences). Finally, they may provide selections from explicit alternatives presented to them. This last includes selections from command menus, selections from subject menus, and responses to offered items or terms. Again, a particular user response may combine categories — e.g., a natural language statement may contain commands as well as subject terms.

Any of the above may occur at any stage in the course of the user-system dialogue and require the user to make decisions based on information presented to her/him by the system as a result of the interaction. Demands made upon the user's intellect determine how trivial or complex the nature of the responses might be. The levels of complexity of the interaction depend on the ways in which

1. users may use system-generated information to help them reformulate some aspect of their problem;
2. systems may provide information intended to prompt or help the user to reformulate;
3. part of the reformulation process may be included as a system function, and the system may therefore invite user responses in an appropriate form.

The above description is phrased as though the 'system' (in the sense defined above) is a machine. If that part of the system visible to the user is human (e.g., a human intermediary), it is likely that a more complex analysis of information types is required (see e.g., Brooks, Daniels and Belkin (1986)). If a machine, then any query reformulation within the system can be regarded as a form of machine learning. The distinction between 'knowledge-sparse' and 'knowledge-rich' learning (Rada, 1987) is a useful one, implicit in much of the discussion below.

3.2.1 Interaction in Boolean systems

Consider the form of feedback given in the interaction with a commercial host at a typical online session. The host computer would usually provide error messages, help facilities, and perhaps command menus.

Error messages are usually terse messages in response to a user command that the system cannot recognise or respond to. An error message may be just a number, e.g. E1234, which means nothing to the user apart from the understanding that something has gone wrong. More often, it would be a 'message in natural text' which is supposed to be self-explanatory. Thus, the error message may look like 'E1002 Unrecognised command - reenter or return to continue with last command' in Data-Star or 'Invalid Argument' in ESA/IRS. So, if the computer does not recognise a command it will always respond in the same way.

Help facilities in online hosts, if they exist, tend to be unsophisticated and not at all context-dependent. For example, Data-Star does not have an online help facility, whereas ?, ?EXPLAIN, HELP ?, ?HELP, .HE, would invoke the help facilities in ESA/IRS and Infoline, Dialog, Blaise, Dimdi, and Télésystèmes/Questel respectively. This will introduce a 'master help menu' from which the user chooses the topics for assistance. (Usually, unless you know the exact name of the topic where you need help you have to refer back to the master menu each time.)

As a contrast to the traditional online IR systems come the help facilities found in online catalogues. These are context-specific although still specific to system-context. For example, the CLSI OPAC offers different help facilities depending on which module of the system and/or stage of the search the user is in. So, if the user asks help before starting a search s/he will get different help information from that if s/he asks help in the middle of, say, a title search. However, this context-dependence relates only to the context as perceived by the system. A request for help after a keyword search will produce the same screen whether the search produced 1000 items or none at all.

Within the online interactive session different support is necessary and required at different levels (Norman, 1984). Studies have shown that help facilities in interactive information retrieval systems are often inadequate (Trenner, 1989). Thus, a more careful design of menus (cf. Shneiderman (1986)) and of the user-interface (cf. Draper & Norman (1985)) would have beneficial effects on the user-system interaction.

On the whole, from the above description of system facilities it is clear that these at present provide only a rudimentary form of feedback to the user. This affects only the user's model of the system and therefore, should not be regarded as feedback in the sense defined so far.

3.3 Search strategy: some definitions

Search strategy development, i.e., the plan devised to deal with the whole search on a requested topic, is the most intellectually demanding aspect of the online search and it is the central subject of discussion in many textbooks and journal articles (see page 10).

Good search strategy development involves the use of one's knowledge about online searching systems, indexing vocabularies and conventions practiced in text database construction. In other words, it requires a good understanding of the DRS model described on page 7. It also requires a full understanding of the information need(s) and an ascertaining of the search objectives, e.g., high recall or high precision. The result of such analysis will eventually determine the subsequent search formulation, which is the statement or set of statements which express the necessary query in a form understandable to the online system.

A significant decision to be made during the search strategy formulation is the correct decomposition of the information need and the identification of the key concepts or facets.[1] Then the choice of a particular search strategy would determine the way the concepts should be combined and would also suggest possible alternative actions.

Search strategy formulation is a highly unstructured problem and it requires a broad range of knowledge (e.g., knowledge of the user's specific problem domain, knowledge of the DRS and its constituent parts, and other common-sense knowledge as well). Therefore, although it is being studied systematically it is still not a well understood process. Consequently, the search formulation process is difficult to automate.

Thus far, machine intermediary mechanisms have aimed to help users by automating the mechanical parts of the search with no attempt to incorporate and use the expertise and search skills of the human intermediary. Recent research efforts are attempting to embody some of this expertise into intelligent front-ends with the use of artificial intelligence and expert system techniques. The discussion on query formulation and query reformulation below concentrates mainly on interfaces for text retrieval.

The online search (for reasons of simplicity in facilitating the discussion) can be reduced to two stages: *initial formulation* and *reformulation.* At query formulation stage, the user first constructs the search strategy and submits it to the system. At the query reformulation stage, having had some results from the first stage, the user manually, the system automatically, or the user with the assistance of the system or the system with the assistance of user try to adjust the initial query and improve the final outcome.

3.3.1 Query expansion

Query expansion may take place in either the query formulation, or query reformulation stages of the online search, or both. Query expansion, i.e., the process of supplementing the original query terms with additional terms, can be done automatically by the system, manually by the user or semi-automatically. (More generally, query modification may involve deletion of terms as well.)

Query expansion, if considered strictly along the lines of the above definition, is not a form of feedback. One could envisage a system which puts query terms through some completely automatic process, which resulted in new terms being added or additional matches being allowed. Indeed, such systems exist; an example would be an automatic suffix stripping algorithm applied to any user-supplied terms, such as that in CUPID

[1] The two terms, i.e., *concepts* and *facets*, are used interchangeably in the discussion below.

(Porter, 1982). Another example would be an automatic matching of natural language terms against thesaurus entries, without reference back to the user, as is sometimes effected in Plexus (Vickery *et al.*, 1986; Vickery *et al.*, 1987). In such a system, the expansion can be treated as part of the retrieval mechanism.

But query expansion may also either be a form of feedback to the user, or be the result of a feedback operation, or both. A manually performed query expansion could involve the various tactics (or moves) suggested by the work of Bates (1979a; 1979b; 1987), Fidel (1985), and Harter and Peters (1985).

Some general considerations apply to any form of query expansion. In particular, it is appropriate to ask the source of any extra terms supplied by the system. Possible sources include:

- a knowledge structure such as a thesaurus;
- an algorithmic process such as a suffix-stripper;
- characteristics of the collection of documents, such as term clusters; or
- terms extracted from documents retrieved in an earlier iteration of the search.

The first three will be regarded in what follows as examples of *knowledge structures* (taking the phrase broadly), independent of the search process. The last example clearly relates to a particular kind of feedback process which is central to the research presented in this thesis.

3.4 Query formulation

It has long been recognised that in the query formulation and query reformulation stages the knowledge structure of the intermediary mechanism (human or machine) plays a significant role for the success of the online search. But, a Boolean system has nothing that could be described as a dynamic cognitive structure (Ingwersen, 1984), therefore any evolution in the interaction has to involve changes in the cognitive structures of the human beings. This is fine in the case of a professional human intermediary because the model is determined by education, training and the development of skills and experiences, both social and individual. The problem arises in the case where either the intermediary mechanism is a machine, or there is none (the end-user interacts directly with the raw database). Leaving aside the end-user temporarily and concentrating on the machine we could say that in most present machine-based intermediary mechanisms, the model would be static or passive depending on the mechanisms' level of sophistication.

An example of a relatively sophisticated approach to this modelling appears in the work of Belkin *et al*, who start by investigating the functions carried out by intermediaries (Brooks, 1986). They propose to proceed by building a number of different expert systems, each dealing with one aspect of the intermediary's knowledge structure, and all integrated as a distributed expert system based interface which would simulate the behaviour of the human intermediary (Belkin, Seeger & Wersig, 1983). A number of Distributed Expert Based Information System (DEBIS) prototypes have been developed. These are based on the MONSTRAT model but have implemented only some of the expert functions, e.g., I^3R (Croft & Thompson, 1987), and CODER (Fox, 1987).

In current approaches to search formulation and search reformulation, we can see some help being offered in the form of query negotiation aids at one end (e.g., human intermediary: presearch interview; user interface design: form filling, menu selection, question-and-answerdialogue, and quasi natural language dialogues) and in the form of query expansion at the other end (e.g., manual, semi-automatic, automatic).

Search formulation is a quite difficult task, even for trained intermediaries, and there have been various attempts to deal with it at the interface level. Such attempts include graphical representations, e.g., I^3R (Thompson & Croft, 1989), hierarchical menu displays, e.g., MenUSE (Pollitt, 1988), automatic matching of natural language terms against thesaurus entries, e.g., Plexus, Tome Searcher (Vickery *et al.*, 1986; Vickery, 1988).

On the whole there seem to be two approaches in search formulation as being employed by front-ends and ES. The first is automatic search formulation and reformulation (Efthimiadis & Robertson, 1989), where the concern entirely lies with mapping the user's input into appropriate search terms, constructing a search strategy and then carrying out the search automatically, e.g., EXPERT (Yip, 1981), CONIT (Marcus, 1983), CIRT (Robertson & Bovey, 1983). The second is to provide assistance in search formulation rather than to take over completely. Here the role of the interface is that of the "advisor", i.e., to monitor and advise the user during the search process. Apart from IIDA (Meadow, 1979), which is more of a tutoring system, little has been done in this domain. This is mainly due to the vast subject domain diversity covered by DRS. Shoval (Shoval1981, ; Shoval, 1985) describes a system designed to assist users in query formulation by suggesting terms using a thesaurus. Prototypes like CALIBAN (Frei & Jauslin, 1983) and CoalSORT (Monarch & Carbonell, 1987) utilise classification schemes and multi-window techniques to present terms to the user who then manually constructs a query. CANSEARCH (Pollitt, 1987) and MenUSE (Pollitt, 1988) use a classification scheme (MeSH) and are oriented directly towards producing a query from the user selections without bothering the user with the search mechanics involved. I^3R, to assist the user adds browsing to the above techniques.

The initial query formulation stage is being addressed in some ways by most IR systems. The forms the query formulation stage can take are often dependent on or determined by the dialogue mode employed at the user interface: for example, menu selection as in MenUSE (Pollitt, 1987), natural language input as in Plexus and TomeSeacher (Vickery, 1988), system prompts as in CAN/OLE (Lamb, Auster & Westel, 1985), a graph representing subjects (McMath, Tamaru & Rada, 1989), or from a non-typing interface (such as a touch screen) (Pollitt, 1981). The query reformulation task is left entirely up to the user in the case of the less sophisticated interfaces, whereas the intelligent interfaces try to tackle the problem in a number of ways which are determined by the retrieval techniques used and the knowledge structures available to hand (Efthimiadis, 1990).

Most, especially commercial, interfaces, whether intelligent or otherwise, use menu selection or command interaction styles for query formulation. Research prototypes and recent in-house developed applications, on the other hand, tend to make use primarily of natural language dialogue, graphical and direct manipulation interfaces.

In general, interfaces of commercial online retrieval systems use Boolean logic and command dialogue styles. The user, in order to interact with the system, must be familiar with the Boolean logic, as well as the command language used by each particular system. In command driven dialogues users must enter their formulations in ways based on their knowledge of the required syntax of the host command language or the CCL used. An alternative approach used in some systems is the input of bare search terms from the user and with the system automatically formulating the Boolean statements. Interfaces of commercial hosts that allow such input of search terms then use Boolean or positional operators to connect the terms. For example, Knowledge Index and ProSearch treat a space between terms as implicit adjacency and After Dark and BRKTHRU as 'OR'. Multiword terms could result in the retrieval of too many or too few references. A compromise in the use of these techniques, i.e., using 'WITH' in BRS and 'N' in Dialog could be proven to be more effective as demonstrated in an interface developed at the University of Illinois (Tenopir, 1988) (see also Mischo (1986)). More sophisticated implementations are found in CIRT (Robertson *et al*, 1986), Questquorum (available on ESA/IRS) and in systems described by Heine (1982; 1988), Radecki (1988) and Salton (1988).

In a menu-driven dialogue users build their request by entering their queries as independent facets in a step by step manner. In some cases the system requires that the user is somehow familiar with the mechanics of the search, in SciMate users have to select Boolean operators themselves (Lamb, Auster & Westel, 1985). In others the system asks the user how the entered terms are to be related to a subsequent entry. A common method used in query formulation by many systems is to lead users through a series of menus and assist them in splitting their query into concept categories. The user is then prompted to enter terms into the concept groups. Terms within each concept group are to be OR-ed and then the concept groups are AND-ed. CONIT (Marcus, 1983), EXPERT (Yip, 1981), IT (Williams, 1985), OAK (Meadow *et al.*, 1989) have all used this method in an explicit manner, i.e., the user creates the facets by him/herself. In TomeSearcher the method is used without user intervention (mainly because of the natural language input) but the concept groups are presented to the user for confirmation before being searched. Some systems also assist in term selection and concept development. Another query formulation method with menu selection is demonstrated in MenUSE (Pollitt, 1988), where the user does not have to type terms but to choose, with the use of a mouse, subjects from a list of subject headings.

Natural language interfaces for bibliographic information retrieval systems are still at an early stage of development despite the 30 odd years of continuous research efforts. Some reasons may be suggested for this state of affairs. Firstly, natural language processing is still very much a research area in linguistic based disciplines and in AI. Secondly, the domain itself is of very different scope from that of facts databases, which usually have very narrow subject domains. Thirdly, the effort involved in the development of natural language interfaces is substantial.

CITE and OKAPI do not employ syntactical analysis techniques to process the user's input. They concentrate on automatic vocabulary mapping by stemming the words and then use 'best-match' retrieval techniques.

IOTA parses the user's natural language query and produces a Boolean expression of user concepts (Chiaramella & Defude, 1987). IR-NLI's natural language interface has modules for reasoning, by using domain specific knowledge, and for understanding and dialogue, by using linguistic rules and a dictionary (Brajnik, Guida & Tasso, 1986). The natural language processing facilities are complemented by using the approaches and tactics for search strategy design described by Bates and Meadow and Cochrane (Bates, 1979a; Meadow & Cochrane, 1981). I^3R seems to be the most comprehensive of all systems mentioned so far (Croft & Thompson, 1987; Thompson & Croft, 1989). The user's natural language query is processed to extract terms from it. The user is then asked to highlight words that are important and to enter any connection that may exist between query terms. If the user has used I^3R before, the system would then examine the model built for the user, i.e., from previous uses, for possible relevant concepts, and then it would conduct the search. Otherwise, the system begins to build such a model.

3.5 Query reformulation

3.5.1 Simple relevance feedback

Apart from the manual/intellectual query reformulation where the task falls to the searcher it is possible for the system to take over this task entirely requiring only some yes-no answer from the user. This automatic query reformulation process is called relevance feedback (Salton & Mc Gill, 1983). Its aim is to improve the retrieved set by removing unwanted documents and adding more wanted documents without the user consciously constructing new search strategies, and by using relevance or nonrelevance information obtained from the user.

The typical automatic relevance feedback operation involves an initial search with a user-supplied query and an initial retrieval of certain documents. Then, from a display (usually of titles or abstracts of the retrieved documents) the searcher identifies/chooses some relevant documents. Those documents are used to modify the query by reweighting and/or adding terms that appear useful and by deleting terms that do not. This process creates a new query which resembles the relevant documents more than the original query does.

If we consider the information involved in the user-system communication in simple relevance feedback, it goes like this:

(a) user gives system initial query;

(b) system gives user document description;

(c) user gives system relevance judgements.

The difference in the form between (a) and (c) suggests that there must be some kind of intermediary mechanism involved, since a raw database would normally accept queries in one form only. This intermediary mechanism may have to transform the initial query into a suitable formulation for searching, but will certainly have to transform the relevance information into such a form, and then combine the two kinds of information.

In its simplest form feedback could be based on one document only. For example, after displaying a single document, the system could invite the user to see more documents like the one on display. Here the intermediary mechanism, e.g., in an OPAC, could use the classification scheme and present the user with books of the same class-mark as the first viewed. (There is no system restricted to this simple method known to me.)

Another simple form is when judgement is based on a set as a whole (as apparently suggested by Pietilainen, 1983). Here, a set derived from a previous search becomes the seed for the new query formulation. The method uses 'searchonyms' (Attar & Fraenkel, 1977; Attar & Fraenkel, 1981), i.e., terms which might be regarded as synonyms for the purposes of a particular enquiry and derived from terms contained in the seed set.

Automatic feedback, generally, could be implemented in various ways depending on the retrieval technique used, e.g., vector space, probabilistic, etc., and also on the methods used to select terms for the feedback query. We could distinguish four term selection methods for query reformulation.

- The first relies entirely on the original query and uses only those terms in the new one (Robertson & Sparck Jones, 1976). This method has been successfully implemented in CIRT (Robertson & Bovey, 1983; Robertson *et al*, 1986).
- The second method uses terms from the original query and also adds terms from some other source, e.g., from all the adjacent terms in the maximum spanning tree (MST) (van Rijsbergen, Harper & Porter, 1981) or nearest neighbour (NN) terms (Smeaton & van Rijsbergen, 1983).
- The third method is a mixed method because it uses combinations of the terms derived from the original query and from the documents retrieved and judged relevant as found in the work of Salton and his colleagues ((Wu & Salton, 1981; Salton, Fox & Voorhees, 1985)) and in the OKAPI online catalogue (Walker & De Vere, 1990).
- Finally, the fourth method abandons the terms from the original query and uses only terms found in the retrieved set of documents (Dillon & Desper, 1980; Dillon, Ulmschneider & Desper, 1983).

In all cases, after the initial query formulation, the only form of feedback to the user is items, and from the user is choices of items. The query reformulation is entrusted entirely to the intermediary mechanism.

3.5.2 Interactive query definition & expansion

This section considers methods where the user is offered search terms as part of the reformulation process. These could be based either on some form of knowledge structure (such as a thesaurus), or on the results of an earlier search.

3.5.2.1 Based on a knowledge structure

The knowledge structure of the intermediary mechanism, as mentioned earlier (section 3.4 plays a major role in the interaction and is clearly of great importance for the success of the search.

However, for most present machine-based intermediary mechanisms, the model is static or passive depending on their level of sophistication. For example, the commercially available intermediary systems like Sci-Mate, Pro-Search, Easy-Net, etc. have automated only the mechanical operations of the online search (Hawkins, 1988). At best, with a fixed or passive knowledge structure (Ingwersen, 1984), the intermediary can select part of that structure to present to the user, thereby inviting the user to choose one or more elements.

Interpreting the phrase in a very broad way (as described above), the knowledge structure could be based in some way on the collection, or it could be independent of the collection.

Collection dependent

Examples of feedback based on the collection are the EXPAND or ROOT commands (available in commercial hosts) and the INSTRUCT term-clustering and morphological expansion modules. The EXPAND and ROOT commands provide a form of feedback from a knowledge structure of the 'raw' database which is the dictionary file. The user is given an alphabetical listing of descriptors and free-text terms to choose from and modify her/his query formulation. The INSTRUCT term clustering technique (Wade & Willett, 1988) identifies keyword stems which are most similar to the query term stem. The morphological expansion (Freund & Willett, 1982; Hendry, Willett & Wood, 1986) calculates a measure of string similarity (measured in trigrams) between a selected query stem and each of the stems in the dictionary file of the database. Then, in both cases, the system displays the twenty most similar stems to the user who chooses the ones to be added in the query.

Collection independent

Query expansion examples based on a knowledge structure which is independent of the collection are found in CITE and in expert systems which use thesauri in one way or another. CITE automatically performs stemming on words entered by the user, then selects medical subject heading (MeSH) terms and terms from the dictionary file by matching the user's words against them. Then it ranks the selected terms and displays them to the user for approval and feedback before commencing the search (Doszkocs, 1983).

In MenUSE (Pollitt, 1988) the emphasis is on the structure of the MeSH thesaurus. The user identifies concepts by starting from the top of the hierarchy and going down the tree till the appropriate entry is recognised and selected. The same process is repeated by the user for each of the concepts. Then the system combines them and presents a review to the user. Another system, which also takes advantage of the MeSH thesaurus, uses a graphical interface to achieve similar goals (McMath, Tamaru & Rada, 1989). The hierarchical structure of MeSH is graphically displayed and the user can move about the hierarchy with the help of the mouse. Queries can be formulated by selecting terms from the hierarchy and placing them in the query window. Multiple windows allow the user to simultaneously see the thesaurus, a histogram of document rankings and the document descriptions themselves.

All expert systems which have pre-search aid modules, such as CANSEARCH, CONIT, etc., although they use knowledge structure independently of the collection and search results, help in the suggestion of terms. Some of these systems maintain a thesaurus as part of their structure while all provide some means for the identification of terms.

EXPERT (Yip, 1981) helps the user in building a query formulation by asking the user to split the topic into concepts and then to suggest terms to express each one of them. The system builds the query by developing these concepts and then it prints them out in a columnar manner with each column containing the terms that form one concept. Thus it has no thesaurus (or any form of semantic knowledge structure); its knowledge structure could be described as purely syntactic, relating to the structure of typical simple Boolean queries.

CANSEARCH (Pollitt, 1981; Pollitt, 1987) uses a subset of the MeSH thesaurus, together with its own knowledge of how cancer queries tend to be structured, to help the user identify terms and incorporate them in the query. TomeSearcher (Vickery, 1988), processes its natural language query input and uses the INSPEC thesaurus to supplement the terms and expand the query. A system specifically designed to assist users in the selection of query terms has been developed by Shoval (1985). Its knowledge base consists of a thesaurus enhanced by additional information about terms, which is organised as a semantic network. The user inputs search terms and the system searches its knowledge base, selects terms that seem to be relevant, evaluates them and presents them to the user who chooses which are useful for the query.

3.5.2.2 Based on search results

An alternative method for assistance in the search process is to present the user with information based on the results of the search. User-system interaction is on various levels of sophistication. For instance, the system presents to the user a list of terms based on their occurrences in an identified set of documents. Then the user feeds back his or her choice of terms. The document set on which this analysis is based may either be simply a set retrieved in the usual fashion (and chosen or accepted by the user as a *suitable set* for this purpose), or it may consist of documents individually selected as relevant by the user.

The ZOOM (Martin, 1982; Ingwersen, 1984) feature on ESA/IRS is a helpful tool for online searching along these lines. It performs term frequency analysis on a number of records from the retrieved set(s). The user is then presented with screen-displays which contain terms in a frequency ranked order. The searcher selects terms which then can use to expand the query. ESA/IRS offers also Questquorum (D'Elia & Marchetti,) as a simple interface to its command driven system for inexperienced users which can also do a semi-automatic query expansion based on terms selected by the user from a ZOOM-like display.

There are also some other systems that have utilised the term frequency analysis function either in a ZOOM-like way or in some other form. IT (Information Transfer system) (Williams, 1984) uses the command EXPLORE to retrieve, organise and present index terms to the user, for addition to the search profile. Analogous is the command TERMS in the MUSCAT online catalogue (Porter & Galpin, 1988). It presents terms and UDC numbers which are extracted from documents inspected and judged relevant and which were not included in the original query. With each term presented the user is asked whether or not to add it to the query. CITE (Doszkocs, 1983) utilises the user feedback also in a similar manner. It automatically performs term frequency analysis on the records marked as relevant, and then it presents the terms in ranked order to the user for selection.

EUREKA (Burket, Emrath & Kuck, 1979) is an experimental full text retrieval system which uses a user specific thesaurus (there is not a system-wide one). Each user can create and maintain a personal thesaurus which is used by EUREKA at search time to find synonyms for the query terms. As additional user aids EUREKA can present on demand either a histogram of term frequencies based on the retrieved documents, or word-lists of terms that are used in many documents or have high average frequencies. From these lists the user selects terms to refine the retrieved set.

The MICROARRAS full text retrieval engine (Smith, Weiss & Ferguson, 1987) retrieves text passages and then informs the expert system (Gauch & Smith, 1989) of the results that satisfy the request. The expert system evaluates them and decides whether and how to reformulate the query. Query reformulation is done by using a thesaurus (i.e., adding terms from the thesaurus to the search terms), by adjusting contextual constraints on the Boolean operators (i.e., alternating the operators from the most to the least restrictive, as for example in the sequence 'ADJ, WITH, SAME, AND, OR'), and by replacing Boolean operators. These techniques can be employed individually or in any combination.

Feedback based on the result of the search and with a dynamic form of search modification is given by Thomas (Oddy, 1977) which treats the document collection as a thesaurus and lets the user browse through it. The user starts with a simple natural language expression of interest, and the program responds by presenting a document representative (i.e., title, author(s), index terms) and asks for the user's judgements. At this point the user can make negative or positive judgements, on the document as a whole, and/or on any index term(s) or author(s) listed. Then Thomas iterates the search while keeping track of its actions by creating a model of the search in the form of a network.

In I^3R (Thompson & Croft, 1989) browsing begins after the user has picked an item as the starting point. The system provides assistance to the user by suggesting paths that should lead to relevant information. Recommendation is given to guide and not to restrict the user options. This is reflected by the displays on the neighbourhood and context maps, which consist of nodes representing concepts, documents and journal issues connected by links. In the centre of the map is the document the user selected as relevant. The surrounding nodes are documents that the system has decided are likely to be interesting. When the user selects a node the document that it represents appears on the screen and the user can mark it as relevant or ignore it. After a number of documents have been judged relevant I^3R will re-iterate the search.

Finally, there are a number of IR systems using AI techniques with the aim of improving their interaction style and effectiveness (Smith, 1987; Vickery and Brooks, 1987; Hawkins, 1988). Some of them use as their knowledge-base a highly structured thesaurus, as described above. Also, some have incorporated knowledge of search heuristics for choosing alternative search strategies, and use browsing and other techniques to help the user choose terms and reformulate the query. Search heuristics drawn from the work of researchers such as Bates (1979a; 1979b) and Fidel (1985; 1986) have been applied in IR-NLI (Brajnik, Guida & Tasso, 1986) and PLEXUS (Vickery *et al.*, 1987). According to the users responses, PLEXUS evaluates the search results then reformulates the query and resubmits it to the DBMS for processing. The result is again assessed and the process repeated if necessary.

It will be clear that a reasonably full implementation of Bates-type tactics, for example, would require use of search result information. However, many of the expert systems are implemented as pre-online-search aids, and therefore depend on the use of other knowledge bases (as discussed earlier).

Chapter 4

Query expansion

Query expansion as defined in section 3.3.1 is the process of supplementing the query with additional terms. The rationale behind this process is that query expansion can be considered as a method for improving performance. The method itself is applicable to any situation irrespective of the RT used. The initial query (as provided by the user) may be an inadequate or incomplete representation of the user's information need, either in itself or in relation to the representation of ideas in documents.

Before going online

In the traditional online search environment the searcher must break down the request into some distinct concepts. Then the searcher must think of how the concepts and the term(s) associated with them correspond to the document representation stored in the database.

Once the terms have been chosen then these can be combined to form the query. However, the mere realisation that one term per concept might sometimes be inadequate to (accurately) express a concept and the effort to find terms to complement the first chosen term is an instantiation of query expansion. The situation requires a change in the searcher's thinking process for choosing terms. It may be necessary to consult a thesaurus, a list of subject headings, a dictionary, or a classification system and its index to aid in the choice of terms. This usually requires specialised training or experience on the part of the searchers, as it foretells poor results for the inexperienced and those who do not consult the search aids.

While online

Another way of getting ideas to choose terms is by browsing the dictionary index of the database that is being searched or the cumulative dictionary index of all the databases available in the retrieval system, like for example Dialindex in Dialog.

This of course requires imagination and experience, on behalf of the user. Choosing terms from unstructured thesauri, dictionary indices, or alphabetic word lists is not an easy thing to do because these lists are usually created as the sum of all the words in all databases that are available in a particular host retrieval service at that time.

4.1 The curse of dimensionality

The *curse of dimensionality*, which was adopted in information retrieval from the pattern recognition literature (van Rijsbergen, 1979, p.130), refers to the problem of selecting a set of search keys which can be legitimately used to predict on relevance.

More specifically this problem refers to the situation whereby a decision should be made as to which terms should be used in query expansion. In the IR situation we are cursed with a high dimensionality term space, i.e., large number of attributes, which in this case are the terms. In other words, we are faced with the complete set of index terms in the document collection, upon which to base the decision and which makes such a decision nearly impossible. A natural way of dimensionality reduction in IR is the acceptance that the query terms themselves are good guides for suggesting the terms to be used by the system for predicting relevance.

In a manual system dimensionality reduction is controlled as well as achieved in a number of ways by the searcher, who, as has been mentioned so far, uses the initial query terms which are then clustered together with other closely associated terms. Decisions upon the association of terms lie entirely at the discretion of the searcher and are usually determined by the quantity and quality of the knowledge structures held by the searcher as well as the availability and use of searching aids.

The *association hypothesis* (van Rijsbergen, 1979, p134) states that:

> "If an index term is good at discriminating relevant from non-relevant documents then any closely associated index term is also likely to be good at this."

If this hypothesis is accepted then in the case of automatic query expansion, in a weighted or associated retrieval system, the problem of dimensionality reduction is being taken care of by computing the weighting function only for the terms specified in the query and maybe their close associates rather than for all possible terms. An important issue, however, is how one defines which are the terms that are closely associated to the query terms.

4.2 Automatic Query Expansion

There is a considerable amount of laboratory experimental work on systems that incorporate some sort of automatic query expansion. However in many cases it is difficult to decide how the query expansion process itself is taking place as the expansion process is *hidden* within the overall process of information retrieval. Such systems often employ weighted or associative retrieval techniques. An example of this can be seen in the SMART system with the use of normalised vectors of information retrieval (Rocchio, 1966; Salton, 1971). Queries and documents are represented by weighted numeric vectors and for retrieval the query vectors are matched against document vectors. The system operates a relevance feedback mechanism and the original query is modified with each iteration of the loop by adding term vectors for all retrieved relevant documents by a given cutoff point and subtracting term vectors for all retrieved non-relevant documents by that cutoff point. This effectively added many terms, some with negative weights. It also increased the weights of some query terms and decreased the weights of others. However, the results were not conclusive because of the general problems encountered in evaluating relevance feedback (for an introduction to the way these techniques were used in SMART see (Salton & Mc Gill, 1983, pp215–243 (clustering) and pp284-287 (syntactic analysis)). Given therefore that there is, firstly, a large number of experimental systems that incorporate query expansion and modification techniques and, secondly, that it is often difficult to separate the expansion process from the overall retrieval process, I intend in this section to look only at a selection of the ideas in the area of automatic query expansion. Another related area for query expansion is the work done on natural language processing and syntactic analysis. This, although interesting, is a very large area and will not be discussed further in this thesis.

During the late 60s and early 70s, query expansion was investigated a great deal. At that time the emphasis was concentrated in clustering single index terms together (using different techniques) before the user submitted a query by constructing a similarity matrix of terms. From this matrix groups of similar objects were identified based on some definitions of what cluster types were being used. Queries were then expanded by simply adding clusters of related terms. Typical examples of this type of research is the work of Spark Jones (1971) and of Minker, Wilson and Zimmerman (1972), which appear to have rather conflicting views.

Sparck Jones did the most extensive work in the area of term clustering (Sparck Jones & Jackson, 1970; Sparck Jones, 1971; Sparck Jones & Barber, 1971; Sparck Jones, 1973; Sparck Jones & Bates, 1977). She formed groups of related word stems based on co-occurrence in documents. Her experiments explored a number of different clustering strategies and she found the effect of all these strategies relatively similar. For as long as the strategies were restricted to forming relatively small clusters their performance was best.

Sparck Jones (1971, p.208) reached the overall conclusion that the use of an automatically obtained term classification does enable better retrieval performance compared to the performance obtained with the unclassified terms alone. She suggested that this occurred when frequent terms were being left un-clustered, infrequent terms were grouped into clusters and strongly connected terms were clustered by applying a matrix threshold (Sparck Jones, 1971, p.195). This result was challenged by Minker, Wilson and Zimmerman (Minker, Wilson & Zimmerman, 1972) who evaluated the effect of adding cluster-related terms to queries. They concluded that the expansion of queries by cluster-related terms was at best marginally useful and generally produced worse results than un-expanded queries. However, Salton (1973) questioned Minker's experimental design. However, similar to Minker's results obtained by other researchers (Lesk, 1969) have for the most part also been marginal. Lesk (1969) concluded that manually constructed thesauri were superior to automatically constructed ones.

Sparck Jones' subsequent research provided rather pessimistic conclusions because retrieval performance was significantly improved by term clustering only on one small collection (Sparck Jones, 1973; Sparck Jones & Bates, 1977).

On the whole, the explanations given for the poor results from term clustering focused on the properties of the collections that were used to derive the clustering. In Lesk's case it was the small size of the collection used (Lesk, 1969), while Sparck Jones in a detailed analysis of her experiments concluded that the fault lay in insufficient differences in the vocabulary between relevant and non-relevant documents (Sparck Jones, 1973). She also stressed the need to use strongly connected non-frequent terms (Sparck Jones & Barber, 1971).

An additional explanation given by Lesk (1969) is that the clusters captured only terms whose meaning was purely local (to the small collection) and did not reflect their general meaning in technical text.

Sparck Jones (1971) also argued that clustering of words limits ambiguity instead of emphasising it. According to her a match on two words from the same class is very suggestive of which word sense is present. However, since matching on two words in the same cluster is much less common than matching on a single word, it seems unlikely that this is an effective disambiguation strategy. Since phrases (bound descriptors) are much less ambiguous than single words, they might be more effective.

The Online Associative Query System (OAQS) (Preece, 1980) is a model which has also had a partial implementation as a prototype system. OAQS modifies search queries (a) by incorporating thesaurus or dictionary linkages between terms and preferred or alternative forms and (b) by generating associations between sets of documents and the vocabulary they contain. The former allows the automatic inclusion in a query of pre-defined equivalent or alternate terms.

OAQS creates document nodes and term nodes. Each document node has links to all the terms contained in that document. Similarly, each term node has links to all documents that it has been assigned to. Thus, OAQS uses those links to provide a network view of documents and terms.

At retrieval, the initial query terms start triggering document nodes which in turn trigger term nodes, and so on. This process is repeated by identifying the documents best matching a set of terms and then the terms best matching that set of documents. The system uses inverse document frequency (IDF) for the weighting of the nodes and the retrieval of documents. The process can be completely automatic or may be controlled by the searcher.

Preece suggests that the simplest way to implement OAQS on a Boolean online retrieval system is to invert both sides of the database so that the list of terms in a given document is as readily accessible as the list of documents containing a given term.

The OAQS system is an application of the spreading activation technique. Similar network techniques are the probabilistic inference networks found in systems like CONSTRUCTOR (Fung *et al.*, 1990), RUBRIC (Mccune *et al.*, 1985) and its commercial offspring TOPIC and, for example, in the work of Croft and his colleagues (Turtle & Croft, 1990).

A procedure for the automatic adjustment of queries which could be employed on retrieval systems that use a structurised thesaurus for indexing has been described by (Vernimb, 1977). The procedure starts with the user having identified at least two relevant documents although as Vernimb suggests, searching effectiveness is improved the higher the number of known relevant documents. The system then examines the descriptors of these documents, which are 5-6 per document in this particular database. Then, it arranges them according to their frequency of occurrence in these documents, with the most frequently occurring descriptors first, and so on. Descriptors with the same frequency of occurrence, i.e., ties, are ordered in increasing order of their collection frequency (n). The one with the smallest collection frequency is placed first and that with the highest last, which according to Vernimb "...takes into account of the fact that the 'information content' is proportional to $ln\frac{1}{n}$ ..." (Vernimb, 1977, p.340).

Partial queries are then generated for each one of the relevant documents. These queries consist of the descriptors (DE) that index the document and which are combined with the AND logical operator. For example, if there are seven DE in the first document then the query is

$$DE1 + DE2 + \cdots + DE7$$

These partial queries are subsequently 'loosened' by stepwise omission of descriptors eliminating the lowest ranked descriptor first until at least 10 new documents are found. The partial queries are then combined in an OR statement to form the 'total query.' The next step involves a procedure for establishing the most useful partial queries based on relevance feedback of documents retrieved. A new query is formed and new documents are retrieved which after relevance judgements are made may be used for re-initiating the whole process. The procedure just described is wholly automatic and hidden from the user who only sees the retrieved documents.

Dillon and Desper (1980) and Dillon *et al.* (1983) have described an algorithm for automatically incorporating search terms into a query using a form of relevance weighting known as prevalence weighting. Positive and negative prevalence is computed based on the occurrence of terms in relevant and non-relevant documents that are retrieved from the initial search query. A number of threshold values for the prevalence weights exist so that groups of terms are assigned to a particular category depending on their prevalence weights. A new Boolean query is constructed by OR-ing together groups of terms according to their position in the prevalence category. Terms in the highest prevalence category are added (ORed) as single terms. Terms from the second highest category are ANDed as pairs of terms and so on. Finally bad (negative weight) terms are employed and these are NOTed. Any document containing one of these terms is not retrieved. Thus a query would be phrased as (Dillon & Desper, 1980, p.202):
a series of positive expressions (P_i) which are ORed:

$$(P_1 \; OR \; P_2 \; OR \; \cdots \; OR \; P_n)$$

a series of negative expressions (N_j) which are ANDed:

$$(N_1 \; AND \; N_2 \; AND \; \cdots \; AND \; N_n)$$

and the final query is the intersection of the above expressions:

$$(P_1 \; OR \; P_2 \; OR \; \cdots) \; AND \; (N_1 \; AND \; N_2 \cdots)$$

This process results in new queries being formed based on the relevance judgements of retrieved documents and according to the calculated prevalence weights. The terms of the original user query are not included in subsequent queries. The terms found in the retrieved document set are ordered by these weights and used in the construction of new Boolean queries as described above. So this process uses the terms in the initially retrieved set as a starting point, abandoning whatever information was present in the user query, and resulting in a complete query reformulation rather than query expansion.

CITE (Current Information Transfer in English), was created as an interface system to Medline (Doszkocs & Rapp, 1979). It allows for a natural language input of the query which is then translated by the system. CITE makes use of weighting and ranking, using inverse document frequency and it has a relevance feedback facility. Its query expansion capability takes the form of modifying the search query using the MeSH headings of the retrieved document set. The MeSH headings of all the selected documents are incorporated into the search query and combinatorial search is performed to retrieve similar documents. Doszkocs and Rapp also mentioned other methods of query expansion that may be implemented in the system, such as using both the additional free text terms from the title, abstract and MeSH headings; or using those free text terms and MeSH headings that appear in relevant documents exclusively, i.e., do not appear in non-relevant documents. These and other possibilities of automatic query expansion as well as a semi-automatic query expansion facility were implemented in subsequent versions of CITE (Doszkocs, 1983) and are discussed in the next section.

Recent versions of the OKAPI experimental online catalogue have been influenced in some ways by the interactional aspects of CITE (Walker & De Vere, 1990). At the query input stage, OKAPI uses a dictionary table of substitution terms which allows common synonyms to be searched automatically, i.e even if the user has not entered them. For example if 'Britain' is entered then OKAPI automatically includes Great Britain, GB, UK, United Kingdom, etc. (Walker & Jones, 1987). OKAPI expands a query by selecting the 'best' terms from a list containing the original query terms together with terms extracted from all the records which the user has judged relevant. Terms are weighted using a scheme based on the F4 formula and which gives a higher weight to terms that occur in more of the relevant document and a lower weight to those that do not. The list of terms is then sorted by descending term weight. Term selection for query expansion starts at the top of the list providing the following conditions are satisfied:

- the user has not already seen all the records indexed by this term;
- the term is not in the list of 'dubious' words, i.e., a list of common words which cannot be stopped but which are not very useful in retrieval either, such as 'analysis', 'introductory', etc.;
- the term weight is positive; and
- fewer than 16 terms have been selected.

The main reason behind the choice of implementing only an automatic query expansion module is in Walker's belief that 'catalogue users would not want a high degree of involvement and would probably not understand what the system was doing' (Walker, 1989).

Most if not all of the work on automatic query expansion so far, with the exception of Sparck Jones (1971), has not been fully evaluated with regard to performance measurement.

At best they have investigated a specific method of automatic query expansion as implemented in a particular system. Relatively recent research on query expansion which tried to systematically evaluate the performance of query expansion using term co-occurrence data is the work of van Rijsbergen and co-workers. In a series of papers they advocated the use of query expansion techniques based on a maximum spanning tree (MST) (van Rijsbergen, 1977; van Rijsbergen & Harper, 1978; van Rijsbergen, Harper & Porter, 1981; Smeaton & van Rijsbergen, 1983).

Experiments on three different test collections demonstrated that relevance feedback searches using expanded queries were superior to simple co-ordination level match searches for which these was no expansion and no relevance feedback information available (van Rijsbergen, Harper & Porter, 1981). But, it was not clear whether this improved performance was due to the feedback or to the expansion. Smeaton (1982) and Smeaton & van Rijsbergen (1983) have investigated the effects of automatic query expansion using three different sources of new terms:

- A Maximum Spanning Tree (MST) which is a term-term dependence structure similar to a thesaurus but derived from statistical associations between terms. It takes the form of a tree structure where every term in the collection points to or is connected to at least one other term.
- Nearest Neighbours (NN). A nearest neighbour to a given term is that term which is the most strongly related statistically to the given term.
- Using index terms from the relevant documents found so far.

The three strategies were tested on the Vaswani test collection. The F4 and EMIM weighting formulae were used and query terms were weighted according to one of the formulae. Relevance judgements were simulated and automatic query expansion was performed using MST and non-MST methods. A test was also performed where no query expansion was performed and where randomly selected terms were incorporated in the strategy. It was found that the above three query expansion strategies all had a detrimental effect on overall retrieval effectiveness (measured by averaging the precision and recall figures computed for each query). The effect in order of increasing degradation (i.e., from best to worst performances) of retrieval was:

1. No modification.
2. Randomly selected terms added to original query.
3. Terms derived from known relevant documents.
4. Terms derived from the MST.
5. Terms as nearest neighbours.

It was also found that the more extra terms that were added the greater the degradation in retrieval effectiveness. Smeaton and van Rijsbergen have put forward several reasons for this detrimental effect on overall retrieval effectiveness. One reason given is that the probabilities were estimated from little relevance information and so could not be estimated accurately and that the situation was compounded by the addition of extra search terms which resulted in a decline in retrieval performance. They also stated that in theory the modifications that should have yielded the best retrieval (i.e., MST and NN), in fact yielded the greatest degradation. This is because the effects that poor probability estimations have on retrieval are 'clustered' into an area around the query. MST and NN terms will be highly related to the original query terms and will co-occur with the original search terms more often and therefore have a significant influence on the top ranked documents. Another reason given is that the query modification strategies may have been implemented too early in the overall retrieval strategy.

A weighting function which utilises the principle of fuzzy sets has been also suggested by Smeaton (1984). This is achieved by assigning a secondary weight as well as a relevance weight to each of the search terms which indicates the terms relative degree of importance to the user's search. The 'degree of importance weights' are combined with the relevance weights. This effectively makes a search term set a partial membership or fuzzy set. New sets of search terms are generated from relevant documents using a matching function whereby all terms from relevant documents are added automatically. In order not to swamp the original query each term is weighted according to how important it is to the overall retrieval. This is achieved by measuring the similarity between the original query and each relevant document and grading the contribution of the new terms obtained from a particular relevant document in accordance with how 'similar' that document is to the query. The retrieval strategy therefore uses the user provided relevance feedback to compute a fuzzy set of search terms composed of the initial query terms and all the terms from all known relevant documents. Smeaton concludes that this strategy has not yielded any significant improvements in retrieval effectiveness over a retrieval strategy that uses conventional weighting of the initial query terms.

Therefore, to date the overall conclusion of the above mentioned research has been that query expansion based on term co-occurrence data is unlikely to bring significant improvements in the performance of document retrieval systems. There is recent evidence, however, which has identified a substantial limitation on the use of term co-occurrence data for automatic query expansion (Peat & Willett, 1991). It has been pointed out that the above limitation arises from the characteristics of the coefficients that are used to measure the similarity between a pair of terms. Peat and Willett have concluded that:

- in a given document collection a given term is likely to be most similar to the terms that have comparable frequencies of occurrence;
- since query terms tend to have high collection frequencies, their NN, which are usually the terms added to the search by the expansion method, are also likely to have high collection frequencies;
- since high frequency terms poorly discriminate between relevant and non-relevant documents, the terms added to the search by the expansion method are unlikely to be effective discriminators.

Their findings provide a rationale for the lack of success attained in the studies which used term co-occurrence data for query expansion. These also explain the findings by Sparck

Jones (1971) (see page 42) that the best results were obtained if only the infrequent terms were clustered and if the more frequent terms were left un-clustered, and by Smeaton and van Rijsbergen (1983) that query expansion by the addition of randomly selected terms gave better results to any of their methods based on co-occurrence data. Apparently, this last point by itself (i.e., that randomly selected terms gave better results) could have been taken as an indication that something might have been wrong.

4.3 Semi-automatic query expansion

Attempts to incorporate a semi-automatic user assisted term selection stage are fewer than their automatic query expansion counterparts. These appeared mainly in the late 70s and during the 80s. The source of the terms for the selection stage, as in the case of automatic expansion, could be based either on the search results or on some knowledge structure which could subsequently be either dependent on the collection or independent of it. The discussion below examines some selected attempts of semi-automatic query expansion which were made mainly on commercial systems. It does not cover recent work which uses AI and ES techniques on machine readable dictionaries (MRD) and lexicons, such as the work of Fox and his co-researchers (Nutter, Fox & Evens, 1990).

The Associative Interactive Dictionary (AID) is a prototype system developed for the Medline and Toxline databases at the National Library of Medicine (Doszkocs, 1978). AID automatically generates and displays related terms, synonyms, broader and narrower terms and other semantic associations for a given search term. These are controlled vocabulary associations from MeSH in Medline or free text word associations in Toxline. The terms are derived from titles, abstracts and/or controlled indexing fields from retrieved documents. These terms are displayed in ranked order according to a 'relatedness' value (R) which is calculated using a modified chi-square value:

$$R = \frac{O - E}{O}$$

where:
R = the relatedness measure, expressing the strength of semantic association between a term and the retrieved set,
O = observed frequency of the term in the retrieved set,
E = expected number of document occurrences of the term in the retrieved set.

The retrieved set is defined as the set of documents retrieved by a given search term or Boolean query. The expected number E is given by

$$E = \frac{nT}{N}$$

where:
n = total number of documents in the retrieved set,
T = number of documents in which the term occurs,
N = total number of documents in the collection.

If the observed document frequency O of a given term in the retrieved set is less than its statistically expected frequency E ($O < E$) then the term is assumed to have no associative value.

The rationale behind the AID system is based upon the assumption that terms showing a considerably higher than expected frequency of occurrence in a retrieved set are assumed to be semantically related to the terms that retrieved that set. Thus AID is said to be a tool that retrieves semantically related terms.

AID operates by storing a subset of the inverted files for the two databases in its in-core hash table. The hash table terms represent all the inverted files' index terms with a frequency of four or more postings. Searches are carried out in the usual Boolean fashion and AID can be implemented at any time through the 'EXECUTE' command.

A typical search on AID requires that the searcher is not only familiar with the terminology of document representation of bibliographic systems (in order to select which field the system should base the analysis on) but has a knowledge of how statistical correlation works since one is invited by AID to decide on a correlation threshold (a number between .0 and .999). Another requirement is that one must also specify the number of records to be used in the associative analysis. Nevertheless, searchers are presented with a list of terms which are ranked according to their relatedness to the initial query term together with an indication on the number of documents each of them would retrieve. The searcher then could select or reject any of the associated terms for query expansion.

A number of the research ideas in AID were subsequently implemented by Doszkocs in the various versions of CITE (Doszkocs, 1983). An interactive session with CITE might be instigated by the searcher who enters an enquiry statement in natural language. CITE parses the input, identifies spelling mistakes, requests their clarification and then suggests to the searcher a set of potentially applicable single words and MeSH headings that match the parsed input and for which CITE proposes to search. These terms are ranked by some weighting formula (which has not been disclosed) and presented to the searcher for selection. Relevance feedback information is also utilised in order to locate other items. CITE performs a frequent analysis on the MeSH headings and presents them to the user who can choose any of these an/or add additional keywords to the search.

Williams has developed a series of systems (Userlink, OASIS, IT) to help both the unskilled user (Williams, 1984; Williams, 1985; Goldsmith & Williams, 1986) and the skilled intermediary (Williams, 1983; Williams, 1984). These systems utilise a microcomputer that acts as a front-end to online bibliographic databases. The system for unskilled users enables them to carry out a search by answering a series of questions asked by the front-end which then formulates a search profile and carries out the search automatically. The skilled user system is primarily concerned with improving the recall of a search by using the microcomputer to assist in the selection of index terms from retrieved documents. This capability has subsequently been incorporated into the Information Transfer (IT) system as the EXPLORE feature (Williams, 1984; Williams, 1985). This query expansion capability will be the feature of these systems considered here.

The steps involved in the use of the skilled user system are as follows.

1. The searcher carries out a search in the normal way and identifies a set of relevant documents.

2. The program then retrieves the titles and descriptors which are contained in the final results set.

3. The retrieved items are then processed to provide two lists of the single words and phrases that are contained in the retrieved documents. The lists are ranked in order of decreasing frequency.

4. The word list and phrase list are then displayed for the user to scan them.

5. Words or phrases can be chosen by the user to be incorporated into the search.

6. The words or phrases which are displayed can be used as triggers to display other selections of terms, such as the inspection of all words which occur in the same document as any chosen word, or inspection of all phrases that contain a chosen word.

7. A simple dialogue is provided which enables the selected terms to be added to the existing search profile.

The words and phrases are displayed in order of decreasing frequency of their occurrence in the retrieved set. The system can handle fairly large numbers of references. Williams suggests that up to 50 references should normally be retrieved and processed to keep the online time to a reasonable value.

A similar system has been introduced into the European Space Agency information retrieval system (ESA/IRS) under the ZOOM command (Martin, 1982; Ingwersen, 1984). This system automatically retrieves and organises terms from retrieved sets of documents. Terms can be in the form of words, phrases or both and can be taken from any field such as the descriptor, title, abstract or author fields which can be specified by the user. Other fields can also be explored using the ZOOM command which will be database dependent, e.g. on Chemical Abstracts the registry numbers. Multiple fields can also be specified.

The term list is organised by frequency of occurrence of the term in the retrieved set. Every occurrence of the term in the record is counted so that the frequency of occurrence for a particular term could be greater than the number of documents being analysed, e.g. in a set of 15 documents a term may be given a frequency of 18. Once the term list has been compiled the user can then scan the list and select relevant terms using the SELECT command.

ZOOM can handle large number of documents. The default value is for 50 records but up to 200 can be analysed. An enhancement of ZOOM called SuperZOOM can analyse up to 20,000 records. Commands similar to ZOOM are now also available in some other hosts, e.g. GET in ORBIT (Maxwell Online), MEMSORT in Questel, EXTRACT in Dimdi.

ZOOM and Williams' system are thus fairly similar in their approach, extracting terms from retrieved records which are subsequently displayed in an organised fashion using frequency information. Using the ZOOM command on ESA/IRS does mean that all scanning is performed online at the host's end while Williams' system allows for the retrieved terms to be saved and scanned offline.

The ZOOM and SuperZOOM features have played an important role in the research described in this thesis. Because of this importance these features are analysed, discussed in detail, and illustrated with examples in a separate section.

Finally, two points were observed in the use of Williams' system (Williams, 1984, p.142) which are also applicable to ZOOM. The first was the way that free language search queries retrieved controlled language terms in databases with authorised vocabulary and that these terms often appeared at the top of the frequency list. This is not surprising since papers on a given topic are likely to be indexed under the same controlled term. However, it does mean that the use of free language will often lead directly to controlled terms. The second point is that it was noticed that many of the terms that a user needs are not at the top of the frequency table. Many terms which users recognised as being useful were present with quite low frequencies. Methods based on ranking terms in this way could therefore miss terms that were useful. However, Williams claims that in using this system users could scan one or two hundred terms very quickly. If term ranking based on frequency of occurrence of terms in the retrieved documents puts useful terms near the bottom of the list but users are able to scan the term lists quickly then this may have implications as to the usefulness of this type of ranking. However, there are a number of research questions involved here which are directly related to the research of this thesis and which are discussed in detail later. Some of the questions are concerned with:

1. the effectiveness of the within-document frequency of terms of the retrieved set (as exemplified by ZOOM) in ranking terms for query expansion; and
2. the willingness and the persistence of users to scan 100-200 terms in order to identify additional terms for the search.

Porter and Galpin (1988) discuss Muscat, an OPAC at the Scott Polar Research Institute, which allows relevance feedback and query expansion in its advanced use of the system. In their words '...the information retrieval system requires a good deal of understanding in order that it may be used to its fullest extent...' (Porter & Galpin, 1988, p.3). However, the system and the ideas it expresses are of interest to this research and therefore it comes up in subsequent discussions.

In the advanced module of the Muscat OPAC a query can be in probabilistic or in Boolean terms. For example,

`query` Alaskan moths

defines a probabilistic query, while a Boolean query is given as:

`boolean` 'Alaskan' `AND` 'moths'

In the probabilistic query extra terms may be added in any of the following ways:

- by browsing through the index and adding in selected terms
- by using the command '`add penguin`' which explicitly adds the term penguin in the search
- by using the command '`terms`' which is a relevance feedback method of query expansion.

The `'terms'` command allows users to view terms found in the relevant documents. These are comprised of stemmed single words and Universal Decimal Classification (UDC) numbers. The terms and UDC numbers are weighted and ranked in decreasing order. A detail discussion of the ranking formula is given in section 7.4.

Another investigation relating to the research of this thesis on the investigation of query expansion is the IRX experiment (Harman, 1988). Part of that experiment looked at some ways of extracting short lists of terms for query expansion gathered from relevance feedback, nearest neighbours, and term variants of the original query terms. IRX used the Cranfield test collection for this bench experiment and user interaction was simulated. The goal of the part of the experiment that looked at relevance feedback was to identify 20 terms that could be included in the 'feedback terms' window of the interface. There is not any theoretical justification for deciding to use the top 20 ranked terms. The main reasons given are the assumption that users will not want to see more terms, and the size of the window itself. According to Harman (1988, p.322) '...the reweighting of terms using feedback, although clearly a desirable technique, was beyond the scope of this experiment.' For the experiment Harman used the following algorithm:

$$w_j = \sum_{k=1}^{Q} \frac{\log Freq_{jk}\,(noise_{max} - noise_k)}{\log M}$$

where
w_j = the weight or score of document j
Q = the number of terms in the query
k = a query term
$Freq_{jk}$ = the frequency of query term k in document j
$noise_{max}$ = the maximum value of $noise_k$ in the database
M = the number of terms in record j. It is a length factor and it includes all significant terms in document j including duplicates.
$noise_k$ = $\sum_{i=1}^{N} \frac{Freq_{ik}}{TFreq_k} \log \frac{TFreq_k}{Freq_{ik}}$
where
N = the total number of documents in the database
$Freq_{ik}$ = the frequency of term k in document i
$TFreq_k$ = the total frequency of term k in the database

The methodology used for feedback was to (Harman, 1988, p.323):

1. run a query, using the algorithm described, and note which of the top ten documents retrieved for that query are relevant.

2. create a file containing all non-common words from those relevant documents, including statistics concerning those terms

3. sort the term list based on one of six statistical techniques, i.e. noise, postings, noise within postings, noise × frequency within postings, noise × frequency × postings, noise × frequency; (these can be derived from the formula presented earlier).

4. add 20 terms from the top of the sorted list to the query.

5. rerank the expanded query against the collection of unretrieved documents (no reweighting of query terms) and evaluate using the 'frozen' method (Salton, 1970).

6. repeat steps 3,4, and 5 for each different statistical technique.

4.3.1 An overview of ZOOM and SuperZOOM

The ZOOM and SuperZOOM features of ESA/IRS have played an important role in the research presented in this thesis. It is therefore appropriate to introduce them here in detail.

ZOOM was first discussed in the literature in Martin (1982) and an analysis of it from a cognitive viewpoint was given by Ingwersen (1984). The discussion below is also based on the IRS QUEST Searcher's Manual, my personal experience of searching ESA/IRS, and from exchanges with IRS technical support both in London and in Frascati, Italy.

ZOOM is based on automatic frequency analysis of phrases, single words, codes, or a combination of these contained in a selected set of references. The objectives underlying this feature are that it:

- replaces record scanning for related terms;
- allows for associative searching;
- helps to improve recall and/or precision depending on how it is used.

Once a set of records is generated in a file you may ZOOM the set. Originally the ZOOM command could analyse only up to 200 records. This was changed recently and now ZOOM can analyse up to a maximum of 20,000 records. However, the basic default analysis sample is 50 records across the last selected set. In determining the sample set of 50 from a set of 200 records the system analyses every fourth record of that set.[1] The new QUEST manual, however, does not provide any information on the sampling method.[2] It merely mentions that the sample of 50 records in the retrieved set is equally distributed throughout that set. In any event, if the set contains less than 50 records then ZOOM analyses all of them.

The fields of the record that can be used to ZOOM on are title (TI), author names (AU), corporate source (CS), coden (CO), journal name (JN), controlled terms (CT), uncontrolled terms (UT), abstract (AB), classification code (CC), meeting title (MT), status (ST), molecular formulas (MF), registry numbers (RN), publication date (PD), source data (SO). ZOOM defaults to the CT and UT fields. However, the searcher may request any of the above fields, either on their one or in any combination. For example:

[1] IRS QUEST User Manual, vol. 2, section 5.1.1.10: Frequency Analysis - ZOOM, February 1983.

[2] ESA-IRS QUEST Searcher's Guide, section 8.7.1/2.

Command :	*Effect :*
ZOOM	analyses indexing phrases in the last set (default 50 records)
ZOOM WORDS	analyses indexing words in the last set (default 50 records)
ZOOM TITLES	analyses titles in the last set (default 50 records)
ZOOM TITLES WORDS	analyses title words in the last set (default 50 records)
ZOOM 8	analyses indexing phrases in set 8 (default 50 records)
ZOOM 8 (90)	analyses indexing phrases from a sample of 90 records in set 8
ZOOM 8 LATEST	analyses indexing phrases from the 50 most recent records of set 8
Z 8(90) LAT W TI,CT,UT	analyses single words from titles and indexing from the 90 most recent records of set 8

A typical search session in INSPEC on ESA/IRS, which includes the use of ZOOM, can take the following form:

```
SET  ITEMS  DESCRIPTION
 1   12867  INFORMATION(1W)RETRIEVAL
 2    2517  FRONT(1W)END?
 3     106  2*1
```

At this point the searcher enters the default ZOOM command which analyses a sample of 50 records out of the 106 records in set 3. It can be seen that the system puts itself into pagemode, thus displaying a screen at a time. At the end of screen it waits for the user's input which can be either a request for looking at the next (or previous) page with terms or a return to the system prompt and search mode.

```
ENTER-z
                                   Text Analysis Results
Frq   Words/Phrases           Frq   Words/Phrases          Frq   Words/Phrases
----- ------------------      ----- ------------------     ----- ------------------
   48 INFORMATION                    SYSTEMS                       SYSTEMS
       RETRIEVAL                   4 DATABASE                    3 MICROCOMPUTERS
   16 INFORMATION                    MANAGEMENT                  3 RELATIONAL
       RETRIEVAL SYSTEMS             SYSTEMS                       DATABASES
   13 INFORMATION                  4 DOWNLOADING                 2 BIBLIOGRAPHIC
       SERVICES                    4 FRONT END                     DATABASES
   13 SOFTWARE PACKAGES            4 IBM PC                      2 BOOLEAN LOGIC
   10 EXPERT SYSTEMS               3 COMPUTATIONAL               2 CHEMICAL
   10 FRONT ENDS                     LINGUISTICS                   INFORMATION
   10 USER INTERFACES              3 DATABASE SEARCHING            SYSTEM
    7 GATEWAYS                     3 DIALOG                      2 COMMODITY TRADING
    6 MICROCOMPUTER                3 DOCUMENT RETRIEVAL          2 COMPUTER
       APPLICATIONS                3 FRONT END SOFTWARE            COMMUNICATIONS
    6 ONLINE DATABASES             3 IBM COMPATIBLE                SOFTWARE
    5 ELECTRONIC MAIL                MACHINES                    2 DATABASE
    5 FRONT END SYSTEM             3 IBM COMPATIBLES             2 END USER
    5 ONLINE SEARCHING             3 IN SEARCH                   2 END USERS
    4 BIBLIOGRAPHIC                3 INTERACTIVE                 2 INDEXING
...Pages.Lines: More= 11.35
```

The phrases and words of the analysis are displayed in columns. All terms are ranked in descending order of their frequency of occurence in the sample set. At the bottom of the screen there is information which refers to the number of terms in the ZOOM list. This information is given in pages (i.e. screens) and lines. In the example above this is Pages.Lines: 11.35 which means that besides the screen we look at there are 11 additional screens with 54 lines of analysis in 3 columns each plus a screen with 35 lines only. This effectively makes a total of 13 screens with terms for the user to browse.

An example of a ZOOM list of single words from the indexing fields (CT, UT) is given below. This is the same set as used earlier since the searcher has entered the command Z W (zoom words).

```
ENTER-z w
                                Text Analysis Results
Frq   Words/         Frq   Words/         Frq   Words/         Frq   Words/
      Phrases              Phrases              Phrases              Phrases
----- ------------ ----- ------------ ----- ------------ ----- ------------
  117 INFORMATION          C              7 GATEWAY            4 ENVIRONMENT
   89 RETRIEVAL       13 IBM              6 AIDS               4 FRIENDLY
   61 SYSTEMS         13 INTERFACES       6 COMMUNICATIO       4 KNOWLEDGE
   39 FRONT           12 ENDS               NS                 4 LIBRARY
   35 END             10 INTERFACE        6 INTELLIGENT        4 LOGIC
   35 SOFTWARE         9 DATA             6 INTERACTIVE        4 MENU
   33 ONLINE           9 ELECTRONIC       6 NATURAL            4 PROGRAM
   30 USER             9 MICROCOMPUTE     6 PROCESSING         4 QUERY
   29 DATABASE           R                5 ACCESS             4 RESOURCES
   27 SYSTEM           8 BASED            5 CONTROL            3 AUTOMATED
   25 SEARCH           8 COMPUTER         5 DOWNLOADING        3 BOOLEAN
   24 DATABASES        8 GATEWAYS         5 ENGINEERING        3 BUSINESS
   21 SERVICES         8 LANGUAGE         5 MAIL               3 COMMAND
   16 SEARCHING        8 MANAGEMENT       5 RELATIONAL         3 COMPATIBLE
   15 EXPERT           8 PC               5 USERS              3 COMPATIBLES
   15 PACKAGES         7 APPLICATIONS     5 VIRTUAL            3 COMPUTATIONA
   14 BIBLIOGRAPHI     7 DOCUMENT         4 CHEMICAL             L
...Pages.Lines: More= 5.53
ENTER-
```

Looking at this list I would like to point out once again that every occurence of a term is counted. Therefore, from the default sample of 50 documents (out of the 106 in set 3) that were used by ZOOM for the analysis we have here a count of 117 occurences for the word 'information', 89 for 'retrieval' and 61 for 'systems'. Although this may seem to be a small detail it is a very significant one for the purposes of my research as will be explained later (e.g., in chapter 9).

In addition, the analysis takes into account all terms including stopwords. However, all punctuation is removed in the display of the ZOOM text analysis.

Finally, if we look at the number of 'Pages.Lines' of the two examples we can observe a rather significant difference in the number of screens mentioned. 'Phrases' show a total of 11.35 whereas 'Words' have a total of 5.53 screens. The reasons here are that the display of phrases is in three columns while the words is in four, and also that most phrases occupy more than one line thus reducing the number of phrases per screen.

SuperZOOM

SuperZOOM was introduced in 1984-85.[3] It is a menu system and provides an alternative way of analysing up. to 20,000 records, which was a significant improvement over ZOOM which then could handle only a maximum of 200 records.

SuperZOOM operates through a menu system, from which the searcher can choose a number of options. It offers all the facilities of the ZOOM command but has some additional features. The most significant is that the searcher can easily add new terms found through SuperZOOM to the search without having to retype the terms as is the case with ZOOM. This is possible because the terms listed in the analysis display are identified by numbers, which are then used to select the terms. The list resulted from a SuperZOOM command can be stored for use at a later time.

As with ZOOM, SuperZOOM is also displayed in pagemode, i.e. a screen at a time. The reason behind a pagemode presentation is that it allows searchers to browse each screen in their own time by letting them move between screens (forwards: P, backwards: P-). This assists users in term selection and subsequent addition of the selected terms into the search. However, although the pagemode is very convenient for humans it is not for a front-end. For this reason I requested from ESA/IRS if a scrollmode, i.e. continuous presentation of terms, could be implemented. This request was accommodated with the introduction of the SuperZOOM Scroll (szs) command. In the example below I discuss the SuperZOOM Scroll version of the SuperZOOM command.

A search session using SuperZOOM Scroll (szs) can take the following form.

```
? find crystals and gravity
   1122626 CRYSTALS
   2 19061 GRAVITY
   3   254 1*2
? run szs
```

To carry out SuperZOOM one has to create a set by using one of the SELECT, FIND, or COMBINE commands. The searcher is then requesting SuperZOOM by typing 'run szs' and the system responds with the SuperZOOM menu. Throughout in the example below user responses are the ones that follow the ESA/IRS prompt (?).

```
? run szs

SETSCROLLMODE Accepted

Please remember: enter HALT to exit anywhere from this service

Do you want to:

 1  submit the zoom command
 2  look at the zoom list
 3  feedback terms into the search
```

[3] Dialtech IRS Newsletter, no.59, January 1985, p1.

```
 4  cancel 01253 SUPERZOOM.OUTPUT MENU
 5  submit other QUEST commands
 6  exit from this service

? 1
```

The user is invited to select from the menu and type in the corresponding number. Here the user selects the option to submit a ZOOM command.

```
Do you want to cancel the previous 01253 SUPERZOOM.OUTPUT MENU? (Yes/No)
? y
```

If in a previous search session a SuperZOOM list of terms had been stored it must be cancelled before a new one is created. Therefore, the searcher must answer 'YES' in the above question. Then the system invites the searcher to submit the ZOOM command which can take any of the forms as explained earlier.

```
Please type the zoom command
? z 3
  Z 3

Zoom submitted: please wait.

Zoom output saved as 01253 SUPERZOOM.OUTPUT MENU

------------------------------------------------

Do you want to:
 1 submit zoom; 2 see the list; 3 feedback terms; 4 cancel the dataset;
 5 submit QUEST cmd; 6 exit.
? 2
```

Once the system has finished processing the ZOOM command it presents the searcher with an abbreviated version of the menu. The searcher here requests to see the list of terms analysed by ZOOM. The list has three columns which represent, from left to right, the rank of the term, the frequency of the term in the sample set, and the term itself. Terms are ranked in descending order. Within ties, i.e. whenever there are more than one term with the same frequency of occurence, terms are ranked in alphabetical order.

```
Please type: P, P- (text scan), X (scan end) or seq. no.s (2/6-8)

                              Text Analysis Results
No. Frq   Words/Phrases
--- ----- --------------------------------------------------------------------
  1    20 ZERO GRAVITY EXPERIMENTS
  2    14 CRYSTAL GROWTH
  3     9 MICROGRAVITY
  4     9 SEMICONDUCTOR GROWTH
  5      8 CRYSTAL GROWTH FROM SOLUTION
  6      7 CRYSTAL GROWTH FROM MELT
  7      6 ELEMENTAL SEMICONDUCTORS
  8      5 CONVECTION
 ...
 11      5 IV VI SEMICONDUCTORS
 ...
 15      4 CONVECTION IN LIQUIDS
 16      4 GRAVITY
 ...
 24      3 CRYSTAL ATOMIC STRUCTURE OF INORGANIC COMPOUNDS
 26      3 DISLOCATION DENSITY
 29      3 GETE PROPERTIES REL TO SUITABILITY AS THERMOELECTRIC MATERIAL
 ...
 35      2 ALLOYS
 36      2 BOUNDARY LAYER
 39      2 CRYSTAL CHEMISTRY
 89      2 X RAY DIFFRACTION
 ...
 90      1 ABSORPTION
 ...
353      1 MOLECULAR SHAPE
 ...

? 1-4
Accepted: 1-4
? x
```

As can be seen, in the list presented here there are more than 353 terms. These terms come from a sample of 50 records out of the 254 records of set 3.

At the prompt (?) the searcher is invited to type in the numbers that correspond to terms that s/he would like to include in the search. This searching ability accounts for the main difference between ZOOM and SuperZOOM. However, the maximum number of terms, from a list of any size, that can be added to the search is 49. If a higher number is requested the program crashes. One could of course ask, why a user would like to search 50 or more terms. The answer is that an ordinary searcher is unlikely to ever request that many terms. A front-end however is very likely to do so and during this research I have come up against this problem with every search.

```
? 1-4
Accepted: 1-4
? x

Command(s) queued:

F ZERO GRAVITY EXPERIMENTS
F CRYSTAL GROWTH
F MICROGRAVITY
F SEMICONDUCTOR GROWTH

    5  1092 ZERO(W)GRAVITY(W)EXPERIMENTS
    6 17393 CRYSTAL(W)GROWTH
    7   636 MICROGRAVITY
    8 15404 SEMICONDUCTOR(W)GROWTH

Service:
 1 zoom; 2 list; 3 terms; 4 cancel; 5 QUEST; 6 exit.
?
```

Once the user has selected the terms (in this case 1-4) SuperZOOM formulates search statements for each term. It then proceeds with the execution of single term searches, with multi-word terms being treated as single terms. When the search is complete the user is presented with the menu. By choosing option 5 from the menu the user can carry out ESA-QUEST commands such as combine sets, etc.

As has been mentioned earlier ESA/IRS was chosen to be used in the research described in this thesis especially because of its unique SuperZOOM feature. The problems encountered/imposed by the ESA/IRS system during a search are discussed in subsequent chapters.

The next chapter introduces CIRT, a front-end that allows weighting, ranking, and relevance feedback during a search with a Boolean system.

Chapter 5

CIRT: a front-end for weighted searching

5.1 Introduction to front-ends for DRS

In the history of online information retrieval, especially in the context of commercial systems, we notice that the basic functional specification of the retrieval mechanism became fossilised very early on. Consequently, command-driven retrieval engines, which allow the selection and manipulation of sets of items, became established as the standard way to do online retrieval. This paradigm allows the creation and manipulation of sets by means of words, Boolean operators, proximity operators and truncation, and it is used for retrieval regardless of whether the database searched contains natural language text, controlled language, or both.

However, despite the existence of some alternatives and despite the difficulty of using Boolean systems by untrained or infrequent searchers this paradigm emerged as the dominant paradigm of online retrieval.

Jamieson (1979ab) has identified three important reasons of why large scale commercial retrieval systems have not implemented the alternative retrieval strategies.

1. Many of the more sophisticated strategies would appear to be too inefficient for the large scale implementations. They would require more computing resources compared with conventional Boolean searching, and consequently the system would be unable to support as many users.
2. The large capital investment that the vendors have committed for establishing and maintaining their existing systems does not allow them to invest additional capital for new software and hardware. Especially not until the original investment has paid for itself.
3. The sophisticated retrieval strategies have been proved effective on small test collections (and since Jamieson's paper on larger collections) but not on large operational collections.

It is unfortunate however that these reasons, especially (1) and (3), still hold today almost as strongly as they did then. The only notable exception of a commercial host implementing alternative retrieval strategies with relevance feedback is that of the Dow Jones News/Retrieval service. This might have been however due to Dow Jones' choice of hardware, i.e. the Connection Machine.

The efforts of many of those who wish to see alternatives to the dominant paradigm have therefore been directed towards the general area of "online searching aids". In other words, the basic retrieval mechanisms are being left intact, however they are being hidden from the searcher by successive layers of intermediary mechanisms, such as front-ends. A review of 'online searching aids', their developments, as well as an analysis of their functional characteristics is given in Efthimiadis (1990).

This chapter examines one attempt to offer an alternative to the dominant (Boolean searching) paradigm. This is based on the principle of developing a front-end that allows term weighting, ranking, and relevance feedback and which interact with a commercial retrieval system, like Data-Star or Dialog. The discussion below does not cover attempts to combine term weighting and Boolean retrieval based on the vector space model, or the extended Boolean methods, such as p-norms, used by Fox, Salton and their co-researchers (Salton *et al.*, 1983a; Salton *et al.*, 1983b; Salton *et al.*, 1984; Salton, 1988; Fox & Koll, 1988), it concentrates only on those attempts based on probabilistic models.

One such attempt is CIRT (Robertson *et al*, 1986) a front-end to Data-Star which was developed at City University, London. An analysis of CIRT's design, implementation and evaluation including the theory that it has been built on is given in the following sections. Such analysis is deemed necessary because the research on interactive query expansion reported in this thesis uses CIRT as its basis for such an investigation. Finally, the analysis of CIRT is based on published and unpublished material as indicated in the text and on my personal experience from using CIRT throughout this research.

5.2 CIRT: background and introduction

CIRT is a prototype software front-end to a remote host, i.e. to a traditional Boolean retrieval system, more specifically to Data-Star. It is based on the probabilistic approach to information retrieval, as applied to search term weighting (Robertson & Sparck Jones, 1976).

The theory (see page 16) leads to a weight for each search term. CIRT derives the weight of each term from the frequency characteristics of this term in relation to relevant and non-relevant documents. The weights may be estimated from partial relevance information (Sparck Jones, 1979a) and may even, in the absence of relevance information, be estimated from the raw frequency of the term in the collection as a whole (Croft & Harper, 1979). The match function according to the relevance weighting theory is the simple sum of weights.

The relevance weighting theory has been the subject of a large number of laboratory experiments (Robertson & Sparck Jones, 1976; Sparck Jones, 1979a; Sparck Jones, 1979b; Sparck Jones, 1980; Bovey & Robertson, 1984) but had not been tested under realistic conditions. Nor had it been compared experimentally with the usual Boolean search methods. The CIRT research projects represented an attempt to conduct such an evaluation in an operational context (using real users, queries, databases, hosts, etc.) Furthermore, it seemed appropriate to compare a retrieval technique based on relevance weighting with conventional retrieval methods using Boolean and pseudo-Boolean techniques.

One method for implementing this type of experiment is via a front-end system, which would provide access to an existing host and which will give the option of employing either a Boolean or a weighted method of retrieval.

In the case of the weighted search the front-end would have the effect of transforming the conventional Boolean system to a non-conventional one. It is worth mentioning here that not all of the information retrieval techniques currently being researched could be implemented in this way. However, there are several that could, for example the probabilistic methods.

One of the first attempts to develop such a front-end was that of Jamieson & Oddy (1979). They proposed an experiment very much like CIRT. They planned to develop the front-end from scratch, using three microprocessors and various peripherals. Unfortunately, they ran into technical difficulties, and did not produce a system which could be tested.

Morrissey & van Rijsbergen (Morrissey, 1981) developed a software version of the front-end on a mainframe. That system although operational had some problems (especially in dealing with bad telephone lines) and it had not been extensively tested.

In order to avoid similar problems the development of CIRT was divided into two parts.

The first part was concerned with the development of the prototype front-end, i.e. CIRT (Robertson & Bovey, 1983; Robertson *et al*, 1986). In the second part CIRT was used to evaluate the two types of retrieval, i.e. weighted, incorporating term weighting, document ranking and relevance feedback, and traditional Boolean [1] in an operational environment (Robertson & Thompson, 1987; Robertson & Thompson, 1990).

CIRT can only talk to one host, the Data-Star service of Radio-Suisse in Switzerland. Furthermore, the number of databases that can be searched using CIRT is limited to Psychological Abstracts (PsycInfo), Inspec and Medline (including all its subdivisions). CIRT has an automatic login procedure that provides access to Data-Star. Then, if the searcher has selected a Boolean search CIRT operates transparently allowing the search to be conducted in the usual Data-Star vernacular. If a weighted search has been selected then CIRT operates opaquely. The search is conducted in CIRT's own language and CIRT itself generates commands and Boolean statements that Data-Star can understand. Data-Star responses are then interpreted by CIRT before being displayed to the searcher.

It must be emphasised that CIRT is an experimental system designed as a tool for the conduct of experiments in particular in testing weighted search as conducted on a commercial database. It is therefore not yet as refined as would be expected of a commercial system. Furthermore, CIRT's user interface is command-driven and terse and thus suitable only for experienced intermediaries.

[1] For the remainder of this thesis 'Term weighting, document ranking and relevance feedback' will be abbreviated to 'weighted searching'. Similarly, traditional methods including Boolean and pseudo-Boolean operators and intermediate search sets will be referred to as 'Boolean searching'.

5.3 Weighting, ranking, and relevance feedback in CIRT

Term weights are estimated according to the relevance weighting theory (see page 16). The search term weight is calculated from the frequency characteristics of the term in relation to relevant and non relevant documents. The weight may be estimated either from partial relevance information derived from viewing references and tagging them as relevant or not, or (in the absence of such information) from raw frequency, i.e. term postings and total size of the collection.

A complete technical specification of the formula used is as follows (Robertson & Thompson, 1987; Robertson & Thompson, 1990):

1. the basic relevance weighting formula is formula F4 (2.2)
2. estimation is by the point-five version of formula F4 (2.4)
3. the non-relevant parameter q is estimated by the complement method (Harper & van Rijsbergen, 1978, p.204)
4. in the case of no relevance information (e.g. the initial search where $r = R = 0$) the simplest estimate ($p = .5$) of the relevant parameter is used (Croft & Harper, 1979).
5. where a search is performed on two databases in succession, the occurrence of each term in any relevant documents is identified in the first database and this contributes to the calculation of the term's weight in the second (Robertson, 1986).

This specification results in the following:

$$p_t = \frac{r + r' + 0.5}{R + R' + 1}$$

$$q_t = \frac{n - r + 0.5}{N - R + 1}$$

where

p_t and q_t are the probabilities of term t occurring in a relevant or a non-relevant document respectively (see discussion in page 15).

N is the size of the collection (in Data-Star given by the docz search)

n the frequency of the term t

R the number of known relevant documents in the database, zero initially

r the number of relevant documents (from the sample R) assigned to the term t, zero initially

R' the number of relevant documents known from searchers on other databases

r' the number of these documents that were assigned to the term

At the initial search, i.e. when terms from the query are first entered into the database, the weight of the terms is estimated from their frequency of occurrence in the database. In other words, terms are given weights in the absence or relevance information since $r = R = 0$ in the weighting formula (see equation 2.7). This leads to terms with lower term frequencies to be given initially higher weights. After the terms have received the initial weight they are searched in Data-Star. CIRT converts the query terms into Boolean expressions as determined by its search algorithm which is discussed in detail in section 5.4. The total weight of each term combination is equal to the sum of the weights of each of the search terms of that combination. Sets of documents retrieved are ranked according to that total weight with the highest total at the top of the ranked list. Records from these sets can then be displayed and relevance judgements are made online. Documents judged relevant are tagged. Re-weighting of the query terms based on the user supplied relevance judgements can then take place. A new search can then be initiated using the new weights. The whole procedure can be repeated until the searcher decides to terminate the search. Figure 5.3 presents a flowchart that demonstrates the iterative nature of weighted searching in CIRT.

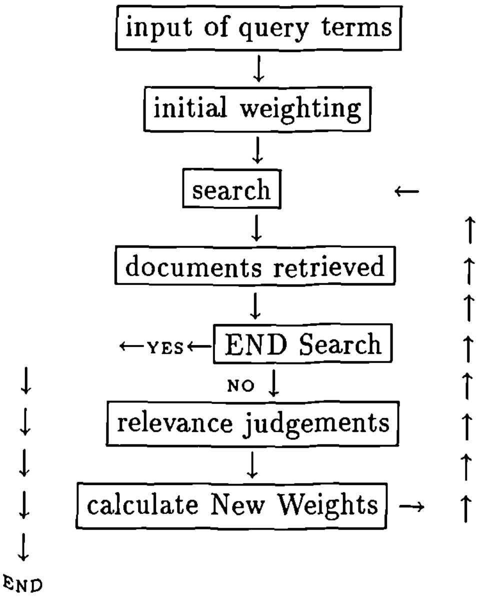


Figure 5.1: CIRT's relevance feedback mechanism

5.4 The search algorithm

For CIRT to conduct weighted retrieval on Data-Star it needs to convert the weighted search into a series of Boolean searches in the way that Data-Star will understand them. CIRT operates a search algorithm in doing this (Bovey & Robertson, 1984). The algorithm generates a series of Boolean statements which are sent one-by-one to Data-Star.

The searches done are stored in the form of a tree which is updated and added to each time a new search is done (see Figure 5.2). Each node of the tree represents a combination of terms for which a search can be done. However, not all search statements are sent to Data-Star.

Furthermore, it is not necessary for the user to know about the search tree in order to do searches. Thus, the search tree relating to the search algorithm is not visible to the searcher at the user interface. Nevertheless while CIRT is processing the request it provides some indication that processing is in progress, with the comment

"SEARCHING..."

followed by a succession of dots being displayed.

After just completing a search the program simply displays its search-tree on the terminal which gives some idea about how the search went. This is done by displaying in descending order the combination of terms that retrieved the sets. The user can then look at the retrieved records. If a new search is requested then enough further search statements will be sent to bring the search-tree up to date.

The program builds a search tree containing nodes with the structure as on Figure 5.2. For a request of any size a very large number of Boolean statements may theoretically be generated. At the n-term level there are:

$$2^n - 1$$

Boolean combinations. However, in practice the number of the theoretical possibilities is radically reduced by making use of search information and of heuristics. Figure 5.2 shows the search tree of the theoretical possibilities for a three term request.

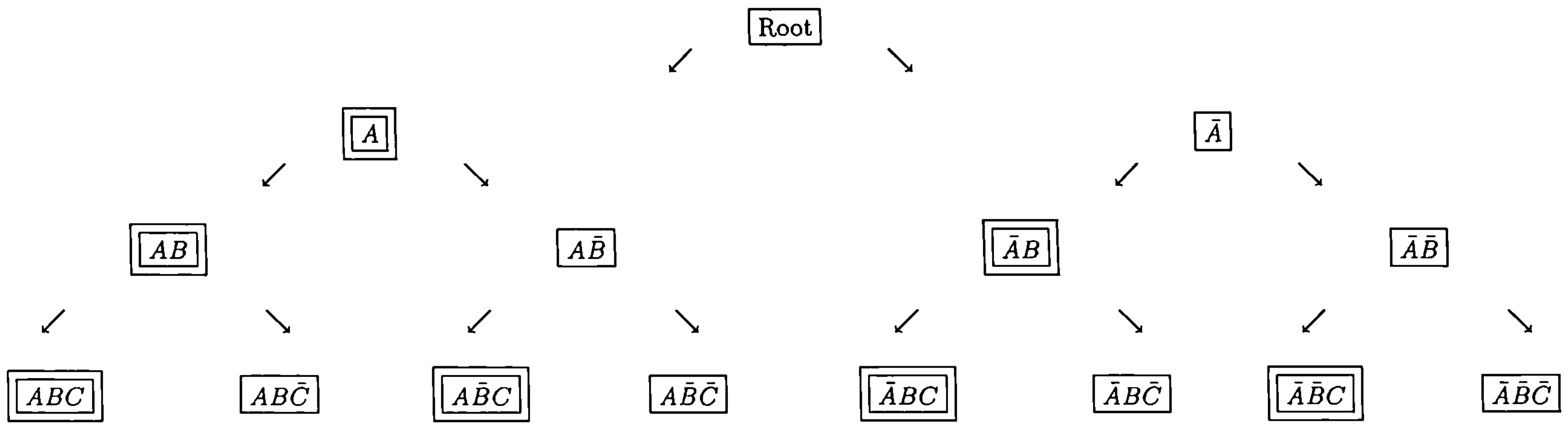


Figure 5.2: Search tree for three terms

The search tree is built up as follows:

1. Single term searches are made for all terms in the query.
2. Search terms are sorted by the program in descending weight order, i.e. the term at the top of the list is the one with the highest weight. This maximises the extent to which large branches can be excluded.
3. A dummy root node is created. The root is a set of documents defined by ORing all the terms in the query.
4. Each node in the tree corresponds to a search statement which is made up of a combination of query terms ANDed or NOTed.
5. Leaves represent all such statements that are possible using all the query terms.
6. The tree stops growing (i.e. the search stops) when it retrieves the default target set of at least the top 15 documents. This default is a limit imposed on retrieval by the `Searchsize` command. However, this limit can be changed to any number so that a larger set can be retrieved. During my data collection the `Searchsize` was set to 50 for reasons explained later.
7. In addition, not all branches of the tree are explored. Criteria for stopping are:
   (a) current search gives zero results, i.e. no documents are retrieved at this node. For example, if node AB retrieves nothing then the searches ABC and $AB\bar{C}$ are not made because they will not retrieve anything either.
   (b) no document on the branch can give a better matching value than the x^{th} best document retrieved so far (x is the `Searchsize`).
   (c) there are no more terms in the query left to be searched.
8. Only searches for the AND branches are done. For the NOT branches searches need not be done explicitly because the resulting search set size can be calculated using the other branches (Bovey & Robertson, 1984, p.85).

The algorithm is performed by a recursive function. The branches in the search tree are terminated by tags either as FREE or NULL. FREE means that this branch has not been searched yet, whereas NULL means that there are no documents in this branch.

At each recursion the function starts from a node that has already been created (the first time this is the root node). It then searches by first ANDing and then NOTing the next query term to that node. This is done with the help of the WITH and WOUT functions. Looking at the search tree of Figure 5.2 it can easily be seen that the leaves of each node are always created in this fashion. For example, node *AB* has two leaves. One is the WITH leaf which ANDs term *C* to terms *AB*, and the other is the WOUT leaf which NOTs term *C* from terms *AB*.

The algorithm works for repeated searches on the same request with new weights or new terms added doing only those searches that are needed.

Looking again at the total number of searches that can be theoretically generated from the algorithm and adding to that number the single term searches and the root search that are required we can establish the maximum number of searches that could be generated.

Search terms n	Number of search combinations $2^n - 1$	Single term searches	Root	Total number of searches required
3	7	3	1	11
4	15	4	1	20
5	31	5	1	37
6	63	6	1	70
7	127	7	1	135
8	255	8	1	264
9	511	9	1	521

However, as mentioned already CIRT's search algorithm does not have to do all these searches. This is illustrated with an example from a real search, here search Q123 from my data collection, that demonstrates the true behaviour of the search algorithm.

Search Q123 had two parts. A first iteration (or initial) search which used four terms and a second iteration where the query was expanded to include a total of nine terms. The number of searches done for these 9 terms were, from start to finish, a total of 116 statements as opposed to the maximum number of 521 searches that could have theoretically been generated.

A point worth noting here is that this total of 116 statements may not be repeatable by other queries which also have 9 query terms. In other words, some other query also consisting of 9 terms may generate fewer or more than 116 statements.

The complete search tree as searched by CIRT on Data-Star is given in Appendix A on page 217. The four terms searched initially were `vision`, `robots`, `transputers`, and `parallel`. Looking at the portion of the search tree that corresponds to the initial search we see that CIRT did the following (NB: this extract is from the network log and users do not see it):

```
 1_: 3579294  docz
 2_:   20082  vision
 3_:   17512  robots
 4_:     580  transputers
 5_:   84978  parallel
 6_:      23  4 and 3
 7_:       4  6 and 2
 8_:       1  7 and 5
 9_:      16  6 not 2 and 5
10_:       3  7 not 5
11_:       4  (C89042735).AN. or (B90010030).AN. or
              (C90006971).AN. or (C89042529).AN. or
              (C89032649).AN.
```

First it established the total number of documents that were available in the database at that moment. The docz search in Data-Star does exactly that and therefore provides the information needed by CIRT for calculating the *N* element of the F4 formula. Then CIRT did single term searches (statements 2-5). The search tree starts at statement 6 and continues through to 9. Looking at the results from the single term searches and at statement 6 one can see that CIRT's searching algorithm has sorted the results by term weight (which in this case is reverse frequency order) and then searched. So the sorted terms look like this:

D-S set	term frequency	term weight	search term
4	580	8.7266	transputers
3	17512	5.3151	robots
2	20082	5.1774	vision
5	84978	3.7165	parallel

This information is used then to build the tree which starts (statement 6) by combining the terms with the lowest frequency (or highest weight) first, e.g. here, `transputers AND robots`. The tree then continues to grow and the search stops at statement 9 because by that point the searching algorithm has identified the 15 documents that are its default target. At this point the search results and the search tree are presented to the user:

```
Searching........Searched

Search tree
No.     seen      weight   terms
1       0       22.94    transputers robots vision parallel
3       0       19.22    transputers robots vision
16      0       17.76    transputers robots parallel
->
```

It is worth noting at this point that CIRT does not do all the NOT searches of the search tree immediately because it can infere the results of such searches from the AND searches. It does however do the NOT searches at a later stage when full records are needed to be displayed. For example, in building the above search tree, CIRT has infered that statement 10, i.e. `vision AND robots AND transputers NOT parallel`, gives 3 documents from the fact that set 7, i.e. `transputers AND robots AND vision`, gives 4 documents and set 8, i.e. `vision AND robots AND transputers AND parallel`, gives 1 document. However, the search tree stops at set 9. When the user asks to see the records, after having seen the top ranked set, i.e. set 8, CIRT has to actually do the search of statement 10 to get the next three documents. Finally, statement 11 is the set of relevant items chosen by the user after seeing the records.

Looking at both search trees, i.e. the network log with the Data-Star statements and the CIRT-to-user display, we can see the correspondence between the weighted and the Boolean statements. Data-Star statements 8, 10, and 9, are presented in this ranked order to the user (though as mentioned statement 10 is not actually searched till required).

In conclusion, the procedure of translating weighted searches to Boolean statements and sending them to Data-Star is protracted. Given also that searchers do not want to wait too long for a search this further limits the number of search terms that can be reasonably used. Furthermore, if a search is to take a reasonably short time then queries need to be restricted to about 7 or 8 terms (Bovey & Robertson, 1984). Time delays, both actual and perceived, resulted in a probably suboptimal use of the weighted search of CIRT by intermediaries as found during the analysis of the evaluation project (Robertson & Thompson, 1987; Robertson & Thompson, 1990). However, as I have already mentioned, besides the significant time delays, CIRT's hardware configuration does not have enough memory to accommodate the large search trees that can be created if more than 8 terms are used. The memory problems result in system crashes for reasons explained in the section 7.3.4. The example presented here was selected on purpose in order to demonstrate that a search with more than 8 terms may be successfully searched only if the resulting search tree is relatively small.

5.5 CIRT's user interface

CIRT, being designed, firstly, to test the technical feasibility of providing a weighted search to a Boolean host via a front-end and, secondly, to evaluate the performance of weighted versus Boolean searches, presents a somewhat terse and restricted user interface, with a limited number of facilities. This means that CIRT has been intended for the use by trained intermediaries who have learned its command language and understood its operating principles. The user can:

1. add and delete terms to the query
2. perform a search
3. examine and evaluate records, one at a time from the top of the ranked list
4. tell the system to adjust the weights in accordance with the new relevance information and then reiterate

5. print citations offline

6. change databases

7. assign search limits (available only in Medline)

8. perform additional actions of a housekeeping nature.

A *term* is any searchable item, e.g. a natural language word, a right-truncated stem, a controlled language word or phrase, or even a Boolean expression of such elements. For practical reasons searching with CIRT is usually limited to eight such terms. Terms can be added or deleted to/from the query before or after the search is performed. Before the search, terms can be deleted from the list completely. However, deleting a term after the search has been performed on a query can be done only by assigning a zero weight to that term. This is because the search algorithm as implemented does not allow term deletion (Robertson & Bovey, 1983, p.21). The only other way being to start again from scratch and exclude that term.

Terms can be added at any stage. This however has to be done manually, that is, there is no automatic query expansion. In a more highly developed relevance feedback system, one might expect the system to be able to extract new terms from the documents judged relevant, calculate the weight of each, and offer it to the user if the weight was sufficiently high. This type of semi-automatic query expansion is investigated in this thesis. Such a facility however was very difficult to incorporate considering CIRT's host/front-end set-up for a number of reasons which are discussed in more detail below including the difficulties to arrange such a facility at the host end.

Initial experiences of the use of CIRT suggested a number of major or minor alterations to the user interface, some of which were included in the version that was evaluated. Several had to do with the granularity of ranking, e.g. although potentially every individual document has a unique rank, in practice (given typical query set sizes) there are considerable numbers of ties. That is, the collection is divided into sets (corresponding to every combination of query terms present and absent), and these sets are ranked. Nevertheless, CIRT's user interface is closely allied to the perceived function(s) of the front-end which was conceived, and has been developed, essentially as an implementation of the relevance weighting theory.

CIRT is divided into two modes, a display mode and a search mode, which incidentally correspond to Data-Star's modes. The search mode is prompted by a " → " whereas display mode is prompted by a " ?? ".

In search mode all commands are in the form

$$\rightarrow \ command \ \ argument_1 \ argument_2 \ldots$$

The following are CIRT's search commands and these can be abbreviated to that portion of the command that is in upper case.

`ADD`	add term(s) to the search
`AOF`	add term(s) to the query list offline
`DELete`	delete terms before or after the search
`LIST`	list the terms presently active on the search
`Look`	display titles of retrieved records
`NewWeights`	recalculate weights including relevance feedback information
`OldQuery`	lists query terms searched so far (usually requested after a reset
`Out`	logoff Data-Star
`RESET`	start a new search or change database
`Quit`	logoff CIRT
`Search`	search terms in query list on Data-Star

Display mode is prompted by a " ? " or " ?? " depending on which of two display levels CIRT is in at that moment. The first display level allows for the display of titles and is indicated by one " ? ". When other fields from the records of the retrieved ranked set are requested then CIRT enters the second level of its display mode which is indicated by the two " ?? ".

Once CIRT has finished searching and has presented the search tree to the user it then offers the searcher a choice of four options for handling the retrieved set(s):

`Ignore, Print, Look,` or `Quit.`

The searcher can view documents one at a time following the search. Each document is retrieved in its entirety, but only the title is displayed on the screen initially.

The `Look` option displays titles of documents in the ranked sets. Titles are displayed automatically, but the searcher may choose to display other fields as required, such as the abstract or descriptor field. The program prompts the user for a reply. `Print` allows the searcher to print offline the set of documents being displayed and `Quit` returns the searcher to the search mode.

`Ignore` allows the searcher to ignore a complete set of documents and skip to the next set down in the ranked list. This facility allows the searcher to indicate to CIRT that the combination of terms in this particular set, even though it produced a high rank position, is not very good for this search. This situation may arise when a number of synonymous terms produces a term combination that is high in rank order which however lacks other vital search terms. The theorical explanation for this behaviour is due to the term independence assumptions of the relevance weighting theory as described in page 15.

While on display mode CIRT also accepts the user's relevance judgements on the displayed documents. When a document is tagged as relevant (with the letter r) then CIRT makes a note of this information and uses it if it is asked to continue searching. In essence after each relevance judgement CIRT updates the information required by the F4 formula in terms of relevancy information, i.e. the formula elements r and R (see 16) needed for the recalculation of the weights. The options available to the searcher are:

?? rl	document is relevant, look at next document's title
?? rq	document is relevant, quit looking and return to search mode
?? r	user indicates document is relevant, CIRT then asks what to do with the entire set, i.e., `Ignore, Print, Look, Quit.`
?? f	user would like to see additional fields (AB, DE, YR, LG, SO, AU+SO)
?? l	look at next document's title
?? i	ignore this set and start looking at the next one
?? p	print the whole set offline in full format
?? q	user chooses to return to search mode
?? CR	asks user what to do with entire set, i.e., `Ignore, Print, Look, Quit.`

5.6 A typical weighted search

A typical weighted search using CIRT is described in this section. The example search used here is the same, i.e. search Q123, as the one used to illustrate the discussion of the search algorithm in section 5.4.

User Q123 is a doctoral student and the topic of his research is "Computer vision for robots using parallel computers (transputers)". The aim of this research is:

> "to investigate algorithms and computer architectures to provide a vision system for an industrial robot. Use will be made of parallel computing, particularly transputers. The vision system must provide real-time (frame rate) response and should be affordable."

The weighted search in the example below demonstrates the search up to the end of the first iteration. There are two reasons for such choice. First, because subsequent search iterations are similar to the first unless query expansion takes place and, secondly, because this style of searching, i.e. one or more iterations without query expansion, was followed by the intermediaries during the CIRT evaluation project. The complete search of this example, which included query expansion, is given in the Appendices.

What follows is the procedure for a typical weighted search on CIRT. User requests and responses are in **bold typeface** whereas CIRT's responses are in `typewriter typeface`. Additional comments appear in square brackets [ ].

Once the CIRT program is invoked the search intermediary may enter an identification number for the search session, e.g. **q123b**, select search mode, i.e. Boolean or weighted, add the search terms offline, check with the `OldQuery` command how CIRT has interpreted his/her input of terms, and finally execute the `li` (login) command to connect to Data-Star.

```
Enter id for offline prints- q123b
Enter query identifier- q123b
Enter y or Y for boolean search; n,N or RETURN for weighted search- N

-> aof vision robots transputers parallel
-> oq
rels     t rels
1.  0        0      vision
2.  0        0      robots
3.  0        0      transputers
4.  0        0      parallel

-> li
```

CIRT's login command automatically logs into Data-Star and to the default database as specified in the `.login` file in the user's home directory. In the case of this example and throughout my experiments the default database was Inspec (INZZ). However, as mentioned earlier one can also access Medline and PsycInfo. CIRT then prompts the user for any limits that should be imposed on the search with the question "`any limits?`" The limits include Year, Language, the main MeSH categories such as human, animal, male, female, and an option to include any other user defined limit applicable to the database being searched.

```
-> li
No host name given - dstar assumed
call established

ENTER YOUR USERID
:yxzadj
ENTER YOUR SECURITY PASSWORD
:
WELCOME TO RSAG
CHOICE OF RSAG SERVICES
1 DATA-STAR
9 DISCONNECT
ENTER YOUR CHOICE_: 1
ENTER DATA BASE NAME_:    inzz

Login successful
Any limits ?   no
doczsrch:  3579294
->
```

At the -> prompt CIRT signifies that it has successfully logged in the database and that it is in command mode awaiting the user's instructions. At this point the user can issue any of CIRT's search commands. If search is to be done to a database other than the default the `reset` command is used to change to the desired database.

Once logged into the desired database search terms are added in. Any natural language or controlled vocabulary term can be used as well as any Data-Star search capability, such as truncation, adjacency, or Boolean combination. If terms have been added offline they can be added in either all at once by using the "add all" command or selectively by identifying the term numbers from the query list, e.g. "add 3 4". Each term is then sent to Data-Star and searched in the database. The single term searches are required for the estimation of the weights which once calculated are displayed to the user.

```
->  add all
2.  "vision"       added, freq=20082, weight= 5.1774
3.  "robots"       added, freq=17512, weight= 5.3151
4.  "transputers"  added, freq=580,   weight= 8.7266
5.  "parallel"     added, freq=84978, weight= 3.7165
->  s
```

At the prompt the searcher may wish to start the search at this point or add in other terms. Once the search command has been issued and the search has commenced CIRT displays the message Searching followed by a series of dots which indicate that searching is taking place. Each dot represents a single search statement, rather than a time interval. Once the search is complete the message is concluded with the word Searched. CIRT then displays the search tree in which the sets are in ranked order. As mentioned earlier the default searchsize is 15 documents. Therefore, only sufficient sets are displayed to retrieve the top 15 documents.

```
Searching........Searched

Search tree
No.     seen     weight   terms
1       0       22.94    transputers robots vision parallel
3       0       19.22    transputers robots vision
16      0       17.76    transputers robots parallel
-> l
```

The search tree provides information about the number of sets that have been retrieved and of the number of documents in each set. Sets are ranked according to their aggregate weights together with the term combinations that retrieved them.

The searcher can now request to see the documents by issuing the Look command. CIRT then enters the display mode, it presents summary information about the first set and prompts the user to take action, i.e. ignore, print, look, quit. Here, the user has chosen to look at the first document of the first set. Consequently, the title is displayed and CIRT awaits for the next command. In all cases below the user has indicated that the records were relevant.

```
1 documents with weight 22.94
transputers(8.7) robots(5.3) vision(5.2) parallel(3.7)
----------------------------------------------------------
ignore,print,look or quit?  l
TI Computer vision for robotics using a transputer array.
??  rl

3 documents with weight 19.22
transputers(8.7) robots(5.3) vision(5.2)
----------------------------------------------------------
ignore,print,look or quit?  l
TI Image processing for robot road following.
??  rl
TI A structured lighting vision system for dynamic
obstacle avoidance with a mobile robot.
??  rl
TI Image processing transputer-based machine for mobile
robot road-following.
??  rq
```

While looking at the retrieved records whole sets can be printed offline whilst other sets can be entirely rejected, as is the case in this example with set 3. The display can be stopped at any point and the system will automatically re-weight the terms according to the relevance information. This procedure can be iterated any number of times until the user decides to terminate the search.

```
There are 4 known relevant documents
old wt.     new wt.      rels.
8.73        10.93     4    transputers
5.32         7.51     4    robots
5.18         7.37     4    vision
3.72         2.87     1    parallel
```

The information presented to the user after the re-weighting includes the query terms and the number of times each query term appeared in the retrieved documents. For example, the term `parallel` occurs only in 1 out of the 4 records that have been judged relevant whereas the `robots` appeared in all 4 of them. The individual information about each term is the element r of the F4 formula while the total number of documents that were judged relevant is the R. The two together are used in the re-calculation of the weights and as can be seen this relevancy information may result in an increase or decrease of the weight of a term.

CIRT, if so desired, can now search again by utilising the information given with the new weights. In addition, query expansion can take place. This option will be discussed in detail in the following chapters.

5.7 The CIRT evaluation project

In the research reported in this thesis I have used some of the data collected by the CIRT evaluation project in a number of ways especially during my Pilot studies. In my final experiment I have also used some of the results of the CIRT evaluation project to compare them to my own results. It was therefore thought appropriate to include a brief description of the CIRT project.

The CIRT evaluation project was one of the projects funded by the British Library Research & Development Department to investigate weighted searching. The evaluation project was carried out between June 1985 and July 1987 and was designed to evaluate weighted searching under operational conditions (Robertson & Thompson, 1987; Robertson & Thompson, 1990).

The aim of this project was to attempt a comparative evaluation of weighted and Boolean searching through the use of a front-end. Both types of searching were conducted through CIRT. The comparison of the two systems was based on independent samples. For each query in the experiment a random choice was made as to whether it was to be conducted using weighted or Boolean searching. This was to avoid the problem of the *learning effect* from repeating the same search on two systems (i.e. the matched pair design), since both intermediary and end-user would be influenced on the second search from the results of the first (Jamieson & Oddy, 1979). Originally, it was hoped to get a sample size of around 500 searches. This size was required in order to satisfactorily determine the statistical significance of any observed differences (Robertson, 1990a). However, only 190 searches were finally obtained due to difficulties in recruiting suitable users to take part in the project.

All searchers were conducted by trained intermediaries. The users were required to be present during the search, and in the case of the weighted searches make relevance judgements. They also had to fill in two questionnaires and evaluate the offline prints of the retrieved records. The intermediary was also required to fill in a questionnaire after the end of the search.

The first user-questionnaire was given to the user after the pre-search interview. This was concerned with user and request characteristics, e.g. personal details such as name, address, etc., and information about the search. The second user-questionnaire was completed immediately after the search and at the same time a similar questionnaire was completed by the intermediary. These questionnaires were concerned with both the user's and the intermediary's overall satisfaction with the search.

The user was then asked to come back after a few days in order to evaluate the offline prints obtained in the search. For the evaluation it was decided to print only a maximum of 50 documents per search. If the search had retrieved more than 50 documents then the first 25 and the last 25 records were only printed and used for the evaluation. The user asked to evaluate the offline prints according to two criteria. The first was concerned with the *topical relevance* of the document whereas the second was establishing its perceived usefulness (*utility*) to the user.

Complete logs of all the searches were automatically kept by CIRT. Two types of logs were produced corresponding to the interaction between the intermediary and CIRT and between CIRT and the host (Data-Star). For the Boolean searches these were virtually identical. For the weighted searches however the two logs differ significantly. The *search log* shows the transaction between the searcher and CIRT. The *net log* shows the transaction between CIRT and Data-Star. The logs provided quantitative assessments of the searches such as the number of terms used in the search, number of terms added or amended, the number of packets sent over the network, online time, and the number of online and offline citations, etc.

The results of the evaluation are given in the project's final report (Robertson & Thompson, 1987) and these are further analysed in (Robertson & Thompson, 1990). Overall it was found that weighted searching can be implemented as a front-end to a remote Boolean host with performance comparable to Boolean searching. Comparisons were made in areas such as retrieval effectiveness, user effort, cost, subjective user reactions, and intermediary's contribution. Little or no difference was found between Boolean and weighted searches under these criteria though online time was slightly greater for weighted searches which would affect cost.

5.8 CIRT: Technical description

5.8.1 Operating environment

For the development of CIRT, in order to avoid the implementation problems encountered by Jamieson and Oddy (1979), it was decided to make use of existing technology of the time as much as possible. An additional decision was to exclude microcomputer technology or dial-up lines because of the inherent limitation of that technology at the time (c.1983). CIRT has been implemented as follows (Robertson *et al*, 1986):

Implementation of CIRT

Hardware	LSI 11/23 256 Kb main memory 20 Mb hard disk storage
Operating system	Unix version 7 (DEC release overlay kernel)
Communications	Front-End Processor (FEP) LSI 11/02 'York Box' (X25) software connected to JANET PAD, file transfer, & mail facilities
Prog. Language	C Unix facilities

The hardware is an LSI 11/23 with 256Kb memory, and the operating system is Unix version 7 (DEC release overlay kernel). The programs comprising CIRT are written in C and make use of other Unix facilities. Telecommunications are provided by a 'York Box' (University of York Unix-X25 packet switched connection). The network connection is to JANET (the UK Joint Academic Network), which allows gateway connections to British Telecom's PSS and to international networks. Some additional considerations, besides Unix itself, for the choice of the C programming language were:

- the need to use a sophisticated system development language, which will have access to low-level and operating system operations as well as the facilities of a high-level language.

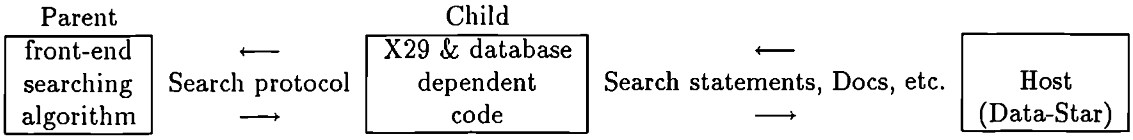
- the 'York Box' provides a range of C library functions for direct access to and control of network operations. Thus the network communications are closely integrated with the program itself.

- an additional Unix/C facility, which was used extensively was LEX (lexical analyser generator). LEX, which is used for pattern matching, is a C pre-processor, i.e. LEX source programs are translated into a C function (`yylex`). Essentially a LEX program scans an incoming stream of characters for particular patterns, and executes appropriate C routines depending on the patterns found. All the analysis by CIRT of incoming messages from the host is done by means of a LEX program. For example, if the incoming message is:
  ```
  DATA-STAR SEARCH MODE
  ENTER QUERY
  ```
  then a variable is set to indicate what mode (search or print) Data-Star is in. If the prompt _ : is received then LEX knows that Data-Star is waiting for input. The LEX function `yylex` will return a value to indicate the current situation.

CIRT's code consists of about 6000 lines of C or LEX source and makes extensive use of the 'York Box' library functions.

As originally developed, CIRT ran as a single program, i.e. one 'process' in Unix terms (Robertson & Bovey, 1983). However, this implementation had some inherent limitations which led to splitting CIRT into two processes running in parallel (in software terms). The version used in the CIRT evaluation project and in the research presented here runs in this fashion. Some of the limitations that led to the split were purely pragmatical. For example, processes on the LSI 11/23 are limited in size to 64Kb. The use of the 'York Box' and LEX both make for large programs. The system was also running out of environment space whenever there was an attempt to provide the necessary additional facilities for the evaluation project.

CIRT was split into two processes as follows (Macaskill, 1987):


Parent
front-end searching algorithm
←
Search protocol
⟶
Child
X29 & database dependent code
←
Search statements, Docs, etc.
⟶
Host (Data-Star)

The nature of the split is that one process (parent) deals with the dialogue with the user and the searching algorithm; the other (child) deals with telecommunications and the dialogue with the host. The implementation of this two-process version again depends on the relative sophistication of Unix, which allows for processes to communicate via 'pipes'. Two pipes connect the two processes (one for communication in each direction).

When CIRT is invoked the parent process establishes a two way communication channel with the child process with the use of two pipes. The flow of control in the child process is as follows:

1. get the next command from the parent process
2. hold dialogue with the host
3. send reply to the parent process
4. go to step 1 and repeat

As an example, let us consider the case of a single-term search. The parent will send a command to the child (i.e. to search the term). When the child receives the request it sends a message to the parent acknowledging receipt and then it will send a message to the network (e.g. the term to be searched) and waits for a reply.

It is worth noting that the two processes run on a single processor. Therefore they do not operate in parallel in a hardware sense, but only in a software simulation of such.

Part II

Interactive Query Expansion: pilot case studies

Chapter 6

Introduction to the pilot case studies

6.1 Aims

Tests on the searching function in information retrieval can be divided in two categories (Sparck Jones, 1981, p.237). One is concerned with operational systems and concentrates on two areas, *search logic* and *search behaviour.* The other tests are concerned with experiments on *query modification by relevance feedback.* The latter tests have been using test collections where the role of the user in detailed decision making in query expansion and query modification either does not exist or is at best simulated. This division of searching tests in information retrieval research is associated with the differences in the retrieval techniques involved, i.e. Boolean vs weighted retrieval, which also corresponds to commercial and experimental systems. Quite often research on searching reported for one side seems to be irrelevant or not applicable to the other, and vice versa. However, there are strong reasons for attempting to bridge the gap in the case of query expansion. This is for a number of reasons including the role that knowledge organisation and representation, semantics, and human information seeking behaviour have to play during the process.

Query expansion as defined in section 3.3.1 and discussed in the previous chapters can be automatic or semi-automatic. With automatic query expansion the user has no control over which terms are added during the search. Furthermore, the query can become swamped with additional terms thus changing the original query considerably and often in rather unpredictable ways. In addition, previous research on query expansion has shown little, if any, improvement in the search (see discussion in chapter 4). In other words, automatic query expansion has not provided any significant results with respect to improving retrieval performance.

Some of the questions that emerge when considering query expansion are:

— what are good terms?

— which are the best terms for query expansion?

— where can we get the query expansion terms from?

— how useful can the query expansion terms be?

— how can we present the terms?

— how can we rank the terms?

— which ranking algorithm or method should we use?

— are searchers able to recognise the good terms?

— how do the searchers select terms?

— what criteria do they use?

— what kind of relationships are there between the original query terms and the terms that the users select?

— is there a difference in what the user selects and what the system suggests as a good term?

All these questions reinforced my decision to conduct an investigation on interactive query expansion.

The overall aim of this thesis is to investigate the process of interactive query expansion from various points of view including effectiveness.

At first, interactive query expansion is investigated in a controlled environment. Then having gained experience and learnt from this I looked at the process while searching with real users, the aim being to investigate the effectiveness of interactive query expansion.

In order to investigate effectiveness, however, one needs a real system. Furthermore, for the study of interactive query expansion in a dynamic user-centred environment one also needs a real system with real users and their real information requests rather than a system which searches a static test collection with fixed (artificial) queries. The plan to use an operational system, which allows weighting, ranking and relevance feedback and searches commercially available databases, led to the use of CIRT as the research tool for conducting this investigation.

The first part of this research consists of three case studies, referred to as *Pilot 1, Pilot 2* and *Pilot 3*. Each of these case studies looked at different aspects of the query expansion process by using a front-end. After the fact I can say that the pilot case studies involve small amounts of data. This became apparent only after the examination of the CIRT search logs was comlete. Nevertheless, the knowledge and experience that was gained from the pilot case studies is regarded necessary for the data collection and was applied to the methodology of the study proper, i.e., in the operational situation with real users, that is described in Part III and is referred to as the *Experiment*.

The aims of the Pilot case studies and the Experiment were:

Pilot 1: to look at the process of query expansion as a whole, within the context of CIRT, and see what can be learned from it. The study investigated the different sources for selecting query expansion terms for the front-end, e.g., ZOOM on certain record fields, such as DE and TI, select terms from the INSPEC thesaurus, etc.

Pilot 2: to look at the process of adding new terms in the search. How people do query expansion without getting any help from the system.

Pilot 3: to look into the questions of: *'What evidence is there that terms taken from relevance judgements of the first iteration search might subsequently be useful?'*, *'How predictive is the ranking of the first iteration (set) in retrieving the documents of the second set?'* and *'Are the terms high-up on both lists?'*

Experiment: using the results of Pilot 1, 2 and 3 to investigate a real system, real requests, and real interaction. The emphasis of this study, which looks at query expansion in a real environment, concentrates on the characteristics of the interaction.

The three pilot case studies answered research questions on query expansion in terms of: How useful are the query expansion terms? Where to get them from for the proper experiment. How to present (rank) them. Which ranking method to use in ranking query expansion terms.

Having investigated query expansion under the controlled experimental conditions of the pilot studies and having gained experience and learned from it I then proceeded with the experiment under real conditions. I used an operational system (CIRT), which uses weighting, ranking and relevance feedback and searches commercially available databases (INSPEC) in order to study query expansion in a dynamic environment.

The aim of studying interactive query expansion from various points of view including effectiveness means that it was broader than just effectiveness. In order to investigate the process of query expansion as well as its effectiveness one needs to have a real system, a dynamic user centred environment.

When looking at a real system, real requests, and real interaction, the emphasis lies on the characteristics of the interaction. For example, the most obvious characteristic is the selection of terms at a particular stage in the process. To look at the characteristics of the interaction as a whole necessitated the recruitment of users for a real interaction. Consequently, I am reporting qualitative as well as quantitative results.

6.2 Rationale

One could argue that an appropriate way to start this investigation would be to use a test collection rather than the real life situation that I have tried in this research. For example, it is reasonable to do everything as has been done in this research, but one could have looked at some aspects of this investigation with a test collection. Indeed, there are some questions that one could address with a test collection, such as evaluate ranking algorithms, design and then evaluate some aspects of the user interface specific to interactive query expansion, etc.

Due to the artificiality involved in test collections, a test collection is not so good for the type of investigation described in this thesis. This artificiality is in part related to the sequence of making relevance judgements during the search and the unreality of that sequence in a test collection environment where one does not get this sequence.

In terms of evaluation methodology information retrieval researchers have attempted to control the large number of unpredictable and disruptive variables encountered in information retrieval (an extensive list of independent variables that affect online information retrieval is given by Fidel and Soergel (1983)). Therefore, variables that are likely to significantly influence evaluation such as characteristics of individual users, information needs, search intermediaries, indexing policies, databases, etc., are controlled. However, human behaviour cannot reliably be reproduced or simulated in laboratory experiments. It is also not feasible to reproduce the information seeking behaviour of a person who has a real information need. Thus, controlled experiments to evaluate information retrieval systems have been criticised for their artificiality (Ellis, 1984; Sparck Jones, 1988). It is for these reasons that I opted for an operational system evaluation, despite the difficulties and the uncertainty involved in such evaluation.

There is however some parallelism in what I have done in the pilot case studies and the artificiality of a test collection environment. The three pilot cases, especially *Pilot 1*, are artificial with respect to the relevance judgements because I did not have all the necessary information available on which to base relevance judgements. Consequently, I had to rely on my personal judgement for relevance decisions and in some occasions on that of experts, in the subject area of the request, whom I used as judges.

In some respects, this is an artificiality which can be set against the artificiality of the test collections. There are research questions which I might have resolved better with a test collection, as for example, in evaluating the suitability of a number of ranking algorithms for term selection on query expansion. In such an evaluation test collections would have provided the conditions for statistically more reliable results, e.g. standardised environment, larger sample of queries.

However, I was using the pilot cases to prepare me for a real situation with real users, i.e. I used the pilot case studies to lead me on to what might be done in a real situation with real users. Taken from this point of view a test collection would not be able to help me in this research because of the artificiality involved.

Furthermore, most of the experiments on query expansion which used test collections and automatic methods for the expansion, have given rather disappointing results. More importantly, in all such experiments there was no real user involvement. The role of the user was either assumed by the investigators or by computer simulation. Therefore, this research follows Sparck Jones' (1988) call for investigations with real user involvement. More specifically, the research looks at

- how one might go about doing such research in a real life environment, and
- what sort of questions would be involved, and then
- some experiments along the lines of the experience gained are carried out in a real online environment and evaluated.

So, even though the three pilot case studies involve some artificiality this fits with the emphasis of the entire thesis. The pilot studies, *Pilot 1-3*, are described in detail in Chapters 7, 8 and 9 and they lay the ground for the discussion of the *Experiment* described in Part III.

Chapter 7

Pilot 1

7.1 Introduction

The main objective of the first part of this case study was to look at the process of query expansion as a whole and see what can be learnt from it. The process of query expansion was seen as pertaining to using CIRT only rather than any other system.

My starting point in looking at the process of how and where CIRT could accommodate query expansion was CIRT itself. It was decided, earlier on in this research, that CIRT will be used as is during this investigation. In other words, it was not attempted to re-write CIRT and the reasons that led to this decision are discussed in section 7.3.4. I started by looking at searches collected during the CIRT evaluation project. The idea being that by looking at the data collected during a search, i.e. questionnaires, transaction logs, relevance judgements online and offline, I would be able to understand enough about it so that I could replicate it and see how I could add a query expansion stage to that search.

Furthermore, I aimed at investigating the different sources where I could select terms for query expansion. Two sources for term selection were identified, the controlled vocabulary of the database to be searched (in the form of a thesaurus or subject headings) and the retrieved records identified as relevant from relevance judgements online.

ZOOM, besides CIRT, was the other most important tool in this research. Its role was essential for performing the term frequency analysis of the retrieved records. Using the CIRT evaluation searches in the investigation meant that suitable searches needed to be identified, i.e. searches on databases that are available on both Data-Star and ESA/IRS. CIRT accesses Data-Star and during the evaluation project searches were conducted on the Medline and INSPEC databases. ZOOM is available only on ESA where the Medline database is not available. Therefore, the INSPEC searches could only be used. For this purpose all INSPEC searches were identified and a sample of these was taken in order to re-run the searches.

It was necessary to reconstruct the same database environment and replicate the CIRT evaluation searches. Most CIRT searches were conducted in 1986-1987. Repeating the same searches in 1989 would give different results because of the addition of new records in the database. Also, the different term frequencies observed between the query terms in, for example, 1987 and 1989 would result in the calculation of different weights which could consequently fail to retrieve the documents of the original set.

Attempting to reconstruct searches that were done during the CIRT evaluation involved many problems which are discussed in the methodology section. It will suffice here to say that once a suitable CIRT evaluation search was selected for this study it was searched in Data-Star as well as in ESA. This required the reconstruction of the same database environment in both hosts, which is a difficult task in itself. Reconstructing the same database environment in ESA was required because the analysis of the INSPEC records was done using the ZOOM command and this was needed in order to render the results of the two hosts compatible.

The ranking of the query expansion terms initially was done with the modified version of the F4 formula (Robertson, 1986). However, the poor quality of the ranking obtained by its use resulted in an investigation which involved six ranking algorithms and is discussed in this chapter as well as in the discussion of the *Experiment.*

7.2 The INSPEC database

INSPEC (Information Services for the Physics and Engineering Communities) is a bibliographic database published by the Institution of Electrical Engineers, London, UK. INSPEC provides worldwide coverage of the literature in physics, electrical/electronics engineering, computers and control equipment. The INSPEC database provides access to all the information appearing in INSPEC's printed abstract journals:

A for Physics Abstracts
B for Electrical and Electronics Abstracts
C for Computer and Control Abstracts.
D for Information Technology (incorporated in C)

Physics Abstracts covers all areas of physics including particle, nuclear, atomic, molecular, fluid, plasma and solid state physics, biophysics, geophysics, astrophysics measurement and instrumentation. Electrical and Electronic Abstracts covers all areas of electronics, radio, telecommunications, optoelectronics and electrical power. Computer and Control Abstracts covers all aspects of computer installation, applications, hardware, software as well as control engineering, robotics, systems theory and artificial intelligence. IT Focus covers office automation, new systems, equipment, applications, electronic mail, facsimile, teleconferencing, viewdata, word processing.

INSPEC's comprehensive information service focuses on journal articles (80% of records), conference proceedings and papers (18%), and books, monographs, reports and dissertations (2%). The sources include some 4000 journals and 1000 conference proceedings worldwide, together with books, reports and other occasional publications. INSPEC provides a number of searching aids such as a User Manual, the INSPEC Thesaurus, the INSPEC Classification and the List of Journals.

The time span of the database covers the period 1969 to date and its size is approximately 240,000 citations per year. INSPEC is available on ESA/IRS as file 8 and on Data-Star as insp (1980 to date), in79 (1969-1979), and inzz (1969 to date).[1] An example of the structure of an INSPEC record is given in Appendix B.2.

7.3 Methodology

7.3.1 Sample searches

During the CIRT evaluation project 190 searches were collected (Robertson & Thompson, 1990). From those only 46 were searched in the INSPEC database on Data-Star and which are further divided into 25 Boolean and 21 weighted (a list of the searches is given in Appendix B.1).

For the purposes of my research I could only use the weighted searches because these were the ones that had relevance judgements made online. The numbers of all 21 searches were put in a lottery and case numbers 62, 140, and 287 were drawn. In order to avoid any confusion that might arise when I am referring to the original CIRT searches and the CIRT searches that were used in this investigation I will refer to them as c62 (c for CIRT) and r62 (r for reconstructed) respectively. The searches that I have collected will be given with the search number only, e.g. 115.

The logs and other related material of the three randomly selected weighted searches were then extracted and studied. The three searches are described in detail in the discussion section. However, it is necessary at this stage to mention the dates these three cases were searched on. The date of search for c62 was 16-3-87, for c140 was 23-10-86, and for c287 was 22-4-87.

7.3.2 Methodology for search reconstruction

The methodology used to study and re-search the CIRT searches is described below. The emphasis throughout is on query expansion, e.g. how and where to implement it. The aim of Pilot-1 was to experiment with the different options that were available in order to get a better understanding of the process. Furthermore, the knowledge accrued from the Pilot cases will be implemented in the methodology of the *Experiment* proper.

The methodology included studying the CIRT evaluation searches, simulating the database environment in which these were originally searched, finding terms for query expansion, weighting and ranking the query expansion terms, and finally repeating the search including an iteration with query expansion. The query expansion terms were taken either from the INSPEC thesaurus or by analysing the relevant set of documents with the ZOOM facility on ESA. Separate searches were performed for each set of query expansion terms and the results of these searches were then compared. The aim of such comparison being to establish which is a better candidate source for supplying query expansion terms in this research.

[1] The Data-Star subfiles have been rearranged after the introduction of INSPEC2 in December 1990. These are: insp (1987 to date), in86 (1980-1986), in79 (1969-1979), and inzz (1969 to date).

The procedures of studying each search and then repeating the search is described below. Search c62/r62 will be used as the example to highlight the discussion whenever necessary.

The methodology can be summarised in the following steps:

Steps

1. Identify suitable CIRT evaluation searches
2. For each search, extract information necessary to allow repetition of search (query terms, relevance judgements, date of search)
3. Do both (a) and (b) below:
   - (a) Search in ESA
     - i. recreate database environment, so it is same as that of the date of the original search
     - ii. select documents
     - iii. analyse documents with ZOOM
     - iv. single term searches for all terms in ZOOM list
     - v. go to step 4
   - (b) Search INSPEC thesaurus for query expansion terms;
     go to step 5
4. Weight and rank all candidate terms for query expansion
5. Select terms for query expansion
6. Search using CIRT
   - (a) recreate database environment of original search in Data-Star
   - (b) repeat search
   - (c) add terms for query expansion and search
   - (d) print retrieved documents
7. Repeat steps 3 to 6 and each time add terms from a different source (relevant documents, INSPEC thesaurus, etc)
8. Evaluate results obtained from each run of step 7.

The records available for each of the three CIRT searches were carefully studied. The records include a set of the user and intermediary questionnaires, CIRT's user log, CIRT's network log, and the offline prints which were evaluated by the users for relevance judgements. Studying these records was necessary in order to establish a better understanding of the search. During this stage I also extracted the relevant information that was necessary for reconstructing the search. The collected information included the date of search; the database searched (i.e., insp, inzz); the limits, if any were imposed on the search; the terms used in the initial query; and the accession numbers (AN) of all records seen online and judged relevant during the initial search.

A weighted search using CIRT goes through the following stages:

initial search: is the first search that CIRT performs with the initial query terms.

first iteration search: is the search that CIRT performs after receiving relevance feedback information from the user but without adding any new terms.

query expansion search: is the search that CIRT performs after the addition of new search terms which are added to the initial query terms.

One source for query expansion terms was the set of records judged relevant by the user during the initial search. The other was the INSPEC thesaurus and is discussed later. The relevant records were then analysed by using ZOOM. The intention was to get the list of all the terms from the analysis of ZOOM and find the term frequency each term had on the date of the original search (16-03-87). Subsequently, the terms were assigned weights using a ranking formula (F4modified). A number of these terms were chosen for query expansion. The search was repeated (in Data-Star using CIRT) and the query expansion terms were added. Finally, the retrieved documents were printed and evaluated.

I was searching ESA and retrieving these records using their accession numbers. In search c62, for example, there were 8 documents that were judged relevant. Each one of them was searched as, e.g., `S NN=A86063402`, and these sets were then ORed together to form a single set (the relevant document set).

At this point in the search the reconstruction of the original database environment was tested. Search c62 was performed on the 16-03-87 and the following method was used to limit the search to that date:

Steps:	ESA commands:	
3-1	`?limit all/69-87`	(limits entire database up to the end of 1987)
3-2	`?expand up=87`	(gives the list of the 24 updates in 1987)
3-3	`?select e8-e27`	(select updates 8705-8724)
3-4	`?run SuperZOOM`	(use ZOOM to analyse the 8 relevant records)
3-5	`?select [term]`	(search one term in the ZOOM list)
3-6	`?combine [term of step 5] not [e8-e27]`	

Steps 3-5 and 3-6 had to be repeated for every term in the ZOOM list in order to restrict the search to the desired date (16-03-87). This is because at Step 3-5 the term frequency of a term is calculated up to the end of 1987, so at Step 3-6 the term frequency is restricted further to match that of the original search.

The entire search on ESA was captured in a file. The log file was then processed and specific information extracted from it. All the terms (as appear in the ZOOM list - step 3-4) and their term frequencies (as found in step 3-6) were extracted from the log file in order to calculate the weights of each term. This process was done with a program (script) that uses DOS-based UNIX utilities and UNIX shell programming. The programs that I wrote are UNIX KornShell scripts and C programs for the Mortice Kern Systems Toolkit which is a UNIX shell for DOS are described in Appendix B.4.1. The programs process an ESA log file by identifying and extracting the portions of the search that correspond to the ZOOM list, the terms that were actually searched, and those sets that give the term frequency of each term limited to the date of the original search. This information was then used by a C program that calculated the weights of each term (see Appendix B.4.2). The output of the C program was a ranked list of all candidate terms for query expansion.

The second source for query expansion terms was the INSPEC thesaurus. I took all the terms searched in the initial query and checked them in the INSPEC thesaurus (1987 in-house edition). For each term, if there was a match, I noted the term(s) it leads to. In this process I included all types of relationships given for the term, i.e., BT, NT, RT as well as USE – USE FOR. This provided a pool of candidate terms for query expansion.

Once the query expansion terms were selected I then used CIRT to log into Data-Star and search (step 6). The methodology used to reconstruct the search using CIRT and to add new terms was:
Steps:

```
 6-1  start CIRT
 6-2  set search size=50
 6-3  aof initial query terms of c62
 6-4  li dstar
 6-5  select database INZZ
 6-6  limits? not (8611.an. or 87$2.an. ···)
 6-7  add all
 6-8  search
 6-9  relevance judgements
6-10  add new terms (ZOOM, thesaurus.de.,etc.)
6-11  new weights (CIRT weighting)
6-12  search
6-13  print first 50 or all if less than 50
```

There were three separate query expansions searches for each CIRT evaluation search used in this case study. In each of the three searches the query expansion terms added in step 6-10 were taken from a different source and were searched either free-text or in the controlled vocabulary:

1. Terms from the combined CT and UT fields of the set of user-judged relevant documents. These fields were analysed by ZOOM and the terms were weighted and ranked.
2. Terms extracted from the INSPEC thesaurus and searched as free-text terms.
3. Terms extracted from the INSPEC thesaurus and searched in the descriptor field only.

In my efforts to reduce bias as much as possible I decided that I would first select the query expansion terms from each source and then search. The results of each of these searches were evaluated for their relevance to the initial request (step 8). Although my intention was to compare the original output to each of the new ones this was not possible because the data was not available. As mentioned in chapter 5 in the CIRT evaluation project if a search was retrieving more than 50 documents the user was given only 50, i.e., the first 25 and the last 25 of the retrieved set. For this reason I did not have the complete output of these searches and consequently I compared the output of the three new searches in terms of precision and overlap.

The sections that follow discuss this methodology in detail. Emphasis is placed on specific issues and on the reasons that led to particular decisions that were taken during the course of this study.

7.3.3 Search reconstruction

The experiments using the CIRT searches described in this chapter require that the database environment is replicated to match that of the time of the original search. In other words, when searching case c140, today the database should contain exactly the same number of records as it did at the time c140 was first searched, i.e. 23-10-86. The aim of wanting to control the database environment in this way is that it guarantees the exact replication of a search. This control can provide an environment which is in many respects similar to that of a test collection. In such a controlled environment I would then be able to try out different search strategies and add query expansion terms in order to observe the effect that these have on retrieval. This task is not as simple nor as easy as one would imagine. Reconstructing a search in Data-Star did not involve many problems, while reconstructing the same search in ESA was very difficult to accomplish. In order to achieve results that could be regarded as satisfying required interaction with the help-desk and the technical support people of both ESA and Data-Star as well as a lot of time experimenting with the online systems.

However, despite my numerous attempts to achieve this goal I can now say that this is almost impossible to accomplish. I could approach the original environment very closely but I could never exactly match it. There are many reasons that account for this and which are discussed in detail below.

Many searchers think that the same database, e.g. INSPEC, is the same irrespective of whether it appears on different online services, e.g. it will be the same whether on ESA/IRS or Data-Star. The main reason behind this assumption is that the database producer is responsible for the content of the database whereas online services just load the database in their system. Therefore under this assumption the only differences one could expect regarding INSPEC on ESA/IRS or Data-Star would be on the searching costs, the commands and the labels for the fields of each record. Unfortunately, these are not the only differences. Among the reasons that significantly affect the searching of the same database in each host are the differences in:

1. the processing and loading of the database
2. the initial set-up of the number of fields per record for the database by the host
3. the database tape loading schedules
4. the agreements signed between online services, database producers and primary publishers with respect to availability of certain publications.

The discussion below concentrates on only those aspects that relate to the INSPEC database in ESA and Data-Star.

7.3.3.1 INSPEC record fields

The structure and the number of fields per record for the INSPEC database is different between the two hosts. There are 19 fields per record in Data-Star and 26 fields in ESA/IRS. A complete list of the fields for each host is found in Appendix B.2.

The fields that I refer to most often are given the same field label in both hosts, e.g. title (TI), author (AU), abstract (AB). However, two fields are named differently and it seems appropriate to address this now as to avoid any confusion later on in the discussion.

The field that refers to the Accession Number of the INSPEC record is called AN in Data-Star and NN in ESA/IRS. The information indicating the origin of each record in the INSPEC database is available in the accession number field as, e.g. A83019439, where A refers to the INSPEC hard copy publication, here, the Physics Abstracts, 83 is the year and the 6-digit number that follows (i.e., 019439) is the unique number for this record assigned by INSPEC. The AN number can be searched and displayed.

INSPEC maintains a vocabulary which includes the thesaurus terms and a list of terms in the form of an uncontrolled vocabulary. These are referred to differently by the two hosts. The thesaurus terms are called descriptors (DE) in Data-Star and controlled terms (CT) in ESA/IRS while the terms in the uncontrolled vocabulary are called identifiers (ID) and uncontrolled terms (UT) respectively. For the remainder of this thesis I will use the terms DE or CT and ID or UT interchangeably.

7.3.3.2 Processing and loading of INSPEC

Online search services can choose from a variety of retrieval software. For example, Data-Star uses a modified version of the IBM STAIRS software whereas ESA/IRS uses a version of the RECON system developed by Lockheed. The software used by each host determines the way a database can be loaded (indexed).

ESA/IRS maintains two types of indexes, the *basic index* and the *additional indexes* as seen in Figure 7.1. The basic index for the INSPEC database contains words or phrases from the Abstract (AB), Title (TI), Controlled terms (CT), Uncontrolled terms (UT), and Corporate Source (CS) fields of the record. The additional index contains words or phrases from the remaining fields of an INSPEC record such as Author (AU), Document Type (DT), etc. The idea behind this division is that the basic index facilitates subject searches whereas the additional indexes are for non-subject related searches, e.g. locating a source or an author.

Data-Star, however, maintains a single index as seen in Figure 7.1. Terms and phrases from all 19 fields are put into this index and sorted in alphanumeric order. There is not an explicit division between terms representing subjects and those that do not, therefore all terms are interfiled.

The differences in loading the database as well as the uneven number of fields create problems in searching INSPEC in the two systems. These problems are exacerbated by the way the hosts index each field. For example, terms in a field can be indexed either as single words, as phrases, or both. The first two indexing methods will make only one entry per term in the alphanumeric index. The latter indexing approach is also referred to as *double posting* of the terms because it makes more than one entry for each term, so that the term `water absorber` will be indexed once as is, i.e. as a phrase, and then there will be separate entries for the terms `water` and `absorber`. ESA indexes the CT and UT fields with both word and phrase indexing whereas Data-Star indexes only the DE in that fashion.

When searching ESA/IRS using a term or a combination of terms (commands: SELECT, FIND) or when browsing the index (command: EXPAND) the database by default looks only in the basic index. For the system to access the additional indexes the user has to specify which index is to be searched, e.g. select au=abbott for an author search. In contrast, when searching or browsing (command: ROOT) in Data-Star the system looks into the one and only available index.

The effects that these different indexing approaches have on the final database index and consequently on searching can be seen by studying Figure 7.1. For example, the term water has a term frequency of 113246 in ESA as opposed to 117922 in Data-Star. Similarly, the term waters retrieves 4039 and 4205 records respectively. The terms water aberration, water absorber, waterfall effect and waters layer are not indexed as phrases by Data-Star because they are identifiers.

The effect on searching can also be demonstrated when truncation is used. If the term water is truncated to match any number of characters (e.g. water? or water$) it will retrieve very different terms in the two hosts. In ESA it will retrieve only subject terms from the basic index whereas in Data-Star it will retrieve any term that is in the index, be it a subject, an author's name, a journal title, etc.

ESA/IRS Basic Index		Data-Star Index		
113246	WATER	117922	WATER	
1	WATER ABERRATION	1	WATER-A-B	
7	WATER ABSORBER		WATER-ABERRATION	*Not Indexed*
115	WATERFALL		WATER-ABSORBER	*Not Indexed*
2	WATERFALL EFFECT	1	WATER-D-G-P	
1	WATERLAND	1	WATER-L-B	
12	WATERLOGGING	115	WATERFALL	
1	WATERPUMPS	1	WATERFALL-B	
4039	WATERS		WATERFALL-EFFECT	*Not Indexed*
1	WATERS LAYER	1	WATERFALL-F-D	
		1	WATERLAND	
Author Additional Index		2	WATERLAND-J-C	
1	AU=WATER, A.B.	12	WATERLOGGING	
1	AU=WATER, D.G.P.	1	WATERPUMPS	
1	AU=WATER, L.B.	4205	WATERS	
1	AU=WATERFALL, B.	1	WATERS-A-D	
1	AU=WATERFALL, F.D.	4	WATERS-B-E	
2	AU=WATERLAND, J.C.		WATERS-LAYER	*Not Indexed*
1	AU=WATERS, A.D.			*by Data-Star*
4	AU=WATERS, B.E.			

Figure 7.1: Index of the INSPEC database on ESA/IRS and Data-Star.

7.3.3.3 Tape loading schedules

Additional discrepancies of the same database when available in more than one host are caused by the different tape loading schedules. Hosts can receive the tape with the updates from the database producer at the same time. However, they load the updates in their system at different time intervals. This means that for some unspecified time (a few days to a few weeks) the database might have been updated in one host and not in the other. The effects on retrieval can therefore be significant, for example, the same query searched in both hosts at the same day will retrieve different results.

In my efforts to reconstruct the CIRT searches as closely to the original environment as possible I took into consideration the loading schedules of ESA and Data-Star. INSPEC produces 24 tapes per year.[2] ESA loads them fortnightly whereas Data-Star monthly. However, the hosts do not strictly adhere to this schedule. Therefore, the loading of the updates can cover more than one INSPEC tape. If this happens then the identification of one INSPEC update from another becomes very difficult. The only way to match updates between hosts is to study their schedules and manually calculate from these the size of the database in each host. This can then be used to limit the database as required. The method can give quite reasonable results.

Appendix B.3.2 presents the calendar of updates of the INSPEC database by the two hosts. In addition to the dates the appendix also presents the update number and the number of records added to the database. This information was necessary in order to be able to construct search statements that would limit the database to the appropriate size. Despite the good results obtained by this method exact replication of the search results was not possible.

7.3.3.4 Search reconstruction in Data-Star

In order to illustrate the process of reconstructing a search I will again use as an example search c62. The search took place on the 16-03-87 in the INZZ database. The search terms and the corresponding term frequencies were as follows: `finite element$1` 13886, `fluid$1` 54879, `interaction` 114714, `incompressible` 7594 and `navier stokes` 3420. The INSPEC database at that time had docz=2805999 records. It was thought that by using the limit command it would be relatively easy to recreate the same database size in the same host.

At first I tried to establish the exact INSPEC update that corresponded to the search data (16-03-87). Access to this information is not available online and it was therefore necessary to contact the Data-Star offices and request this information. The calendar of the Data-Star updates was obtained (see Appendix B.3.1) and the dates were matched to the information that is received from the ROOT command (`ROOT 87$2.an.`). From this information I could say that the search date (16-03-87) was before Data-Star loaded the update 8703 (21-03-87). Therefore, I should limit the database before update 8703.

After experimenting with a number of candidate methods two approaches seemed most promising.

Method 1: is to search for docz and then limit by selecting all records before a certain update, e.g.

[2] Jeff Pache, INSPEC, personal communication 1989.

```
D-S - SEARCH MORE - ENTER QUERY
1_: docz
2_: ..1/1 up<8703
```

Method 2: is to search for docz, then select all the updates after update 8702 and finally exclude the updates from docz, e.g.

```
D-S - SEARCH MORE - ENTER QUERY
1_: docz
2_: (8703 or 8704 8705 8706 8707 8708 8709 87$1).an. 88$2.an. 89$2.an.
3_: 1 not 2
```

Both methods are correct however Method 2 was preferred because it is much faster than Method 1.

Once the method was chosen I then searched the terms and compared the results to those of the original search. To my surprise these did not exactly match.

I started searching again by using other updates but I could not get the figures right. After studying the calendar of updates and consulting with Data-Star, I was informed that there are occasions where a tape may be loaded more than once or merged with previous updates. This was the case with update 8703. Data-Star loaded the INSPEC tapes on the 6/1/87 and 14/1/87, however, because of some technical problems these were merged and reloaded with newer tapes as 8703. Therefore, it would be impossible to recreate the exact environment for this search.

7.3.3.5 Search Reconstruction in ESA

Reconstructing the same database environment in ESA is not possible because of the different loading methods used by the hosts as mentioned earlier. Despite this I had to find ways to approach the original environment as closely as possible. The reason for this is that following the analysis of the retrieved records by using SuperZOOM I would then feed into the search all the SuperZOOM terms and get their term frequencies. The term frequency of each term would be used to calculate a weight and then rank all the terms in descending order. Therefore, a deviation from the original term frequency would result in weight estimation errors. This will affect the ranking of the terms and possibly the choice of the terms for query expansion.

Therefore, I looked at the updates online (EXPAND UP=86?) and at ESA's calendar of updating INSPEC which I received from Italy and tried to establish which update corresponded to each search (see Appendix B.3.2).

ESA does not have a command similar to the docz of Data-Star. Consequently, one cannot establish the exact size of the database at any given moment. This increases the difficulties of comparing the size of the database between the two hosts.

Limiting on ESA has its advantages and disadvantages. One can limit on whole years or ranges of years (e.g., limit set#/84, limit set#/69-86, limit all/69-87). However, a search can be limited only to one update period (e.g., l set #/UP) and therefore one cannot directly limit to a range of updates. Using c62 as example, the method employed to search ESA is the following:

```
-----------------------------------------------------------------------
? select NN=A86063402
...
? select NN=C86005359
? combine 1-11/OR
   1     8 1-11/OR         (select the 8 documents of search c62 to ZOOM on)
                                (Original date of search c62: 16-3-87)
? lall/69-87
           LIMIT ALL/69-87     (limit all subsequent searches to
                                  the period 1969-end of 1987)
? e up=87
           EXPAND UP=87
REF  ITEMS   INDEX-TERM
...
E3 *         UP=87
E4      7059 UP=8701
...     ...
E10     7853 UP=8707         (expand the update index for 1987 and
...     ...                   select updates 8707 to 8724
E27     ...                   which cover the period
                              mid-March to December 1987)
? s e10-e27
 2196042 E10-E27
        E3: UP=87

? run szs                       (run the SuperZOOM program and
Do you want to:                   ZOOM on set 1)
 1  submit the zoom command
 2  look at the zoom list
...
? 2
 1-38                            (select the first 38 of the 76 terms
  Accepted: 1-38                   of the ZOOM list)
? x

F NAVIER STOKES EQUATIONS
F FINITE ELEMENT ANALYSIS
...

3  3341 NAVIER(W)STOKES(W)EQUATIONS
4 11694 FINITE(W)ELEMENT(W)ANALYSIS
...
? 6
  SETSCROLLMODE Accepted                     (end of SuperZOOM session,
Program execution completed                    return to Quest)

? combine 3-2; c4-2;
 38  3028  3-2                  (by combining "set 3 NOT set 2" the
 39 10508  4-2                   results are now limited to the period
...                              1969 up to update 8707 which represents
```

```
                                the date of the original search)
? c37-2
 72   292 37-2

? .delete 3-72                (Because of the ESA limitation of 99 sets,
 DELETED SETS 3 - 72            delete sets 3-72, and
?                               continue the search by...

-----------------------------------------------------------------------
? 1all/69-87                    ...repeating the entire process
? run szs                          for the remaining 38 terms
? 39-76                            in the ZOOM list)
  Accepted: 39-76
? ...
-----------------------------------------------------------------------
```

7.3.3.6 Overall comments on limiting

Problems similar to those described above when searching ESA/IRS and Data-Star were encountered when replicating all of the searches. Despite the time and effort spent to replicate the exact database environment I came very close to the original results for most terms and in some occasions matched them, but I could not have an identical environment because of the changes incurred on the database. I had come as close to the original environment as one could get considering the circumstances and I was satisfied as I believed it would not distort my results. Having achieved this, I then conducted the searches and incorporated a stage of query expansion.

7.3.4 Problems

The existence of some problems were considered at the design of the research methodology. However, there were some additional unforseen problems of technical nature over which I had no control. A number of unanticipated problems occurred during the course of the CIRT projects which also affected the data collection or my PhD research (i.e. 1983-1990). The problems encountered related to:

- the York Box and the LSI 11/23 program size limitation
- accessing and searching Data-Star
- accessing and searching ESA/IRS
- telecommunications problems
- staffing
- capturing searches
- term deletion
- speed of search

- computer system crashes

In the discussion below I address only those problems that had a direct or indirect effect on my PhD research. Details of the effect of these problems on the CIRT projects are given in (Robertson & Thompson, 1987; Robertson & Thompson, 1990).

7.3.4.1 The York Box and the LSI 11 size limitation

Adapting CIRT to Release 2.1 of the York Box software during the CIRT evaluation project was problematic. This was mainly because of the process-size limitations.

The LSI 11/23 design of hardware has a built-in limitation on the size of program that can be run, namely 64 Kb of main memory. (This is not a Unix limitation: processes of any size can be run under Unix on 68000-based hardware. The limitation reflects the age of the LSI 11 design.) In practice, for a C program, this means 56 Kb of instruction and data space; the remainder is reserved for operating system use. (Hardware higher up in the LSI range allow instruction and data spaces to be separated, allowing 56 Kb for each however the Departmental LSI 11/23 does not come into this category.)

In any event, the LSI hardware could not be replaced because of lack of financial support. Consequently, any research involving CIRT had to be tailor-made to fit the above hardware limitations. Furthermore, the Department of Information Science did not have at the time any other hardware that such development could be done and there were not plans for acquiring any such hardware due to lack of funds. The Department of Information Science acquired its first SUN Sparc station in late 1989 and by then the data collection of the research reported in this thesis was well under way and near its completion.

7.3.4.2 Speed of search

When CIRT is doing a weighted search it translates the weighted search into a series of Boolean searches which are sent one by one to Data-Star, waiting each time for a response. This procedure is somewhat protracted, given typical Data-Star response times and transmission speeds. Thus, making CIRT's responses to the user slower than one would like.

On the whole communications are slow, even with the direct X25 connection. This may still have to do with network constraints, but also relates to the host's use of the network. This problem was exacerbated when searching ESA/IRS.

7.3.4.3 Term deletion

Terms can be deleted if they have not been searched by CIRT. If a term has been searched and then deleted its weight gets equal to zero thus making the term ineffective for searching. However, the searching algorithm has a bug and distinguishes between documents with or without the term, even if they have the same matching value (Robertson & Thompson, 1987, p.7).

7.3.4.4 System crashes

The LSI 11/23 was originally bought in 1981 and had been showing signs of age for some time. In addition, adequate backup was not available despite efforts to provide it. A tape system which would allow such backups was installed but malfunctioned shortly after its installation. Attempts to repair it were unsuccessful.

1. A major system crash occurred in March 1986, before the short-lived tape system was operational. A problem developed with the root disc system; the disc had to be reformatted. This delayed the CIRT evaluation project for two months. There have since been a number of serious problems.

2. The most serious problem however occurred during my data collection (1989). The disc controller of the LSI 11/23 failed and attempts to repair it were unsuccessful. The maintenance company announced that they no longer serviced LSI 11s. After an almost six month delay (May-October 1989) a compromise solution was reached. This involved using parts from the Department's LSI 11/73 computer to revive the LSI 11/23. A version of the LSI 11/23 operating system was installed in the LSI 11/73. Then, the disc controller of the 11/73 was installed on the 11/23. So, by 'merging' the two computers CIRT was revived and I was able to continue my data collection.

3. Another serious problem that causes CIRT to crash can occur during searching.

   This problem is caused by the searching algorithm when there are more than about eight terms to be searched because the system runs out of memory.

   Whenever a new search mode is to be created, a call to the function `malloc` grabs enough extra memory to accommodate it. In this way, the program grows in size until this space is freed by either logging out of Data-Star altogether or by reinitialising the structures using the command `reset`.

   However, there is a limit beyond which a program cannot grow. The effective limit to the size of a program on the LSI 11/23 using Unix Version 7 is 56K (the available address spaces is 64K, but 8K of this is used by the system). This limitation was the main reason for splitting CIRT into two processes. However, although it solved other problems it did not solve the search algorithm growth problem which can be reached with approximately eight search terms.

CIRT system crashes were a major impediment during my research. The delays suffered during the Pilot searches and especially during the data collection searches were great and created a difficult situation because of the end-users who had to be present throughout the search. Quite often only the determination of those end-users who wanted their searches completed enabled me to complete the data collection process.

7.3.4.5 Changes in Data-Star's transmission sequences

CIRT relies heavily on being able to detect various situations, i.e. type of data being sent from Data-Star, by pattern recognition via LEX. Such example patterns are:

```
(the D-S prompt)        _:
                        SEARCH MODE
(the D-S field labels)  AU
                        TI
                        AB
```

Minor changes in such patterns can cause major problems in a system like CIRT. Apparently, during my data collection (in 1989) Data-Star changed its sequences. It was a change in the characters sent at the login stage which did not affect any human users. However, it was a major problem for CIRT because the LEX program could not recognise the patterns and consequently was failing to connect to Data-Star. It took a lot of troubleshooting before it was discovered that Data-Star had made the change.

This type of problem cannot be easily alleviated because hosts do not notify their users when such changes take place, since these changes have often effect only on output screen formatting.

7.3.4.6 Problems Searching ESA

Telecommunication Problems

When I started searching ESA in 1988 a telecommunications problem occurred which disrupted my data collection for a considerable amount of time.

The network connection to access ESA (or any other host) from my computer at the Department was via X25 lines. The Departmental PAD (Packet Assembly Disassembly) was connected to City University's CPSE-X25-link to University of London Computer Centre (ULCC) and from there via JANET and other networks to ESA in Frascati, Italy. The problem that I encountered was intermittent and totally unpredictable. Whenever it occurred it resulted in 'hanging' the PAD and thus disabling any communication between my computer and the ESA computer.

Most of the time this problem occurred at the login stage although it did happen during searching as well. The immediate result was the disabling of the Departmental PAD completely, thus requiring the PAD to be reset in order to allow communication to any other destination.

Despite the investigation of this problem by the network engineers of both the City University's Computer Unit and the ULCC the cause could not be found. The Dialtech representatives in London were particularly helpful but their efforts were also fruitless. The cause was found when ESA network engineers in Frascati monitored the line while I was attempting to login to the ESA computer. It was then established that ESA's transmission of the M-bit was causing the PAD to hang. An M-bit is part of an OCTET of the header of a Data Packet that is transmitted over the network. The M-bit is the '`more data`' mark and it is a 1-bit field; when it is 1, additional packets follow and when it is 0, no more packets will follow this unit. The M-bit sent by the ESA computer was apparently upsetting a hardware device in the City University's Computer Unit which in turn was causing the PAD to hang.

An additional cause was that the PAD line was badly configured. That hardware device was eventually replaced by a newer model but it took almost a year between the first occurrence of the problem and the replacement of that hardware. Apparently, as I was informed by the Dialtech staff, this same problem occurred in other British Universities when trying to access ESA via JANET.

ESA software problems

There were some additional problems relating to ESA's search software limitations which when seen in isolation or from a human user's perspective could be regarded as insignificant. However, a closer look at these limitations and their effect on searching demonstrates that these can affect searching capabilities especially when a front-end is used. Some of these limitations and the problems that these can create are described below:

1. Each ESA user is allocated a fixed amount of disc space which is equivalent to 2.4 million records. When the search reaches this mark then the message `'disk storage overflow'` notifies the user that the search cannot be continued. In order to continue searching the user would have to delete a number of search statements which can effectively destroy a search session.

2. The size of each set cannot exceed a maximum of 2 million records.

3. The maximum number of sets that can be created in a session is 99 (sets 1-98 plus the KEEP set which is set 99).

   These three limitations are very restrictive for doing the type of searches CIRT requires. In addition, the output from SuperZOOM usually exceeds this limit. Consequently, feeding terms back to the search has to be done in blocks of terms, so, that a block of X terms (e.g., 40) is searched, the results noted, and then sets are deleted so that a new block of terms can be searched.

4. Another problem is caused by a limitation of the maximum number of characters a search statement is permitted to have. When using the `FIND` command the line length of each statement is limited to 40 characters and statements longer than that return error messages such as `'Term string too long'` or `'Invalid adjacency'`. This problem is particularly noticeable when SuperZOOM is feeding phrases back to the search because it does not have a mechanism to compensate for this.

Problems using SuperZOOM

While I was experimenting with SuperZOOM and investigating which record fields would be the candidate sources of the query expansion terms, I came across a bug in SuperZOOM that could create many problems when searching. The bug was that when in SuperZOOM mode, a request to ZOOM on more than one field, e.g., `'z ti, ct, ut'`, resulted in ZOOM-ing only on the first field, e.g. here the TI. This problem was reported to ESA and was fixed as of 22 May 1989.

Another problem that I discovered was that SuperZOOM seems to have a limit on the number of terms that can feed into the search at any one time. The limit is 50 terms and if this limit is exceeded SuperZOOM crashes returning the message `'Array subscript out of range'` and also destroys the entire search session up to that point because the only way out of it is to logoff ESA. This problem could not be fixed and consequently I carefully avoided this problem during my researching.

7.3.5 Source of query expansion terms and term selection

In interactive query expansion as opposed to automatic query expansion there are two parties responsible for determining and selecting terms for the expansion. One is the retrieval system itself which, as in the case for automatic query expansion, is designed to select terms from a number of predetermined fields of the bibliographic record and then weigh and rank those terms. The other is the user, who is presented with the ranked list of terms and has to decide which are the terms to be added to the search.

So, in a weighted search with interactive query expansion there is a joint responsibility of system and user. The reasons of the success or failure of the search becomes therefore even more difficult to pin-point because in interactive query expansion we are increasing the uncontrollable variables.

If we temporarily exclude the matching function (which is discussed in section 7.4), among the remaining variables that affect the search are the source of terms and the relevance judgements of these terms during the term selection process for query expansion.

Both variables are very important and often influence the search results in quite unpredictable ways. Therefore, I aimed first to investigate the possible sources that I could use to get the terms for query expansion. Then I repeated the search and each time I added query expansion terms from a different source. The results were then compared in order to establish which source to use when searching on behalf of real users during the study proper.

Sources

Of the four methods for term selection that were described in section 3.5.1 (page 35) the one that seemed to be the most appropriate method that could be adopted in this investigation, combines the original query terms with terms from documents retrieved and judged relevant online.

However, besides the user-judged relevant set of documents there are also some other sources for query terms that can be considered. One such source is the INSPEC thesaurus. A question that can be raised is, if a searcher does not look into the retrieved set of documents, and does not analyse it with ZOOM, but instead takes the initial query terms and goes directly to the INSPEC thesaurus, 'where in the thesaurus are the query terms going to lead the searcher?' and 'do the initial query terms suggest any new terms in the thesaurus?' By using the INSPEC thesaurus (1987 in-house edition), I aimed at taking the original query terms, look each one of them in the thesaurus, and follow the terms that were suggested.

This process generates a number of terms from which some can be then used for query expansion. The terms chosen for inclusion were then searched once in the descriptor field and once as free text terms.

Therefore, in this pilot case study, which investigates query expansion as part of CIRT, I used these two different sources for supplying the search with new terms.

The two sources supplied three sets of query expansion terms which were then searched separately. The retrieved sets were then evaluated in terms of overlap of retrieved records and of their relevance to the subject searched.

Returning to the relevant document set as the source of the query expansion terms we are faced with two questions. The first is concerned with the way that the relevant document set is analysed in order to extract there terms.

Such analysis is usually based on term occurrence information. It is for this reason that ZOOM was selected for this research, since it is available online on a host and can be used by a front-end, like CIRT, for query expansion.

The other question concerns the document representations which can be used as the source of the terms. In other words which record field(s) like TI, DE, ID, AB, etc. are to be used in searching and therefore to be analysed by ZOOM.

The effectiveness of different document representations in retrieval has been the subject of numerous investigations in IR research. However, there is not a document representation that has distinguished itself from all the others in terms of recall or precision (Katzer *et al*, 1982, p.262). Most conclusions are rather non-specific and have added little to the understanding of why document representations work the way they do in retrieval. The only clear finding from previous research on controlled vocabulary and free text searching is that different representations tend to retrieve different documents (Svenonius, 1986).

When searching ESA the options available, in terms of record fields as sources of query expansion terms, are limited to the fields of the basic index. For the INSPEC database these fields are TI, AB, CT, UT, and CS.

In investigating which of these would be suitable for my research I used ZOOM to analyse the fields, singly and in combination. For the analysis I used search c68 which is not contained in the sample of searches for Pilot 1. This was decided in order to reduce the possibility of searching bias. Had I used any from the projected Pilot 1 searches at this stage I would have formed an opinion that would have influenced my subsequent searches and increased the bias in my conclusion.

Search c68 has a relevance feedback set of four documents. These documents were located in ESA and analysed using the ZOOM command on the following field combinations:[3]

[3] ZOOM is discussed in detail in section 4.3.1

	ZOOM command	Pages.Lines	Comment
1	z	1.7	default fields: CT, UT
2	z words	2.6	
3	z ti		compete titles
4	z w ti	1.13	
5	z ab		first line of the abstract
6	z w ab	10.16	
7	z w ti ab ct ut	11.10	
8	z w ti ct ut	2.16	
9	z ti ct ut	1.11	

The analysis obtained from the title and abstract fields was the complete titles and part of the first line of each abstract (up to 80 characters) respectively and were therefore rejected as unsuitable.

Combinations 2, 4, 6, and 8 present a list of single terms because all phrases are being split into their constituent parts. The list also includes stopwords whenever abstracts are included. As seen from 6 and 7 a large number of terms is generated (e.g., pages.lines 10.16 and 11.10) mainly because of the inclusion of the abstract.

Combinations 1 and 9 present the terms as they appear in the fields, i.e., as phrases or single terms. In combination 9, as with 3, the titles are presented in the form of a single entry and for this reason it was also excluded.

Having looked carefully at these options and the way the terms are presented I decided to ZOOM on the CT and UT fields as phrases (combination 1). The important factor that influenced the decision to choose phrases as opposed to single words is that during interactive query expansion in a real search phrases will be more easily identified by the users with respect to their meaning rather than if they are split into single terms.

An example that illustrates this point is that the phrases "colour coding", "colour vision" and "hue matching" would convey more information to the user than the single terms "colour", "hue", "coding", "vision" and "matching" would. In addition it would be easier for a user to choose a word by taking it out of a phrase rather than create a phrase by looking at a list of single words. The latter point becomes very difficult because the ZOOM analysis ranks terms according to their frequency of occurrence. Therefore, the terms "colour" and "coding" might be a few screens apart thus decreasing the likelihood of being found and combined by a user.

Term selection

The selection of terms for the expansion, from the different sources mentioned above, was an intellectual process that varied depending on the situation. Whether terms were taken from the INSPEC thesaurus or from the relevant documents I was evaluating them by simulating the user as closely as possible. The artificiality involved during this simulation stems from the use of the search intermediary's knowledge and the subject knowledge during the term selection process.

To illustrate this further I will refer to the selection of query expansion terms, from the relevant document set, for search r140 (page 120). A user with no searching knowledge might have selected any of the terms presented in the list. However, myself, just by looking at the list I excluded the first three terms at the top. The reason for not considering these terms is that their frequency is $n = 1$ and therefore these terms cannot retrieve any additional documents.

Furthermore, from the information given in that search, the initial query terms, my understanding of what was required after having studied the retrieved documents, and the terms in the list, I chose terms that I though were good for the search irrespective of their rank position. For example, after selecting `automatic vehicle classification` and `automatic vehicle identification` I chose `road traffic` which is at the bottom of the list. That is because the other terms were judged either as general or too specific. Therefore, I was selecting terms as a user would, i.e. based on subject knowledge, but I was also combining this knowledge with other information.

Similar approaches that rely on subject or expert knowledge were followed in the selection of terms in all of the searches. These I allowed to vary depending on the search requirements.

7.3.6 Relevance judgements

Through the Pilot case studies relevance judgements (a) for the suitability and choice of terms for query expansion and (b) for the evaluation of the search results were my responsibility.

This was a difficult task because besides the fact that relevance is by nature very much a subjective process, I did not have available to me enough information about the individual requests.

During the CIRT evaluation project the intermediaries and the end users had to fill in a number of questionnaires. Unfortunately, most searchers did not keep any detailed information either with respect to the pre-search interview process or with their own decision making process of term selection. There were only a few instances where a detailed description of the topic was provided. In the majority of the cases where this information is given search topics were described in a one line sentence. This makes much of the analysis rather difficult.

In addition to the information supplied by the questionnaires and the pre-search interview, I also based my judgements on the relevance information that was provided on the offline prints by the users. For each of the three searches I studied all documents, both relevant and non-relevant, in order to form a better understanding of the search and of what was requested.

Therefore, I based relevance judgements purely on my own subjective intuition, just like any search intermediary who searches on a topic without the user being present during the search process. However, in cases where I felt in doubt about the subject matter and of what was requested I sought the expert opinion of researchers in that subject area. For example, when searching r62 I asked the expert opinion of a researcher from the Department of Civil Engineering, City University.

Once the three searches for each case study were completed I evaluated the results and assigned relevance judgements. In doing this I followed any leads that I had from the study of the CIRT evaluation searches. For example, in search c287 the user has marked as non-relevant all foreign language material. Therefore, in order to compare the results, I did the same to the results obtained for r287. At first, relevance was decided on a binary scale, so that a retrieved record was either relevant or not relevant. In subsequent searches however I used a scale with three levels so that a document could be relevant, partially relevant, or not relevant.

7.4 Ranking algorithms and term selection for query expansion

Interactive query expansion requires a term selection stage where the system presents the query expansion terms to the user in some reasonable order. The order should preferably be one in which the terms that are most likely to be useful are close to the top of the list. In addition, heuristic decisions can also be applied during this stage, for example, poor terms are excluded from the term list instead of being given low weights.

This section discusses term weighting and ranking as it relates to term selection for query expansion and only within the context of this investigation.

In information retrieval various formulae have been proposed which attempt to quantify the value or usefulness of a query term in retrieval. Formulae may estimate the term value based on some qualitative or quantitative criteria. The qualitative arguments are concerned with the "value" of the particular term in retrieval whereas the quantitative argument may involve some specific criterion such as a proof of performance. The relevance weighting theory discussed in section 2.2.2.1 is an example of the latter.

Robertson (1986) suggested a modification to the F4point-5 formula (referred to as F4modified) which takes into consideration the addition of new terms to the original query. The modified formula which is also derived form the same 2×2 table as the F4 (see page 16) is:

$$w_t = \log \frac{(r+c)(N-n-R+r+1-c)}{(n-r+c)(R-r+1-c)} \tag{7.1}$$

where $c = n/N$ and r, R, n, N are the same as defined in the F4.

It was further suggested that the F4modified could be used in two ways (Robertson 1986, p.86). In automatic query expansion every term from the relevant document would be weighted using $c = n/N$ and added to the search. In interactive query expansion the terms would be weighted in the same fashion and those selected by the user would then be weighted with the F4point-5 formula.

Another aim of my research was to test the F4modified by using it as the weighting function of the query expansion terms. Apart form the theoretical argument for using the F4modified there were not any empirical data available. Therefore, any first step was to compare its behaviour against that of the F4point-5 and look at what effect these formulae have on the ranking of the terms. The weighting of all the terms of the three searches in this pilot case study (c62, c140, c287) gave rather unexpected results. An example of the weighing of the terms of search c287 is given in Table 7.1.

The table presents the terms found in the eight relevant documents of search c62. In the left column the weights of the terms have been calculated using the F4modified formula and in the right column by the F4point-5 formula. Besides the terms and their weights the table also gives the frequency of the term (n) in the collection and the number of times that the term occurs in the relevant document set (r), which was used by the formulae to calculate the weights.

By observing the two ranked lists one can very easily establish that they are almost identical. Out of the 52 terms in the list only 5 terms have different ranks and the difference is negligible (these terms have been marked with " * ". At the top of the lists are terms with low collection frequency (n) as well as low frequency in the relevant document set (r). At the bottom of the list are terms with high collection frequencies and which are ranked in ascending order (from low to high frequencies).

This type of ranking, however, is because of the nature of the relevance weighting theory which assigns a higher degree of importance to the low frequency terms and therefore brings them at the top of the ranked list.

What is known from IR research with respect to the relationship that holds between term frequency and term value and the effect on retrieval can be summarised as follows (Sparck Jones, 1971; Salton, 1975; van Rijsbergen, 1979):

- very frequent terms are not very useful,
- middle frequency terms are quite useful,
- infrequent terms are likely to be useful but not as much as the middle frequency terms,
- very infrequent terms are useful terms in the sense that when they are present are good indicators of relevance. However, since these terms are not present for most of the time they do not help in retrieving very many documents.

From this knowledge it can, therefore, be hypothesised that a good term ranking algorithm would bring the middle frequency terms near the top of the list.

7.4.1 In search of a term ranking algorithm for query expansion

The almost identical ranking of the F4modified to the F4point-5 was both a surprise and a disappointment because it was thought that the former was an improvement on the latter. This led to the decision to investigate the suitability of other formulae that have been proposed in the literature as candidate algorithms for the weighting of terms for query expansion in this research project. There is a plethora of ranking algorithms reported in the literature (Sager & Lockmann, 1976; McGill *et al*, 1979;

f4modified				f4			
weight	r	n	term	weight	r	n	term
27.65	1	1	vhf tetrode transmitters	13.95	1	1	vhf tetrode transmitters
27.65	1	1	uhf tv equipment	13.95	1	1	uhf tv equipment
27.65	1	1	th 582 20 kw tetrode	13.95	1	1	th 582 20 kw tetrode
27.65	1	1	pyrobloc pyrolitic graphite grids	13.95	1	1	pyrobloc pyrolitic graphite grids
27.65	1	1	low thermal emission	13.95	1	1	low thermal emission
27.65	1	1	fast standby operation	13.95	1	1	fast standby operation
27.65	1	1	aural and visual conditions	13.95	1	1	aural and visual conditions
27.65	1	1	50 kw tetrode	13.95	1	1	50 kw tetrode
12.79	1	2	uhf tv tetrodes	12.85	1	2	uhf tv tetrodes
12.79	1	2	no secondary emission	12.85	1	2	no secondary emission
12.09	1	3	tv klystron efficiency	12.34	1	3	tv klystron efficiency
12.09	1	3	pyrobloc grids	12.34	1	3	pyrobloc grids
12.09	1	3	performance at uhf	12.34	1	3	performance at uhf
12.09	1	3	coaxial tetrodes	12.34	1	3	coaxial tetrodes
12.09	1	3	coaxial cavity circuit	12.34	1	3	coaxial cavity circuit
12.09	1	3	10 1 kw	12.34	1	3	10 1 kw
11.40	1	5	vapour phase cooling	11.75	1	5	vapour phase cooling
11.40	1	5	low expansion coefficients	11.75	1	5	low expansion coefficients
11.18	1	6	20 2 kw	11.55	1	6	20 2 kw
10.99	1	7	klystrode	11.38	1	7	klystrode
10.84	1	8	hypervapotron cooling	11.24	1	8	hypervapotron cooling
10.59	1	10	* tv amplifier	11.02	8	801	* television transmitters
10.35	6	189	* tetrodes	11.00	1	10	* tv amplifier
10.27	8	801	* television transmitters	10.61	6	189	* tetrodes
10.15	1	15	high power tetrodes	10.58	1	15	high power tetrodes
10.08	1	16	anode dissipation	10.51	1	16	anode dissipation
9.89	1	19	collector depression	10.34	1	19	collector depression
9.84	1	20	band iv v	10.28	1	20	band iv v
9.38	1	31	uhf tv transmitters	9.84	1	31	uhf tv transmitters
9.27	4	220	tv transmitters	9.49	4	220	tv transmitters
9.00	1	45	rf losses	9.46	1	45	rf losses
8.35	1	85	tv stations	8.82	1	85	tv stations
8.23	1	96	compact construction	8.70	1	96	compact construction
8.18	1	101	band iii	8.65	1	101	band iii
7.85	1	140	safety margin	8.32	1	140	safety margin
7.31	1	239	high thermal conductivity	7.78	1	239	high thermal conductivity
7.21	1	266	beam modulation	7.67	1	266	beam modulation
7.05	1	312	video amplifiers	7.51	1	312	video amplifiers
7.02	2	733	* electron tubes	7.48	1	322	* rf signal
7.01	1	322	* rf signal	7.31	2	733	* electron tubes
6.29	2	1510	radio equipment	6.59	2	1510	radio equipment
6.06	1	837	klystrons	6.53	1	837	klystrons
5.93	2	2171	television equipment	6.23	2	2171	television equipment
5.58	1	1346	basic design	6.05	1	1346	basic design
5.49	2	3355	television broadcasting	5.79	2	3355	television broadcasting
4.73	1	3163	test data	5.20	1	3163	test data
4.61	2	8095	uhf	4.91	2	8095	uhf
4.36	2	10448	tuning	4.67	1	5342	vhf
4.20	1	5342	vhf	4.65	2	10448	tuning
2.47	1	30265	gain	2.93	1	30265	gain
2.37	1	33468	cooling	2.83	1	33468	cooling
2.07	1	45012	reliability	2.52	1	45012	reliability

Table 7.1: QE terms for r287 ranked by F4modified and F4point-5.

Ro, 1988). The ones that I selected to examine were the ZOOM term frequency ranking, F4point-5, F4modifed, EMIM, Porter's and $w(p-q)$. The first three algorithms have already been discussed whereas the last three are introduced below.

Porter's algorithm

Porter and Galpin (1988) in the MUSCAT online catalogue used the following ranking formula:

$$Porter = \frac{r}{R} - \frac{n}{N} \tag{7.2}$$

where r, R, n, N are defined as in the F4point-5 weight (see page 16). In the paper there is not any justification given for the formula nor any explanation of how they arrived at it. However, looking at the formula it can be established that the weight is influenced by a term's occurrence (r) in the relevant document set (R) as well as the term's frequency (n) in the collection (N). Because this formula is used to rank terms for query expansion the r/R portion:

1. never becomes 0,
   because in order to use the formula there must always be at least one document containing the term that has been judged relevant, and
2. has a maximum value of 1,
   this happens whenever $r = R$

In other words, this portion of the weight can take values that fall within the range:

$$0 < \frac{r}{R} \leq 1$$

The n/N fraction however is influenced by a term's frequency (n). So that, the higher the term frequency (n) the higher the result of the fraction. The Porter formula seems to place more emphasis on terms that occur frequently in the relevant document set.

The EMIM algorithm

EMIM (expected mutual information measure) is a term weighting model incorporating relevance information in which it is assumed that index terms may not be distributed independently of each other. (van Rijsbergen, 1977; Harper & van Rijsbergen, 1978; van Rijsbergen, Harper & Porter, 1981)

$$EMIM = E_{iq} = \sum_{t_i, w_q} \Delta_{iq} P(t_i, w_q) \log \frac{P(t_i, w_q)}{P(t_i)P(w_q)} \tag{7.3}$$

or more generally

$$G_{iq} = \sum_{t_i, w_q} \Delta_{iq} D_{iq} P_{iq}$$

where

t_i indicates the presence (1) or absence (0) of a term,

w_q indicates that a document is relevant (1) or non-relevant (0),

Δ_{iq} indicates the value of a term as a relevance discriminator, and it is 1 if $t_i = w_q$ or -1 if $t_i \neq w_q$,

D_{iq} is the "degree of involvement", i.e. one of the four cells of the 2 × 2 contingency table (see page 16), and

P_{iq} is the "probabilistic contribution" given by the log expression.

EMIM reduces to the F4 weight when the "degree of involvement", i.e. the joint probabilities, are all unity. Assuming the same definitions for *n, N, r, R,* as those already used earlier, the EMIM weight of a term is calculated as follows:

$$\begin{aligned} E_{iq} &= p_{11}i_{11} - p_{12}i_{12} - p_{21}i_{21} + p_{22}i_{22} \\ &= \log\frac{rN}{Rn}\cdot r \\ &\quad - \log\frac{(n-r)N}{(N-R)n}\cdot(n-r) \\ &\quad - \log\frac{(R-r)N}{(N-n)R}\cdot(R-r) \\ &\quad + \log\frac{(N-n-R+r)N}{(N-n)(N-R)}\cdot(N-n-R+r) \end{aligned}$$

The EMIM weight is claimed to make use of the statistical dependence between index terms over the entire collection.

The $w_t(p_t - q_t)$ algorithm

The results obtained from my empirical investigation of the F4point-5 and F4modified meant that some other algorithm based on the relevance weighting theory might offer a better alternative. As a result of the discussion regarding the behaviour of these two formulae Robertson developed an algorithm which I used for the empirical testing in this research. The theoretical arguments supporting the algorithm have been reported in Robertson (1990b).

The independence assumption of the relevance weighting theory is that terms are distributed independently of each other in relevant documents and also that terms are distributed independently of each other in non-relevant documents (see page 15).

In the query expansion stage of search an additional assumption should be made which considers statistical independence between the query expansion term and the terms in the entire previous search formulation.

The distribution of the relevant items for the initial formulation, for example, is further divided into two distributions: one which describes the relevant items that contain the term and one that describes the relevant item that do not contain the term. These two new distributions are identical according to the independence assumption. In other words, the presence or absence of the query expansion term does not affect the initial distribution. When the entire collection is considered, these assumptions predict a positive association between an initial query formulation and a good new term for query expansion.

The inclusion of term t in the search formulation with weight w_t will increase the effectiveness of retrieval by

$$a_t = w_t(p_t - q_t) \tag{7.4}$$

where

w_t is a weighting function, which in this case is the F4point-5,

p_t is the probability of term t occurring in a relevant document, and

q_t is the probability of a term t occurring in a non-relevant document.

This means that irrespective of the weighting function (w_t) used the rule for deciding the inclusion of a term in a query expansion search should be based on the ranking of a_t instead of w_t. Substituting the weighting function and the probability of relevance in a_t with r, R, n, N we get:

$$a_t = \log \frac{(r+.5)(N-n-R+r+.5)}{(n-r+.5)(R-r+.5)} \cdot (\frac{r}{R} - \frac{n-r}{N-R}) \tag{7.5}$$

The relevance weighting theory attempts, via the Probability Ranking Principle (Robertson, 1977b), to optimise the entire length of the search curve from high-precision to high-recall. This is expressed by the w_t component of the above formula which assigns greater importance to the infrequent terms. However, a model that determines which term(s) to add for query expansion will have to lead to a preference somewhere between the very infrequent terms that lead to high precision and the frequent terms that lead to high recall.

In the above formula this is achieved by $p_t - q_t$. This component, like the Porter formula (7.2), is influenced by the frequency of occurrence of a term in the relevant document set as well as the term's frequency in the collection. Therefore, the multiplication of the two components results in the effect which seems to be required by the model for term selection in query expansion.

7.4.1.1 Selecting a ranking algorithm for query expansion

Having identified the six ranking algorithms I then looked at the way these rank the terms of searches c62, c140 and c287. The purpose of this test was to find out, in general terms and irrespective of value, how much difference there is between the six algorithms. The results of this test should be treated with caution and were only indicative of the performance of the algorithms in this context. These results were however important as I used them to help me decide which formula to use for ranking the terms for query expansion in the remainder of this research.

Although I used all six algorithms to rank the terms from the relevant document set of the three searches (c62, c140, c287) in the discussion below I will only use search c287 as an example. Table 7.2 and Table 7.3 present the terms of the relevant document set of search c62 which are ranked according to ZOOM, Porter and EMIM, $w(p-q)$ algorithms respectively.

The ranking of ZOOM, in Table 7.2is based on the within document frequency. Furthermore, within ties terms are ranked in alphabetical order.

The ranking of Porter's algorithm as seen in Table 7.2 is predominantly affected by the frequency of occurrence of the term in the relevant set. A term's collection frequency, which is also considered by the algorithm, assists in the further ranking of the terms. Its effect is seen on the way it handles ties which are resolved by ranking the terms according to their term frequency.

In Table 7.3 the terms are ranked using EMIM and $w(p-q)$. The lists are very similar. This can be explained by looking at the components of the two formulae. Both are influenced by the F4point-5 weight as well as the term's frequency (r) in the relevant document set (R).

Another way of looking at the effect of ranking from these algorithms is to look at Table 7.4. For each term the table presents the rank position of that term as given by each of the six algorithms.

The pattern that emerges from this table is that $w(p-q)$ and EMIM are very similar as are F4point-5 and F4modified. Porter's is more similar to the $w(p-q)$ and EMIM whereas ZOOM's ranking does not resemble any of the others (chiefly because of the alphabetical method of tie-breaking).

This method of looking at the ranking of terms can be misleading because it can overlook important information. For example, in trying to establish the difference between the algorithms by looking at the way these suggest terms for query expansion one needs the information conveyed by r and n. As it has been mentioned a good ranking algorithm should bring to the top of the list the middle frequency terms. Therefore, when looking at the six ranked lists (Tables 7.1, 7.2, 7.3) with this in mind, in order to select the algorithm that looks best, the following can be observed. F4point-5 and F4modified look bad because of the emphasis they place on the low frequency terms. ZOOM also looks bad because it cumulates the within document frequencies over documents. This is strongly influenced by, and strongly correlated with, r. Porter's appears to fall between ZOOM on the one side and $w(p-q)$ and EMIM on the other. Thus, $w(p-q)$ and EMIM look the best of all these algorithms since they seem to bring out some good terms and on this basis were the main candidate algorithms.

Selecting a ranking algorithm for information retrieval experimentation process can be difficult. There are a number of interacting factors (both theoretical and empirical) which may be relevant to the choice of a formula. I selected $w(p-q)$ on the grounds of the analysis of the behaviour of the six algorithms presented here, and because Porter and Galpin (1988, p.11) and van Rijsbergen, Harper and Porter (1981, p.82) have acknowledged that they have no theoretical justification for the weighting formulae they have suggested (i.e., Porter and EMIM).

The adoption of $w(p-q)$ has been made largely on subjective grounds because there is no firm evidence that it performed better than some of the other ranking formulae. A positive sign however, towards its adoption is its theoretical justification.

ZOOM ranking			Porter			
r	n	term	weight	r	n	term
8	801	television transmitters	0.9997194810	8	801	television transmitters
6	189	tetrodes	0.7499338101	6	189	tetrodes
4	220	tv transmitters	0.4999229536	4	220	tv transmitters
2	733	electron tubes	0.2497432954	2	733	electron tubes
2	1510	radio equipment	0.2494711815	2	1510	radio equipment
2	3355	television broadcasting	0.2492396921	2	2171	television equipment
2	2171	television equipment	0.2488250423	2	3355	television broadcasting
2	10448	tuning	0.2471650425	2	8095	uhf
2	8095	uhf	0.2463409962	2	10448	tuning
1	16	anode dissipation	0.1249996498	1	1	vhf tetrode transmitters
1	1	aural and visual conditions	0.1249996498	1	1	uhf tv equipment
1	101	band iii	0.1249996498	1	1	th 582 20 kw tetrode
1	20	band iv v	0.1249996498	1	1	pyrobloc pyrolitic graphite grids
1	1346	basic design	0.1249996498	1	1	low thermal emission
1	266	beam modulation	0.1249996498	1	1	fast standby operation
1	3	coaxial cavity circuit	0.1249996498	1	1	aural and visual conditions
1	3	coaxial tetrodes	0.1249996498	1	1	50 kw tetrode
1	19	collector depression	0.1249992996	1	2	uhf tv tetrodes
1	96	compact construction	0.1249992996	1	2	no secondary emission
1	33468	cooling	0.1249989494	1	3	tv klystron efficiency
1	1	fast standby operation	0.1249989494	1	3	pyrobloc grids
1	30265	gain	0.1249989494	1	3	performance at uhf
1	15	high power tetrodes	0.1249989494	1	3	coaxial tetrodes
1	239	high thermal conductivity	0.1249989494	1	3	coaxial cavity circuit
1	8	hypervapotron cooling	0.1249989494	1	3	10 1 kw
1	7	klystrode	0.1249982489	1	5	vapour phase cooling
1	837	klystrons	0.1249982489	1	5	low expansion coefficients
1	5	low expansion coefficients	0.1249978987	1	6	20 2 kw
1	1	low thermal emission	0.1249975485	1	7	klystrode
1	2	no secondary emission	0.1249971983	1	8	hypervapotron cooling
1	3	performance at uhf	0.1249964979	1	10	tv amplifier
1	3	pyrobloc grids	0.1249947468	1	15	high power tetrodes
1	1	pyrobloc pyrolitic graphite grids	0.1249943966	1	16	anode dissipation
1	45012	reliability	0.1249933460	1	19	collector depression
1	45	rf losses	0.1249929958	1	20	band iv v
1	322	rf signal	0.1249891435	1	31	uhf tv transmitters
1	140	safety margin	0.1249842405	1	45	rf losses
1	3163	test data	0.1249702321	1	85	tv stations
1	1	th 582 20 kw tetrode	0.1249663798	1	96	compact construction
1	10	tv amplifier	0.1249646287	1	101	band iii
1	3	tv klystron efficiency	0.1249509705	1	140	safety margin
1	85	tv stations	0.1249162996	1	239	high thermal conductivity
1	1	uhf tv equipment	0.1249068439	1	266	beam modulation
1	2	uhf tv tetrodes	0.1248907342	1	312	video amplifiers
1	31	uhf tv transmitters	0.1248872321	1	322	rf signal
1	5	vapour phase cooling	0.1247068734	1	837	klystrons
1	5342	vhf	0.1245286161	1	1346	basic design
1	1	vhf tetrode transmitters	0.1238922828	1	3163	test data
1	312	video amplifiers	0.1231291732	1	5342	vhf
1	3	10 1 kw	0.1144008661	1	30265	gain
1	6	20 2 kw	0.1132791405	1	33468	cooling
1	1	50 kw tetrode	0.1092363055	1	45012	reliability

Table 7.2: QE terms for r287 ranked by the ZOOM and Porter algorithms.

EMIM				$w(p-q)$			
weight	r	n	term	weight	r	n	term
87.42	8	801	television transmitters	11.0	8	801	television transmitters
76.49	6	189	tetrodes	8.0	6	189	tetrodes
51.55	4	220	tv transmitters	4.7	4	220	tv transmitters
24.57	2	733	electron tubes	1.8	2	733	electron tubes
22.76	2	1510	radio equipment	1.7	1	1	vhf tetrode transmitters
22.13	1	1	vhf tetrode transmitters	1.7	1	1	uhf tv equipment
22.13	1	1	uhf tv equipment	1.7	1	1	th 582 20 kw tetrode
22.13	1	1	th 582 20 kw tetrode	1.7	1	1	pyrobloc pyrolitic graphite grids
22.13	1	1	pyrobloc pyrolitic graphite grids	1.7	1	1	low thermal emission
22.13	1	1	low thermal emission	1.7	1	1	fast standby operation
22.13	1	1	fast standby operation	1.7	1	1	aural and visual conditions
22.13	1	1	aural and visual conditions	1.7	1	1	50 kw tetrode
22.13	1	1	50 kw tetrode	1.6	2	1510	radio equipment
21.87	1	2	uhf tv tetrodes	1.6	2	2171	television equipment
21.87	1	2	no secondary emission	1.6	1	2	uhf tv tetrodes
21.85	2	2171	television equipment	1.6	1	2	no secondary emission
21.57	1	3	tv klystron efficiency	1.5	1	3	tv klystron efficiency
21.57	1	3	pyrobloc grids	1.5	1	3	pyrobloc grids
21.57	1	3	performance at uhf	1.5	1	3	performance at uhf
21.57	1	3	coaxial tetrodes	1.5	1	3	coaxial tetrodes
21.57	1	3	coaxial cavity circuit	1.5	1	3	coaxial cavity circuit
21.57	1	3	10 1 kw	1.5	1	3	10 1 kw
21.08	1	5	vapour phase cooling	1.5	1	5	vapour phase cooling
21.08	1	5	low expansion coefficients	1.5	1	5	low expansion coefficients
20.88	1	6	20 2 kw	1.4	2	3355	television broadcasting
20.75	2	3355	television broadcasting	1.4	1	6	20 2 kw
20.71	1	7	klystrode	1.4	1	7	klystrode
20.55	1	8	hypervapotron cooling	1.4	1	8	hypervapotron cooling
20.27	1	10	tv amplifier	1.4	1	10	tv amplifier
19.75	1	15	high power tetrodes	1.3	1	15	high power tetrodes
19.66	1	16	anode dissipation	1.3	1	16	anode dissipation
19.43	1	19	collector depression	1.3	1	20	band iv v
19.36	1	20	band iv v	1.3	1	19	collector depression
18.74	1	31	uhf tv transmitters	1.2	2	8095	uhf
18.51	2	8095	uhf	1.2	1	31	uhf tv transmitters
18.21	1	45	rf losses	1.2	1	45	rf losses
17.85	2	10448	tuning	1.1	2	10448	tuning
17.28	1	85	tv stations	1.1	1	85	tv stations
17.10	1	96	compact construction	1.1	1	96	compact construction
17.03	1	101	band iii	1.1	1	101	band iii
16.54	1	140	safety margin	1.0	1	140	safety margin
15.75	1	239	high thermal conductivity	1.0	1	239	high thermal conductivity
15.59	1	266	beam modulation	1.0	1	266	beam modulation
15.35	1	312	video amplifiers	0.9	1	322	rf signal
15.30	1	322	rf signal	0.9	1	312	video amplifiers
13.87	1	837	klystrons	0.8	1	837	klystrons
13.16	1	1346	basic design	0.8	1	1346	basic design
11.86	1	3163	test data	0.6	1	3163	test data
11.05	1	5342	vhf	0.6	1	5342	vhf
8.23	1	30265	gain	0.3	1	30265	gain
8.05	1	33468	cooling	0.3	1	33468	cooling
7.50	1	45012	reliability	0.3	1	45012	reliability

Table 7.3: QE terms for r287 ranked by the EMIM and $w(p-q)$ algorithms.

$w(p-q)$	EMIM	F4	F4mod	Porter	ZOOM	Term
1	1	22	24	1	1	television transmitters
2	2	24	23	2	2	tetrodes
3	3	30	30	3	3	tv transmitters
4	4	40	39	4	4	electron tubes
5	6	1	1	10	48	vhf tetrode transmitters
6	7	2	2	11	43	uhf tv equipment
7	8	3	3	12	39	th 582 20 kw tetrode
8	9	4	4	13	33	pyrobloc pyrolitic graphite grids
9	10	5	5	14	29	low thermal emission
10	11	6	6	15	21	fast standby operation
11	12	7	7	16	11	aural and visual conditions
12	13	8	8	17	52	50 kw tetrode
13	5	41	41	5	5	radio equipment
14	16	43	43	6	7	television equipment
15	14	9	9	18	44	uhf tv tetrodes
16	15	10	10	19	30	no secondary emission
17	17	11	11	20	41	tv klystron efficiency
18	18	12	12	21	32	pyrobloc grids
19	19	13	13	22	31	performance at uhf
20	20	14	14	23	17	coaxial tetrodes
21	21	15	15	24	16	coaxial cavity circuit
22	22	16	16	25	50	10 1 kw
23	23	17	17	26	46	vapour phase cooling
24	24	18	18	27	28	low expansion coefficients
25	26	45	45	7	6	television broadcasting
26	25	19	19	28	51	20 2 kw
27	27	20	20	29	26	klystrode
28	28	21	21	30	25	hypervapotron cooling
29	29	23	22	31	40	tv amplifier
30	30	25	25	32	23	high power tetrodes
31	31	26	26	33	10	anode dissipation
32	33	28	28	35	13	band iv v
33	32	27	27	34	18	collector depression
34	35	47	47	8	9	uhf
35	34	29	29	36	45	uhf tv transmitters
36	36	31	31	37	35	rf losses
37	37	49	48	9	8	tuning
38	38	32	32	38	42	tv stations
39	39	33	33	39	19	compact construction
40	40	34	34	40	12	band iii
41	41	35	35	41	37	safety margin
42	42	36	36	42	24	high thermal conductivity
43	43	37	37	43	15	beam modulation
44	45	39	40	45	36	rf signal
45	44	38	38	44	49	video amplifiers
46	46	42	42	46	27	klystrons
47	47	44	44	47	14	basic design
48	48	46	46	48	38	test data
49	49	48	49	49	47	vhf
50	50	50	50	50	22	gain
51	51	51	51	51	20	cooling
52	52	52	52	52	34	reliability

Table 7.4: Rank position of the terms in search r287 for each of the 6 algorithms.

7.5 Results and discussion

The three searches r140, r62 and r287, that constitute this pilot case study are discussed in this section. Each search is presented separately. The summary of the request characteristics is followed by a brief description of the selection of the query expansion terms and the three query expansion searches performed for each case study. All the searches were performed according to the methodology described earlier.

7.5.1 Search r140

A summary of the request characteristics of search c140 is presented here. This information has been obtained from the questionnaires that were filled by the user and the search intermediary. Additional information about the search process was extracted from the search logs.

Topic: Axle loading in pavement design.

Date of search: 23-10-86; **Database:** insp, inzz; **docz=N=** 1327757; **R=** 11

User characteristics: Post-graduate student in the Department of Civil Engineering, Highways and Transport. At the pre-search questionnaire, the user assessed the nature of his enquiry as precise and indicated that he wanted only a few specific references, i.e. precision search. He did not have any previous experience with online literature searching.

User's comments: The user's overall assessment of the search was that it was difficult and that the number of references retrieved was less than expected. It was also indicated that the results obtained were poor.

Intermediary's Comments: The intermediary indicated that overall this was a poor search. The search process was difficult but the results, i.e. the one document retrieved, seemed satisfactory. According to the intermediary the search "found all there appeared to be".

Apparently there was only one document retrieved in the initial search. Any additional documents retrieved in subsequent searches were ignored. A closer look at the search reveals the following.

At the pre-search interview nine terms were identified:

```
pavement adj design     axle adj load  axle adj load adj distribution
commercial adj vehicle  pavement       developing adj countries
heavy adj vehicle$1     highways       roads
```

However, only three were selected to be used in the search. The initial query was searched in the INSP database (1980-1986) and consisted of the terms:

freq.	weight	term
5	12.4	axle adj load
29	10.7	heavy adj vehicle$1
41	10.4	pavement

Search tree for the initial search of c140

No.	seen	weight	terms
1	0	23.1	axle adj load heavy adj vehicle$1
4	0	12.4	axle adj load
28	0	10.7	heavy adj vehicle$1

The search retrieved the three sets with 1, 4 and 28 records respectively. However only the first set with one document combined any of the query terms. The other two sets were retrieved by single terms. `Axle load` which is an important term for this search has a collection frequency of 5 documents and all of them were retrieved in the first two sets.

The way this search was conducted is of a surprise. After looking at the first document which was judged relevant the searcher did not continue to browse but reset CIRT and searched the INZZ database (1969-1986).

The same search was repeated in INZZ and because it did not give any better results searching was terminated. This apparently was decided solely by looking at the search tree because the searcher did not look at any of the retrieved documents.

QE terms drawn from the relevant document set

The CT and UT fields of the document that was judged relevant were analysed with ZOOM and the terms were then ranked using the $w(p-q)$ formula. For the calculation of the weights the values N=1327757 and R=1 were used.

rank	rel	freq	term
15.8908	1	1	freight traffic monitoring
15.8908	1	1	dynamic axle load measurement
15.8908	1	1	cargo shipments
13.6935	1	5	automatic vehicle classification
13.0575	1	9	automatic vehicle identification
12.2797	1	19	fleet management
11.4245	1	44	help system
11.1989	1	55	road traffic control
9.8852	1	203	economic feasibility
8.6937	1	665	traffic computer control
8.1639	1	1126	road traffic
7.5850	1	1998	computerised monitoring

The terms automatic vehicle classification, fleet management, automatic vehicle identification and road traffic were chosen for the query expansion search. The general approach to term selection was discussed in section 7.3.5. A summary of the search is given below:

Summary of search r140rel – QE terms from ranked list
(There are 1 known relevant documents)

new wt.	rels.	term
13.69	1	axle adj load
11.85	1	heavy adj vehicle$1
9.27	0	pavement
		Query Expansion Terms
13.69	1	automatic adj vehicle adj classification
13.06	1	automatic adj vehicle adj identification
12.00	1	fleet adj management
8.14	1	road adj traffic

Search tree for QE search r140rel
Source of QE terms: relevant document

No.	weight	terms
3	21.84	automatic adj vehicle adj classification, road adj traffic
3	21.20	automatic adj vehicle adj identification, road adj traffic
5	20.14	fleet adj management, road adj traffic
10	19.99	heavy adj vehicle$1, road adj traffic
5	17.42	pavement, road adj traffic
		Sets printed offline (total 27 docs)
4	13.69	axle adj load
1	13.69	automatic adj vehicle adj classification
5	13.06	automatic adj vehicle adj identification
19	12.00	fleet adj management

The search retrieved a total of 55 documents. However, 27 were only printed offline (1 from the initial search and 26 from the query expansion search) because the remaining documents were retrieved by single terms.

Apparently, the term axle adj load retrieved a set of 4 documents. This was not printed, however, for two reasons. The same set was retrieved in the original set but had been rejected without looking at any of the retrieved records and I have come to the conclusion that sets retrieved by single terms should not be considered at all.

Query expansion terms derived from the INSPEC thesaurus

All nine terms that were discussed at the pre-search interview were looked up in the INSPEC thesaurus. None of then however was found in the thesaurus neither as main term nor as a load-in term. Since most of these terms were multi-word terms it was decided to look each word that makes up the term separately in the thesaurus and see whether any useful terms can be suggested. Out of all these words there were entries only for `load distribution`, `load`, and `vehicles`. However, the two former terms were related to electricity in the thesaurus.

The terms that were chosen for query expansion from the INSPEC thesaurus were `road`, `traffic`, `vehicle$1`, `load`, and `distribution`. These terms were then used in two separate query expansion searches: one where these were searched free-text and another where these were searched in the descriptor field. A summary of the two searches and the corresponding trees are:

Summary of QE search r140ftx
QE terms from INSPEC thesaurus & searched free-text

(There are 1 known relevant documents)

old wt.	new wt.	rels.	terms
13.69	13.69	1	axle adj load
11.85	11.85	1	heavy adj vehicle$1
9.27	9.27	0	pavement
....	*Query Expansion Terms*	....	
5.06	7.26	1	road
4.14	6.34	1	traffic
3.86	6.06	1	vehicle$1
3.00	5.20	1	load
1.67	1.67	0	distribution

Search tree for QE search r140ftx
Terms from INSPEC thesaurus & searched free-text

No.	seen	weight	terms
1	0	33.19	heavy adj vehicle$1 road traffic vehicle$1 distribution
1	0	32.22	axle adj load road vehicle$1 load
10	0	31.51	heavy adj vehicle$1 road traffic vehicle$1
1	0	30.61	pavement road traffic vehicle$1 distribution
5	0	28.94	pavement road traffic vehicle$1
1	0	26.85	heavy adj vehicle$1 road vehicle$1 distribution
2	0	25.17	heavy adj vehicle$1 road vehicle$1
2	0	24.86	road traffic vehicle$1 load
1	0	24.25	heavy adj vehicle$1 traffic vehicle$1
1	0	23.11	heavy adj vehicle$1 vehicle$1 load
1	0	22.88	pavement road traffic
2	0	22.60	pavement road vehicle$1
1	0	21.68	pavement traffic vehicle$1
....		*Sets printed offline (total 30 docs)*	....
26	0	21.34	road traffic vehicle$1 distribution

This search retrieved a total of 55 documents. However, only 30 documents were printed offline (1 from the initial search and 29 from the query expansion search). This is because I felt that the term combination that retrieved the set with the 26 documents did not pertain to the query.

Summary of QE search r140de
Terms from INSPEC thesaurus & searched in DE field
(There are 1 known relevant documents)

old wt.	new wt.	rels.	terms
13.69	13.69	1	axle adj load
11.85	11.85	1	heavy adj vehicle$1
9.27	9.27	0	pavement
.... *Query Expansion Terms*			
5.62	7.82	1	road.de.
4.95	4.95	0	load.de.
4.87	7.07	1	traffic.de.
4.68	4.68	0	vehicle$1.de.
4.12	4.12	0	distribution.de.

Search tree for QE search r140de
Terms from INSPEC thesaurus & searched in DE field

No.	seen	weight	terms
1	0	28.85	pavement road.de. traffic.de. vehicle$1.de.
10	0	26.74	heavy adj vehicle$1 road.de. traffic.de.
1	0	26.20	axle adj load road.de. vehicle$1.de.
2	0	24.35	heavy adj vehicle$1 road.de. vehicle$1.de.
3	0	24.16	pavement road.de. traffic.de.
1	0	21.78	pavement road.de. vehicle$1.de.
.... *Sets printed offline (total 19 docs)*			
42	0	19.57	road.de. traffic.de. vehicle$1.de.

This search retrieved a total of 61 documents. However, only 19 documents were printed offline (1 from the initial search and 18 from the query expansion search) because the term combination that retrieved the set with 42 documents was too general to be relevant to this request.

7.5.1.1 Analysis

This was the very first search for this study and therefore every step of it was a learning experience. For this reason it was unfortunate that it was also a particularly difficult search. Difficult because of the lack of information about it and the fact that there was only 1 document printed offline. In addition to the subject difficulty and the fact that there was only one document printed offline, there was no information available from the research interview. Unfortunately, the significance of these data was not realised at the time of the CIRT evaluation project and they were not available for my indepth analysis.

After the completion of the query expansion searches four sets of documents were formed that contained:

c140: the documents retrieved in the original search.

r140rel: documents retrieved using query expansion terms from the relevant document set.

r140de: documents retrieved using query expansion terms from the INSPEC thesaurus and searched in the descriptor field.

r140ftx: documents retrieved using query expansion terms from the INSPEC thesaurus and searched free text.

(this convention for naming the retrieved sets has been followed for all searches in this pilot case study.)

Each of these sets was then evaluated for its relevance to the original request. I read all the records in each one of these sets and decided their relevance on a binary scale, i.e. relevant, not relevant. In retrospect, I can say that I was rather strict in judging relevance in a binary scale. For this reason in the subsequent searches (i.e. 287, 62) relevance was evaluated on a three-level scale (yes - partially - not).

The accession numbers (AN) of the retrieved records in each of the four searches, including the originally retrieved relevant documents, are given in Appendix B.5. The results of the query expansion searches could be summarised as follows. Set r140rel retrieved 27 documents and 6 were judged relevant. Set r140de retrieved 19 documents with 1 document judged relevant and set r140ftx retrieved 30 documents 2 of which were relevant. The precision of these searches was 22%, 6% and 7% respectively. These figures are rather low and should be attributed to the overall difficulties encountered in this search. When looking at the results the highest precision is given by the set retrieved with query expansion terms taken from the relevant document set (r140rel).

The next step in the analysis of the results compares the output of these searches in terms of absolute overlap of documents without any reference to relevance.

The overlap in terms of documents between all the searches of pilot case 140 is presented in Appendix B.6. Overlap was measured by examining the retrieved sets in all searches, i.e. c140, r140rel, r140de, r140ftx. It was calculated in absolute numbers only, that is without considering relevance. Documents, in the table in the appendix, are represented by their accession number (AN).

The four searches retrieved a total of 77 documents. This total is made up of 41 unique documents. Since search c140 had only 1 document printed offline there was only 1 core document for all four searches. Therefore, the overlap shown is basically that between the query expansion searches. There were 14 documents appearing in the three of the query expansion sets, which accounts for 34% of the 41 unique documents or for 18% overlap of the total. There were 5 documents appearing in any two of the three query expansion sets, and 21 documents that could be found only in one of the retrieved sets. The 21 documents that appear only once consisted of 11 from set r140rel and 10 from set r140ftx. Search r140de retrieved 19 documents, 15 common to all sets and 4 common with r140ftx.

7.5.2 Search r62

The characteristics of search c62 are summarised below. This information was extracted from the search logs and the questionnaires filled before and after the search.

Topic: No topic was given.
However, after considering the query terms and reviewing the relevance judgements it seems that the user wanted:

"to solve Navier Stokes equations assuming 2-D incompressible flow. One method for solving Navier Stokes equations with the above assumption is the finite element method".

Date of search: 16-3-87; **Database:** inzz; **docz=N=** 2805999; **R=** 8

User characteristics: Post-graduate student in the Department of Aeronautics, Imperial College, London. At the pre-search questionnaire the user assessed the nature of her enquiry as precise and indicated that she wanted a broad search, i.e., all references on the subject including peripheral material. She did not have any previous experience with online literature searching.

User's comments: The user responses in the post-search questionnaire indicate that the results were good, the search was fairly close to the original request, and that the number of references retrieved was about as expected.

Intermediary's comments: It was indicated that the level of difficulty of the search was average (three point scale) and that the results were good.

The search was done in the INZZ database and the initial query terms were:

term	freq.	weight
finite adj element$1	13886	5.3
fluid$1	54879	3.9
structure$1	344920	2.0
interaction	114714	3.2
incompressible	7594	5.9
navier adj stokes	3420	6.7

This search had three feedback iterations with 23 positive relevance judgements online, 8 of which were found during the initial iteration. Furthermore, in the third iteration query expansion was attempted by including the term two adj dimension$2 or 2d. This search retrieved a total of four documents that contained the term but all of them had already been seen by the user. Because of this attempt I decided that this term should be included in my query expansion searches. There were 73 documents retrieved but only 50 were printed as defined by the CIRT project. The search tree of the initial search was:

Search tree for initial search of c62

No.	seen	weight	terms
3	0	25.0	navier adj stokes incompressible finite adj element$1 fluid$1 interaction
2	0	23.8	navier adj stokes incompressible finite adj element$1 fluid$1 structure$1
55	0	21.8	navier adj stokes incompressible finite adj element$1 fluid$1

Query expansion terms drawn from the relevant document set

The CT and UT fields of the eight documents that were judged relevant online during the initial search were analysed with ZOOM. For the calculation of the weights the values of N=2805999 and R=8 were used. The list of the terms as ranked by $w(p-q)$ is:

rnk	r	wt	freq	term	rnk	r	wt	freq	term
9.6	8	9.6	3028	navier stokes equations	1.2	2	5.1	6194	laminar flow
8.3	8	8.4	10508	finite element analysis	1.2	1	9.8	31	steady viscous flow
2.9	2	11.6	11	broyden method	1.2	1	9.7	34	finite element comp
2.5	2	10.2	41	two dimensional incompressible flow	1.2	1	9.6	38	zero viscosity
2.1	3	5.8	4965	finite element method	1.1	1	9.2	52	complex flows
2.0	3	5.5	7160	physics computing	1.1	2	4.6	10174	boundary layers
1.9	2	7.8	414	newton raphson method	1.1	1	8.9	75	nonlinear solutions
1.7	1	13.9	1	standard six node triangle	1.1	1	8.8	78	incompressible visc
1.7	1	13.9	1	piccard method	1.0	1	8.7	89	elastically supporte
1.7	1	13.9	1	nonsymmetric linear problem	1.0	1	8.6	94	simple harmonic m
1.7	1	13.9	1	nine nodes quadrilateral element	1.0	1	8.5	112	cavity flow
1.7	1	13.9	1	nachos	1.0	1	8.5	114	subdomains
1.7	1	13.9	1	monochromatic dynamic behaviour	1.0	1	8.4	117	constant viscosity
1.7	1	13.9	1	macro element structure	1.0	1	8.2	150	harmonic oscillatio
1.7	1	13.9	1	incompressible fluids flows	1.0	1	8.1	159	iteration methods
1.7	1	13.9	1	harmonic navier stokes problem	1.0	1	8.0	179	complex geometry
1.7	1	13.9	1	free convection form	0.9	1	7.7	249	high reynolds numb
1.7	1	13.9	1	bulb function	0.9	1	7.5	290	shear layers
1.7	1	13.9	1	asea version	0.9	1	7.5	292	incompressible visc
1.6	1	12.8	2	finite difference finite elements	0.8	1	7.1	452	euler equations
1.5	1	12.3	3	penalty finite element model	0.8	1	7.0	495	lagrange multipliers
1.5	1	12.3	3	2d incompressible flows	0.8	1	6.8	607	periodic boundary
1.4	1	11.9	4	uzawa algorithm	0.8	1	6.6	717	incompressible flow
1.4	1	11.9	4	isothermal form	0.8	1	6.6	718	solid body
1.4	1	11.7	5	curved isoparametric elements	0.7	1	5.6	1948	incompressible fluid
1.4	1	11.5	6	sf sub 6 gas flow	0.6	1	5.3	2656	software package
1.4	1	11.2	8	quadratic basis functions	0.6	1	5.2	2898	shear flow
1.3	1	10.9	10	spectral element method	0.5	1	4.8	4579	flow instability
1.3	1	10.9	10	preprocessor program	0.5	1	4.6	5390	sulphur compounds
1.3	2	5.4	4437	fluid dynamics	0.5	1	4.4	6545	iterative methods
1.3	1	10.8	11	pressure evaluation	0.4	1	3.5	15654	convection
1.3	1	10.5	15	high velocity gradients	0.4	1	3.5	16604	viscosity
1.2	1	10.3	19	pressure calculations	0.2	1	2.0	71908	stability
1.2	1	10.3	19	fluid mechanics problems	0.1	1	1.4	127692	flow

Because the initial query consisted of six terms I could only add up to a maximum of another three terms due to CIRT constraints. However, my choice was further limited to two terms since the term `two dimension$2` was used for query expansion in the original search and I had decided that it should be part of any subsequent query expansion. Therefore, the query expansion terms chosen were:

query expansion term	frequency
`two adj dimension$2 or 2d`	45977
`incompressible adj fluid$1`	2169
`incompressible adj flow$1`	936

A summary of the query expansion search and the search tree is given below:

Summary of search r62rel – QE terms from ranked list
(There are 8 known relevant documents)

old wt.	new wt.	rels.	terms
9.54	9.54	8	navier adj stokes
8.75	8.75	8	incompressible
8.12	8.12	8	finite adj element$1
6.75	6.75	8	fluid$1
2.20	2.20	2	interaction
0.35	0.35	1	structure$1
.... *Query Expansion Terms*			
5.19	8.02	4	incompressible adj flow$1
4.34	6.22	2	incompressible adj fluid$1
1.28	5.72	7	two adj dimension$2 or 2d

Search tree for search r62rel
Source of terms: relevant documents

No.	seen	weight	terms
3	0	53.12	navier adj stokes incompressible finite adj element$1 fluid$1 incompressible adj flow$1 incompressible adj fluid$1 two adj dimension$2 or 2d
1	0	47.40	navier adj stokes incompressible finite adj element$1 fluid$1 incompressible adj flow$1 incompressible adj fluid$1
5	0	46.90	navier adj stokes incompressible finite adj element$1 fluid$1 incompressible adj flow$1 two adj dimension$2 or 2d
10	0	45.10	navier adj stokes incompressible finite adj element$1 fluid$1 incompressible adj fluid$1 two adj dimension$2 or 2d
5	0	45.00	navier adj stokes incompressible fluid$1 incompressible adj flow$1 incompressible adj fluid$1 two adj dimension$2 or 2d
10	0	41.18	navier adj stokes incompressible finite adj element$1 fluid$1 incompressible adj flow$1
8	0	40.15	navier adj stokes incompressible finite adj element$1 incompressible adj flow$1 two adj dimension$2 or 2d
1	0	39.63	navier adj stokes incompressible fluid$1 structure$1 incompressible adj flow$1 incompressible adj fluid$1
10	0	39.38	navier adj stokes incompressible finite adj element$1 fluid$1 incompressible adj fluid$1

The search retrieved a total of 62 documents (8 in the initial search and 54 in the query expansion search). All documents were then printed and judged for relevance.

Query expansion terms derived from the INSPEC thesaurus

All the terms in the initial query, except for `interaction`, `structure` and `incompressible`, have entries in the thesaurus. However, these terms are found as part of multi-word descriptors that are associated with the subject, e.g. incompressible fluid, flow interaction, fluid structure.

The two terms chosen for query expansion from the thesaurus were flow and fluid structure. These were searched once with the descriptor term two-dimension$2 or 2d.de. and once free text with the term two adj dimension$2 or 2d. The terms and their frequencies were:

two-dimension$2 or 2d.de.	2075
two adj dimension$2 or 2d	45977
fluid-structure$	204
flow.de.	54547

A summary of the two searches and the corresponding trees are:

Summary of QE search r62ftx – QE terms searched free text
(There are 8 known relevant documents)

old wt.	new wt.	rels.	terms
9.54	9.54	8	navier adj stokes
8.75	8.75	8	incompressible
8.12	8.12	8	finite adj element$1
6.75	6.75	8	fluid$1
2.20	2.20	2	interaction
0.35	0.35	1	structure$1
.... *Query Expansion Terms*			
6.71	6.71	0	fluid-structure$
1.28	5.72	7	two adj dimension$2 or 2d
1.10	3.48	3	flow.de.

Search tree for QE search r62ftx
QE terms from INSPEC thesaurus & searched free text

No.	seen	weight	terms
11	0	42.36	navier adj stokes incompressible finite adj element$1 fluid$1 two adj dimension$2 or 2d flow.de.
15	0	38.87	navier adj stokes incompressible finite adj element$1 fluid$1 two adj dimension$2 or 2d
1	0	38.60	incompressible finite adj element$1 fluid$1 interaction structure$1 fluid-structure$ two adj dimension$2 or 2d
9	0	36.64	navier adj stokes incompressible finite adj element$1 fluid$1 flow.de.
1	0	36.43	navier adj stokes incompressible fluid$1 interaction two adj dimension$2 or 2d flow.de.
6	0	35.61	navier adj stokes incompressible finite adj element$1 two adj dimension$2 or 2d flow.de.
.... *Sets printed offline (total 46 docs)*			
54	0	34.23	navier adj stokes incompressible fluid$1 two adj dimension$2 or 2d flow.de.

This search retrieved a total of 100 documents. 46 of them were printed offline (8 from the initial search and 38 from the query expansion search) in order to simulate the 50 document limit of the original CIRT search.

Summary of QE search r62de - QE terms searched in DE field
(There are 8 known relevant documents)

old wt.	new wt.	rels.	terms
9.54	9.54	8	navier adj stokes
8.75	8.75	8	incompressible
8.12	8.12	8	finite adj element$1
6.75	6.75	8	fluid$1
2.20	2.20	2	interaction
0.35	0.35	1	structure$1
.... *Query Expansion Terms*			
6.71	6.71	0	fluid-structure$
4.39	4.39	0	two-dimension$2 or 2d.de.
1.10	3.48	3	flow.de.

Search tree for QE search r62de - Terms from INSPEC thesaurus & searched in DE field

No.	seen	weight	terms
20	0	36.64	navier adj stokes incompressible finite adj element$1 fluid$1 flow.de.
38	0	33.15	navier adj stokes incompressible finite adj element$1 fluid$1

This search retrieved a total of 61 documents (8 from the initial search and 53 from the query expansion search). All documents were printed offline and judged for relevance.

7.5.2.1 Analysis

The accession numbers (AN) of all records retrieved in the original search (c62) and the three query expansion searches (r62rel, r62de, r62ftx) are given in Appendix B.7.

Set c62 had 50 records and the query expansion searches retrieved 62 (r62rel), 61 (r62de) and 46 (r62ftx) records each. The offline prints of search c62 had been evaluated by the user. The relevance judgements provided by the user were used in evaluating the query expansion searches. Because I had to provide relevance judgements for the three sets I decided that for consistency the records that appear in more than one set should be assigned the same relevance judgements. Since the user had provided relevance judgements on the offline prints I started from that set and matched it against the new sets. Then, I evaluated set r62rel and I matched the results to the two remaining sets. This process was continued till all sets were evaluated. The results of the relevance judgements were:

	Relevance judgements for search:							
	c62		r62rel		r62de		r62ftx	
Relevance	docs	%	docs	%	docs	%	docs	%
YES	19	38	20	32	17	28	11	24
PAR	26	52	26	42	32	52	23	50
NO	5		16		12		12	
Total	50		62		61		46	

The precision of the three query expansion searches based on the (YES) judgements shows that query expansion terms that were taken from the relevant document set (r62rel) performed slightly better. When the results from the documents judged as partially relevant (PAR) are combined with the precision becomes 74%, 80% and 74% for r62rel, r62de and r62ftx respectively. However, the precision obtained from the original search (c62) seems to be better.

The overlap in terms of documents between all the searches of pilot case 62 is presented in Appendix B.8. Overlap was measured by examining the retrieved sets in all searches, i.e. c62, r62rel, r62de, r62ftx, and it was calculated in absolute numbers only, i.e. without considering relevance. Documents, in the appendix, are represented by their accession number (AN).

The four searches retrieved a total of 219 documents. This total is made up of 93 unique documents. There were 17 core documents appearing in all four searches which accounts for 18% of the 93 unique documents or for 31% overlap of the total retrieved. There were 29 documents appearing in any three of the retrieved sets (31% of the unique), 17 documents appearing in any two of the sets (18% of the unique), and 30 documents that were found only in one of the retrieved sets (32% of the unique).

Another way to measure overlap is by considering the overlap between the original search (c62) and each of the query expansion searches and to take into account the relevance judgements. The results of such analysis are presented in the table below:

Document	Original search	Query expansion searches								
	c62	r62rel			r62de			r62ftx		
Relevance		overlap	new	total	overlap	new	total	overlap	new	total
YES	19	14	6	20	14	3	17	9	2	11
PAR	26	16	10	26	20	12	32	12	11	33
NO	5	4	12	16	4	8	12	2	10	12
Total	50	34		62	38		61	23		46

The retrieved records from c62 were matched, for example, against those of r62rel. I then checked how many records of the former search were retrieved by the latter and what was the proportion of the relevant, partially relevant and non-relevant records. So, from the 19 relevant documents (YES) of c62 14 of them were retrieved by r62rel, which has a total of 20 relevant documents. Overall, r62rel and r62de seem to be similar with respect to their overlap with c62 as well as with the proportion of retrieved records in terms of relevance.

7.5.3 Search r287

The information gathered from study of the questionnaires, the search logs and the offline prints of search c287 is summarised as:

Topic: No topic was given.

From the study of the query terms and the online and offline relevance judgements I concluded that the user was interested in:

"Broadcast transmitters, especially television (TV) transmitters. Broadcast transmitters provide the radio-frequency (RF) power that is radiated. There are solid-state, tetrode, and klystron amplifiers. From the query terms it seems that the user's interest concentrated in high power amplifiers, especially tetrode amplifiers, and with particular emphasis on their reliability, life span and efficiency".

Date of search: 22-4-87; **Database:** inzz; **docz=N=** 2855422; **R=** 8

User characteristics: Third year undergraduate student in the Electrical Engineering Department, Imperial College, London. The information from the search was to be used in his final year research project. At the pre-search questionnaire he assessed the nature of his enquiry as general and indicated that a broad search (i.e. all references including peripheral material) was required. The user was familiar with the process of online literature searching and had had a search done for him.

User's comments: The user responses in the post-search questionnaire indicated that the results were good. He felt that the search was fairly close to the original enquiry and that it retrieved more records than expected.

Intermediary's comments The intermediary indicated that the level of difficulty of the search was average (three point scale). The search results were good and the search was finished because it found what was required.

The INZZ database was searched for this request. There were seven terms in the initial query:

term	freq	weight
tetrode$1	389	8.9
uhf	8138	5.9
high adj power	12184	5.5
life	25473	4.7
television$1 or tv	30594	4.5
reliability	46648	4.1
efficienc$3	64072	3.8

The search had three feedback iterations with a total of 21 positive relevance judgements online, 8 of which were made during the initial search. Query expansion was not attempted in any of the search iterations. The search retrieved a total of 32 records. The tree of the initial search was:

Search tree for the initial search of c287

No.	Weight	Term-combination
1	37.3	tetrode$1, uhf, high adj power, life, television$1 or tv, reliability, efficienc$3
1	33.2	tetrode$1, uhf, high adj power, life, television$1 or tv, efficienc$3
1	29.4	tetrode$1, uhf, high adj power, life, television$1 or tv,
1	28.8	tetrode$1, uhf, high adj power, television$1 or tv, reliability
2	27.8	tetrode$1, uhf, life, television$1 or tv, efficienc$3
1	27.1	tetrode$1, uhf, television$1 or tv, reliability, efficienc$3
1	26.9	tetrode$1, high adj power, life, reliability, efficienc$3
2	24.7	tetrode$1, uhf, high adj power, television$1 or tv
4	23.1	tetrode$1, uhf, television$1 or tv, efficienc$3
1	23.0	tetrode$1, high adj power, television$1 or tv, reliability

Query expansion terms drawn from the relevant document set

The CT and UT fields of the eight records of the online relevance judgements during the initial search were ranked using the $w(p-q)$ formula. The list of terms is given in Table 7.3. The terms for query expansion were:

```
(television or tv) adj transmitter$1 (n=953)
television adj broadcasting
high adj power adj tetrodes (n=17)
```

There were many problems of a technical nature with this search. Adding three new terms to the existing seven query terms caused CIRT to crash. After removing the term `television broadcasting` I tried again and the search was successfully completed. Besides the system crashing this search was very time consuming due to the number and the nature of the terms. A summary of the query expansion search and the search tree is given below:

Summary of search r287rel - QE terms from ranked list
(There are 8 known relevant documents)

old wt.	new wt.	rels.	term
11.76	11.76	8	tetrode$1
8.69	8.69	8	uhf
5.45	5.45	4	high adj power
3.75	3.75	2	life
7.36	7.36	8	television$1 or tv
3.14	3.14	2	reliability
3.77	3.77	4	efficienc$3
			.... *Query Expansion Terms* ....
9.20	11.20	2	high adj power adj tetrodes
5.20	9.65	7	(television or tv) adj transmitter$1

Search tree for QE search r287rel
source of terms: relevant documents

No.	seen	weight	terms
1	0	45.42	tetrode$1 high adj power television$1 or tv high adj power adj tetrodes (television or tv) adj transmitter$1
11	0	37.46	tetrode$1 uhf television$1 or tv (television or tv) adj transmitter$1
2	0	36.36	uhf life television$1 or tv reliability efficienc$3 (television or tv) adj transmitter$1
4	0	34.93	uhf high adj power television$1 or tv efficienc$3 (television or tv) adj transmitter$1
4	0	34.30	uhf high adj power television$1 or tv reliability (television or tv) adj transmitter$1
1	0	34.23	tetrode$1 high adj power television$1 or tv (television or tv) adj transmitter$1
1	0	32.61	uhf television$1 or tv reliability efficienc$3 (television or tv) adj transmitter$1
1	0	32.59	uhf life television$1 or tv reliability (television or tv) adj transmitter$1
1	0	32.54	tetrode$1 television$1 or tv efficienc$3 (television or tv) adj transmitter$1
1	0	32.18	tetrode$1 high adj power efficienc$3 high adj power adj tetrodes
1	0	31.91	tetrode$1 television$1 or tv reliability (television or tv) adj transmitter$1
26	0	31.16	uhf high adj power television$1 or tv (television or tv) adj transmitter$1

A total of 62 documents were printed offline, 8 from the initial search and 54 from the query expansion search.

Query expansion terms derived from the INSPEC thesaurus

All the terms in the initial query except for `efficienc$3` have entries in the INSPEC thesaurus. After considering the terms that were suggested by the thesaurus I selected the following three terms:

```
television-transmitters
amplifiers.de.  (n=16399)
life-testing (n=1168)
```

Following another unsuccessful attempt to use all three terms, because of CIRT crashing, I eliminated the term `television transmitters`. Because of technical problems experienced during this search I was able to complete only one search using terms from the INSPEC thesaurus and search them in the descriptor fields. For some reason all attempts to search the same terms free text resulted in system crashes. A summary of the search is given below:

Summary of search r287de
QE terms from INSPEC thesaurus
(There are 8 known relevant documents)

old wt.	new wt.	rels.	terms
11.75	11.75	8	tetrode$1
8.69	8.69	8	uhf
5.45	5.45	4	high adj power
3.75	3.75	2	life
7.36	7.36	8	television$1 or tv
3.14	3.14	2	reliability
3.77	3.77	4	efficienc$3
.... *Query Expansion Terms*			
4.97	4.97	0	life-testing
2.33	3.55	1	amplifiers.de.

Search tree for QE search r287de
Terms from INSPEC thesaurus & searched in DE field

No.	seen	weight	terms
1	0	31.35	tetrode$1 uhf television$1 or tv amplifiers.de.
1	0	30.26	uhf life television$1 or tv reliability efficienc$3 amplifiers.de.
1	0	29.44	tetrode$1 uhf high adj power amplifiers.de.
3	0	28.82	uhf high adj power television$1 or tv efficienc$3 amplifiers.de.
2	0	28.19	uhf high adj power television$1 or tv reliability amplifiers.de.
1	0	28.11	tetrode$1 high adj power television$1 or tv amplifiers.de.
15	0	27.80	tetrode$1 uhf television$1 or tv
1	0	27.75	tetrode$1 uhf life amplifiers.de.
1	0	27.12	uhf life television$1 or tv efficienc$3 amplifiers.de.
1	0	26.71	uhf life television$1 or tv reliability efficienc$3
1	0	26.63	tetrode$1 life television$1 or tv efficienc$3
1	0	26.43	tetrode$1 television$1 or tv efficienc$3 amplifiers.de.
3	0	25.27	uhf high adj power television$1 or tv efficienc$3
1	0	25.22	uhf high adj power life efficienc$3 amplifiers.de.
5	0	25.05	uhf high adj power television$1 or tv amplifiers.de.
2	0	24.64	uhf high adj power television$1 or tv reliability
3	0	24.56	tetrode$1 high adj power television$1 or tv
1	0	23.99	tetrode$1 uhf amplifiers.de.
8	0	23.37	uhf television$1 or tv efficienc$3 amplifiers.de.

There were 60 documents printed offline, 8 from the initial search and 52 from the query expansion search.

7.5.3.1 Analysis

The results of the original search (c287) and the two query expansion searches (r287rel, r287de) are presented in Appendix B.9.

There were 32 records printed for search c287 and the query expansion searches r287rel and r287de had retrieved 62 and 60 records respectively. The offline prints of search c287 had been evaluated by the user and this information was used when judging relevance for the query expansion searches. However, as mentioned earlier, from the study of the relevance judgements that were supplied by the user on the offline prints I noticed that most of the records that were marked as not relevant (NO) were records which were written in a language other than English. By looking at the language field of the remaining records, i.e. the relevant and the partially relevant, it was established that all of them were written in English. It was therefore obvious that the user wanted English language material only, possibly because of the nature of the research project (a third year final project). Since I had established this before I started searching it would have been easier for me to limit each query expansion search to English language only. However, such decision would have been inappropriate and if I had limited the search it would have made it impossible to reconstruct the original search. Therefore, I searched without imposing any limits and I marked all non-English language material as not relevant. The results of the relevance judgements are:

Relevance judgements for search:						
	c287		r287rel		r287de	
Relevance	docs	%	docs	%	docs	%
YES	14	44	17	27	17	28
PAR	4	12	11	18	8	13
NO	14		34		35	
Total	32		62		60	

The precision of the two query expansion searches based on the (YES) judgements is almost identical. When the results are combined with the results from the documents judged as partially relevant (PAR) then precision for r287rel (45%) is slightly higher than that for r287de (41%). The original search (c287) with 56% precision seems to give better results than the query expansion searches.

The overlap in terms of documents between all the searches of pilot case 287 is presented in Appendix B.10. Overlap was measured by examining the retrieved sets in all searches, i.e. c287, r287rel, r287de and is inclusive of the original relevant documents. It was calculated in absolute numbers only, i.e. without considering relevance. In the appendix, documents are represented by their accession number (AN).

The three searches retrieved a total of 154 documents. This total is made up of 87 unique documents. There were 25 core documents appearing in all three searches, which accounts for 29% of the 87 unique documents or for 49% overlap of the total retrieved. In addition, 17 documents appear in any two of the retrieved sets (20% of the unique), and 45 that were found only in one of the sets (52% of the unique).

The overlap between the original search (c287) and each of the query expansion searches by taking into consideration the relevance judgements is:

Document Relevance	Initial search c287	Query expansion searches					
		r287rel			r287de		
		overlap	new	total	overlap	new	total
YES	14	14	3	17	14	3	17
PAR	4	3	8	11	4	4	8
NO	14	8	26	34	14	21	35
Total	32	25		62	32		60

Overall, r287rel and r287de seem to be similar with respect to their overlap with search c287 as well as with the proportion of retrieved records in terms of relevance.

7.6 Concluding remarks

Pilot 1 provided very useful information in shaping up the methodology for conducting the experiment that is described in Part III.

The aim of this pilot study was to look into the process of searching within the context of CIRT and see what could be learnt from it and then apply this knowledge to the methodology in searching an operational situation with real users.

Pilot 1 had its own problems which had to be solved before proceeding any further with the study. Reconstructing old searches was proven to be problematic at first. However, the solutions adopted were satisfying.

The programs that process the downloaded data had to be modified often during the searches because the nuances of text processing are difficult to be anticipated. However, the modifications resulted in a rather smooth processing during the data collection.

The two important decisions that were made in this pilot study are the choice of ranking algorithm and the choice of the source for the query expansion terms.

The adoption of $w(p-q)$ as the ranking formula was based on its theoretical justification and its relatively better performance over most of the other algorithms. However, the question will be opened again when additional data is available to facilitate an evaluation (see chapter 11).

The comparison of the results of the three cases described in this pilot study shows that overall these results look very similar. From these three cases, which is not very much in terms of supporting the conclusions in a quantitative manner, there is nevertheless some evidence that the terms drawn from the relevant set (which was analysed by ZOOM) retrieved some documents which are not retrieved by either of the other methods (i.e., thesaurus terms searched as descriptors or free text). However, it should be pointed out that these are not substantially better than the original results.

The overlap in terms of core documents retrieved by all searches for each case were 20% (r140), 31% (r62) and 49% (r287). This finding comes in accord to that of Katzer *et al* (1982), i.e. that different representations retrieve different subsets of the collection.

Although the number of documents retrieved by using query expansion terms from the relevant document set is somewhat larger, precision figures are on the whole higher than that achieved by the other methods.

The terms from the INSPEC thesaurus did not perform markedly better than the terms taken from the relevant document set. Consequently, I decided to use the latter as the source of the query expansion terms in the study proper.

The fact that the query expansion terms taken from the relevant document set seem to have given better results is a positive sign. Nevertheless, even if the terms from the thesaurus had given better results it does not mean that I had to use the thesaurus as the source of the terms. This is because the aim of the investigation is to look for an automatic procedure or one that requires very simple input and little effort from the user. It is believed that such goal can be achieved by the query expansion process described in this thesis.

Chapter 8

Pilot 2

8.1 Introduction

The aim of Pilot 2 was to investigate the process of adding new terms in the context of a weighted search which uses a front-end. More specifically it was to study the CIRT evaluation searches and establish whether query expansion took place in those searches. CIRT did not offer help on query expansion and the task was left entirely on the user. So, one of the questions is how searchers do query expansion in a weighted search and without any help from the system. In other words, how do searchers go through the search process in a system that does not offer any means of system generated query expansion and whether the searchers are trying to get at terms which query expansion might provide to them?

Having identified the CIRT searches in which query expansion took place, I then looked at the query expansion terms that were used and compared them with the list of terms generated from the relevant document set of the initial search. The next question was to consider whether the query expansion terms used would have been suggested by the ranked list of terms that is derived from the set of relevant documents.

The assumption behind this question can be found in most text books for online searching where it is suggested that the searcher should look into some of the retrieved records and get ideas (extract terms) for query expansion. In a weighted search on CIRT the searcher is required to look at documents so that relevance judgements can be made. Titles are automatically displayed though other fields can also be looked at. Having to look at retrieved documents as part of the search process was thought to offer the searcher a good opportunity for new search terms to be suggested and added in the search.

8.2 Methodology

The logs of the 21 weighted searches that are mentioned in Appendix B.1 were examined to ascertain which of them had more than one search iteration. These searches were then looked at to further identify the ones that had term(s) added to them, i.e., those where there was an attempt for query expansion.

Having selected the searches I noted the terms used for the query expansion and I extracted the accession number of the documents that were judged relevant up to the point where query expansion took place.

The accession numbers were used in order to search and retrieve the documents in ESA. These were subsequently analysed using ZOOM and the terms in the list were ranked using $w(p-q)$.

Finally, the terms used for query expansion were matched against the ranked list in order to establish whether there was any correspondence between them.

In summary the steps involved were:

1. identify CIRT searches where query expansion occurred
2. extract query expansion terms
3. extract AN of online relevance judgements up to the point where query expansion took place
4. locate documents in ESA and analyse with ZOOM
5. rank terms
6. evaluate query expansion terms and terms from ranked list

8.3 Results and Discussion

From the 21 weighted searches in INSPEC there were only five that had extra terms added into the search (c55, c68, c193, c278, c291). In three of these, the terms added into the search were discussed or included at the pre-search stage. That is the terms were entered in CIRT during the `add offline` (`aof`) stage, but were not searched in the first iteration. These were therefore excluded from the investigation because it was obvious that the query expansion terms could not have been suggested by the relevant documents seen online. The remaining two searches that had new terms added to them were c68 and c291 and are described below.

8.3.1 Search c68

Topic: No topic was found in the questionnaires.
However, after considering the query terms and reviewing the offline prints it seems that user was interested in "the ergonomic issues of colour coding in screen displays."

Initial terms: `color$1 or colour$1`, `coding`, `psycholog$4`

Query Expansion terms: `vdu$1, screen$1, crt$1`

There were four relevance judgements made before the query expansion took place. The AN of these documents is given in Appendix C.1. The ZOOM analysis generated a list of 24 terms which were ranked using $w(p-q)$ and are presented in Appendix C.1.

The comparison of the terms used for query expansion and the terms in the ranked list shows that the new terms do not appear in the list. A closer look at the terms of the ranked list reveals that there is one term, `visual displays`, that is related to the query expansion terms. In addition, the questionnaire filled by the intermediary shows that 23 terms were discussed during the pre-search interview and seven terms were used in the search. Because a record of the terms was not kept I can only speculate that the query expansion terms were known in advance. So, the search was initially approached at a general level, i.e., colour coding and psychology, and then it was directed to a more specific level.

8.3.2 Search c291

Topic: No topic was given in the questionnaires.
From the query terms used and after reviewing the offline prints it seems that the user was interested in "functional programming languages" more specifically, in the language "hope".

Initial query terms: `functional adj database$1, functional adj data adj model$1, polymorphism, abstract adj data adj type$1, hope, adaplex, efdm`

Query expansion terms: `functional with language$1`

There were 22 documents judged relevant online before the query expansion search. The AN of these documents is given in Appendix C.2. The ZOOM analysis generated a list of 171 terms which were ranked using $w(p-q)$ and presented in Appendix C.2.

The term used for query expansion is found in the ranked list. However, the information obtained from the questionnaires reveals that the query expansion term was among the 12 terms discussed at the pre-search interview. Furthermore, from the search logs it can be deduced that the initial search was that of a general approach to the subject. This is because `hope`, `adaplex`, and `efdm` are programming languages that support complex data typing, polymorphism, inheritance and are used in object-oriented systems.

The initial search was followed by two feedback iterations and then query expansion took place. However, in the query expansion search a complete restructuring of the query was performed. Six terms from the initial query were deleted, leaving `hope` as the only term to be combined to `functional adj language$1`.

8.4 Concluding remarks

This pilot study aimed at looking at the process that users of weighted searches using CIRT went through in conducting query expansion. From the two cases involved in this study there is no evidence available that the searchers tried to get at terms which query expansion based on the relevant documents might provide for them. In part this means that for whatever reasons, which have to deal partly with the constraints of the CIRT experiment, searchers did not try to expand queries very much.

On the whole, from the data collected from the CIRT evaluation project it is very difficult to arrive at some firm conclusion of how searchers do query expansion in a weighted environment like that of CIRT. There are many reasons that account for this.

Some are obvious technical problems that relate directly to CIRT as already mentioned in section 7.3.4. The slow response time, the 8 term maximum query size and the system crashes imposed a burden on the intermediaries that seems to have been reflected in their searching. The length of time it takes CIRT to search with a lot of terms was a disadvantage which the searchers soon discovered for themselves.

For the CIRT evaluation project intermediaries were given some initial training and instructions of how to perform a weighted search. The project investigators did not have any specific requirements or guidelines for query expansion. Searching was left entirely to the discretion of the intermediaries who unfortunately did not do very much.

This is surprising in that one expects professional intermediaries involved in searching would use query expansion routinely. Most text books in online (Boolean) searching discuss the benefits associated with the method of query modification that looks at the retrieved documents in order to get new ideas and provide terms for query expansion. *Citation pearl growing* is the search strategy named after this technique (Meadow & Cochrane, 1981).

Furthermore, the literature has numerous accounts of this method and the work by Bates, Fidel, Harter and others on search techniques and heuristics is an attempt to formalise and provide a theoretical framework to the empirical knowledge. In that sense and because the CIRT searches were done by professional intermediaries it is even more surprising that they did not do any query expansion.

However, one could only speculate for the reasons.

For example, the searcher's familiarity (or lack of it) and comfort with weighted searching might have also contributed in choosing not to do query expansion. Another reason might be that the searchers, despite their initial training, did not have a good grasp of the underlying theory of CIRT's weighted searching. So, they did not have an adequate understanding of what the CIRT does, how it works and how to take advantage of such system to its fullest. This assumption, if correct, has apparently many implications for research and pin-points the need for user studies in searching weighted systems.

The fact that searchers did not make much use of query expansion in the CIRT experiment may be taken as evidence of the necessity to provide searchers with help for query expansion.

Chapter 9

Pilot 3

9.1 Introduction

In Pilot 3, as in Pilot 2, I continued to look into the CIRT searches that involved some form of query expansion. During this pilot case study I focused on the relevant documents that were identified from the online as well as offline relevance judgements. The question addressed was "what evidence can be found that terms taken from the relevance judgements of the initial search might subsequently be useful?"

In a weighted search one can identify two stages of relevance judgements. A series of relevance judgements that take place online, during the search iterations, and the final set of relevance judgements that takes place during the evaluation of the offline prints. Successive relevance judgements imply some continuity in the criteria of relevance. Thus, some terms from the relevance judgements of the first iteration, other than the query terms, are good for retrieving the documents of subsequent iterations which will eventually form the final set of retrieved documents.

I will refer to the set of relevant documents that are identified during the initial search as the *initial set.* Similarly, I will call the relevant documents on the offline evaluation as the *final set.* The final set usually, but not necessarily, contains all the documents of the initial set plus some additional ones. It is these latter documents that are of interest to this study. More specifically the interest is directed to the relationship between the relevant documents in the initial set and the additional relevant documents of the final set. Because these two sets of documents are different the interest is on the relationship, if any, that may exist among them. For example, are there any similarities between the terms that are found in these sets?

How did the initial set of relevant documents retrieve the final set? One way to look at this is by examining the terms in the documents of each set. Obviously, retrieval is determined by the query terms. Therefore, if the query terms are excluded from the initial list are any of the remaining terms useful? Furthermore, if the terms in the initial and the final set are ranked how predictive is the ranking of the initial set in retrieving the documents of the final set? Are there any other terms that could be extracted from the initial set that could help in retrieving the final set? Finally, are the same terms high up on both lists?

Answers to these questions were sought by looking at the weighted searches that included more than one search iteration. A simple way of comparing the two lists of terms is by measuring their overlap.

9.2 Methodology

The logs of the 21 weighted searches (mentioned in Appendix B.1) were examined in order to establish which of them had more than one iteration. In the examination of the searches I excluded the five used in Pilot 1 and Pilot 2. I looked at the remaining searches noting how many iterations were performed during each search, how many documents were judged relevant online for each of these iterations, how many documents were retrieved, and finally how many documents were printed offline. The detailed search history was necessary in order to be able to identify those searches that could have in their logs all the information needed for this case study.

For these searches however, I excluded all those that had:
(a) printed offline only as many documents as were seen online,
(b) retrieved more than 50 documents.

The former were excluded on the basis that these documents were the same in both sets (initial and final) and therefore did not add any new information that could be considered under the objectives of this pilot. The latter had to be excluded because of lack of information, due to the rule to print a maximum of 50 documents during the CIRT evaluation project. In other words, because of the print limitation documents that were judged relevant online were not always printed. Consequently, it was not possible to create the initial set.

Having selected the searches I then extracted the accession numbers of the documents that were judged relevant (a) during the initial search (initial set) and (b) on the offline prints (final set). I then excluded all the documents that were common with the initial set, thus creating two distinct sets.

The accession numbers of each set were used to search and retrieve the records in ESA. The fields with the controlled and uncontrolled terms (CT, UT) were subsequently analysed using ZOOM and the terms were ranked using $w(p-q)$. The two ranked lists of terms, one for the initial set and the other for the final set, were then evaluated.

The steps involved in the methodology were:

1. identify CIRT searches with more than one iteration
2. relevant documents online
   (a) get AN of the online relevance judgements of the initial search iteration.
   (b) locate records in ESA and analyse the CT and UT fields with ZOOM
   (c) rank terms
3. relevant documents offline

   (a) get AN of the offline relevance judgements
   (b) exclude common records with online relevance judgements
   (c) locate records in ESA and analyse CT and UT fields with ZOOM
   (d) rank terms

4. compare the two ranked lists
   (a) measure absolute overlap of the lists
   (b) compare 10 top ranked terms from each list

9.3 Results

From the 21 weighted searches in INSPEC only three were selected (c60, c69, c70). The rejected searches were excluded for the following reasons. Three had no additional search iterations to the initial search, five had retrieved more than 50 documents, five had printed offline only the documents judged relevant online, and five were excluded because they were used in the earlier case studies, The results of the three searches are presented below.

9.3.1 Search c60

Topic: No topic was given in the questionnaires

Initial query terms: `(metal or copper) adj colloid$1, glass or silicon, electroless adj plating, reduction or reductant$1`

Initial set: 11 documents were judged relevant during the initial search

Final set: 5 documents were identified in addition to those in the initial set.

The 11 documents of the initial set generated a list of 128 terms. The 5 documents of the final set generated a list of 68 terms. This brings a total of 196 terms in both lists. A comparison of the two lists shows 189 unique terms and only 7 terms common to both, i.e. an overlap of 4%. The common terms are presented below.

initial set (total 128 terms)					final set (total 68 terms)			
rank				term	rank			
posit	wt	r	term	freq.	posit	wt	r	term
35	0.8	3	copper	56315	33	1.5	2	copper
117	0.3	1	electroless deposition	265	31	1.7	1	electroless deposition
1	3.3	4	electroless plating	261	1	5.9	3	electroless plating
3	1.8	3	electroplating	2100	3	2.8	2	electroplating
119	0.2	1	etching	16592	55	0.8	1	etching
28	0.9	1	nickel	22128	61	0.8	1	nickel
104	0.3	1	printed circuits	3382	42	1.2	1	printed circuits

The terms are presented in alphabetical order. Each term is given with the rank position in its respective list. The terms appear as found in the CT and UT fields, i.e. single terms or phrases. Normalisation, i.e. stemming, of the terms was not considered. There is very little to study from such low overlap however there are two query terms (copper, electroless plating) among the seven common terms. The ten top ranked terms for each list are given below.

initial set					final set				
rank	wt	r	n	term	rank	wt	r	n	term
1	3.3	4	261	electroless plating	1	5.9	3	261	electroless plating
2	1.9	2	30	reductants	2	2.9	2	1732	abrasion
3	1.8	3	2100	electroplating	3	2.8	2	2100	electroplating
4	1.2	1	2	pyridine	4	2.7	1	2	semiconductor
5	1.2	1	2	he ion implantation	5	2.7	1	2	photoresists
6	1.2	1	2	cu sup 2 solutions	6	2.6	1	3	wear resistant coatings
7	1.2	1	2	competitive exchange equilibria	7	2.6	1	3	electroless ni deposition
8	1.1	2	2641	ion exchange	8	2.6	1	3	ceraform
9	1.1	2	2621	colloids	9	2.5	1	4	flexi rigid boards
10	1.1	1	3	hypophosphite	10	2.5	1	4	cu inks

In addition to the rank and the weight, the term frequency and r frequency is also given for each term. Two terms are common to both lists (electroless plating, electroplating) and one of them (the first) is a query term. The majority of the terms in both lists, but especially in the final, are of very low frequency.

9.3.2 Search c69

Topic: No topic as given in the questionnaires

Initial query terms: bubble$1, drop$4, coalescence, electrophore$3, mobility, zeta adj potential$1, thin adj liquid adj film$1, laser adj doppler, two adj phase adj flow$1

Initial set: 12 documents were judged relevant during the initial search.

Final set: 4 documents were identified in addition to those in the initial set.

The 12 documents of the initial set generated a list of 115 terms. The 4 documents of the final set generated a list of 37 terms. This brings a total of 152 terms in both lists. A comparison of the two lists shows 139 unique terms and only 13 terms common to both, i.e. an overlap of 9%. The common terms are presented below.

initial set (115 terms)			final set (37 terms)		
rank posit	wt	term	rank posit	wt	term
11	1.7	bubble	34	1.2	bubble
1	7.3	bubbles	1	5.1	bubbles
31	0.9	coalescence	28	1.6	coalescence
58	0.7	draining film	12	2.7	draining film
18	1.1	drop	35	1.1	drop
2	4.3	drops	32	1.2	drops
16	1.3	electrostatic forces	22	2.1	electrostatic forces
8	2.3	liquid films	3	3.7	liquid films
115	0.1	liquid phase	17	2.4	liquid phase
3	3.8	london van der waals forces	13	2.6	london van der waals forces
26	1.0	rupture	36	1.1	rupture
44	0.8	stability	9	2.9	stability
64	0.6	van der waals forces	2	3.8	van der waals forces

The terms are presented in alphabetical order. Each term is given with the rank position in its respective list. The terms appear as found in the CT and UT fields, i.e. single terms or phrases. Normalisation, i.e. stemming, of the terms was not considered. The overlap has slightly been improved in comparison to the previous search (c60). However, there is very little to study from such low overlap. There are three query terms (`bubble`, `coalescence`, `drop`) among the 13 common terms. Two terms are the plural form of two of the query terms which were retrieved because of the truncation. Finally, one term is part of a query terms (`liquid films`). This brings to six the terms that are associated in some way to the query terms.

The ten top ranked terms for each list are given below.

initial set (115 terms)			final set (37 terms)		
rank	wt	term	rank	wt	term
1	7.3	bubbles	1	5.1	bubbles
2	4.3	drops	2	3.8	van der waals forces
3	3.8	london van der waals forces	3	3.7	liquid films
4	3.1	two phase flow	4	3.4	navier stokes equations
5	2.8	hydrodynamic theory	5	3.1	horizontal solid surface
6	2.5	fluid fluid interface	6	3.1	electric double layer forces
7	2.4	dimpled liquid film	7	2.9	condensation
8	2.3	liquid films	8	2.9	london van der waals
9	2.3	thin liquid film	9	2.9	stability
10	1.8	coalescence time	10	2.8	thin liquid films

The rank and the weight for each term is given in the table. Three terms of the initial list are query terms and another three are associated with a query term (e.g., `liquid film`). The final list contains two query terms. Finally, there are three terms that are common to both lists, `bubbles`, `van der waals forces` and `thin liquid film$1`.

The majority of the terms in both lists, but especially in the final, are of very low frequency.

9.3.3 Search c70

Topic: No topic was given in the questionnaires

Initial query terms: `picture adj processing, encoding, compress$4, codec, videotelephony, videoconferenc$3, teleconferenc$3`

Initial set: 14 documents were judged relevant during the initial search.

Final set: 13 documents were identified in addition to those in the initial set.

The 14 documents of the initial set generated a list of 130 terms. The 13 documents of the final set generated a list of 104 terms. This brings a total of 234 terms in both lists. A comparison of the two lists shows 214 unique terms and only 20 terms common to both, i.e. an overlap of 9%. The common terms are presented below.

initial set (130 terms)			final set (104 terms)		
rank	wt	term	rank	wt	term
116	0.3	ccitt	87	0.4	ccitt
6	2.1	codec	70	0.5	codec
46	0.7	codec system	29	0.8	codec system
3	5.6	codecs	4	3.0	codecs
56	0.7	cost 211	16	1.0	cost 211
117	0.3	data communication equipment	34	0.7	data communication equipment
14	0.9	digital communication systems	46	0.6	digital communication systems
24	0.8	digital transmission	51	0.6	digital transmission
4	5.3	encoding	6	2.0	encoding
37	0.7	error correction	72	0.4	error correction
55	0.7	graphics transmission	7	1.6	graphics transmission
88	0.5	motion compensation	26	0.8	motion compensation
12	1.1	picture processing	5	2.7	picture processing
103	0.4	speech transmission	103	0.1	speech transmission
1	11.0	teleconferencing	1	10.9	teleconferencing
87	0.5	transmission errors	60	0.5	transmission errors
7	2.0	video codec	8	1.4	video codec
13	1.1	video signals	2	5.0	video signals
5	2.4	videoconferencing	3	4.2	videoconferencing
25	0.8	voice communication	80	0.4	voice communication

The terms are presented in alphabetical order. Each term is given with the rank position in its respective list. The terms appear as found in the CT and UT fields, i.e. single terms or phrases. Normalisation, i.e. stemming, of the terms was not considered. There is very little to study from such a low overlap however there are five query terms (`codec`, `encoding`, `picture processing`, `teleconferencing`, `videoconferencing`) among the twenty common terms. Many of the remaining terms are associated with the query terms.

The ten top ranked terms for each list are given below.

initial set (130 terms)			final set (104 terms)		
rank	wt	term	rank	wt	term
1	11.0	teleconferencing	1	10.9	teleconferencing
2	8.0	videotelephony	2	5.0	video signals
3	5.6	codecs	3	4.2	videoconferencing
4	5.3	encoding	4	3.0	codecs
5	2.4	videoconferencing	5	2.7	picture processing
6	2.1	codec	6	2.0	encoding
7	2.0	video codec	7	1.6	graphics transmission
8	2.0	european videoconferencing experiment	8	1.4	video codec
9	1.8	data compression	9	1.0	work program
10	1.2	video teleconferencing	10	1.0	video tape

The rank and the weight for each term is given in the table. There are five out of the seven query terms at the top of the initial list. In the final list there are four query terms. Four terms are common to both lists.

9.4 Analysis and discussion

When considering which methodology to follow for comparing the two lists I investigated the possibility of using a correlation measure. The reason was that I had two ranked lists and consequently a statistic that measures correlation could help identify the strength of the association between them.

There are two questions associated with the kind of data taken from the lists that make it very difficult to choose a measure as well as to make sense of the results. The main question is how and whether two ranked lists of unequal size (length) can be compared. If they can be correlated, how much use correlation is? Finally, do the correlation measures consider the importance of ranking in these lists. In other words, the terms in the two rank lists are ranked in decreasing order of probability of relevance of the query as estimated by the system. This order is important and the information carried with it should be maintained and taken into consideration when comparing the two lists.

Ranking of terms implies that the terms at the top of the list are more important than those at the bottom of it. A measure that takes equal account of the rank order at either end of the lists (i.e., top and bottom) is not very useful. It would be better to find a measure of rank correlation which at least takes more account of the top of the list rather than at the bottom.

I considered a number of possible measures that I could use. However, the two requirements needed, i.e., to be able to handle unequal lists and take into consideration the importance of ranking, made the search rather difficult.

Many information retrieval researchers had suggested the need for using a rank correlation measure for the testing of systems with ranked output. Measures which have been proposed include Kendall's τ correlation coefficient and Spearman's ϱ. The major disadvantage of Kendall's τ as well as Spearman's ϱ is however that the entire list is used for comparison. An interchange of two terms with rank 2 and 6 will be taken as equally important by the statistic as the interchange of terms with ranks 92 and 96.

A possible solution to this problem is a measure that assigns weights reflecting the relative importance of the parts of the list. Two such weighted rank-correlation measures that have been proposed are Pollock's (1968) and Kaye's weighted rank correlation coefficient (1973). Pollock discussed the M-V statistic which is based on Kendall's τ while Kaye's is based on Spearman's ϱ. Kaye [p.381] criticised Pollock's statistic because

> "...only the first M items of the list are ranked according to the original ranking, whilst the remaining items are regarded as being tied. This has the effect of regarding misplaced rankings at the lower end of the list with indifference, but it does not weight the M items that remain ranked."

Kaye's weighted rank correlation coefficient does not suffer from this problem by taking care of ties and misplaced ranks in the list. However, the two statistics assume that one of the two lists, that are to be compared, is the reference list. This is inappropriate for the purposes of this case study because the terms in the lists are derived from different sets of documents and we are interested to see whether there is any correlation between them without assigning superiority to one of them.

Furthermore, by assuming that one list is a reference list either statistic shows a bias towards positive correlation which is always expected given that the "system" is trying to achieve it (Kaye, 1973, p.386). Therefore, these weighted coefficients are only empirical statistics and cannot be used like Spearman or Kendall for the testing of statistical significance. In addition, both statistics assume that the two lists are equal which is the main reason for not using either statistic in this study.

Leaving the issues of the applicability of the correlation coefficient aside the question becomes not whether the lists are correlated at all but of how much use the correlation is? In other words, whether the initial list can be used in a useful and valuable way to predict something about the final list, i.e., to predict value in retrieval. One form of analysis is to take the top 10 terms of the initial list and then see if there are many terms among them in the final list. If the terms that have been ranked high up in the final set are bad then that could be an argument against automatic query expansion and not semi-automatic query expansion.

However, the simplest way to start the comparison of the two lists is by doing a simple overlap between them. The reason being that if the two lists have a high percentage of common terms, even if they are not of equal length, then there is a strong possibility in retrieving additional documents. The analysis in this chapter has been mainly concentrated in the overlap between the the two lists of terms.

The analysis of the three pairs of ranked lists shows that for each pair (initial-final) a large number of infrequent terms has been retrieved that is found only either in one list or the other. There are also a number of single frequency terms (n = 1). These are terms which necessarily belong only in one list. The number of single frequency terms in the lists is given in the table below.

Single posted terms in the ranked lists						
	initial list			final list		
Search	# of terms (a)	n=1 (b)	terms in list (a-b)	# of terms (c)	n=1 (d)	terms in list (c-d)
c60	135	7	128	75	7	68
c69	124	9	115	38	1	37
c70	146	16	130	119	15	104

For each search the table shows how many terms were found in the initial list and the final list, how many of these were single posted terms and how many terms were eventually included in each list after eliminating the single frequency terms.

The overlap of terms between the initial list and the final list for each search in this case study is given in the table below. Single frequency terms were excluded from the lists. The overlap which was measured as the number of terms common to both lists (initial-final) is very low and ranges from 4% - 9%.

Overlap excluding single posted terms from the lists					
search	initial	final	total	overlap	%
c60	128	68	196	7	4
c69	115	37	152	13	9
c70	130	104	234	20	9

I would like to consider now the single frequency terms. In fact, I want to discuss any term in the lists, whose collection frequency (n) is the same as r, i.e., $n = r$. What kind of terms are these and what is their role in retrieval?

At this point I would like to re-state the difference between r and the ZOOM term frequency count. In the various formulae used, like $w(p - q)$, F4, EMIM, Porter, r is the number of relevant documents (from the sample R) assigned to term t.

The ZOOM term frequency count that is given for a term t is not the same as r. ZOOM counts every occurrence of the term in the document set and reports a cumulative count, for example in a set of 8 documents a term might get a count of 11. This can interfere with the calculation of the weights. In order to prevent such problem I included, in the programs that calculate the term weights, a conditional statement so that

$$\texttt{if } r > R \texttt{ then } r = R$$

Now, returning to the discussion of $r = n$ and the single posted terms we are faced with the fact that any term whose collection frequency (n) is the same as r is expendable because it cannot retrieve any additional documents.

There are some interesting questions that emerge from this in relation to both the initial list and the final list.

The initial list is drawn from terms from the online relevance judgements and the final list is drawn from terms from the offline relevance judgements. A term in the initial list with $r = n$ would, by a decent algorithm, be excluded from query expansion, because it is not going to retrieve any additional documents. This means that I should be excluding from the ranked lists all the terms with frequency $r = n$. However, ZOOM, which is used for the analysis of the relevant document set, gives cumulated frequencies of a term over all documents. For this reason I decided to exclude only terms where $n = 1$. It should be mentioned at this point that the case of $r = n$ is one instance of a more general case that could occur in the initial set. The general case refers to terms that have all been seen in the initial set (whether or not judged relevant). In such case, these terms cannot be of any use in the final set.

A term in the final (offline) list with $r = n$, what kind of term is that? It is a term which is actually terribly good in terms of the final list, but one cannot possibly have found it in the initial list. This is because in either case a term with $r = n$ cannot occur in the other list given that the initial set and the final set are exclusive sets of documents. So, a term with $r = n$ from the final list would be an ideal term, for query expansion except that it cannot be found with this kind of query expansion, i.e., the one that is based on the relevant document set.

Semi-automatic query expansion that is based on the relevant documents involves two stages:

- extract the terms that are given a good ranking from the relevant documents. The system here is "saying" that these are good candidate terms.
- the user should be able to recognize and choose the so good terms.

The question discussed in this pilot case study relates only to the first of the above. In other words, are there good terms in the list irrespective of whether the user would recognise them?

In automatic query expansion the question becomes how useful are the top ranked terms. In interactive query expansion the question focuses on whether the list contains some good terms that are high up in the ranked list. If we assume that the users are able to recognise which are the good terms then it does not matter if the list would contain some lousy terms as well. Most importantly, the absolute positioning of the terms at the top of the list becomes less of an issue in interactive query expansion than it is with automatic query expansion. In conclusion, the purpose of Pilot 2 and Pilot 3 were to look at CIRT searches, analyse them and see what can be learnt from them that could be used when conducting the searches and studying interactive query expansion. These issues were considered in the methodology of the study proper and are discussed in more detail in Part III.

9.5 Concluding remarks

Pilot 3 looked into the questions of 'what evidence can be found that terms taken from relevance judgements of the initial search might subsequently be useful?' of 'how predictive is the ranking of the terms in the first set in retrieving the documents of the final set?' and of 'are the terms high-up on both lists?'

The overall conclusion that can be drawn from the three cases presented here is that the overlap between the terms taken from the online and offline relevance judgements is very low and ranged from 4% to 9% (average 7%). Query terms accounted for an average of 26% of the overlap. The similarity of these documents is predominantly determined by the query terms. The low overlap therefore means that the documents in the final set were retrieved because they contained the query terms but they treat a different aspect of the subject which is not addressed by the documents of the initial set. Whether these documents are peripheral to the subject of the query, and if yes, how peripheral, is an issue that the user can only answer.

Finally, it is established that terms from the initial set that have frequency of $r = n$ are expendable. On the other hand, terms in the final set with frequency of $r = n$ are very good but unfortunately cannot be found with this type of query expansion and are not accessible during the search.

Part III

Interactive Query Expansion: the experiment

Chapter 10

The experiment

10.1 Introduction

Part III of this study describes interactive query expansion using a real system (CIRT) and real users. The emphasis of this investigation, which looks at query expansion in a real environment, concentrates on the characteristics of the interaction. All the searches were conducted in the INSPEC database on Data-Star using the CIRT front-end and on ESA.

The three pilot case studies described in Part II answered research questions on query expansion in terms of the usefulness of the query expansion terms, the choice of the ranking algorithm, the selection of the source for the query expansion terms, etc. The pilot cases should be taken as part of the overall methodology because they were very useful in shaping up the methodology used in the Experiment.

The main objective of the Experiment was to look at the process of query expansion as a whole in very much the same way as discussed in the Pilots. By capitalising on what was learnt from the pilot studies and applying this knowledge in a real environment situation I was able to concentrate on the investigation of interactive query expansion and on the characteristics of its interaction without having to be concerned with methodological issues. However, as is discussed in more detail later, there is no such thing as a fool proof and rigid methodology when real users are involved and when one looks at the interaction process.

Real user-system interaction is unpredictable, it cannot be anticipated and therefore it cannot be modelled in detail and placed in some firm way in a methodology. Furthermore, it requires that whenever a new situation arises the searcher should resolve it immediately by following some rule. If such rule, i.e. methodological procedure, does not exist, then a solution is given based on the information at hand. Thus, this would make the solution the rule to be applied in subsequent cases.

10.2 Methodology

10.2.1 Experimental design

The experiment has been designed with a view to the inclusion of the semi-automatic query expansion module in an operational environment which uses relevance feedback. More specifically the design is intended as a module that can be incorporated in CIRT. The major decisions taken were:

1. The initial set of documents displayed to the user should be obtained using CIRT and searching the INSPEC database in Data-Star.
2. The user should only be asked to judge a small number of documents for relevance during the relevance feedback process.
3. At least one relevant document must be identified by the user during the online relevance assessment, although I would aim for a minimum of five relevant documents.
4. The ZOOM facility on ESA/IRS would be used to analyse the relevant documents identified in the online assessment.
5. The $w(p-q)$ ranking algorithm would be used for the ranking of terms for query expansion.

Each of these decisions will be discussed in detail in section 10.2.5. The above decisions serve also as an outline of the search process and they have been presented at this point in order to give an overview of the process and to facilitate the discussion in the sections that follow.

10.2.2 Sample and participants

Having in mind the difficulties encountered during the data collection phase of the CIRT evaluation project I aimed for a number of searches that would be proportionally comparable. To illustrate the difficulties involved in the CIRT project I will mention that over a period of two years and with four sites collecting data it reached only a total of 190 searches. Therefore, a total of 25 searches for my data collection was felt to be a reasonable target to achieve and fell well within the imposed limits. In fact, I collected 30 searches in total. However, 5 were not used because the searches were interrupted due to technical difficulties and were not completed because the users did not return to continue the search. The time required to complete a search from the pre-search interview to the final evaluation of the offline prints was at least four hours. Thus, time as well as the technical difficulties were the main deterrents during the data collection.

In order to attract users I offered completely free searches. The searches were all confined to the INSPEC database for the reasons mentioned in Part II. The free searches would require the user to come to the Information Science Department at The City University for an appointment; for the search and for the evaluation of the offline prints. Throughout the data collection I aimed at completing the entire process (search and evaluation) in one session. However, in many occasions, because of technical problems, the process was completed over two sessions.

The searches were all performed by myself and the users were present throughout the search process. Advertising was distributed in the City University Library and within the various Departments, such as Computer Science, Engineering, etc. The response to the advertising was not high so I made personal calls in the City University Departments to recruit users.

A complete record of the search includes four questionnaires, all the search logs and the evaluated offline prints. The offline prints were evaluated on site immediately after the search thereby reducing the possibility of users not returning them if the prints were taken away for evaluation. At end of the session users were given a copy of the offline prints that they had evaluated.

10.2.3 Variables

The variables examined are divided into seven categories and are evaluated by the questionnaires, logs and relevance evaluations. The categories include:

1. Retrieval effectiveness.
2. User effort.
3. Subjective user reactions.
4. User characteristics.
5. Request characteristics.
6. Search process characteristics.
7. Term selection characteristics

Some of these variables relate directly to query expansion and the results obtained are discussed in section 10.3.1.1. The results from the variables that did not directly relate to query expansion are discussed in section 10.3.2. However, all variables are listed below.

10.2.3.1 Retrieval effectiveness (V1)

This is based on relevance judgements for all documents retrieved and which were printed in full format. Relevance judgements are made on a three-point scale, i.e. 'Relevant', 'Partially relevant', 'Not relevant'. *Relevance1* indicates that the middle group, i.e. the partially relevant, has been classified as not relevant. *Relevance2* indicates that the partially relevant documents have been included with the relevant group. The parameters to be evaluated were:

(a) total number of items retrieved;

(b) number of relevant items retrieved; and

(c) precision (b/a);

the latter two being calculated both for Relevancel and for Relevance2.

Because the experiment was conducted in an operational environment, and because each search was conducted only once, it was not possible to establish a recall base. Therefore number of relevant retrieved was used instead of recall (Robertson & Thompson, 1987).

Additional relevance data was gathered during the evaluation of offline prints (see section 10.2.4.2). This information was concerned with whether the user had seen the documents prior to the evaluation and could be used for determining novelty ratios. The questions covered both the issue of subject relevance as well as that of the usefulness of each document to the user. In addition to precision the following relationships were also examined:

(a) relationship (proportion) of online relevance judgements to total number of documents seen online.

(b) relationship (proportion) of total number of documents seen online to total retrieved.

(c) relationship of online relevance judgements to offline relevance judgements. How online relevance judgements correlate to overall results? Is there a consistency between online-offline relevance judgements?

10.2.3.2 User effort (V2)

This is based on the interaction between the intermediary, i.e. myself, and the user at the pre-search interview stage and between the front-end, the intermediary and the user during the search. It includes: time to prepare the search, pre-search terms, query terms, query expansion terms, online time, online citations and number of relevance judgements made.

User effort could also be measured on how many documents one has to see in order to arrive at 5 relevant documents online. This aspect of the variable is closely related to retrieval effectiveness and therefore the two are discussed in the results in parallel rather than separately. There are two stages in looking at documents, one is online and the other is offline. Traditionally we calculate precision based on the results of the offline evaluation. However, the relevance judgements online provide us with information that may relate to precision. For example, during the searches I aimed to base the analysis on a sample of at least 5 relevant documents. Therefore, one could relate the user's effort to precision. In other words, this variable may be linked to retrieval effectiveness.

The hypothesis is that the longer it takes (in terms of documents examined online) to arrive at the 5 relevant documents the lower the precision of the final retrieved set will be. To put it in another way, the question could be reformulated to how the proportion of the relevant documents seen online to the total number of documents seen online affects the final result, i.e. could the precision that is achieved online be an indication for the expected precision to be achieved from the final results. Could such a relationship be established, and if yes, what does it mean?

10.2.3.3 Subjective user reactions (V3)

This category is mostly concerned the user's overall reactions to the search, impressions of effort involved, and reaction to search results - not offline prints. This also included variables from other categories such as how close was the search to the original/intended enquiry (question 17); and whether the expected number of references were retrieved (question 18).

10.2.3.4 User characteristics (V4)

This category covers personal data about the individual which would relate to the search process, such issues as areas of subject expertise, the level of their work and number of previous online searches either with or without an intermediary.

Categorical questions on education/academic status, occupation/subject studied, etc. give data on individual differences that have been shown to be related to information retrieval system performance (Borgman, 1986).

10.2.3.5 Request characteristics (V5)

I elicited scaled data on context and problem characteristics: e.g., subject area of the request, nature of enquiry e.g. accurate or vague, type of search required i.e. broad or narrow. These data allowed me to relate such issues as specifiability of problem, clarity of problem, work done on problem, etc. to type of search required and success of search.

10.2.3.6 Search process characteristics (V6)

The questionnaires on the outcome of the search would elicit scaled data on the effectiveness and efficiency of the search. These would allow me to make estimates of the degree of success or failure of the interaction, from the point of view of the user and thus to relate searching behaviour, problem context and types of uses to various measures of success.

10.2.3.7 Term selection characteristics (V7)

This questionnaire elicited information about the query expansion terms that were selected by the user. The questions concentrated on how the user perceived the relationship between the query terms and the term that s/he selected for query expansion. They were asked to identify whether the query expansion terms were chosen because they were thought of as being synonyms of query terms, related terms, the best alternative to express the subject that they could find in the list, or representing new ideas.

We have a list of terms taken from the relevant documents in which terms are ranked. There are two ways to look at the ranked lists:

1. the ranking algorithm is not in question, and
2. the ranking algorithm is in question.

In chapter 10 I discuss the results assuming that the ranking algorithm is not in question. In chapter 11 the results are discussed by assuming that the ranking algorithm is in question.

Assuming that the ranking algorithm is not in question then we accept that the terms are ranked by the system according to their usefulness for their particular query, with the best terms at the top of the list and the bad terms at the bottom. We also have the user who looks at the list and identifies all those terms that s/he thinks are good terms for potential inclusion in the query. The question then becomes whether there is any correlation between what the system suggests (measured through its ranking of the terms) and what the user chooses as potentially good term.

10.2.4 Data collection instruments

The data collection instruments used in the study can be divided into three categories which include questionnaires, transaction logs and offline prints. A summary of the instruments is given below and the discussion that follows is presented by category.

1. Pre-search questionnaire
   Background information and context (purple - 11 questions). Personal data, assessment of subject enquiry, online experience, etc.
2. Search process log
   Search summary and notes, terms, online time, etc.
3. Query expansion questionnaire
   (a) Term selection from ranked list, and (b) Term relationships
   (blue - 2 questions)
4. End of search questionnaire
   User's satisfaction, impression of search, assessment of query and results, and of the number of references expected (green - 5 questions)
5. Evaluation of offline prints
   Relevance judgements on a 3 point scale and subdivisions (yellow - instructions)
6. Final questionnaire
   Final assessment of search as a whole and user remarks (pink - 2 questions)
7. Other logs
   Logs of the interaction between intermediary-CIRT & CIRT-host, intermediary-ESA, etc.

10.2.4.1 Questionnaires

The questionnaires provided a qualitative assessment by the user and the intermediary (myself) of a range of variables. The questionnaire was divided into four parts, and each was colour coded for easier identification and processing. All parts were completed by the enquirer in my presence. I provided additional information when necessary and following the completion of the questionnaires I immediately checked the answers in order to clarify any points and eliminate possible misunderstandings.

In addition there was an introductory form which briefly explained the project, what was required of the user, and stressing that all the information given was strictly confidential and data protected (see Appendix D).

The first questionnaire (colour: purple) given at the pre-search interview dealt mostly with user and request characteristics. It enquired about the users' background and the context of the information request. It asked for combined user and search information such as status (student, faculty, researcher, etc.); what they hoped to use the search results for (coursework, research project, teaching); whether they had done any online searches before, either on their own or with an intermediary. There were also questions regarding request characteristics. Some questions elicited from the users information on how much they thought they knew about the subject (1=nothing – 5=a lot) as well as at which stage of the project they were at (1=beginning to think – 5=end of project). Other questions asked whether the user wanted a broad or narrow search, and whether they viewed the nature of their search request as precise, general or vague. This last question was trying to ascertain how the request and the terms used in the search related to the subject domain of the query.

The next questionnaire (blue) asked questions relating to the choice of terms for query expansion. For example, if the user had selected some terms from the ranked list s/he was asked whether these terms were thought of as: variant expressions or synonyms, or alternative (related) terms to the original query terms; or whether these were chosen because the user could not find a better term(s) to express the subject of the enquiry; or whether the chosen terms represented new ideas that were not part of the original request. For each query expansion term chosen the user had to identify if it corresponds to an existing query term and whether it was a broader, narrower or related term.

The remaining two questionnaires were completed after the search. One, the post-search assessment (green), was completed immediately after the search while the user was still sitting at the terminal and the other, after the evaluation of the offline prints (pink). Remaining at the terminal became important because users often asked how could they answer these questions without first seeing the offline prints. It was explained that this questionnaire was concerned with the search process and the offline print results would be separately evaluated at a later stage.

These two questionnaires were concerned primarily with the users' overall satisfaction with the search, their impression of the ease or difficulty of the search and a consideration of the results based upon what they had seen during the search. Users were also asked how close was the search to the original enquiry and did they retrieve the number of references they had anticipated. The questions that were asked after they had evaluated the offline prints were concerned with their satisfaction from the references they evaluated and whether in retrospect there were any other concepts that they would like to search for.

10.2.4.2 Evaluation of offline prints

The offline prints included all the documents that were viewed online and judged relevant and all the documents that were retrieved after the query expansion search. CIRT was set to aim for a search size of 50 documents. This meant that CIRT would often need to create large search trees in order to accommodate this target. Aiming for an output set of 50 documents was not chosen at random. Although there is no theoretical justification for this cutoff point it seems that this size is favoured by many researchers probably for practical reasons, i.e. it is a size that is manageable by the users and the cost associated with it is not prohibitive for whomever pays the bill. For example, the Information Transfer front-end (Williams, 1984; 1985), the QUESTQUORUM facility of ESA/IRS, Personal Librarian's default cutoff level, and the CIRT evaluation project all have used the same target of 50 documents. It has also been argued that online searchers in Boolean systems aim at manipulating their final output in order to achieve a set of a similar size (Bates, 1984). The offline prints obtained were also used for relevance evaluation for the effectiveness measurement.

The primary consideration here was whether or not the references retrieved by the search were relevant. In order to give the user as much information about the reference as possible it was decided to supply full format offline prints which included all the information provided by the database producers. The evaluations were to some extent complicated by the users' previous knowledge of any of the retrieved documents. Following the practice of the CIRT evaluation project I adopted the questions for the relevance assessment of the offline prints. In order to overcome this problem two questions were asked about each reference to be evaluated:

A. From the information given, is the document an answer to, or about your subject enquiry? to which the user would reply 'Yes', 'Partially' or 'No'.

B. The user was asked to select any one of the following four categories which best applied to the document under consideration. The categories were:

   1. I have SEEN THE DOCUMENT itself before, and it WAS USEFUL.
   2. I have SEEN THE DOCUMENT itself, but it was NOT USEFUL.
   3. I have NOT SEEN the document represented by this reference, but I WOULD LIKE to see it.
   4. I have NOT SEEN the document represented by this reference and I would NOT LIKE to see it.

The users were asked to answer the questions on the copy of the offline prints. For each bibliographic reference they should use a combination of one letter (Y, P, N) and of one number from the ones above (1-4). A model answer was also provided, e.g. Y3. The evaluations were marked directly on a separate copy of the offline prints. The user was subsequently given their own complete copy of the offline print set to keep.

10.2.4.3 The Logs

Complete logs of all searches were kept automatically by CIRT. These include the intermediary-CIRT interaction and the CIRT-Data-Star interaction and are referred to as user logs and net logs respectively. The user logs print out the searchers transactions with CIRT, and the net logs show CIRT's transactions with Data-Star.

The logs provide the quantitative assessments such as number of terms, number of relevance judgements, number of online/offline prints, etc.

Logs were also kept for all the searches in ESA. Similarly, a record was also kept of all the files created from the programs that processed the downloaded data from ESA and weighted and ranked the terms.

10.2.5 Procedure for data collection: summary

The methodology is summarised and presented in a step-by-step manner below. The summary is then followed by a discussion of the steps which describe the rules and guidelines that were employed as well as how these worked in practice.

Steps:

1. pre-search interview, user completes questionnaire (purple, questions 1-11)
2. select initial query terms (white, search notes)
3. search using CIRT
   (a) start CIRT
   (b) set search size ss=50
   (c) `aof` initial query terms
   (d) `li dstar`
   (e) select database INZZ
   (f) `add all` terms
   (g) `search`
   (h) online relevance judgements
   (i) logoff
4. search in ESA
   (a) login to ESA
   (b) retrieve in one set all the relevant documents found in step 3(h)
   (c) SuperZOOM the set and analyse the CT and UT fields
   (d) do single term searches for all the terms in the list
   (e) logoff

5. weight and rank all candidate terms (from step 4(d)) for query expansion

6. print (and display on the screen) the ranked list of terms

7. user evaluates the ranked list

    (a) user identifies all terms thought to be useful for the search

    (b) user selects query expansion terms

    (c) user completes questionnaire on term selection (blue, questions 12–13)

8. search using CIRT

    (a) repeat search, i.e. steps 3(a)-3(h)

    (b) add query expansion terms

    (c) calculate new weights

    (d) search

    (e) print retrieved documents

    (f) logoff

9. user completes questionnaire assessing the search process before seeing and evaluating the search results (green, questions 14-18)

10. offline evaluation

    (a) user evaluates the search results assigning relevance judgements based on a three point scale with subdivisions (yellow)

    (b) user completes final questionnaire (pink, questions 19-20)

10.2.6 Procedure for data collection: discussion

The steps presented above together with a list of guidelines helped in providing a consistent method for data collection.

Throughout the process, from the pre-search stage to the final evaluation of the offline prints, I had been trying to make the search realistic. In other words, I was facilitating rather than dictating the search. Therefore, I was necessarily allowing myself to respond to particular situations in ways that I did not have rules for. This was appropriate, however, because I had given myself guidelines of how to do certain things. Whenever these did not work or some new situation arose then I would decide what to do at that time. Following that I would reconsider the situation and if it was appropriate I would revise the old guideline or introduce a new one. I will present the guidelines at the steps that correspond in the discussion of the methodology.

When a user was asking for a search I would make sure three criteria were met: firstly that it was a subject search (as opposed to a known item search, such as looking for an author or a specific source); secondly that the user would be present during the search process; and thirdly that the subject of the enquiry was suitable for a search in the INSPEC database.

On the day of the search the users were given the introductory form explaining the project which they were asked to sign. The intention behind the signing of this form was that users would realise their commitments and act in a serious and responsible manner throughout the entire process.

Each user was given a query identification number, e.g. 101. This id number provided the means of keeping all the data (questionnaires, logs, processed files, and offline prints) for each query together.

The user was then given the pre-search questionnaire (purple, appendix D). After the completion of the questionnaire I then examined it and discussed the subject with the user in order to identify the concepts of the request and the candidate query terms. Once the query terms were agreed these were recorded in the search notes (white, Appendix D), i.e., a log that I kept throughout the search and where I noted any decision made or action taken.

10.2.6.1 Query Terms

In IR experiments terms, both initial query terms and query expansion terms, have been combined in many different ways. Most experiments have used all the terms of the initial query and have added a number of query expansion terms. The number of terms added for query expansion has been varied in the experiments depending mostly on the objectives of the experiment. In some IR experiments immediately after the initial search the query terms were renegotiated and only a portion of them, e.g., one third, was included in the expanded query. This latter option was considered but it was abandoned because of CIRT's problem in deleting terms. Therefore, it was decided to adopt the former procedure, i.e., to use all the initial query terms and add a number of terms for query expansion.

The number of initial query terms and the query expansion terms however could only total 8 or 9 because of the CIRT limitations. This lead to the decision to select up to five terms for the initial query, thus allowing for at least 3-4 terms for query expansion.

This situation should also be seen in parallel with the form of the input of the query terms. In other words, a query term can usually be a single term or a multi-word phrase; it can be truncated to match a number of variant word forms as thought best for the search; or it can be a set of synonymous terms ORed together, etc.

Since, the input of query terms could vary considerably in its form it was decided to try the following two approaches:

a1) For the first 12 searches (nos. 101-116) initial query terms, i.e. phrases, single terms, truncation, synonyms ORed, etc., presumed most suitable to the search were chosen.

a2) Similarly, query expansion terms were first selected from the ranked lists but were input to the system as thought to be best, i.e. single terms or phrases or parts of phrases.

b1) For the remaining 13 searches (nos. 117-129) query terms were searched as single terms only, i.e. all phrases were split.

b2) However, query expansion terms were taken as they appeared in the ranked list.

These guidelines were followed as closely as possible although it was difficult to strictly adhere to them. So, some leeway was allowed depending on the situation. The application of the guidelines was left to my discretion.

The reason for using single terms was to simulate a basic system that uses some kind of a pseudo-natural language parsing of the input query. Such system usually splits the input into single words, although one could have a form of input which invites "phrases". However, all these were attempted within the restrictions imposed by the CIRT limitations mentioned earlier. For example, I tried to use only up to five terms in the initial search and if there were any phrases I would split them, but if by splitting the phrases there were more than 5 terms then I would try to do term combinations that would have similar effects.

10.2.6.2 Online Relevance Judgements

Once the initial query terms had been selected, I began the search through CIRT. This was conducted in the way described earlier and which is also summarised in step 3. The methodology that was followed for obtaining online relevance judgements (step 3h) was based on three criteria. The first relates to the questions of which parts of the record relevance judgements should be based on, the second is concerned with the online relevance assessment of the documents, and the third considers the size of the sample of relevant documents to be used for relevance feedback and query expansion.

10.2.6.3 On which document representation should relevance judgements be based on?

Previous research has shown that relevance judgements are influenced by form, i.e. by different document representations, for example, title, citation, abstract and/or full text. Saracevic (1975, p.340) summarised the conclusions of research on the comparative effects on relevance judgements of different document representations in the sixties and seventies as follows:

- Relevance judgements for the same article may be expected to differ from titles to full texts; titles should be utilised with considerable scepticism.
- Relevance judgements for the same article may be expected to differ somewhat from abstracts to full texts, depending upon the abstract's type, length, detail, etc.

Based on these experimental findings it was decided that users should judge relevance of documents by looking at both titles and abstracts. During the data collection, I made sure that users were reading both titles and abstracts before deciding on the relevance of a document. I was particularly insistent that they read the abstract of a document they had doubts about or which they thought not relevant.

10.2.6.4 Relevance assessments

In CIRT, like with most relevance feedback systems, after seeing a document or parts of it, the user is prompted to assess the document's relevance. This judgement is based on a binary value of relevance, so that the outcome can only be either "yes, it is relevant" or "no, it is not relevant".

Relevance is a continuous variable and it has been established in major studies of relevance judgements, that it is an over-simplification to collapse a variety of degrees of relevance into yes/no decisions (Cuadra & Katter, 1967; Rees & Schultz, 1967). However, as discussed in chapter 2 the adoption of the binary definition of relevance is a required simplification of this complex notion which facilitates the calculation of relevance weights.

During the online relevance assessments of documents by users we are faced with two issues. One is that the system requires relevance judgements on a binary scale. The other is that the user must respond on the question of the relevance of a document by judging each document for its relevance, i.e. the extent to which the subject matter of the document is about the query, rather that its usefulness. This treatment of relevance is necessary for relevance feedback systems to be able to provide an estimate of the probability of relevance that will be effective. In other words a document should be judged as relevant to a query only on the merits of the information it conveys and not by comparing it with what we already know or with what we have learnt from the previous documents we have seen during the search, or with how useful we perceive it to be.

The distinction of topical relevance, as defined here, and usefulness is that a document which is relevant but not useful to a user, because for example, the user knows about it, or the document is outdated, or it is written in a foreign language, is very important for relevance feedback systems. The importance of this issue lies in the fact that the online relevance judgements obtained are used by the weighting scheme for the estimation of the relevance weights. Although one can argue that feedback systems would try to predict usefulness, including factors such as date and language, the work on this thesis has concentrated on topical questions therefore it seemed appropriate to ask the users for topical relevance judgements as opposed to usefulness.

In helping users understand the difference and make the relevance judgements I adapted the question "is this that sort of thing you are looking for?" that has been attributed to Robertson and used in the OKAPI online catalogue (Walker & De Vere, 1990, p.26). When CIRT displayed the first document I asked the user:

> "regardless of whether you have seen this document before
> is this the sort of thing you are looking for?"

I then suggested to them that they should give a "YES" or "NO" answer and advised them that if the document was partially relevant or if they were for some reason in doubt they should answer "YES". An additional reason behind this suggestion is that it seems better that the system retrieves marginal documents rather than excludes them. The user then, while online or offline, has the opportunity to decide again about the relevance and usefulness of a document.

Furthermore, the ranking together with CIRT's search trees provide a better way of retrieving and presenting documents according to their relative importance. This allows the user to identify sets with potentially relevant documents by scanning the trees. If a set is ranked high on the list but from the term combination it seems to contain non-relevant documents then the user can easily bypass it.

10.2.6.5 Sample size of relevant documents for relevance feedback

The third criterion associated with the online relevance judgements is the size of the sample of relevant documents on which the system should base its estimation.

The retrieval system with the help of the user (through the online relevance judgements) is trying to model the probability distributions of the relevant and the non-relevant documents, so that the probability of relevance of the documents can be calculated. In section 10.2.1 it was indicated that at least one relevant document must be identified by the user in order to be able to continue with the search.

The sample size of just one document is the absolute bare minimum required for a relevance feedback or query expansion search. A variety of sample sizes has been used in IR experiments. A commonly used method for getting a sample is explained below.

In many relevance feedback experiments the sample is defined at a cutoff level of the 10 or 20 top-ranked documents. The retrieved documents are then examined for relevance and those found relevant are then taken to be the sample for the feedback iteration and the query expansion.

The sample that is of interest here is the latter. The former is of concern only in terms of how many documents should be retrieved by the initial search or of how many documents should the user have to see while online. For example, Harper and van Rijsbergen (1978), Harper (1980), van Rijsbergen *et al* (1981), Smeaton (1981) and Sparck Jones (1979a; 1979b; 1980) have used in relevance feedback and query expansion experiments the 10 or 20 top-ranking documents.

A conclusion of these experiments was that a small sample of relevant documents could be adequate as the basis of the reweighting of terms. Spark Jones (1979b, p.143) used a sample of 3-4 documents and in another experiment (1980, pp.328-329) 1-3 documents and Harper (1980, p.6-3) suggested that at least one document is needed. Nevertheless, this conclusion does not exclude the use of a longer sample. On the contrary, it is believed that the longer the sample of relevant documents the better the estimation should be. However, the problem of selecting an optimal sample size is still very much an open IR research issue. Some suggestions however exist and for some reason all tend to converge to recommending a sample size of 5 relevant documents, for example, Harper (1980, p.6-7). It is also worth mentioning here that Martin (1982, p.73) and White (1989, p.34) in discussing ZOOM both propose without providing further justification, that ZOOM will be more effective if its analysis is based on 5 relevant documents.

Consequently, for the data collection I decided to aim for at least 5 relevant documents. However, the question then became how many documents should the user see before the 5 documents are reached. It was further decided that there should be some flexibility on the number of documents seen rather than strictly following the 10 or 20 cutoff level of earlier IR experiments.

One should also consider that this cutoff level is implemented in CIRT through the command `set searchsize` which has the default value of 15 documents. Since CIRT presents a search tree that includes the number of sets that have been retrieved and the number of documents that are contained in each set it was decided that the user should stop looking at documents either at the end of the set where the total of 5 relevant documents has been reached or before starting a new set that contains a large number of documents, even if the target of 5 documents has not been reached.

Getting a sample set of 5 relevant documents was not always possible, but was quite consistent throughout. There are 17 cases that have a sample of 5 or more positive relevance judgements online. In six cases there are 4 documents (searches: 102, 108, 111, 115, 116, 123) which were judged relevant and in two cases only 2 documents (searches: 105, 129).

10.2.6.6 Identifying, weighting and ranking candidate terms

Once the sample set of relevant documents was identified, the accession number of the documents was noted, these were then searched in ESA, analysed using ZOOM and for each term its ZOOM frequency count and its collection frequency was established (step 4).

Two conditions were imposed on the data following the experiences gained from the Pilot case studies.

1. all terms with ZOOM frequency greater than R, i.e. greater than the size of the sample relevant documents, were assigned the same value as R, i.e. if $r > R$ then set $r = R$.
2. terms in the ranked list that have collection frequency of 1 are expendable, i.e. if $n = 1$ then exclude.

Terms with frequencies $n = r$ should also be expendable, but because ZOOM accumulates within document frequencies over all documents in the set there is no way for one to know by looking at the figures of $n = r$ that these frequencies occur all in the same or in different documents (see also discussion in chapter 9).

For this reason it was decided to keep in the ranked list the terms with $n = r$ and exclude from the list all terms where $n = 1$. This decision was implemented in two stages. During the first stage, i.e. for the first 12 searches (nos. 101-116), single posted terms were left in the ranked list in order to see what proportion of these was selected by the users as terms for query expansion. At the second stage, 13 searches (nos. 117-129), single posted terms were excluded from the list.

10.2.6.7 User selection of terms for query expansion

After the data was processed by the programs (step 5; see also example in Appendix B.4.2) the ranked lists of terms were displayed on the screen as well as printed on paper (step 6).

The list was printed because it was thought it would be easier for the user to go through 2-3 pages of continuous-form paper rather than a number of screens. Users were also given all the time they needed to browse the list, go back and forth and decide on the terms to choose for query expansion. Of course this is not a real environment situation but it was preferred in order to study the term selection process. The users were asked to go through the list of terms twice:

(a) to identify ALL the terms that they considered as being good for the purpose of the search;

(b1) to select the 5 best terms of those identified as good in step (a);

(b2) to rank the 5 terms they selected in descending order of importance to them.

They were asked to be thorough and were told to take as long as they needed to finish this task, i.e. no time constraints were imposed on them.

It could be argued that the ranking and its presentation order had an effect on the users in choosing query expansion terms. Users are more likely to choose terms from the top of the ranked list that is presented to them rather than from the bottom. In other words, users are most likely to stop at some point while going down the list rather than to maintain their concentration till the end of it. There might be a number of factors that make someone decide to quit, e.g. they got satisfied with the terms they found on the list so far, they got disappointed by the length of the list and chose terms from the top, etc.

However, it was thought that term selection bias would be reduced if the user had a clear visual idea of the length of the list as opposed to have to go through an unspecified number of screens. Furthermore, I did not suggest to them that the ranking is significant in any way. What I told them throughout was the (a) and (b) above. Some users understood or inferred the significance of ranking because the list contained also the weight of each term. To those who asked about the significance of ranking I told them to pay no attention to the ranking and to choose terms according to the way they perceived the term's value to be for the search. Therefore, I believe that with this approach the order effect was kept to a minimum if not eliminated.

The evaluation of the ranked list and the selection of the query terms was followed by the completion of the questionnaire on term selection (step 7).

10.2.6.8 Completing the search process

After having selected the query expansion terms I then re-initiated CIRT, repeated the search and included the new terms (step 8). In order to be consistent with searching and try to minimise the variability between searches it was decided to have only one search iteration for the initial search as well as with the query expansion terms.

Once the query search was completed the retrieved documents were not assessed online but were printed. At that point the user completed a questionnaire that elicited the user's assessment of the search process (step 9). The user was then given the printed references and was asked to assess their relevance by following the instructions described in section 10.2.4.2 (step 10). When the evaluation of the offline print was completed users were given the last questionnaire. This asked about their satisfaction with the results and whether in retrospect, now that the search was over, there was some aspect of it that they would like to search on.

10.3 Results and discussion

The results discussed in this chapter are based on the 25 searches finally obtained.

The analysis was essentially directed at the query expansion aspects of the searches, on the ranking algorithm and on retrieval effectiveness. These results are discussed in the section 'main results'. Other aspects of the search including an analysis of the data collected by the pre-search and post-search questionnaires are discussed in the section 'other findings'.

The discussion throughout is mostly qualitative in nature. The data have also been subjected to appropriate statistical analysis using various tests. However, because the sample is small there are occasions where the results are presented only with the intention to demonstrate some trends and to facilitate the discussion and there are no claims of any statistical significance.

10.3.1 Main results

10.3.1.1 Query expansion terms

During the first part of the data collection, i.e. the first 12 searches (nos. 101-116) as mentioned in section 10.2.6.6, single posted terms were left in the ranked list in order to see what proportion of these was selected as terms for query expansion.

Table 10.1 presents the number of query expansion terms used in each of the 12 searches and the number of those that had frequency $n = 1$. For the comparison of the searches the following are also included in the table: the number of the terms selected by the user as potentially useful, the number of terms found in the entire ranked list, the number of single posted terms found in the list, and the number of single posted terms that were selected among the potentially useful terms.

The percentage of searches where a single posted term was among the query expansion terms is 42%. This figure is low especially when considered together with either the proportion of all terms in the entire list (column c in Table 10.1) to the number of single posted terms found in the list (column d) which gives an average percentage of 11%, or the proportion of all the terms selected by the users as potentially useful (column e) to the number of single posted terms that were among those selected as useful (column f) which gives an average percentage of 18%.

The fact that single posted terms that occur in the sample of relevant documents are useless in query expansion, because they will not retrieve any additional documents, had been established in Pilot 3. These results demonstrate that some of these terms are selected

Table 10.1: Totals for query expansion terms and single posted terms in the ranked lists

Search	Number of QE terms (a)	QE terms with $n = 1$ (b)	Terms in ranked list (c)	All $n = 1$ terms in list (d)	All useful terms (e)	$n = 1$ terms in (e) (f)
101	3	0	62	0	10	0
102	4	1	38	2	9	1
103	2	0	137	12	11	1
105	4	1	77	9	14	1
108	4	1	33	3	8	1
110	4	2	61	9	10	2
111	3	0	48	2	31	1
112	4	1	93	11	6	3
113	4	0	65	12	24	7
114	4	0	62	6	15	3
115	4	0	42	9	9	4
116	3	0	77	12	3	0

by users for query expansion. Though their proportion is low it was thought that the exclusion of single posted terms would eliminate any possible interference that these may cause to the searches. For example, the CIRT limitation on the number of query terms that can be used in the search does not allow the luxury of the waste involved by using a single posted term.

10.3.1.2 Term selection characteristics

The users, after the evaluation of the ranked list (step 7 in section 10.2.5), completed the questionnaire on term selection. The variables involved and the procedure for collecting the data have been described in section 10.2.3.7 and section 10.2.6.7 respectively.

The results are presented in two stages. First the answers to the questionnaire are analysed and discussed and then the user preferences of query expansion terms are compared to the system's suggestions.

10.3.1.3 User selection of terms for query expansion

The number of terms that were contained in the ranked lists in each of the 25 searches together with the number of those terms that the users selected as potentially useful are presented in Table 10.2. The percentage of the proportion of the terms chosen by the users to the total number of terms in the list is also given in Table 10.2. The percentage of terms chosen ranged from a minimum of 4% to a maximum of 87% with a mean of 28%.

This implies that on average about one third of the terms given in the lists is thought to be useful. This is approximately an average of 18 terms. If however search 128 is excluded as being an extreme case, because 98 terms were selected out of 113, then the average becomes 15 terms.

Table 10.2: Totals and percentages of terms in the ranked-lists and of terms chosen by the subjects

User	Terms in list	Terms chosen	%[†]
101	62	10	16
102	38	9	24
103	137	11	8
105	77	14	18
108	33	8	24
110	61	10	16
111	48	31	65
112	93	6	6
113	65	24	37
114	62	15	24
115	42	9	21
116	77	3	4
117	64	25	39
118	55	32	58
119	117	34	29
120	44	9	20
121	61	17	28
122	60	13	22
123	61	12	20
124	80	27	34
125	41	21	51
126	39	7	18
127	62	11	18
128	113	98	87
129	34	8	24

[†] N.B. All percentages have been rounded to the nearest integer.

	N	MEAN	STDEV	SEMEAN	MIN	MAX
%	25	28.44	19.19	3.84	4.00	87.00

The questionnaire on query expansion asked the users two questions, each corresponding to some aspect of the term selection process they had followed (see section 10.2.6.7).

At first users identified all the terms in the ranked list that they thought as being useful for the purpose of the search, i.e. they selected terms that would be acceptable to use in the search. The results of this part of the selection process are given in Table 10.2 which presents the total number of terms in each list and the number of terms selected by the users. The users were then asked for the reason(s) that made them chose those terms. How did they think the selected terms related to the search and to the original query terms. They were asked to consider such relationship(s) for all terms collectively rather than for each term individually. Users were given four options to choose from and they could select as many as they thought appropriate. The options were:

[ ] variant expressions or synonyms

[ ] alternative (related) terms

[ ] couldn't find better term(s) to express the subject of the enquiry

[ ] representing new ideas (i.e. not part of your original request)

Figure 10.1 summarises the results of the user perceived association between the terms identified from the ranked list to the query. The percentages given in the figure do not amount to a total of 100% because users were asked to select as many of the four options as applied to the terms. Users thought of the terms they selected as being alternative (related) terms to the query terms for 88% and as variant expressions or synonyms for 64% of the time. These two categories account for the majority of the responses. A very small percentage (4%) chose the terms because they could not find a better term from those on the list to express the subject of the query. A rather interesting result comes from terms that do not relate directly to the original query terms and which represent new ideas. These accounted for 44% of the responses.

This result demonstrates the unpredictability that is involved in subject searching and the difficulties imposed on information retrieval. Additional information was not collected for this category. In retrospect I think some questions could have been included to elicit information about the terms that represent new ideas. Further research is therefore needed into this area. More specifically about the relationships of the 'new ideas' to the original query. What was the reason for choosing these terms? Was the user aware about these new concepts/ideas at the beginning of the search? If yes, why were these not expressed at the pre-search interview? Was the reason for the exclusion interview related?, e.g. communication failure, or did the user chose to exclude them at the interview stage because s/he thought of them as peripheral? If users had not thought of these concepts at an earlier stage, did they knew about them before? Did they recognise and chose these concepts as the result of a learning process during the search? On the whole, what were the reason(s) and stimuli that made them choose these new terms. Answers to these questions I believe will contribute to the understanding of the users' searching behaviour.

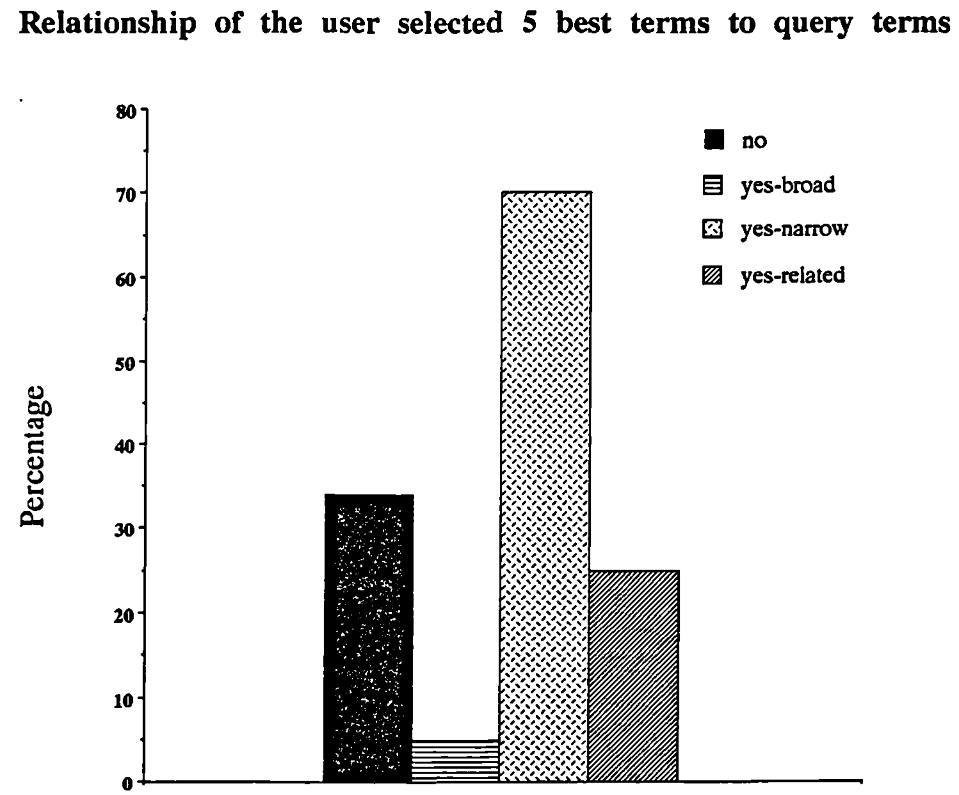


Figure 10.2: Relationship of the user selected 5 best terms to query terms

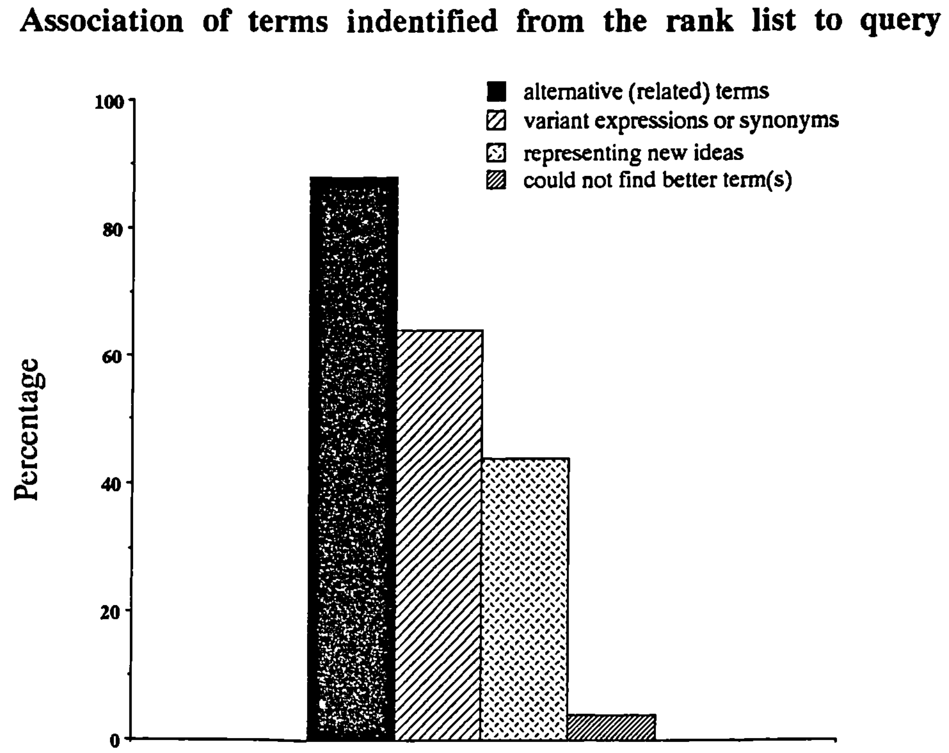


Figure 10.1: Association of terms identified from the rank list to query

The second question asked the users how they perceived the relation of the five best terms to the original query terms.

As mentioned in section 10.2.6.7 users were asked to select the 5 best terms of all those they had identified as useful. The questions on term relationship concentrated on whether there is a correspondence between each of the 5 new terms and the query terms. If the user identified that there was a correspondence between them the type of correspondence was noted. The term relationship that users were asked to select from are among the standard types of relationships found in thesauri, i.e. broader term and narrower term for hierarchical relationships and related term for affinitive/associative relationships.

The relationships of the 5 best terms to the query terms are shown in Figure 10.2. For 34% of the chosen terms there is no relationship or other type of correspondence to a query term. The remaining 66% of the terms is divided as follows. A narrower term relationship between a selected term and a query term accounts for 70% of the responses. A broader term relationship accounts for 5%. An associative relationship (i.e. related term) holds for 25% of the terms.

From these results it can be established that approximately 75% of the term associations fall within a hierarchical relationship. Users have overwhelmingly selected narrower terms as the terms for query expansion. This finding is very important and emphasises the possible advantages that may be involved if an online thesaurus is used. Such a thesaurus could assist users in looking-up terms, establishing their relationships and deciding on term inclusion, as well as for exploiting hierarchical relationships. A possible way to include terms could be in clusters or in hierarchies. For example in a fashion similar to that of the 'explode' command in Medline, where an 'exploded' term retrieves itself as well as all the terms in the hierarchy beneath it. However, all these should be user-controlled or 'machine-aided' operations rather than entirely automated. This suggestion comes from the poor results achieved from automatic query expansion in earlier information retrieval experiments.

10.3.1.4 Evaluating the ranking algorithm

To evaluate the effectiveness of a ranking algorithm, for ranking terms for query expansion, IR experiments to date have focused on retrieval effectiveness. In other words, the effectiveness of ranking is measured through retrieval effectiveness expressed in terms of recall and precision. This means that an algorithm would be effective if during the experiment the top ranked terms which are used for query expansion will achieve high recall levels, if tested only by itself in a test collection, or a higher recall level than some other algorithm, if it is a comparative test. This approach has been the focus for evaluating algorithms for automatic query expansion.

Interactive query expansion introduces new dimensions in the way algorithms could be evaluated. One such dimension comes from the users themselves, who are the ultimate judges of the performance of the system, and it is introduced here. Evaluation of the ranking algorithm was performed through the user selections of terms.

As mentioned earlier, an effective ranking algorithm, will bring the good terms at the top of the list. Users, on the other hand, studied the lists and identified all the useful terms as well as the 5 best terms.

The method employed for the evaluation was by assessing the distribution of the user selected terms over the ranked list. The lists were divided into three equal parts, i.e. top third, middle third, bottom third, and into two equal parts, i.e. top half and bottom half. For each list, the terms identified by the users were tallied according to their distribution in each third and in each half. In other words, the concentration of terms in each part of the list was examined.

The term distribution, of all the terms chosen by the users as being good terms to be included in the search, over the ranked lists, divided into three parts is given in Figure 10.3. The data used for the figure are given in Table D.2 in Appendix D. Figure 10.3 presents the mean percentages for the 25 searches. At the top third there is a concentration of 49% of the terms. The middle third and the bottom third have 31% and 19% of the terms respectively. Figure 10.4 (data given in Table D.1 in Appendix D) presents the term distribution of all the terms chosen over the ranked lists which was divided into two parts. The top half contains 68% of the terms and the bottom half 33%.

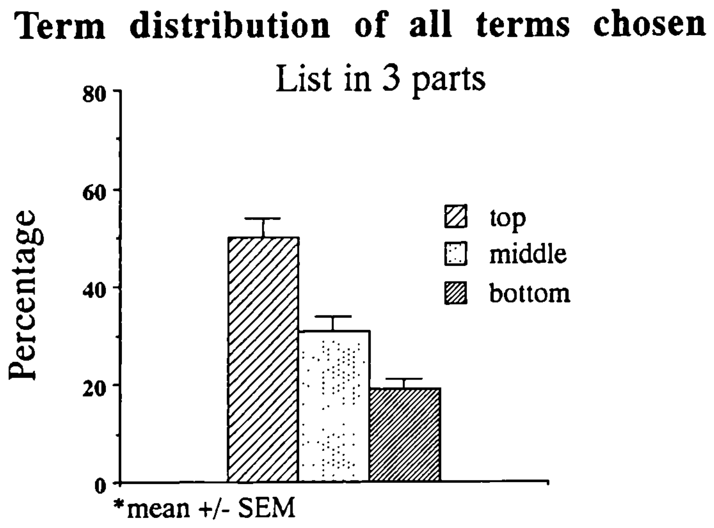


Figure 10.3: Term distribution of all terms chosen: list in 3 parts

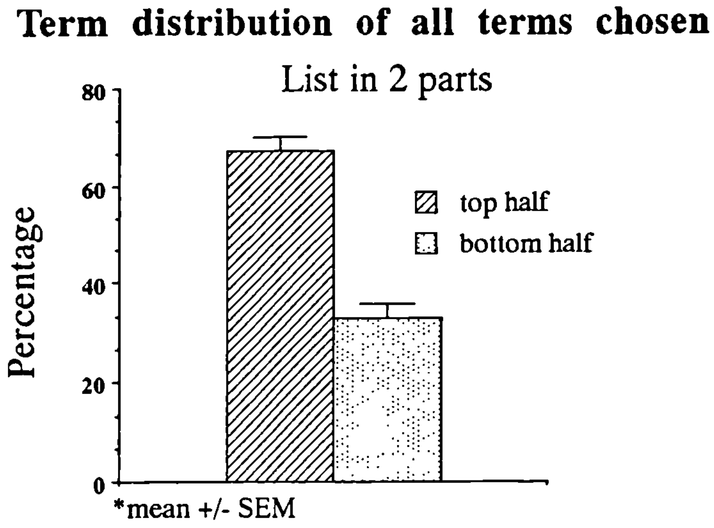


Figure 10.4: Term distribution of all terms chosen: list in 2 parts

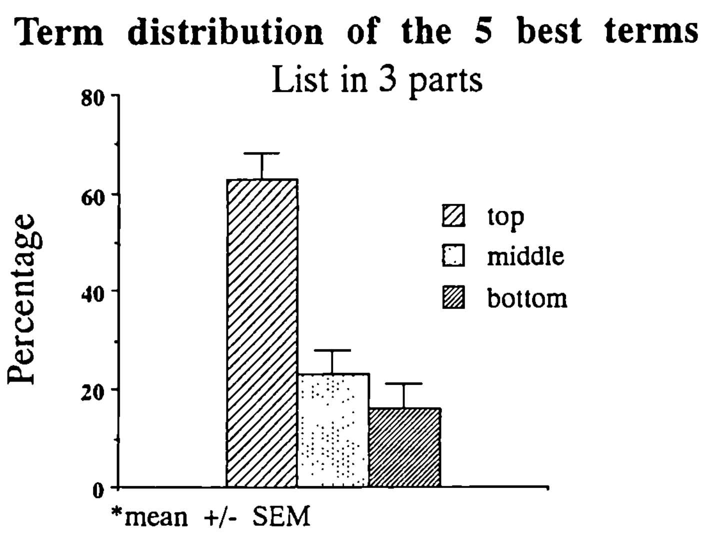


Figure 10.5: Term distribution of the 5 best terms: list in 3 parts

From these averages it can easily be established that the ranking algorithm has performed as it should. That is, that the ranking has separated the terms and on the whole has brought the good terms at the top of the list. However, in order to strengthen these findings a second level was added to the evaluation. At this stage the distribution of the 5 best terms over the ranked lists was assessed. The assumption being that if the concentration of the 5 best terms, as identified by the users, was also found to be at the top of the list, then this would corroborate and enhance the strength of the findings.

The results are given in Figure 10.5. The data are taken from Table D.3 in Appendix D. The concentration of the 5 best terms at the top third is 63%, in the middle third 23% and at the bottom third 14%. This implies a positive correlation between what the ranking list suggests as good terms and what the users choose as good terms. The results obtained from this evaluation indicate that the ranking algorithm is working well. This also seems to justify the choice for using it in this experiment. However, full justification of the algorithm cannot be qualified beyond this point. In order to do so the performance of the algorithm should be evaluated against the performance of the other algorithms that were considered in section 7.4. Such an evaluation is presented in chapter 11.

10.3.1.5 Retrieval effectiveness

Retrieval effectiveness, as mentioned in section 10.2.3.1, is based on relevance judgements for all documents retrieved and which were printed in full format. The results from the relevance assessments are discussed below, first for the offline prints and then for the correspondence of the online to the offline relevance assessments.

The number of documents assessed offline for each of the 25 searches is given in Table D.4 in Appendix D. The number of citations evaluated ranged from 16 to 66 with an average of 30 citations per search.

As discussed in section 10.2.4.2 relevance was evaluated first in a three-point scale ('Yes', 'Partially' or 'No') and then one of four choices was selected that reflected the usefulness of the citation to the user. A detailed breakdown of the relevance assessments for the 12 resulting categories is given in Table D.4. For each search the table gives the user assessments by category and the corresponding totals and percentages. A summary of the results is given in Figure 10.6. It is observed that there is reasonable correlation, i.e. everything is correlated as expected. In other words, the 'want-to-see' decline from the relevant to the non-relevant scale (33%, 22%, 11%) whereas the 'do-not-want-to-see' increase from the relevant to non-relevant (11%, 18%, 30%).

There is therefore a strong sense of distinction between relevance and usefulness, i.e. relevance and 'want-to-see'. There are a few not-relevant documents indicated as 'want-to-see' (11%) and a few relevant indicated 'do-not-want-to-see' (11%). These two are interesting results that raise the question of what is the reasoning behind the decisions. Data to answer this type of questioning have not been collected. The latter case is perhaps more obvious because one can think of good reasons for it whereas for the former it is only the user who could identify what triggered his/her interest to that particular item. By looking at the questionnaires and the citations identified as relevant and 'do-not-want-to-see' I can say that the apparent reasons for not wanting to see them were mainly because the citations were rather old or in a language other than English.

The average percentage of the relevant, partially relevant and not-relevant documents over the 25 searches is 42%, 30%, and 29% respectively. Retrieval effectiveness was calculated for both relevancel, i.e. only relevant documents, and relevance2, i.e. relevant and partially relevant documents. Table D.5 gives figures for relevancel and relevance2 of each search, the total number of retrieved records and the calculated precision for each search which is estimated for relevancel and relevance2. The overall precision (averaged over the 25 searches) for relevancel is 42% (for an average of 12 relevancel documents retrieved) and for relevance2 is 71% (for an average of 21 relevance2 documents retrieved). The figures achieved for precisionl and precision2 are very similar to those reported at the CIRT evaluation project which were 50% (for an average of 15 relevancel documents retrieved) and 73% (for an average of 23 relevance2 documents retrieved) respectively (Robertson & Thompson, 1987, p.26). However, the figures for CIRT are combined for both the Medline and INSPEC searches. The discussion of the effectiveness of query expansion follows the discussion of the comparison of the online and offline relevance judgements.

10.3.1.6 Correspondence of online and offline relevance judgements

During a search the user judged the relevance of a number of documents online. The positive judgements were used by CIRT to train itself and find additional documents. When the search was completed the user was presented with all the citations that were retrieved and was asked to evaluate them. Among the citations were also the ones that were given a positive relevance judgement online. The issues that are raised relate to how these documents were assessed offline. Were they judged relevant? Is there a consistency between the users' online and offline relevance evaluations? How do the relevance judgements break down? The two sets of relevance judgements are expected to correlate, therefore, the question is not whether they correlate but how well do they correlate, i.e. how strong the correlation is. These issues relate to the variables of retrieval effectiveness and user effort discussed in sections 10.2.3.1 and 10.2.3.2. The methodology is described by the following steps:

1. identify the positive online relevance judgements (get Accession Number)
2. match the above positive judgements in the offline prints
3. extract user assessments and tally by category
4. analyse results

The results are given in Table D.6 and Table 10.3. A breakdown of the correspondence of the online positive relevance judgements to the offline judgements is presented in Table D.6.

Table 10.3.presents the total number of those identified as relevant and the precision ratio that corresponds to each of them. In addition, the table gives the figures of the online relevant judgement that fall in the offline relevancel and relevance2 categories as well as their corresponding precision ratios.

Among the studies on relevance assessments the study by Resnick and Savage (1964) assessed the consistency of human judgements of relevance. They studied the ability of

Table 10.3: Results of the correspondence of the online to offline relevance judgements

	seen	relevant	precision	rel1	rel2	prec1	prec2	Prec1	Prec2
search	online	online		offline assessments		relevant online		seen online	
			(2:1)	of column (2)		(4:2)	(5:2)	(4:1)	(5:1)
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
101	5	5	1.00	4	5	0.80	1.00	0.80	1.00
102	4	4	1.00	4	4	1.00	1.00	1.00	1.00
103	10	6	0.60	3	6	0.50	1.00	0.30	0.60
105	8	2	0.25	0	1	0	0.50	0	0.12
108	15	4	0.26	2	2	0.50	0.50	0.13	0.13
110	11	8	0.72	6	8	0.75	1.00	0.55	0.73
111	5	4	0.80	3	4	0.75	1.00	0.60	0.80
112	11	5	0.45	2	5	0.40	1.00	0.18	0.45
113	12	5	0.41	3	5	0.60	1.00	0.25	0.42
114	13	6	0.46	5	6	0.83	1.00	0.39	0.46
115	12	4	0.33	3	4	0.75	1.00	0.25	0.33
116	15	4	0.26	4	4	1.00	1.00	0.26	0.26
117	11	5	0.45	2	3	0.40	0.60	0.18	0.27
118	10	7	0.70	4	6	0.57	0.86	0.40	0.60
119	9	5	0.55	0	2	0	0.40	0	0.22
120	10	8	0.80	0	8	0.88	1.00	0	0.80
121	14	5	0.35	3	5	0.60	1.00	0.21	0.36
122	14	8	0.57	5	8	0.63	1.00	0.36	0.58
123	4	4	1.00	4	4	1.00	1.00	1.00	1.00
124	10	6	0.60	2	4	0.33	0.66	0.20	0.40
125	12	5	0.41	0	5	0	1.00	0	0.42
126	19	5	0.26	3	4	0.60	0.80	0.16	0.21
127	9	7	0.77	4	7	0.57	1.00	0.44	0.78
128	14	12	0.85	12	12	1.00	1.00	0.86	0.86
129	16	2	0.12	1	2	0.50	1.00	0.06	0.12

humans to judge consistently the relevance of documents to their general interest from different document representations, i.e. citations, abstracts, keywords and full text. They concluded that their subjects were able to make such judgements consistently.

Looking at the results of the online to offline correspondence of relevance judgements, i.e. columns (2) & (4) and (2) & (5) in Table 10.3, it can be established that the assessments are also quite consistent. The Pearson's product moment correlation shows that the correlation is high between online assessments and relevance1 (r = 0.853) and becomes even stronger between online assessments and relevance2 (r = 0.932).

The average precision of the positive online assessments (column 2 in Table 10.3) against the corresponding offline relevance assessments of relevance1 (column 4) is 59% (column 6) and of relevance2 (column 5) is 89% (column 7). The average precision achieved from the online relevance judgements (column 3) is 56%.

It would be of interest, from a relevance point of view, to analyse the documents that were seen online but were rejected as non-relevant. That is, the negative relevance judgements online could be printed and re-assessed for relevance by the users offline. However, this kind of analysis could not be done with the CIRT's current setup as documents judged non-relevant online are automatically excluded from the offline prints.

10.3.1.7 Retrieval effectiveness of the query expansion search

Following the previous section of the correspondence of the online and offline relevance judgements the next question that is addressed relates to the retrieval performance of the intial search as compared to the query expansion search. It is necassary to re-iterate the procedure followed. Documents seen in the initial search and judged relevant were included in the offline prints; those judged non-relevant online were excluded. All documents retrieved in the query expansion search were included. However, for the purpose of the present analysis, we should consider the documents judged non-relevant online to have been retrieved.

First, the online relevance assessments are matched to the offline assessments for relevance1 and relevance2. Then precision is calculated for the offline assessments against the total number of documents seen online, as a measure of precision of the initial search. Secondly, the online positive relevance assessments are excluded from the final sets of assessments of relevance1 and relevance2. The result of this operation is two sets which contain the number of relevant documents that were retrieved from the query expansion search. The two sets are used for the calculation of the precision achieved in the query expansion searches. These figures are then compared to the precision of the initial search as described above in order to establish the levels of performance of the different stages of the search and see the effect of query expansion.

The number of documents assessed online for each of the 25 searches is given under the column 'seen online' of Table 10.3. The number of citations assessed ranged from 4 to 19 with an average of 11 citations per search.

The number of positive relevance judgements (column 2 of Table 10.3) ranged from 2 to 12 with an average of 5 citations per search. As discussed earlier, this average, i.e. 5

Table 10.4: Results of the query expansion searches

search	relevant	rel1	rel2	Rel1	Rel2	QERel1	QERel2	total	total	QEprec1	QEprec2
	online	offline assessments		total offline				printed	QEretrieved	offline	offline
		of column (1)		printed		(4-2)	(5-3)	offline	(8-1)	(6:9)	(7:9)
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
101	5	4	5	23	44	19	39	46	41	0.46	0.95
102	4	4	4	14	20	10	16	25	21	0.48	0.76
103	6	3	6	19	26	16	20	26	20	0.80	1.00
105	2	0	1	14	31	14	30	66	64	0.22	0.47
108	4	2	2	10	17	8	15	57	53	0.15	0.28
110	8	6	8	18	24	12	16	53	41	0.29	0.39
111	4	3	4	6	17	3	13	18	14	0.21	0.93
112	5	2	5	5	9	3	4	48	43	0.07	0.09
113	5	3	5	5	9	2	4	17	12	0.17	0.33
114	6	5	6	8	15	3	9	17	11	0.27	0.82
115	4	3	4	14	24	11	20	25	21	0.52	0.95
116	4	4	4	18	37	14	33	37	33	0.42	1.00
117	5	2	3	2	9	0	6	18	13	0	0.46
118	7	4	6	10	13	6	7	21	14	0.43	0.50
119	5	0	2	20	44	20	42	48	43	0.47	0.98
120	8	0	8	0	8	0	0	21	13	0	0
121	5	3	5	12	18	9	13	21	16	0.56	0.81
122	8	5	8	18	32	13	24	35	27	0.48	0.89
123	4	4	4	17	27	13	23	28	24	0.54	0.96
124	6	2	4	10	17	8	13	20	14	0.57	0.93
125	5	0	5	7	14	7	9	16	11	0.64	0.82
126	5	3	4	3	4	0	0	18	13	0	0
127	7	4	7	7	12	3	5	25	18	0.17	0.28
128	12	12	12	29	29	17	17	32	20	0.85	0.85
129	2	1	2	8	15	7	13	21	19	0.37	0.68

documents, is the sample size of relevant documents I aimed at getting for the analysis and extraction of query expansion terms. The precision achieved online, averaged over the 25 searches, is 56% (for an average of 5 relevant documents retrieved).

The precision of the positive online assessments as identified offline for relevance1 (column 4) and for relevance2 (column 5) against the number of documents seen online (column 1) is given in Table 10.3. The average precision, over the 25 searches, for relevance1 (column 8) is 34% (for an average of 3 relevance1 documents retrieved) and for relevance2 (column 9) is 52% (for an average of 5 relevance2 documents retrieved).

Table 10.4 presents the data of the query expansion part of the results. In order to facilitate the discussion some data presented earlier are repeated in this table.

Column 1 gives the total number of positive relevance judgements online. Column 2 and column 3 present the results of the offline assessments of column 1 for relevance1 and relevance2, i.e. the number of those documents identified as relevant online and which were also judged to be relevant or partially relevant offline. Column 4 and column 5 give the total number of offline relevance judgements for relevance1 and relevance2 respectively. Column 6 gives the number of relevance1 documents retrieved by the query expansion search and is derived by subtracting column 2 from column 4. Similarly, column 7 presents the number of relevance2 documents retrieved by the query expansion search and is derived by subtracting column 3 from column 5. The total number of documents printed offline in each search is given in column 8 whereas column 9 presents the total number of documents attributed to the query expansion searches. The precision of the query expansion part of the searches is given in column 10 (QEPrec1) and column 11 (QEPrec2) respectively. QEPrec1 is calculated by dividing QErel1 (column 6) over QEretrieved (column 9) and QEPrec2 by dividing QErel2 (column 7) over QEretrieved.

The average precision for QErel1, over the 25 searches, is 37% (for an average of 9 relevance1 documents retrieved), and for QErel2 is 65% (for an average of 16 relevance2 documents retrieved). These results demonstrate that the query expansion search has been effective and that precision has been increased slightly for a substantial increase in the documents retrieved.

The level of precision for relevance1 is very similar for the initial and query expansion search with 34% and 37% respectively. However, there is a threefold increase of the number of relevant documents retrieved for the query expansion search (i.e., from 3 to 9 documents). Precision for relevance2 has been increased from 52% in the initial search to 65% in the query expansion search. In addition, there is also a threefold increase in the number of documents retrieved (i.e., from 5 to 16).

10.3.1.8 Discussion on retrieval effectiveness and online vs offline judgements

One reason behind any differences between the online to offline assessments lies behind the instructions that the users were given for making the online assessments. As mentioned earlier users were asked to mark as relevant the references that seemed to them as peripheral or they had doubts about.

Overall, these results demonstrate that there is a consistency between the online and offline relevance judgements. The consistency in judgements is also seen through the figures of the average precision achieved online and offline.

A reason for differences between online and offline assessments and between relevance1 and relevance2 is the combination of internal information, (e.g. knowledge on the subject, how they intend to use the information, etc.) that the users brought in at the onset of the search and the learning that took place during the search and during the evaluation of the offline prints. In other words, there are a few references that users see online which they judge for relevance with their state of knowledge at that moment. However, with each new piece of information they are exposed to their knowledge changes/restructures. The magnitude of the change depends on the amount of previous knowledge they had and can be coupled to the anticipated use of the stimulus. As restructuring takes place the relevance assessment on an item may vary mainly because of the potentially different usefulness perceived for it. The twelve options given to the users for assessing relevance capture changes and provide a detailed breakdown not found in other scales. However, changes due to learning are difficult to measure because they require in-depth interviews and are case specific. To answer those questions more studies like the present are required. Such studies, however, are difficult because at the moment there are not many relevance feedback systems available for large operational environments.

The final question is whether these measures of performance are meaningful to the users of an interactive retrieval system? As Cleverdon (1974) has commented, precision and recall may serve well for testing effectiveness of system components in a controlled environment but they do not necessarily serve well in information retrieval situations with real users. A recent study reports that "...user satisfaction with completeness of search results or value of search results as a whole appear to be the best single measure of successful IR performance..." (Su, 1989). Some of the variables addressed in the questionnaires dealt with these issues and are presented in the following section.

10.3.2 Other findings

The results obtained from the analysis of the questionnaires that did not relate directly to query expansion are presented and discussed in this section. These would provide additional information that complete the picture of the searches and would assist in the better understanding of the results. The data are presented in summary form and the results are discussed in general terms under. Results are presented by questionnaire (pre-search and post-search) and for each question separately.

Pre-search questionnaire: results and discussion

10.3.2.1 User status

Data on the user status are given in Figure 10.7. The data correspond to question 1 of the purple questionnaire (Appendix D page 246) and relates to variable (V4) user characteristics as defined in the methodology. The user population was exclusively from an academic environment (lecturers, researchers and students). This when coupled with the subject matter of the INSPEC makes a rather homogenous and well defined sample population. Doctoral students account for 56% of the user population, which together with faculty (16%) and researchers (4%) represent 76% of the population. The remaining 24% was comprised of 3rd year undergraduate students.

10.3.2.2 Intended use of information

The results from the question of the user intentions of usage of the information that the search would provide are given in Figure 10.8. The data are from question 2 of the purple questionnaire and relate to variable (V4) user characteristics. PhD dissertation research (56%) and research projects (20%), i.e. research related use, accounts for 76% of the intended use. The remainder 24% is for course related use, i.e. 3rd year projects. This pattern corresponds exactly to that from the results of the use status.

10.3.2.3 User's assessment of the nature of the enquiry

Figure 10.9 presents the results of the users' general assessment of the nature of their enquiry. Data derived from question 3 of the purple questionnaire and relate to variable (V5) request characteristics. An accurate or precise assessment of the enquiry was indicated by 64% of the users. The remaining 36% indicated the nature as general and no one thought of it as vague.

10.3.2.4 Work done on the problem

Data that indicate the user's progress on the project, that was identified by the question on the intended use of the information received from the search, is given in Figure 10.10. The data correspond to question 4 of the purple questionnaire and relate to variables (V4) user characteristics and (V5) request characteristics.

Users were given a scale from 1 to 5, where 1 represents beginning to think about the project and 5 is the end of the project. The results in the figure show a slightly skewed distribution with 56% indicating that they are at early stages of their projects. 20% indicated to be half way through, another 20% approaching the end of the project and 4% indicated as being at the very last stage of it.

10.3.2.5 Clarity of the problem

Data that express the users' own assessment of their level of knowledge about the subject of the project or reason that made them request the search were collected from question 5 of the purple questionnaire. The results are given in Figure 10.11. Users could assess their knowledge on a 5-point scale where 1 represents 'know-nothing' on the subject and 5 'know a lot'. The responses tend to concentrate in the centre with 44% indicating the mid-scale value 3, 28% the value 2, 20% the value 4 and 8% were very confident about their knowledge.

The responses that represent the first half of the scale amount to 72% which corresponds very well to the 76% that represents the first half of the scale for the work done on the problem.

10.3.2.6 Type of search required

The results from the question of what type of search is required are presented in Figure 10.12. The data were derived from question 6 of the purple questionnaire and relate to the variable (V5) request characteristics. The choice for broad searches, i.e. one that is aimed to retrieve all the references including peripheral material, accounted for 68%, whereas 32% chose a narrow search, i.e. only very specific references.

10.3.2.7 Familiarity with the process of online searching

Questions 7 and 8 of the purple questionnaire collected data on the users familiarity with online searching; relate to variable (V4) user characteristics. Question 7 asked if the users have had online searches done for them before and, if they had, how many did they have. The results show that 56% of the users never had a search done for them. The 44% that replied that they had searches done for them, was further divided into 73% that had 1 to 3 searches done for them and 27% that had 4 to 10 searches done for them.

Question 8 asked the users about their personal experience with online searching, i.e. whether they had ever searched for themselves. The results show that the majority of the users, 84%, had never searched by themselves while 16% had.

Post-search questionnaire: results and discussion

10.3.2.8 User's satisfaction with the search

The user's satisfaction with the search, i.e. the impression they got immediately after the online session was completed but before the evaluation of the offline prints, was measured on a 5-point scale: excellent, good satisfactory, poor, bad. Figure 10.13 reports on these data that were collected from question 14 (green questionnaire, page 249, relates to variable (V6) search process characteristics). The majority of the users, i.e. a total of 96%, expressed a positive satisfaction. The breakdown is 12% for 'excellent', 72% for 'good' and 12% for 'satisfactory'. Negative satisfaction as negligible with 4% for 'poor' while no one selected 'bad'.

10.3.2.9 User's assessment of the search

The user's impression of the ease or difficulty of the search was elicited from question 15 (green questionnaire, variable search process characteristics). This question was concerned with their impression about the search technique, i.e. weighted retrieval and relevance feedback, rather than with the technical aspects of the search, i.e. set up and implementation of the interface, speed, etc. Although this difference was explained to the users it seems that many of them could not easily separate the two. That is the technical limitations of CIRT influenced their decision. Nevertheless, as seen in Figure 10.14 almost half of the users (44%) thought that search was 'easy' and 28% that it was 'average'. Finally, the remaining 28% thought of it as 'difficult '.

From the study of the responses on the questionnaires it is established that those who judged the search as 'difficult' had very low familiarity with computers. These who said that it was 'average' commented that they did so mainly because of the overall complexity of the set up for the experiment and the time requirements for the search rather than for the search technique.

10.3.2.10 User's assessment of the results

The user's overall impression on the quality of the results was elicited in question 16 (green questionnaire). Again, in this question, as with the previous ones of the same questionnaire, the user was asked to comment judging from what was seen during the search, i.e. before having a chance to look at the offline prints. The responses were measured on a 5-point scale: excellent, good, satisfactory, poor, bad. The results are shown in Figure 10.15. The majority of the responses 92% were on the positive end of the scale, 20% 'satisfactory', 68% 'good' and 4% 'excellent'. Negative feeling about the results was very low (8% for 'poor', and there were no responses for the scale 'bad').

The user's impression of the results is very positive (92% on the positive end of the scale). The total of the most positive responses, i.e. excellent and good, amounts to 72%. This corresponds exactly to the 71% overall precision level (calculated for relevance2). Judging from the correspondence of the online to offline relevance judgements and of the pre- and post-evaluation of the results it seems that the user responses are rather consistent throughout.

10.3.2.11 Match of search to enquiry

The user's feeling about the closeness of the online search to their original or intended enquiry was addressed by question 7 (green questionnaire). The responses were measured on a 3-point scale and are presented in Figure 10.16. The closeness was 36% 'exact' and 64% 'fairly close' and no one felt that the search 'considerably altered' their enquiry.

10.3.2.12 Expected references

The last question of this questionnaire (green, question 18), which was given before the evaluation of offline prints, asked whether the users felt that the number of references retrieved was as they expected it to be. The responses were measured on a 3-point scale and the results are given in Figure 10.17. Almost half of the users (48% thought that the number of references retrieved was 'more than expected', 32% said it was 'less then expected', and 20% found it 'as expected'.

The comparison of user status (Q1) with the intended use of information(Q2), the clarity of the problem (Q5) (i.e. user's knowledge on the subject) and the expected references (Q18) shows that those users who felt they knew a lot about the subject could provide better estimates of how many references they were expecting to get from the search. On the other hand, users who did not know the subject matter well (most of those who were at the beginning stages) were underestimating or overestimating the number of references.

10.3.2.13 User's satisfaction with the results

After users had completed the relevance assessments of the offline prints they were given the last questionnaire (pink questionnaire on page 250). This contained two questions that elicited information about their satisfaction with the references and whether, in retrospect, they thought of any other areas or concepts that they would like to search on.

An overwhelming 96% responded that they were satisfied with the references. A 68% said that there was nothing else in relation to the original/searched query that they would like to search on. The 32% that expressed an interest in another search is further divided in those who wanted to repeat the search in another database (16% of the total), to search additional (usually more specific) concepts (12%) and to do the same search all over again (4%). Those who suggested to search on another database felt that the subject matter of their queries was multidisciplinary and that INSPEC's coverage was addressing only one aspect of it. Such subjects included industrial chemistry (search 101, 126) and medical informatics (search 108).

10.3.3 Concluding remarks

The *Experiment* has provided useful information about interactive query expansion and relevance feedback through a front-end system to online databases. The aim was to look at the process of interactive query expansion when searching in an operational situation with real users and see what can be learnt from it.

The methodology as formed through the experiences gained from the Pilot case studies provided consistency during searching and was proven to be effective. There were not any surprises during the data collection and ambiguities were resolved by following the guidelines set by the methodology or as the occasion demanded.

The results demonstrate that single posted terms should be eliminated from the list of the candidate terms for query expansion presented to users. In 42% of the searches users chose a single posted term. Therefore, the elimination of them will prevent users from selecting such terms which besides being useless, if used will provide poor results and cause disappointment or frustration to users.

A pattern that emerges from the user selection of terms for query expansion suggests that about one third of the terms from a list of candidate terms are potentially useful. This finding has design implications for a facility for the selection of candidate terms for query expansion and their presentation to the user. Such facility can be applied to both types of query expansion, interactive or automatic.

Although this finding can be further quantified to approximately 15-18 terms per search I will be inclined to use in a facility for query expansion the one third of the terms of the list for each search rather than a fixed number of terms. However, an alternative might be to use a combination of the two, that is to display a fixed number of terms and also give the option, upon request, to browse more terms.

Users identified mostly as 'related terms' or 'synonymous terms' the relationship between the terms they chose from the list and the initial query terms. A breakdown of the relationship of the 5 best terms chosen by the users to the query terms reveals that the hierarchical relationship predominates. Query expansion terms were mainly narrower terms to the corresponding query terms.

The evaluation of the $w(p - q)$ algorithm was performed through the user choices and demonstrated the effectiveness of the algorithm, for the ranking of terms for query expansion.

The average precision ranged from 42% to 71% for relevancel and relevance2 respectively. When these figures are matched to the responses concerning the users satisfaction about the results it is concluded that users were satisfied with this outcome. The correspondence of the online to the offline relevance judgements demonstrates an overall consistency in user judgements.

The results provide some evidence for the effectiveness of interactive query expansion. The initial search produced on average 3 highly relevant documents at a precision of 34%; the query expansion search produced on average 9 further highly relevant documents at slightly higher precision.

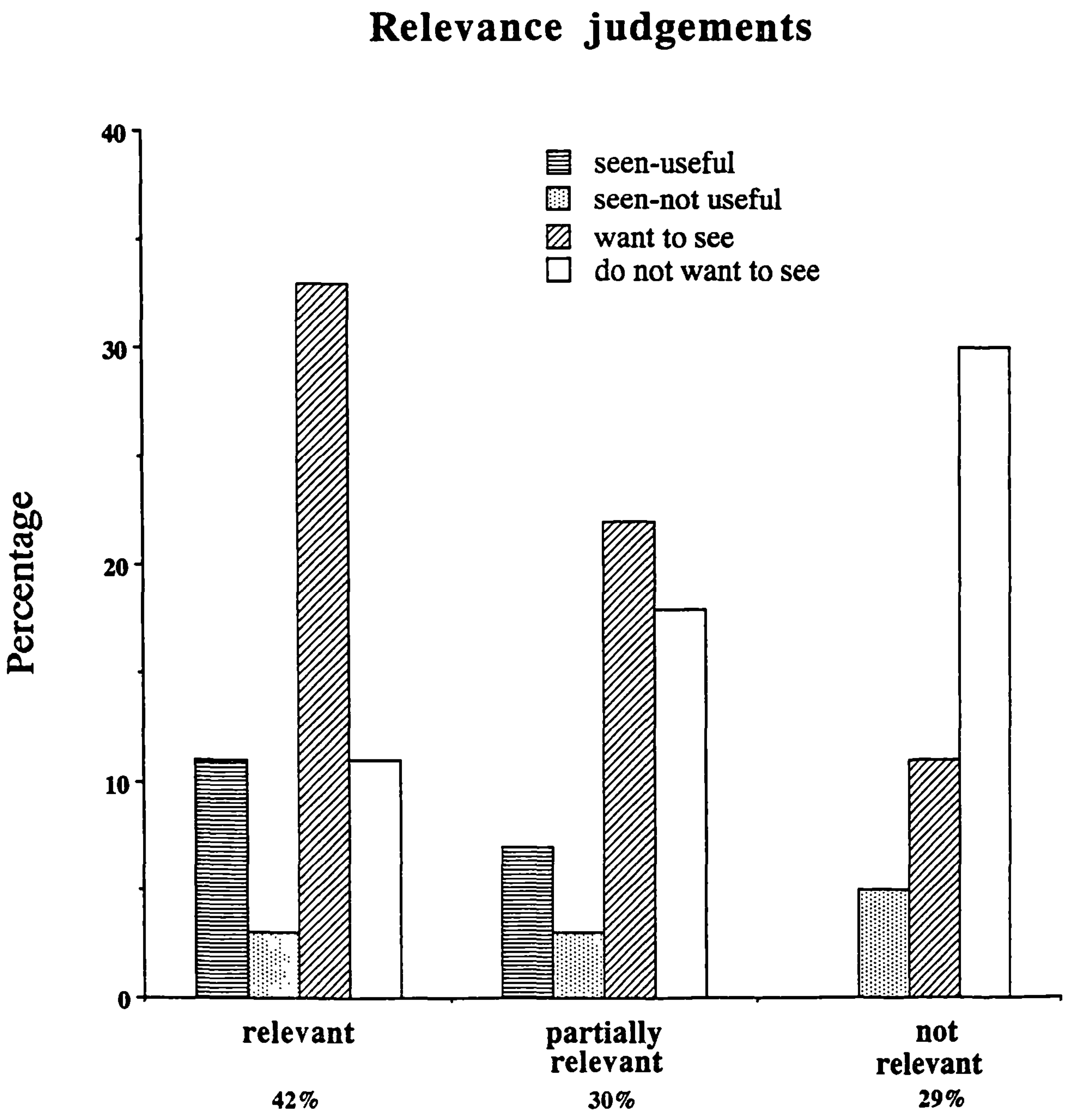


Figure 10.6: Relevance judgements

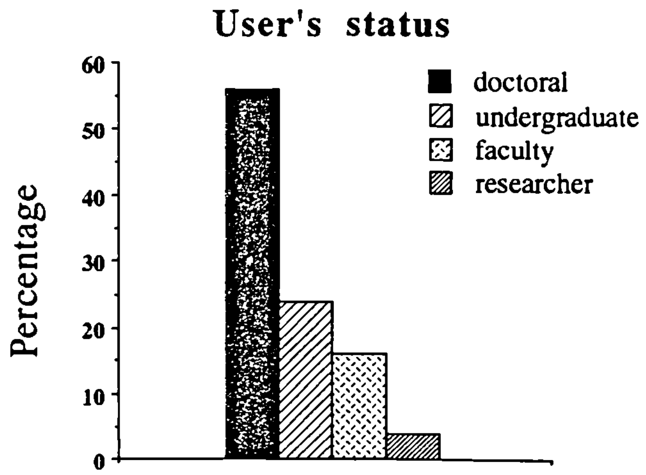


Figure 10.7: User's status

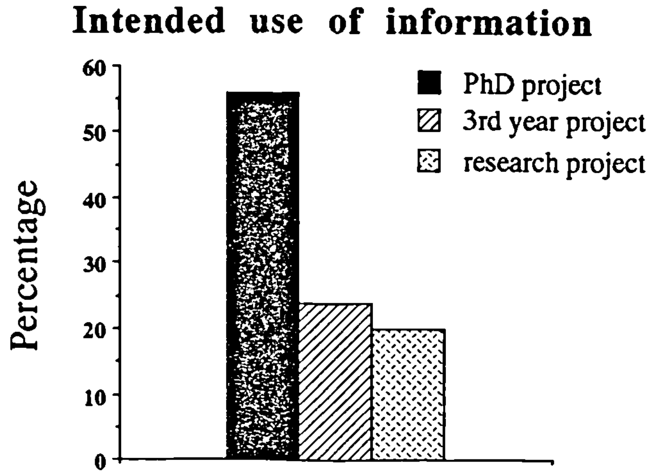


Figure 10.8: Intended use of information

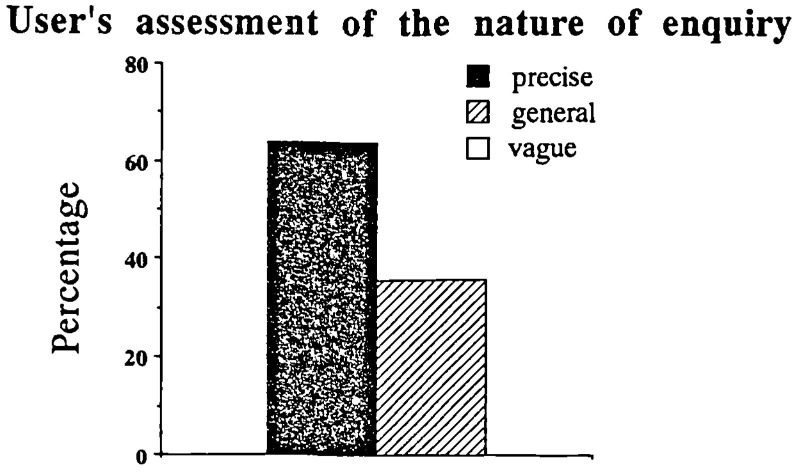


Figure 10.9: User's assessment of the nature of the enquiry

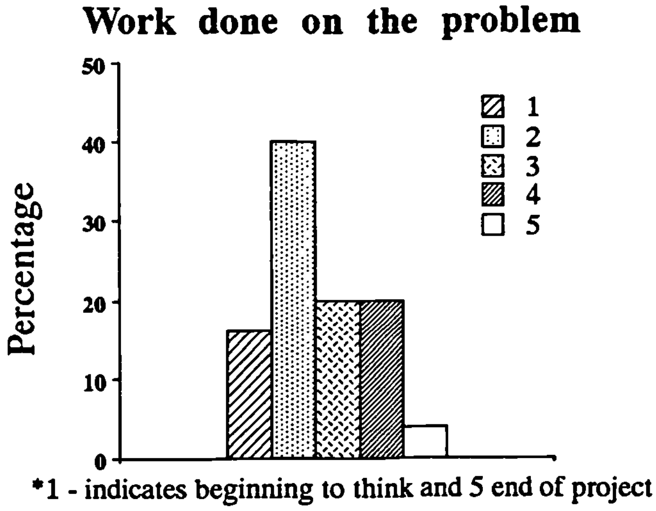


Figure 10.10: Work done on the problem

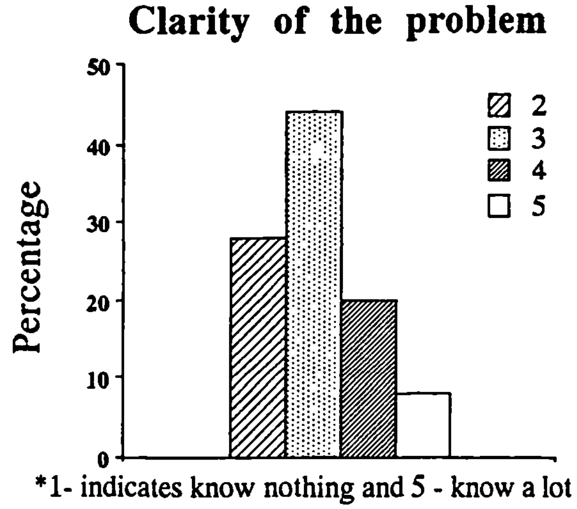


Figure 10.11: Clarity of the problem

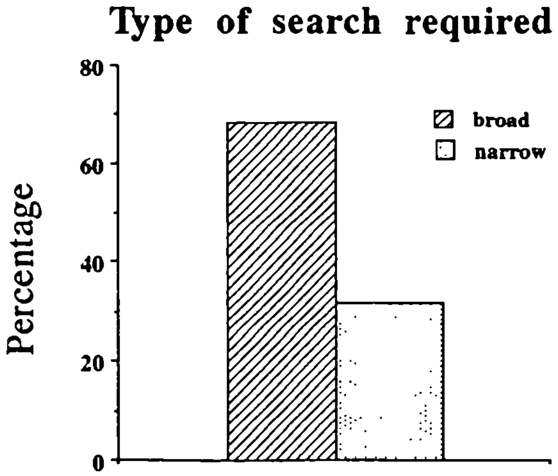


Figure 10.12: Type of search required

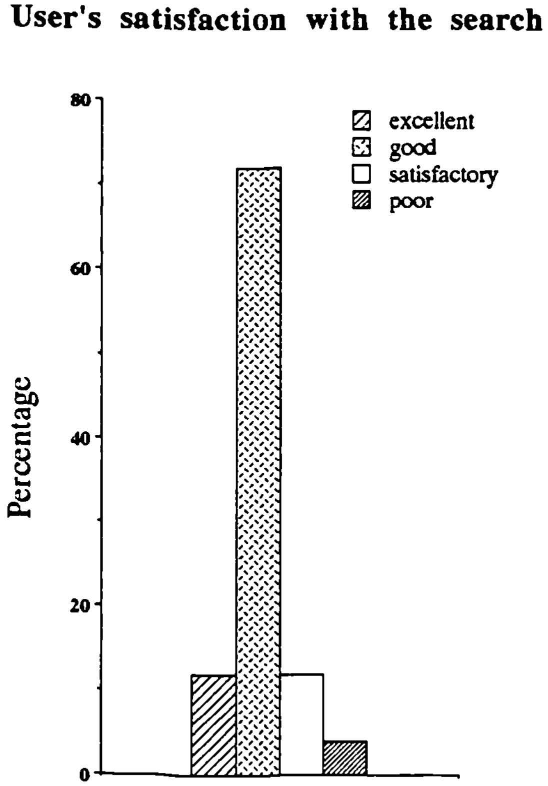


Figure 10.13: User's satisfaction with the search

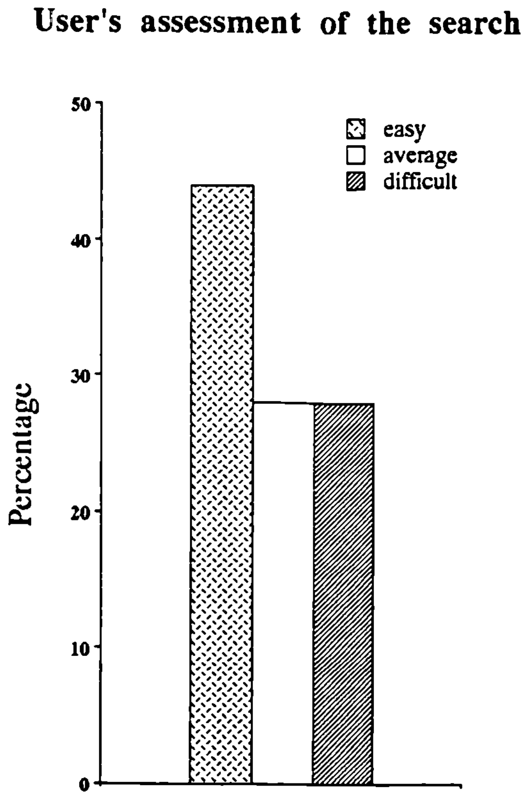


Figure 10.14: User's assessment of the search

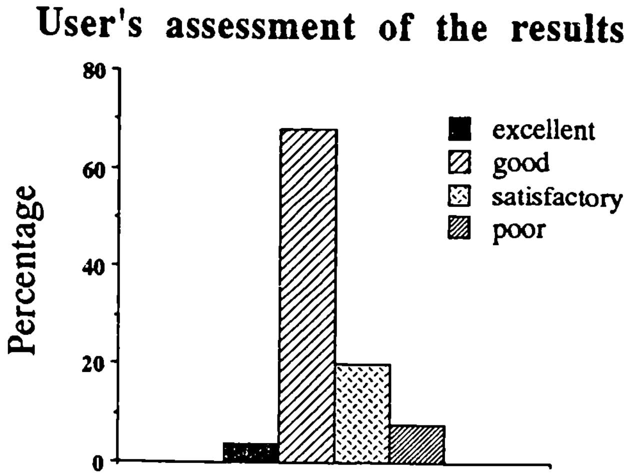


Figure 10.15: User's assessment of the results

Chapter 11

Evaluation of the six ranking algorithms

11.1 Introduction

During the presentation and discussion of some of the results in chapter 10 it had been assumed that the ranking algorithm, i.e. $w(p-q)$, was not in question. This assumption was made in section 10.2.3.7 in order to facilitate the study of the term selection characteristics. In discussing the evaluation of the $w(p-q)$ algorithm, within the context of this study, as presented so far, and with the above assumption, it was concluded that the algorithm seemed to be effective.

The $w(p-q)$ algorithm was adopted as the algorithm to be used in this study for the reasons explained in section 7.4 That was the state of knowledge about the six algorithms at that point and the decision was justified. However, now that the data collection of the experiment has been completed there are data available that do have some bearing on the questions of ranking. This opens up again the question, raised in section 7.4, of which ranking method is better.

Are there any similarities in the performance of the six algorithms? How do algorithms with similar performance rank the terms? In order to answer these questions a number of tests were performed and are described below.

11.2 Methodology

Because of the emphasis on the interactive aspect of query expansion, in this investigation, the use of test collections for the evaluation of the efficiency or effectiveness of the algorithms was excluded (see chapter 6). How then are the algorithms going to be evaluated? What should the criteria be? The solution was found in the user supplied judgements. Users had seen lists of terms and had provided relevance judgements for each term. They selected terms in two groups. One contained all the terms that they thought to be useful for the search. The other group contained the 5 best terms from those in the first group. These term choices were taken as the basis of the evaluation. In other words, given the user preferences for terms, how do the algorithms rank them? The user responses, presented in section 10.3.1.4, were matched against the ranking of the five remaining algorithms.

The methodology followed for the ranking of the terms of each search by the remaining five algorithms was:

1. extract terms presented to users for each search (N=25)
2. calculate weights for the terms of every search with each of the five algorithms, i.e. F4, F4modified, ZOOM, Porter, EMIM
3. divide each of the resulting ranked lists into:
   (a) 2 parts (top half, bottom half), and
   (b) 3 parts (top third, middle third, bottom third)
4. match the user choices of terms to each ranked list
5. for each list, tally the distribution of all the terms over each part, i.e. over:
   (a) top half, bottom half, and
   (b) top third, middle third, bottom third
6. for each list, tally the distribution of the user designated 5 best terms over the top, middle and bottom thirds

The second stage of the evaluation of the six ranking algorithms was to study the top 5 ranked terms of each list. I would like to point out the differences between the 5 top ranked terms and the user designated 5 best terms. The latter are the 5 terms that users identified as being the best terms out of the terms that they checked as useful. These terms were used for query expansion. The former, however, are the 5 terms at the top of the ranked list which are the best terms according to each of the algorithms used to rank them.

The objective of this part of the evaluation was to look at the top ranked terms and compare them in a qualitative manner. This type of analysis reports on the general impression obtained about the terms.

The last part of the evaluation concentrated on the user designated 5 best terms. Since these terms are the most important for the users they should be studied in more depth. Therefore, the emphasis that each algorithm has placed on these terms was measured through the rank position of each of them.

The ranks of the terms were added and the sum was used for comparisons. The rationale is that the sum of the ranks of the chosen terms would indicate the relative importance that each algorithm gives to the user preferences. Then by comparing the differences between the sums of pairs of the algorithms it can be established which algorithm comes closer to user preferences and whether there are any significant differences between the algorithms. In addition, correlation can be used as a measure of association between each pair of the algorithms. The following methodology was used:

1. assign ranks (from 1 to N) to the terms in all ranked lists (6 × 25)
2. establish rank position for each of the 5 best terms in each of the ranked lists
3. add the rank position information for the 5 terms of each list
4. use the Wilcoxon test to find the statistical significance on the performance of the algorithms.
5. calculate the Pearson correlation coefficient r for pairs of algorithms.

For the statistical analysis the Wilcoxon test was chosen because this test is based on ranks, it is for matched-pairs and it is distribution-free (Lehmann, 1975). The Pearson product moment correlation coefficient, r, was used because it operates on each pair of data as they are and not on ranks as does Spearman's ρ. For each search there is a pair of values (the sum of the ranks) that correspond to two algorithms and this information is maintained with Pearson's r.

11.3 Results and Discussion

The results for each of the three tests are discussed below.

11.3.1 Distribution of the terms chosen by the users

The distribution of the terms chosen by the users is given in the tables in Appendix E.

Tables E.1 to E.6 present the distribution of all the terms chosen by the users, as being potentially useful for each algorithm, over the lists which are divided in two parts. A summary of the statistics of these tables are given in Table 11.1. This table presents for each of the six algorithms the mean percentage value over the 25 searches for the top and the bottom halves. In order to facilitate the comparison of the algorithms the data of Table 11.1 have been used in Figure 11.1. The concentration of terms at the top half ranged from as low as 57% for ZOOM to 68% for $w(p-q)$, EMIM and Porter. F4 and F4modified with 65% were not far behind the three top rated algorithms.

Tables E.7 to E.12 in Appendix E give the distribution of all the terms chosen by the users over the ranked lists which are divided in three parts. A statistical summary of these tables is presented in Table 11.2. This table gives for each algorithm the mean percentage values for the top, middle and bottom thirds. The results presented in Table 11.2 are illustrated by Figure 11.1.

As seen in this figure the three way division of the ranked lists is more sensitive to the user preferences. It highlights the part of the list where the highest concentration is and in this way it points out where each algorithm places emphasis on the list.

The concentration of terms at the top third ranges from 39% for ZOOM to 49% for $w(p-q)$ and EMIM. In the middle third the concentration ranges from 31% to 35% and

Table 11.1: Summary statistics: Percentage distribution of all terms chosen by the users. Ranked lists divided into 2 parts.

TOP HALF (All Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	68	15	3	43	100
emim	25	68	16	3	43	100
f4	25	65	19	4	36	100
f4mod	25	65	19	4	36	100
porter	25	68	16	3	43	100
zoom	25	57	18	4	25	100

BOTTOM HALF (All Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	33	15	3	0	57
emim	25	31	16	3	0	57
f4	25	35	19	4	0	64
f4mod	25	35	19	4	0	64
porter	25	32	16	3	0	57
zoom	25	42	18	4	0	75

Table 11.2: Summary statistics: Percentage distribution of all terms chosen by the users. Ranked lists divided into 3 parts.

TOP THIRD (All Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	49	18	4	0	83
emim	25	49	19	4	0	86
f4	25	45	19	4	0	83
f4-mod	25	46	20	4	0	86
porter	25	47	17	3	0	83
zoom	25	39	17	3	13	82

MIDDLE THIRD (All Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	31	17	3	9	100
emim	25	31	18	4	0	100
f4	25	35	20	4	11	100
f4-mod	25	35	20	4	0	100
porter	25	32	17	3	0	100
zoom	25	33	13	3	11	67

BOTTOM THIRD (All Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	19	12	2	0	43
emim	25	19	12	2	0	43
f4	25	19	14	3	0	43
f4-mod	25	19	14	3	0	43
porter	25	19	12	2	0	43
zoom	25	27	15	3	0	63

Table 11.3: Summary statistics: Percentage distribution of the 5 best terms chosen by the users. Ranked lists divided into 3 parts.

TOP THIRD (5 Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	63	24	5	0	100
emim	25	64	25	5	0	100
f4	25	54	25	5	0	100
f4mod	25	54	25	5	0	100
porter	25	60	25	5	0	100
zoom	25	43	20	4	0	80

MIDDLE THIRD (5 Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	23	24	5	0	100
emim	25	22	24	5	0	100
f4	25	31	23	5	0	100
f4mod	25	30	24	5	0	100
porter	25	25	25	5	0	100
zoom	25	29	22	4	0	80

BOTTOM THIRD (5 Terms)						
ALGORITHM	N	MEAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	14	17	3	0	60
emim	25	14	17	3	0	60
f4	25	16	17	3	0	60
f4mod	25	16	17	3	0	60
porter	25	15	17	3	0	60
zoom	25	28	21	4	0	75

in the bottom third the concentration of terms is 19% for all the algorithms but for ZOOM which is 27%.

From these results a pattern emerges for all algorithms except ZOOM. This is that the distribution of all the terms over the lists is on average proportional to 20% – 30% – 50% for the bottom, middle and top thirds, whereas for ZOOM it is to 30% – 30% – 40%. In any case, however, $w(p-q)$ and EMIM algorithms, with 49%, have the highest concentration of terms at the top third.

Having looked at the distribution of all the terms I now turn to the distribution of the 5 best terms. Tables E.13 to E.18 present the distribution of the user designated 5 best terms over the lists which are divided into three parts. A statistical summary of these tables is given in Table 11.3. This table presents the mean percentage values of the term distribution in the top, middle and bottom thirds for each algorithm.

The results presented in Table 11.3 are also given in figure 11.1. The breakdown of the user identified 5 best terms as seen in Figure 11.1 provides a finer way of looking at the results. Not only it highlights the part of the list where the concentration is, but it is a better method for bringing out the differences between the algorithms.

The distribution of the 5 best terms concentrates at the top third and ranges from 43% for ZOOM to 54% for F4 and F4modified, to 60% for Porter, to 63% for $w(p-q)$, to 64% for EMIM. The distribution of $w(p-q)$ and EMIM is the best and it is very similar. This is followed closely by Porter.

11.3.2 The 5 top ranked terms of each algorithm

Tables E.19 to E.43 in Appendix E give the 5 top ranked terms for each algorithm in every search. Each table is divided into six smaller tables one for every algorithm. For every term its weight, term frequency (n), and frequency within the online relevant document set (r) are given in order to facilitate comparisons.

The overall impressions from the study of the 25 tables are summarised below. By looking at the lists the emerging pattern is that the terms between $w(p-q)$ and EMIM, F4 and F4modified, and Porter and ZOOM are very similar. That is the ranking between these pairs of algorithms is very similar, for an example see Table E.28.

The observation of $w(p-q)$ and EMIM show that the top 5 positions contain almost the same terms throughout. However, the rank orders differ slightly. The terms and the rank order found between F4 and F4modified are so close as to be almost identical. Porter and ZOOM, which are highly influenced in their rankings by r, give very similar terms and rank positions in almost all searches. The few cases, like in Table E.22, where differences occur are due to the way ties were resolved by the algorithms. As has been mentioned earlier and seen in the example, ZOOM sorts ties in alphabetical order. Porter and Galpin (1988) have not given any information of how ties should be resolved. The processing programs (in Appendix B.4.2) used in the data collection and for the ranking of the lists in this chapter have sorted the weights in reverse numerical order which apparently ranked the ties in reverse alphabetical order.

The most noticeable differences seem to appear between the terms of F4 and F4modified and the terms of the remaining algorithms. Furthermore, there appears to be a marked distinction between the two groups. The differences are mainly due to factors that influence the ranking in each algorithm. F4 and F4modified are influenced primarily by n while the remaining algorithms are influenced by r. Therefore, high frequency terms do not do well in F4 and F4modified as they do in $w(p-q)$.

11.3.3 Sum of ranks of the user designated five best terms

Table 11.4 gives the sum of the ranks of the 5 best terms selected by the users for each search and for each algorithm. Columns represent the six algorithms and rows give the sum of the ranks for a search under each algorithm. A statistical summary that describe these data, averaged for each algorithm, is also given in Table 11.4. The summary gives the mean, median, standard deviation, standard error of the mean, and the minimum and maximum value of sums of ranks for each algorithm. The best mean values are given by $w(p-q)$ and EMIM which are followed by Porter, F4modified, F4 and ZOOM, in this order.

The results shown in Table 11.4 were subjected to statistical analysis with the Wilcoxon test in order to test whether there is any significant difference in the performance of the algorithms. The algorithms were taken in pair combinations and were analysed with the Wilcoxon test using the Minitab[1] Data Analysis Software version 7. The results of the tests in terms of level of significance for each pair of algorithms are given in Table 11.5. From these results it is established that pairs of algorithms that have significant difference are all combinations of ZOOM, i.e. $w(p-q)$ vs ZOOM and EMIM vs ZOOM at the 1% level, F4 vs ZOOM, F4modified vs ZOOM and Porter vs ZOOM at the 5% level. All other pair combinations of the algorithms show no significant difference.

The Pearson product moment correlation coefficient, r, calculated for pair combinations of the algorithms is given in Table 11.6. Since, correlation measures the strength of the association between two variables, the results in Table 11.6 show the strength of the relationship between the algorithms. Strong positive relationship is given by all pairs that have a value of $r > 0.800$. The strongest association is found between F4 and F4modified ($r = 1.000$), $w(p-q)$ and EMIM ($r = 0.999$), $w(p-q)$ and Porter ($r = 0.998$) and EMIM and Porter ($r = 0.998$).

11.4 Concluding Remarks

In this chapter the six algorithms, that were discussed in section 7.4, were reconsidered and evaluated for their effectiveness in ranking terms for query expansion.

In developing algorithms for the ranking of terms, in this case for the ranking of query expansion terms, the IR researcher focuses on the question of which might be the best terms to add during query expansion. What IR research tries to get from a ranking algorithm for that purpose is:

[1] Minitab is a registered trademark

Table 11.4: Sum of the ranks of the 5 best terms

Subject	$w(p-q)$	EMIM	F4	F4mod	PORTER	ZOOM
101	148	144	107	107	149	169
102	76	80	101	100	82	69
103	209	207	230	229	220	403
105	115	117	159	160	116	205
108	61	63	72	71	60	64
110	71	71	82	79	83	156
111	32	33	101	98	32	97
112	112	112	92	92	117	253
113	70	73	53	53	78	236
114	45	45	108	106	49	133
115	54	54	75	75	57	82
116	104	99	92	92	107	69
117	125	124	94	95	133	182
118	82	82	86	86	85	179
119	199	198	207	205	203	439
120	43	41	31	31	46	83
121	68	68	80	79	71	164
122	115	116	116	116	123	89
123	174	174	199	199	176	159
124	100	101	178	176	102	89
125	112	109	95	94	115	118
126	61	63	55	55	74	122
127	80	79	119	117	84	92
128	227	226	224	222	237	284
129	43	43	40	40	43	114

Statistical summary of the sum of the ranks

ALGORITHM	N	MEAN	MEDIAN	STDEV	SEMEAN	MIN	MAX
$w(p-q)$	25	101.0	82.0	54.1	10.8	32.0	227.0
EMIM	25	100.9	82.0	53.4	10.7	33.0	226.0
F4	25	111.8	95.0	56.1	11.2	31.0	230.0
F4mod	25	111.1	95.0	55.8	11.2	31.0	229.0
Porter	25	105.7	85.0	55.3	11.1	32.0	237.0
ZOOM	25	162.0	133.0	98.3	19.7	64.0	439.0

Table 11.5: Significance levels for the Wilcoxon test on pairs of the algorithms

Algorithms	p–value
$w(p-q)$ vs EMIM:	$p = 0.99$
$w(p-q)$ vs F4:	$p = 0.49$
$w(p-q)$ vs F4mod:	$p = 0.52$
$w(p-q)$ vs Porter:	$p = 0.77$
$w(p-q)$ vs ZOOM:	$p = 0.010$
EMIM vs F4:	$p = 0.48$
EMIM vs F4mod:	$p = 0.51$
EMIM vs Porter:	$p = 0.76$
EMIM vs ZOOM:	$p = 0.0096$
F4 vs F4mod:	$p = 0.96$
F4 vs Porter:	$p = 0.70$
F4 vs ZOOM:	$p = 0.033$
F4mod vs Porter:	$p = 0.73$
F4mod vs ZOOM:	$p = 0.030$
Porter vs ZOOM:	$p = 0.017$

Table 11.6: Pearson's r correlation for pairs of algorithms

algorithm	$w(p-q)$	EMIM	F4	F4mod	Porter
EMIM	0.999				
F4	0.850	0.856			
F4mod	0.855	0.861	1.000		
Porter	0.998	0.998	0.841	0.846	
ZOOM	0.729	0.733	0.613	0.615	0.737

(a) to get the best terms at the top of the list;

(b) to second-guess the user on which terms to choose

These are the ideal goals that a ranking algorithm tries to achieve. However, these two goals are not necessarily compatible. For example, a ranking algorithm that treats terms according to some theoretical argument might not necessarily propose what the users will choose. To put it in another way, the terms people choose to search with might not be the best terms for query expansion according to some theoretical argument. However, since it is the users who search and them that we try to second-guess the current evaluation has tested which terms users choose to search and how these are ranked by the six algorithms.

The methodology that was followed in measuring the distribution of terms chosen by the users as seen in Figure 11.1 is effective in demonstrating how the algorithms ranked the user preferences. The three way division of the ranked lists is particularly sensitive to the preferences. The breakdown of the user identified five best terms provided a fine look at the results.

There appears to be very little difference especially between $w(p-q)$ and EMIM and between F4 and F4modified, because they produce very similar ranking. Overall, there are significant differences in order between the former pair of algorithms and the latter, However, there are not any significant differences in performance.

Porter's algorithm has a very similar performance to $w(p-q)$ and EMIM. However, the $\frac{r}{R}$ component of the algorithm dominates to such an extent that the $\frac{n}{N}$ becomes noise and gets lost in the rounding, i.e. it is so small that its information content is lost. This means that actually Porter's algorithm produces a ranked list very much like ZOOM, i.e. dominated by r. The difference in performance that is reported here between the two algorithms is explained by the resolution of ties, Porter does not specify how ties should be resolved. In my version of the Porter algorithm, ties were sorted in frequency order from high to low whereas ZOOM sorts ties in alphabetical order.

In conclusion, the $w(p-q)$ and EMIM algorithms have outperformed all others in the ranking of the user preferred terms for query expansion. The concentration of user preferred terms at the top parts of the list achieved by these two algorithms is high, 68% at the top half for all terms and 63% at the top third for the 5 best terms. Such concentration levels are acceptable for interactive query expansion because the user can browse the list and can recognise terms. However, for automatic query expansion such level might not be acceptable. Additional testing is needed for further justification of the findings.

A final result that is concluded from this evaluation and the study of the behaviour of the algorithms is the proposal for a new ranking algorithm. By inspection of the rankings all algorithms discussed here might reasonably be replaced by a simpler algorithm that will:

- rank terms according to r, i.e. their frequency of occurrence in the relevant document set, and
- resolve ties according to their term frequency, n, from low to high frequency.

This algorithm is proposed as the result of the observation of the behaviour of the six algorithms studied. It seems that within the setup of the current study the proposed algorithm would have an almost identical ranking to Porter and a performance approaching that of $w(p-q)$ and EMIM. More differences between the algorithms may occur if the size of the set of relevant documents (R) gets larger. Conclusions about the algorithm however cannot be drawn before it is fully evaluated against the six algorithms.

The proposed algorithm seems to be easier and probably cheaper to implement than the other algorithms presented here. Therefore, it can be easily implemented for the ranking of terms for query expansion. Furthermore, such algorithm is independent of a retrieval technique and can be used by vendors of Boolean systems for the ranking of terms for query expansion.

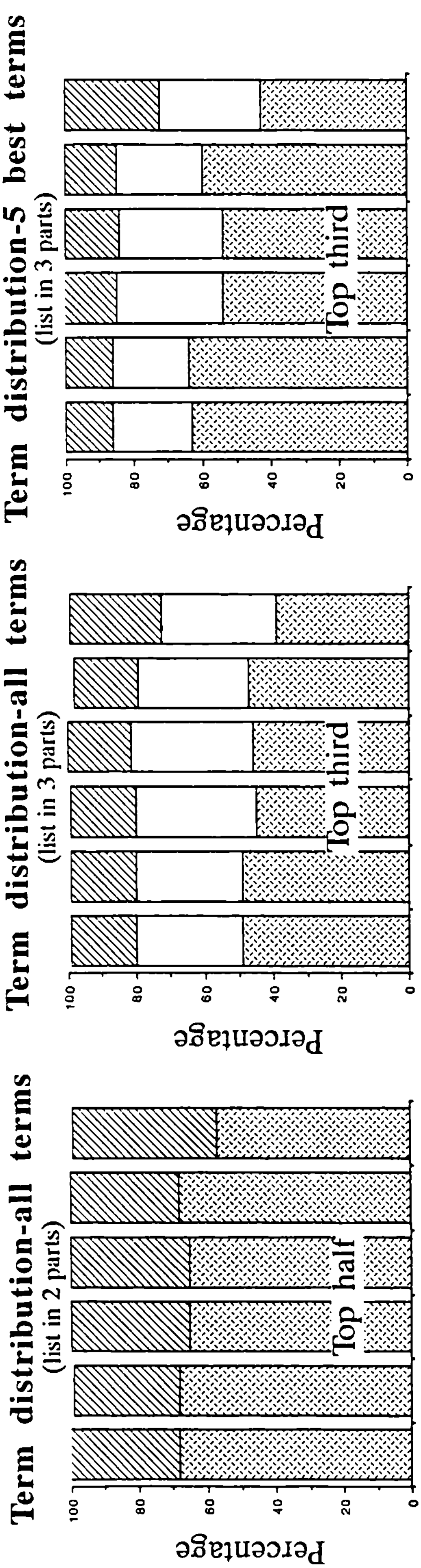


Figure 11.1: Distribution of the terms chosen by the users for each algorithm

Chapter 12

Conclusions and Recommendations

This study aimed at investigating interactive query expansion within the context of a system that is based on relevance feedback, and that uses term weighting and ranking in searching online databases that are available through hosts. So far the evaluation of relevance feedback systems has been established in laboratory conditions and not in a real operational environment.

To investigate the process of query expansion as well as its effectiveness one needs a real system, real requests and real interaction. The CIRT front-end system, that allows relevance feedback, weighting and ranking, was used for searching the information requests of real users. This investigation of interactive query expansion has been performed mainly within the context of the CIRT system.

In order to be able to conduct such an investigation a number of issues pertaining to it had to be resolved. For that reason the study was divided in two stages. The *Pilot* case studies answered questions that were necessary for the formulation of the methodology used during the *Experiment.* The *Pilots* also assisted in building experience, checking on the options available for certain tasks, selecting from the choices offered and exploring avenues in building and testing the methodology. For these reasons the *Pilot* studies are part of the methodology.

The findings of the Pilot case studies are summarised below.

Pilot 1:

1. The $w(p-q)$ algorithm was adopted for the ranking of the terms for query expansion because of its theoretical justification and its relatively better performance over the other algorithms.
2. The set of relevant documents that is derived from the online relevance judgements of the first search iteration was selected as the source for the query terms.

Pilot 2:

1. The study of the CIRT evaluation data revealed that the search intermediaries did not make use of query expansion during their searching. It was therefore very difficult to arrive at some firm conclusion of how searchers do query expansion in a weighted environment like that of CIRT. Consequently, the finding that query expansion was not used in searching can be taken as evidence of the necessity to provide searchers with help for query expansion.

Pilot 3:

1. The overlap between the terms taken from the documents of the online relevance judgements (initial set) and that of the documents of the offline relevance judgements (final set) is very low. The overlap is determined mainly by the query terms. This indicates that the treatment of the subject matter in the document of the final set is different from that of the initial set. The significance of the similarities and differences can only be established in qualitative terms by the users.

2. Terms in the initial set that have a frequency of $r = n$ are not useful for query expansion. Terms in the final set with frequency $r = n$ could be very useful for query expansion but these cannot be accessed through a process of query expansion that is based on the online relevant document set.

Experiment:

The main conclusions drawn from the results of the *Experiment* as pertaining to term selection for query expansion by the users, and to the evaluation of the ranking algorithms are summarised below:

1. one third of the terms presented in the list of candidate terms was on average identified by the users as potentially useful for query expansion.

2. these terms in their majority were thought by the users to be either variant expression (synonyms) or alternative (related) terms to the initial query terms. However, a substantial portion of the selected terms were identified as representing new ideas.

3. The relationship of the 5 best terms chosen by the users for query expansion to the initial query terms was defined as:

   (a) 34% no relationship or other type of correspondence with a query term

   (b) 66% of the query expansion terms have a relationship which makes them:
   (b1) narrower term (70%)
   (b2) broader term (5%)
   (b3) related term (25%)

4. The results provide some evidence for the effectiveness of interactive query expansion. The initial search produced on average 3 highly relevant documents at a precision of 34%; the query expansion search produced on average 9 further highly relevant documents at slightly higher precision.

5. The results demonstrate the effectiveness of the $w(p-q)$ algorithm, for the ranking of terms for query expansion, within the context of the *Experiment.* This conclusion is reached judging from the ranking of the terms, the distribution in the ranked lists of the terms designated by the user as potentially useful, the average precision level achieved, and the users' expressed satisfaction with the final results.

6. The results of the comparative evaluation of the six ranking algorithms, i.e. $w(p-q)$, EMIM, F4, F4modifed, Porter and ZOOM, are:

   (a) $w(p-q)$ and EMIM performed best

   (b) the performance between $w(p-q)$ and EMIM and between F4 and F4modified is very similar.

   (c) ranking of the Porter and ZOOM algorithms are influenced by r with the Porter rankings approaching the performance of EMIM and $w(p-q)$.

7. A new ranking algorithm is proposed which will

   (a) rank terms according to r, and

   (b) use the term frequency n, from low to high, for tie-breaking.

These results are derived from this investigation specific to CIRT. Some of them, however are applicable in a wider context.

12.1 Proposals for future research

The conclusion from this investigation of interactive query expansion in an operational environment suggest many directions for future research. The main research directions are highlighted below.

12.1.1 Ranking algorithms

1. The ranking algorithm, that I proposed as the result of the six algorithm evaluation, i.e. to rank terms by r, tie-breaks by n from low to high frequency, needs to be evaluated in order to establish its level of performance vis-à-vis the other six algorithms.

   If its performance is proven to be comparable to the levels of $w(p-q)$ and EMIM then this algorithm would be easier to implement in an operational system than the other algorithms.

2. A comparative evaluation of the six ranking algorithms in laboratory conditions with test collections, would provide additional evidence about the performance of the algorithms.

   In addition, by taking into consideration the previous work by Smeaton and van Rijsbergen (1983) and the recent suggestions given by Peat and Willett (1991) for the reasons of the poor performance of earlier experiments in automatic query expansion, such comparative evaluation would provide new insights concerning the performance of automatic query expansion.

12.1.2 User Studies and Query Expansion

More research is needed into how users implement query expansion especially on the specific aspects of interactive query expansion that pertain to the user selection of terms for query expansion. As mentioned in the discussion of section 10.3.1.3 the process of term selection by the users is of particular interest for understanding the users' searching behaviour and its implication for the design of the user interface.

For example, of particular interest are the query expansion terms that were identified that represent new ideas. What was the reason that users choose these terms? Were users aware about these new concepts/ideas at the beginning of the search? If yes, why were these not expressed at the pre-search interview? Was the reason for the exclusion interview related?, e.g. communication failure, or did the user chose to exclude them at the interview stage because s/he thought of them as peripheral? If users had not thought of these concepts at an earlier stage, did they knew about them before? Did they recognise and choose these concepts as the result of a learning process during the search? On the whole, what were the reason(s) and stimuli that made them choose these new terms? Answers to these questions I believe will contribute to the understanding of the users' searching behaviour.

The finding that about 75% of the query expansion terms were identified as being hierarchically related to the query terms points to the following. For query expansion based on the relevant document set, as presented in this study, a thesaurus could be used for displaying the relationships of the selected terms to other terms. This can be done, for example, by displaying the hierarchical tree that the term belongs (like in the INSPEC or MESH tree displays) or by presenting broader, narrower or related terms under such headings on the screen for the user to browse and choose from.

However, another set of important research questions is: Are the users able to recognise the good terms during the search and especially at the onset of the search? At what stage should interactive query expansion be implemented? Could semi-automatic query expansion be useful at the query input stage, or would it be better to provide it after the initial search, or should it be an option available on request at any stage of the search? If interactive query expansion is implemented at the query input stage then it presupposes some other source for drawing the expansion terms than the relevant document set. What would that source be? Would it be a thesaurus or some other knowledge structure? Furthermore, how does automatic query expansion compare to interactive query expansion during the query input stage? Which of then is preferable for that stage of the search?

The limitations imposed on the query size by CIRT provided a controlled environment whereby up to 5 terms were used in the initial query and up to 4 for query expansion. A set of questions therefore arises: 'What would have happened if the restriction on the query size was not there? Would the searchers have used more terms for the initial search and more for the query expansion? How many more? and what effect would that have had for the search? Furthermore, users on average selected the one third of the terms in the lists, how many of these terms would they have used for query expansion? How much difference would that have made?'

12.1.3 User studies and relevance feedback systems

The findings from Pilot 2 point out to the need for user studies in searching weighted systems. Research on users and their searching behaviour and styles so far has concentrated on Boolean systems (Fenichel, 1981; Bellardo, 1985; Borgman, 1986). The findings of that research are very useful and have contributed to the understanding of the searching process in general. However, these cannot be directly applied to systems with relevance feedback, because different retrieval techniques require different approaches to searching.

Fenichel (1981) reported that even experienced searchers use only the most basic techniques of selecting and combining terms. She further found that the majority of the searchers enter only one search strategy. These results are from Boolean systems and we need to know what is the corresponding situation in weighted systems. Are Fenichel's results due to the difficulties associated in dealing with Boolean systems, and how much difference does it make when a user moves to a relevance feedback environment? How does searching behaviour change? Does the finding that there are individual differences among searchers have an effect on using weighted systems? The hypothesis is that since individual differences were strongly associated to the use of Boolean searching the effect would be reduced or even eliminated with the use of weighted systems.

Relevance feedback systems are simpler to use than Boolean systems and therefore more suitable to most users. Extensive training is required for understanding and fully exploiting the Boolean systems. Is there any training necessary for weighted systems? What difference would it make? Would it have an effect on searching? Comparative studies of the two retrieval techniques and with different types of users would provide much needed information about how these two retrieval techniques behave.

12.1.4 A module for interactive query expansion: a proposal

The research on interactive query expansion has demonstrated that such a module for interactive query expansion can be incorporated in CIRT and that it would be of benefit to the search. It is believed that if state-of-the-art technology is used including a window-based user interface the benefits would be easier to demonstrate.

I think that an interface where many search processes can be seen concurrently on different windows will simplify the task involved as well as giving a better overall mental picture of the entire process to the user.

For example, during the term selection process for query expansion the screen could have at least three windows, an 'initial query' window, a 'query terms' window, and a 'ranked list' window. The 'ranked list' window could occupy most parts of the screen so that the user could browse more easily. Terms could be presented in columns so that the number of screens is limited to 1 or 2.

With the help of a pointing device, such as a mouse, the user can point-and-click on a term to select it for query expansion. The term, once selected, would be included in the 'query expansion' window and would be also highlighted on the ranked-list so that the user would be always aware of which terms were chosen and where was their rank position. When the selection has been completed the user can then review the chosen terms in the 'query expansion terms' window. This stage gives a second chance to the user to reconsider the terms before requesting a search. However, when appropriate this stage could be bypassed.

An optional weighting can be incorporated at the review stage. This is a user weighting of the query expansion terms. For example, users can be asked to rank the query terms according to their preferences or priorities. This weighting can then be added to the relevance weighting so that user preferences are incorporated into the search. Although this suggestion is not obviously compatible with the theory of relevance weighting I think that it is worth investigating.

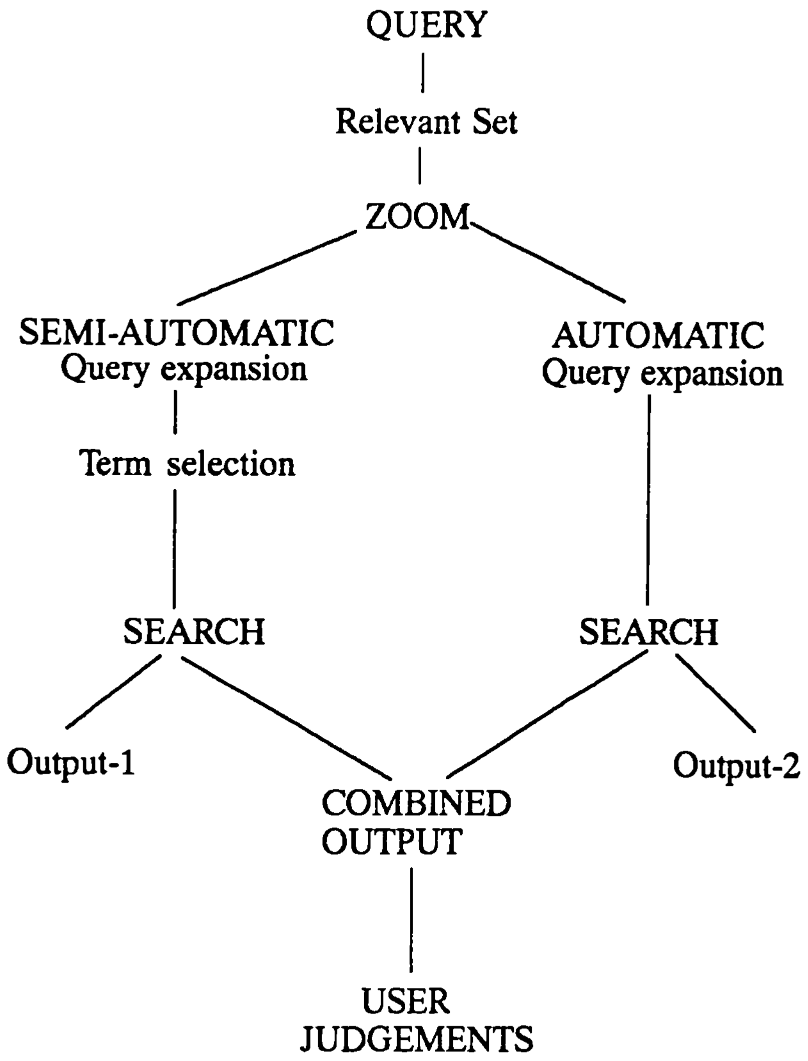
The main points to be studied would be:

- whether user supplied term weighting interferes with the relevance weighting,
- does user weighting affect retrieval performance
- if it does, is it in real terms or is it user perceived, i.e.
  how different are the retrieval sets in terms of overlap?,
  is the ranking altered?,
  are the users that utilised user term weighting 'happier' with the results?, etc.

Some of these questions can be evaluated using matched-pair experiments while others require non-matched pair experiments. In any case such a combined weighting scheme incorporates the preferences of the user in an explicit manner thus increasing the utility of the results to the user. Of course, one could argue that the relevance weighting theory and the $w(p-q)$ algorithm do consider the user preferences. This is true, and any system based on the relevance weighting theory learns from the user responses and re-adjusts itself. The explicit user weighting of the query expansion terms would allow the accommodation of individual preferences which would complement the relevance weights and thus increase the usefulness of the output to the user.

12.1.5 Automatic vs Interactive Query Expansion: a research proposal

A proposal for a comparative evaluation of automatic vs interactive query expansion is presented in Figure 12.1 and discussed below.



Future Research: Evaluation of
Automatic vs Semi-Automatic Query Expansion

Figure 12.1: Evaluation of automatic vs semi-automatic query expansion

Studies on query expansion have mainly concentrated on the issues surrounding automatic query expansion. Little is known about interactive query expansion and, to my knowledge, there are not any studies that have attempted to evaluate both. This proposal outlines a retrieval test for a comparative evaluation of the two types of query expansion. A retrieval system where both automatic and semi-automatic query expansion modules are implemented is assumed. A typical session could be as follows:

1. The user approaches the system and inputs the query.
2. The system searches the database and presents the results to the user for relevance judgements
3. User assesses relevance. The relevant document set is then analysed, for example, with ZOOM, and the candidate terms for query expansion are ranked using for example, $w(p - q)$.
4. Query expansion
   (The process of both interactive query expansion and automatic query expansion takes place simultaneously)
   (a) Interactive query expansion
   user selects terms and initiates search
   (b) Automatic query expansion
   The user is unaware of it happening. System selects terms and searches.

   Search output
   Every document in the output set for each search is separately identified, as to which type of query expansion has retrieved it. The two sets are then merged and the duplicate items eliminated. The unique output is then presented to the user.
5. user makes relevance judgements
6. evaluation

For the evaluation the user supplied relevance judgements would be matched against the two sets of documents that were retrieved by each type of query expansion. The combination of the overlap of documents between the two sets and the distribution of the relevance judgements would be the basis of the evaluation.

In this proposal the source of terms for query expansion is controlled and is the same for both types, i.e. only the relevant document set is used to draw terms from. Therefore, the performance of the two types of query expansion could be studied in detail. In addition to the evaluation a system with a configuration like the one proposed would act as the testbed for the study of other aspects of query expansion.

Appendices

Appendix A

CIRT's search tree

A.1 Search tree for request Q123

The tree presented here is the complete search tree as searched by CIRT on Data-Star for user Q123. It has been extracted from the network log and it should be noted that the user does not see this at all during the entire search. In order to facilitate the reading of this log I have embedded comments which are embraced in square brackets [...] so that the different stages of the search can be identified.

```
D-S - SEARCH MODE - ENTER QUERY
                         [docz corresponds to the size of the entire
                          database and it is element N of the F4 formula.
                          It is required for the calculation of the weights]
  1_: 3579294  docz
                         [beginning of single term searches; required
                          for weight estimation and for the search tree]
  2_:   20082  vision
  3_:   17512  robots
  4_:     580  transputers
  5_:   84978  parallel
                         [beginning searches for building the search tree]
  6_:      23  4 and 3
  7_:       4  6 and 2
  8_:       1  7 and 5
  9_:      16  6 not 2 and 5
                         [search stopped; CIRT reached target of
                          retrieving 15 documents which are now
                          displayed to user for relevance judgements]
 10_:       3  7 not 5
                         [4 documents have been judged relevant  and
                          CIRT is creating a set for future reference
                          (see also set number 116 below)]
 11_:       4  (C89042735).AN. or (B90010030).AN. or
               (C90006971).AN. or (C89042529).AN. or
               (C89032649).AN.
                         [query expansion initiated;
```

```
                     all new terms are added to the search with
                     single term searches for weight estimation;
                     then the tree is being updated and explored further]
12_:    6574  computer adj vision
13_:       4  11 and 12
14_:    8422  industrial adj robots
15_:       1  11 and 14
16_:       7  transputer adj based adj architectures
17_:       1  11 and 16
18_:   11387  parallel adj processing
19_:       1  11 and 18
20_:    6162  multiprocessing adj systems
21_:       1  11 and 20
22_:       0  9 and 16
23_:       0  9 and 20
24_:       0  9 and 12
25_:       2  9 and 14
26_:       2 25 and 18
27_:      10  9 not 14 and 18
28_:       0  6 not 2 not 5 and 16
29_:       1  6 not 2 not 5 and 20
30_:       0 29 and 12
31_:       1 29 and 14
32_:       0 31 and 18
33_:       0  6 not 2 not 5 not 20 and 12
34_:       2  6 not 2 not 5 not 20 and 14
35_:       0 34 and 18
36_:      38  4 not 3 and 2
37_:      29 36 and 5
38_:       0 37 and 16
39_:       0 37 and 20
40_:      26 37 and 12
41_:       0 40 and 14
42_:      15 40 and 18
43_:       0 36 not 5 and 16
44_:       1 36 not 5 and 20
45_:       1 44 and 12
46_:       0 45 and 14
47_:       0 45 and 18
48_:       7 36 not 5 not 20 and 12
49_:       0 48 and 14
50_:       0 48 and 18
51_:     365  4 not 3 not 2 and 5
52_:       3 51 and 16
53_:       0 52 and 20
54_:       0 52 and 12
55_:       0 52 and 14
56_:       0 52 and 18
57_:      27 51 not 16 and 20
58_:       0 57 and 12
59_:       0 57 and 14
60_:       6 57 and 18
61_:       0  4 not 3 not 2 not 5 and 16
62_:      19  4 not 3 not 2 not 5 and 20
```

```
 63_:       0 62 and 12
 64_:       0 62 and 14
 65_:       0 62 and 18
 66_:   17489  3 not 4
 67_:    2726 66 and 2
 68_:     186 67 and 5
 69_:       0 68 and 16
 70_:       3 68 and 20
 71_:       1 70 and 12
 72_:       0 71 and 14
 73_:       1 71 and 18
 74_:     134 68 not 20 and 12
 75_:      30 74 and 14
 76_:       8 75 and 18
 77_:       0 67 not 5 and 16
 78_:      11 67 not 5 and 20
 79_:       7 78 and 12
 80_:       2 79 and 14
 81_:       0 80 and 18
 82_:     442 66 not 2 and 5
 83_:       0 82 and 16
 84_:      16 82 and 20
 85_:       0 84 and 12
 86_:       4 84 and 14
 87_:       1 86 and 18
 88_:       0 66 not 2 not 5 and 16
 89_:      74 66 not 2 not 5 and 20
 90_:       0 89 and 12
 91_:      38 89 and 14
 92_:       0 91 and 18
 93_:   17314  2 not 4 not 3
 94_:     980 93 and 5
 95_:       0 94 and 16
 96_:      17 94 and 20
 97_:      14 96 and 12
 98_:       0 97 and 14
 99_:       4 97 and 18
100_:       0 93 not 5 and 16
101_:      27 93 not 5 and 20
102_:     16 101 and 12
103_:      0 102 and 14
104_:      0 102 and 18
105_:  82959   5 not 4 not 3 not 2
106_:      2 105 and 16
107_:      1 106 and 20
108_:      0 107 and 12
109_:      0 107 and 14
110_:      1 107 and 18
111_:      1  16 not 4 not 3 not 2 not 5
112_:      0 111 and 20
113_:      0 111 and 12
114_:      0 111 and 14
115_:      0 111 and 18
116_:  ..s
```

```
                    [the search has been completed and CIRT
                     requests from Data-Star the 28 documents
                     that should be retrieved]
116_:     28  11 or 73 or 76 or 42
117_:  ..o                  [logoff Data-Star]
```

Appendix B

Pilot 1

B.1 CIRT searches in the INSPEC database.

The list gives the case numbers of the 46 searches in the INSPEC database as assigned during the CIRT evaluation project. There were 21 weighted and 25 Boolean searches.

52
53
w55
56
57
58
59
w60
w61
w62
65
66
67
w68
w69
w70
w71
w73
75
w76
77
80
w140
186
w193
w199
w200
209
w247
256
268
w278
279
281
284
w285
286
w287
288
w289
290
w291
293
w294
296
297

B.2 INSPEC record fields

B.2.1 Data-Star record fields for the INSPEC database

Downloaded from Data-Star in 1989. This information is part of a record that describes INSPEC in the NEWS database of Data-Star.

```
PS LABEL  BIBL  CONTENTS                            USER FUNCTION
   -----  ----  --------                            -------------
   AB           ABSTRACT                               SEARCH
   AN      X    ABSTRACT NUMBER                        SEARCH/LIMIT
   AU      X    AUTHOR/S                               SEARCH
   AV           AVAILABILITY STATEMENT                 SEARCH
   CC           INSPEC SECTIONAL CLASSIFICATION
                   SUBJECT CODES                       SEARCH
   CD           CODEN                                  SEARCH
   DE (*)       SUBJECT INDEX TERMS (COMPOUND
                   TERMS ARE HYPHENATED)               SEARCH
   ID (*)       FREE INDEX PHRASE (COMPOUND
                   TERMS ARE NOT HYPHENATED)           SEARCH
   IN           AUTHOR AFFILIATION                     SEARCH
   LG (*)       LANGUAGE                               SEARCH/LIMIT
   PA           PATENT #, COUNTRY OF ISSUE ASSIGNEE    SEARCH
   PP           ORIG. PATENT APPL. # COUNTRY, DATE     SEARCH
   PT           PUBLICATION TYPE                       SEARCH
   RN           REPORT NUMBER; US GOVT.
                CLEARING-HOUSE #, ISSN,
                STANDARD BOOK #, CONTRACT #            SEARCH
   SO      X    SOURCE                                 SEARCH
   TC           TREATMENT CODES (1971+)                SEARCH
   TI      X    TITLE                                  SEARCH
   TR           TITLE OF COVER-TO-COVER TRANSLATION    SEARCH
   YR           YEAR OF PUBLICATION                    SEARCH/LIMIT

   (*) These paragraphs are not affected by the stopword lists.
```

B.2.2 ESA/IRS record fields for the INSPEC database

Downloaded from ESA/IRS in 1989 using the command ?FIELDS8.

Prefixed fields are those that create the additional indexes. Suffixed fields are those included in the basic index.

```
 Searchable prefixed fields are:

Search field      Prefix Example (EXPAND/SELECT)
----------------- ------ -----------------------
Author            AU=    E AU=ABBOTT
ISBN              BN=    S BN=66 70 04 1
Classific. Code   CC=    S CC=B0170E
CODEN             CO=    S CO=ABACEJ
Corporate Source  CS=    S CS=KOBE(W)CS=STEEL
Document Type     DT=    S DT=DISSERTATION
Journal Name      JN=    E JN=ACCESS (USA)
Language          LA=    S LA=JAPANESE
Chemical Indexing MF=    E MF=AG
Meeting Date      MD=    S MD=1985
Meeting Location  ML=    E ML=AMSTERDAM
Meeting Title     MT=    S MT=SPACE(F)MT=CLIMATE
Numeric Indexing  NI=    E NI=POWER
Native Number     NN=    E NN=C73002869
Publisher         PB=    E PB=AGARD
Publication Date  PD=    S PD=1983
Report Number     RN=    E RN=AECL-4273
Subfile           SF=    S SF=C
ISSN              SN=    E SN=8756-2324
Treatment Code    TR=    S TR=E?
Update            UP=    S UP=8610

 Searchable suffixed fields are:

Search field       Suffix Example
------------------ ------ -------------------
Abstract           /AB    F SPACE AGENCY/AB
Title              /TI    F DIGITAL DATA/TI
Controlled Terms   /CT    F SPACE VEHICLE?/CT
Uncontrolled Terms /UT    F SPACE STATION?/UT
Single Terms       /ST    S OLYMPUS/ST
```

B.3 INSPEC updates

B.3.1 INSPEC updates on Data-Star

INSPEC distributes 24 tapes per year but Data-Star updates the database monthly. The two columns below show the updates of the INSPEC tapes on Data-Star. The left column has been retrieved by searching INSPEC with the ROOT command as indicated in the text (e.g. `root 86$2.an.`). The update `8602.an.` is interpreted as being the second update of the INSPEC database on Data-Star in 1986 and as having 20174 records. The right column gives the actual dates of when the INSPEC tapes were actually loaded in Data-Star for the period 1986-1989. These dates were provided by Data-Star's London based technical support staff in 1989. [1]

```
D-S - SEARCH MODE - ENTER QUERY
    1_:  root 85$2.an.

                ROOT 85$2.AN.
R1     144955 DOCS          8500
R2       8820 DOCS          8508
R3      17137 DOCS          8509
R4      21493 DOCS          8510
R5      15196 DOCS          8511
R6       9667 DOCS          8512
END OF ROOT

D-S - SEARCH MODE - ENTER QUERY
    1_:  root 86$2.an.

                ROOT 86$2.AN.                Day/Month
R1      31078 DOCS          8601          7,15,23/ 1
R2      20174 DOCS          8602             11,19/ 2
R3      19624 DOCS          8603             12,18/ 3
R4      21707 DOCS          8604             15,19/ 4
R5      20648 DOCS          8605             14,22/ 5
R6      17207 DOCS          8606             10,17/ 6
R7      17984 DOCS          8607              5,16/ 7
R8      14527 DOCS          8608              5,13/ 8
R9      14247 DOCS          8609             10,20/ 9
R10     16701 DOCS          8610                12/11
R11     14665 DOCS          8611              5,13/12
END OF ROOT

D-S - SEARCH MODE - ENTER QUERY
    1_:  root 87$2.an.

                ROOT 87$2.AN.                   Day/Month
R1      69446 DOCS          8703       6,14/1 & 21/ 3 (consolidated in one)*
R2      25575 DOCS          8704                23/ 4
```

[1] With the introduction of INSPEC2 in March 1991 the entire database was reloaded. This resulted in many changes in the database. One such change is that access to the updates for the 1969-1990 period is no longer possible in the manner described here. For these data only the year of update is searchable, e.g. 870000.AN. will retrieve all the records in 1987. Updates since March 1991 are searchable in the year/month/day format, e.g. ..1/1 ud>910800.

```
R3     17675 DOCS          8705               9,14/ 5
R4     21879 DOCS          8706                 16/ 6
R5     36442 DOCS          8707            4,15,29/ 7
R6     11795 DOCS          8708                  7/ 8
R7     20844 DOCS          8709                5,9/ 9
R8     22862 DOCS          8710                 14/10
R9     19795 DOCS          8711               6,13/11
R10    28223 DOCS          8712               9,29/12
END OF ROOT

D-S - SEARCH MODE - ENTER QUERY
    1_:  root 88$2.an.

             ROOT 88$2.AN.                      Day/Month
R1     10423 DOCS          8801                 19/ 1
R2     20447 DOCS          8802               5,13/ 2
R3     18739 DOCS          8803                5,9/ 3
R4     20329 DOCS          8804                  7/ 4
R5     21673 DOCS          8805              10,19/ 5
R6     17608 DOCS          8806               2,17/ 6
R7     28562 DOCS          8807             5,8,28/ 7
R8     11754 DOCS          8808                 25/ 8
R9     21176 DOCS          8809               6,22/ 9
R10    20130 DOCS          8810               5,14/10
R11    19458 DOCS          8811               3,12/11
R12    28735 DOCS          8812             2,3,24/12
END OF ROOT

D-S - SEARCH MODE - ENTER QUERY
    1_:  root 89$2.an.

             ROOT 89$2.AN.
R1     18512 DOCS          8901
R2     22339 DOCS          8902
R3     22188 DOCS          8903
END OF ROOT
```

* By looking at the calendar of updates for 1987 on both Data-Star and ESA we can see that there was a problem with the updates at the beginning of that year. This resulted in reloading and consolidating these updates which in turn affects the reconstruction of the database *environment*.

B.3.2 INSPEC updates on ESA/IRS

The two columns below show the updates of the INSPEC database on ESA/IRS. The left column has been retrieved by using the `EXPAND` command as indicated below. The two rightmost columns indicate the update number and the actual date of the update. The dates were provided by ESA/IRS in Frascati, Italy. The ESA updates correspond to the 24 tapes that are distributed by INSPEC. However, there are a few cases where the updates have been merged or missed and these are indicated by an asterisk (*) next to the date.

```
? e up=86

REF  ITEMS  INDEX-TERM              Update:    Date:
E1     17296 UP=8511
E2     17297 UP=8512
E3 *         UP=86
E4      9667 UP=8601                86,01      12-12-86
E5     12658 UP=8602                86,02      19-12-86
E6      8443 UP=8603                86,03      22-01-86
E7      9977 UP=8604                86,04      07-02-86
E8      8646 UP=8605                86,05+              *
E9     11528 UP=8606                86,06      04-03-86 *
E10     8448 UP=8607                86,07      14-03-86
E11    11176 UP=8608                86,08      18-03-86
E12     8548 UP=8609                86,09      17-04-86
E13    13159 UP=8610                86,10      23-04-86
E14     9365 UP=8611                86,11      16-05-86
E15    11283 UP=8612                86,12      21-05-86
E16     7452 UP=8613                86,13      16-06-86
E17     9755 UP=8614                86,14      20-06-86
E18     7687 UP=8615                86,15      07-07-86
E19    10297 UP=8616                86,16      14-07-86
E20     6903 UP=8617                86,17      06-08-86
E21     7624 UP=8618                86,18      11-08-86
E22     6533 UP=8619                86,19      11-09-86
E23     7714 UP=8620                86,20      17-09-86
E24     8457 UP=8621                86,21      16-10-86
E25     8244 UP=8622                86,22      21-10-86
E26     6305 UP=8623                86,23      13-11-86
E27     8360 UP=8624                86,24      20-11-86

? e up=87

REF  ITEMS  INDEX-TERM              Update:    Date:
E1      6305 UP=8623
E2      8360 UP=8624
E3 *         UP=87
E4      7059 UP=8701                87,01      25-02-87
E5      9178 UP=8702                 [I was not given  *
E6      7027 UP=8703                  any days         *
E7      7732 UP=8704                  for the          *
E8      8670 UP=8705                  updates          *
E9     10605 UP=8706                  8702-8707]       *
E10     7853 UP=8707                                   *
```

```
E11    11322 UP=8708            87,08     25-03-87
E12    10973 UP=8709            87,09     27-04-87
E13    14602 UP=8710            87,10     29-04-87
E14     7711 UP=8711            87,11     11-05-87
E15     9964 UP=8712            87,12     20-05-87
E16     8734 UP=8713            87,13     10-06-87
E17    13145 UP=8714            87,14     14-06-87
E18    12802 UP=8715            87,15     08-07-87
E19    13176 UP=8716            87,16     13-07-87
E20    10464 UP=8717            87,17     04-08-87
E21    11795 UP=8718            87,18     11-08-87
E22    10249 UP=8719            87,19     14-09-87
E23    10595 UP=8720            87,20     23-09-87
E24    10417 UP=8721            86,21     14-01-87
E25    12445 UP=8722            87,22     26-10-87
E26     8504 UP=8723            87,23     25-11-87
E27    11291 UP=8724            87,24     26-11-87


? e up=88

REF  ITEMS   INDEX-TERM         Update:   Date:
E1      8504 UP=8723
E2     11291 UP=8724
E3 *         UP=88              88,01+
E4      8596 UP=8801            88,02+
E5     10012 UP=8802            88,03     13-01-88   *
E6      9615 UP=8803            88,03     19-01-88   *
E7     10423 UP=8804            88,04     21-01-88
E8      8366 UP=8805            88,05     07-01-88
E9     12081 UP=8806            88,06     18-02-88
E10    11000 UP=8807            88,07     14-03-88
E11     7739 UP=8808            88,08     17-03-88
E12     6744 UP=8809            88,09     06-04-88
E13    13585 UP=8810            88,10     19-04-88
E14    10259 UP=8811            88,11     18-05-88
E15    11414 UP=8812            88,12     26-05-88
E16     8125 UP=8813            88,13     07-06-88
E17     9483 UP=8814            88,14     14-06-88
E18     9696 UP=8815            88,15     13-07-88
E19    11550 UP=8816            88,16     20-07-88
E20     7316 UP=8817            88,17     13-08-88
E21    11754 UP=8818            88,18     18-08-88
E22     9843 UP=8819            88,19     12-09-88
E23    11333 UP=8820            88,20+               *
E24     8911 UP=8821            88,21+               *
E25    11219 UP=8822            88,22     04-11-88   *
E26     8658 UP=8823            88,23     09-11-88
E27    10800 UP=8824            88,24     16-11-88

? e up=89

REF  ITEMS   INDEX-TERM         Update:   Date:
E1      8658 UP=8823
```

```
E2     10800 UP=8824
E3 *         UP=89
E4      7639 UP=8901                  89,01     01-12-88
E5     11944 UP=8902                  89,02     14-12-88
E6      9152 UP=8903                  89,03     29-12-88
```

B.4 Programs for processing log files

B.4.1 Shell scripts for processing ESA log files

The scripts presented here are written in the KornShell command and programming language. These run on the Mortice Kern Systems (MKS) Toolkit which provides UNIX utilities for DOS-based personal computers. MKS and the KornShell are fully compatible with the IEEE POSIX.2 standard.

The scripts process a log file from an online search on ESA/IRS. The programs identify and extract the portions of the search that correspond to the ZOOM list, the terms that were actually searched, and those sets that give the term frequency of each term limited to the date of the original search. This information is then used by a C program that calculates the weights of each term (see Appendix B.4.2). The output of the C program is a ranked list of all candidate terms for query expansion.

```
# ******************** reconst.ksh  ***************************
#
# Test the number of arguments and if not 2 then take action

if [ "$#" -lt 2 -o "$#" -gt 2 ]
then
echo "Incorrect number of arguments"
echo "Usage: command input_file output_file"
echo "Usage: output_file is used as file extension and \
must be 3 characters long"
exit 1
fi

args1=$1
args2=$2
numchars=`echo $args2 | wc -c`

if [ $numchars -gt 4 ]
then keep=$args2
typeset -L3 keep
args2=$keep
echo "Your second argument has been abbreviated to: $args2"
fi

#    The following section prompts the user to give values for
#    the variables N (total number of docs in the collection)
#               and R (total number of known relevant docs.)
```

```
#
#    These are stored in the file "UNIVERSE" and are then
#    used by the program "FORM.C" to calculate the weigths
#    of the "zoom-ed" terms.
#

answer=y
while [ "$reply1" = "" -o "$reply1" -eq 0 -o "$answer" != "y" ]
do
typeset -i reply1
read reply1?"Please give a value for N: "
if [ $reply1 -ne 0 ]
then
echo "           Reconfirming N:" $reply1.00
echo "Is N correct? answer (y/n) \c"
read answer?
fi
done
echo $reply1.00 > universe

echo "\n"

answer=y
while [ "$reply2" = "" -o "$reply2" -eq 0 -o "$answer" != "y" ]
do
typeset -i reply2
read reply2?"Please give a value for R: "
if [ $reply2 -ne 0 ]
then
echo "           Reconfirming R:" $reply2.00
echo "Is N correct? answer (y/n) \c"
read answer?
fi
done
echo $reply2.00 >> universe

echo "Thank You."
echo "      ...Please wait..."
#
# This portion eliminates the ESA field delimiter problem where
# the set number and term frequency are not separated by a space.
# eg: 1 123 term-A
#     21234 term-B
#     3 123 term-C
# The awk program does the following:
# if the value of the first field in a line is
#      greater than 99
# then it checks the length of the first field of the previous line
#      and depending on the result it adds a space
#      after the first two digits
# or else
#   after the first digit.
#
awk '{
```

```
sub(/^  */,"");
if ( $1 > 99 )
if (length(prev) > 1)
sub(/[0-9][0-9]/, "& ", $1);
else
sub(/[0-9]/, "& ", $1);
if ( $1 < 100 && $1 > 0)
prev = $1;
print $0 }' $args1 > $args2

sed -n '/^Command/,/?/p' $args2\
  | egrep "(^[0-9]* *[0-9]* [A-Z0-9()*]*$|^Inva|^Term)" > search.$args2
#
#  Reads all commands queued for searching by SuperZOOM and
#  selects the ones that were searched;
#  deletes the DS (Display Set) output;
#  then gets only search terms.
#

sed  -n '/^? c[ 1-9]*/,/^? ds/p' $args2\
  | egrep "^[0-9]* *[0-9]* *[0-9-]*$" > combine.$args2
#
# delete the DS (Display Set) output if any and
# then get only combined sets
#

sed '/^Accepted/q' $args2\
  | egrep "^[0-9]* *[0-9]* [A-Z0-9 ]*$" > zmlist.$args2
#
# read file till ACCEPTED and quit. This gets the ZOOMLIST
#

if egrep "( AND | OR | NOT )" zmlist.$args2 > /dev/nul
then echo "WARNING: There is an BOOLEAN operator in the ZOOM-LIST"
echo "Please check and correct the following files:"
echo "combine.$args2 and search.$args2"
wc *.$args2
cat combine.$args2 > lpt1
echo "\014" > lpt1       # send a FF between the files
cat search.$args2 > lpt1
exit 1
else zoom2 $args2
fi


# ***********    reconst2.ksh   ***************
#
#         continues from reconst.ksh
# -------------------------------------------
# This is the second part of the script which
# process downloaded data from ESA sessions
#
#   If RECONST.KSH does not exit
# then this script is executed without any interruption.
#   If RECONST.KSH is interrupted
# then this script should be executed after editing the files
# "combine.$args2" and "search.$args2"
#

wc combine.$1 search.$1 zmlist.$1
#
# The files ZMLIST and SEARCH should have the same number of lines,
# whereas COMBINE should have the same number of lines or fewer lines
# BUT NEVER MORE lines than the other two files.

paste -d' ' search.$1 zmlist.$1 > se_zm.$1
sed '/^Inva/d' se_zm.$1 | sed '/^Term/d' > se_zm-ed.$1
wc se_zm.$1 se_zm-ed.$1
paste -d' ' combine.$1 se_zm-ed.$1 > se_zm_co.$1
#
# First paste the files,
# then check for the words Invalid or Term and if found delete them,
# then count lines, then paste again and send the result to awk.

# awk  '{print $8, $2, $9, $10, $11, $12, $13}' se_zm_co.$1 |\
# form |\
# sort  -b -r -f  +0n -1 | tee weigths.$1 | tee lpt1 | cat 1> con | pg
# DOS cannot cope with the above three lines thus...

awk  '{print $8, $2, $9, $10, $11, $12, $13}' se_zm_co.$1 |\
pq |\
sort  -b -r -f  +0n -1 > weigths.$1
mv error error.$1
mv universe universe.$1
/dos/print weigths.$1    # print ranked list
pg weigths.$1            # display ranked list
```

B.4.2 Program for calculating the F4modified weights

This program calculates weights for search terms according to the F4modified formula. It is used by a script (like the reconst.ksh) that processes downloaded files from ESA searches. The program takes its input from the `stdin` and prints its output in a file.

```
/*        F4MODIF.C
 *
 *  Modified F4 formula:
 *
 *              (r+c)(N-n-R+r+1-c)
 *          w = ------------------
 *              (n-r+c)(R-r+1-c)
 *
 *   where c = n/N
 *
 *   see SER: JDOC, 42(3):182-187; 1986
 */
```

```
#include <ctype.h>
#include <stdio.h>
#include <float.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

FILE *file_ptr;
FILE *file1_ptr;
char inline[81];
char f_error[] = "error";    /* formula calculations for debugging */
char f_name[] = "universe"; /* contains the values for Ntdoc & Rtdoc */

main()
{
FILE *get_line;
double termfreq;        /* term frequency */
double Ntdoc; /* total number of docs in collection */
double Rtdoc; /* total num. relevant documents in search so far */
double rel;    /* relevance judgements */
double f4modif_weight; /* term weight */
double varia_1 = 1.0;

char str[81], *str_ptr, *ptr, *inline_ptr;
str_ptr = str;


    file1_ptr = fopen(f_error, "w");
    if (file1_ptr == NULL){
        printf("\nproblem with error file\n");
        exit(1);
    }
    file_ptr = fopen(f_name, "r");
    if (file_ptr == NULL){
        printf("\nNo input file\n");
        exit(1);
    }
        if (fgets(inline, 20, file_ptr)){
            inline_ptr = inline;
            Ntdoc = atol(inline_ptr);
            fgets(inline, 20, file_ptr);
            inline_ptr = inline;
            Rtdoc = atol(inline_ptr);
                    fclose(file_ptr);
        }
        else    {fprintf(stderr,"\nWARNING: file %s not found", f_name);
        exit(1);}

/*
```

```
   printf("rel\t f4modif_weight\t freq\t  term\n");
*/
   while (gets(inline) ≠ NULL){
      if (strlen(inline) > 1){
         inline_ptr = inline;
         ptr = inline;
         inline_ptr = strchr(inline_ptr, ' ');
         if (isdigit(*ptr) == NULL){
            printf("Non-Numeric field error");
            break;
         }
         *inline_ptr = '\0';
         rel = (double) atol(ptr); /* get leading num, i.e. r */
            if (rel > Rtdoc){
               rel = Rtdoc;
            }
         inline_ptr++;
         ptr = inline_ptr;
         inline_ptr = strchr(inline_ptr, ' ');
         if (isdigit(*ptr) == NULL){
            printf("Non-Numeric field error");
            break;
         }
         *inline_ptr = '\0';
         termfreq = (double) atol(ptr); /* get second number, i.e. n */
         inline_ptr++;
         strcpy(str_ptr, inline_ptr); /* get rest of line, i.e the term */
         f4modif_weight = Ntdoc - Rtdoc - termfreq + rel + varia_1 - (termfreq / Ntdoc);
         fprintf(filel_ptr, "%lf\n", (double) f4modif_weight);
         f4modif_weight / Rtdoc - rel + varia_1 - (termfreq / Ntdoc);
         fprintf(filel_ptr, "%lf\n", (double) f4modif_weight);
         f4modif_weight / termfreq - rel + (termfreq / Ntdoc);
         fprintf(filel_ptr, "%lf\n", (doublc) f4modif_weight);
         f4modif_weight *= rel + (termfreq / Ntdoc);
         fprintf(filel_ptr, "%lf before log: %s\n", (double) f4modif_weight, str);
         f4modif_weight = log(f4modif_weight);
         printf("%5.21f, %4.01f, %7.01f, %s\n", (double) f4modif_weight, rel, termfreq, str);
      }
   }
}
```

B.5 Retrieved records for search 140

The accession numbers of the records retrieved by the searches c140, r140rel, r140de and r140ftx are given below (see chapter 7, page 123).

No	c140	r140rel	r140de	r140ftx
1	C86037973	C86037973	C86037973	C86037973
2		C84027882	B85020269	A82097289
3		C83028799	C83008046	B86049936
4		C82029960	A82109115	C83008046
5		C86037978	A82097289	A82109115
6		B85037576	A82097278	A82097278
7		B82021813	A82020500	A82020500
8		C84001129	A81100249	A81100249
9		C81030104	A80079050	A81072909
10		C81029945	A80055965	A80079050
11		C81028783	C80014716	A80055965
12		C81028782	A80045838	C80014716
13		C83008046	B86049936	A80045838
14		A82109115	B83064762	A86066452
15		A82097289	B81022644	C86037262
16		A82097278	C86037262	B85020269
17		A82020500	A84087510	A84087510
18		A81100249	B82060055	B82060055
19		A80079050	B83031739	A82004645
20		A80055965		B81022644
21		C80014716		B83064762
22		A80045838		B83064749
23		C86037262		A84035979
24		A86066452		C82037135
25		B85020269		A83044171
26		A84087510		B85012278
27		B82060055		C85026467
28				B83031739
29				B80025905
30				C81019966

B.6 Overlap of retrieved documents in searches of case 140

The overlap of the documents in all the searches of pilot case 140 was measured by examining the retrieved sets in all searches, i.e. c140, r140rel, r140de. The overlap was calculated in absolute numbers without considering relevance (see chapter 7, page 123).

The table lists the overlap for each document which is represented by accession number (AN).

Overlap / AN	Overlap / AN	Overlap / AN
4 C86037973	3 C86037262	1 B85037576
3 A80045838	2 A86066452	1 C81019966
3 A80055965	2 B81022644	1 C81028782
3 A80079050	2 B83031739	1 C81028783
3 A81100249	2 B83064762	1 C81029945
3 A82020500	2 B86049936	1 C81030104
3 A82097278	1 A81072909	1 C82029960
3 A82097289	1 A82004645	1 C82037135
3 A82109115	1 A83044171	1 C83028799
3 A84087510	1 A84035979	1 C84001129
3 B82060055	1 B80025905	1 C84027882
3 B85020269	1 B82021813	1 C85026467
3 C80014716	1 B83064749	1 C86037978
3 C83008046	1 B85012278	

B.7 Retrieved records for search 62

The accession numbers of the records retrieved by the searches c62, r62rel, r62de and r62ftx are given below (see chapter 7, page 129).

	c62	r62rel	r62de	r62ftx
	AN	AN	AN	AN
1	A86126007	A87037425	A87037425	A87037425
2	A86121911	A87031792	A87031792	A87031792
3	A86105154	A87026165	A87026165	A87026165
4	A86085290	A87016142	A86126007	A87016142
5	A86063402	A87006180	A86121911	A86126007
6	yA86063474	A86126007	A86105154	A86121911
7	A86050145	A86121911	A86085290	A86085290
8	yA86050205	A86121910	A86063402	A86063402
9	A86050149	A86105154	A86050145	A86063474
10	yA86044422	A86085290	A86050149	A86050145
11	A86030648	A86063402	A86030648	A86050149
12	A86009121	A86050145	A86009121	A86030648
13	A85120154	A86050149	A85100757	A86009121
14	yA85100757	A86030648	A85059953	A86000830
15	yA85085090	A86009121	A85051876	A85001998
16	A85059953	A85120154	A85020951	A84087797
17	A85051876	A85089802	A84112801	A84049816
18	yA85020951	A85059953	A84087797	A84049799
19	yA85006700	A85051876	A84049816	A83113403
20	A84112801	A85006698	A84049799	A83079058
21	yA84087797	A84112801	A84034095	A83002018
22	yA84049816	A84049799	A83113403	A82070895
23	A84049799	A84034095	A83079058	A82054487
24	A84034095	A83046517	A82070895	A82055587
25	yA83113403	A82070895	A82054487	A82050311
26	A79032442	A82054487	A82055587	B81039684
27	A79032441	A82055587	A81104549	A81060046
28	A79001149	A81060046	A81060046	A81024781
29	A79001147	A81019805	A81024781	A81019805
30	yA78092956	A81018026	A81018028	A81018028
31	A78077721	A80106774	A81018027	A80074824
32	yA78042585	A80052032	A81018026	A80069762
AN continue...				

AN continued

No.	c62	r62rel	r62de	r62ftx
33	yA77089088	A80038427	A80074824	A80052032
34	A77072266	A80033554	A80069762	A80019318
35	yA77065730	A80019318	A80058967	A80017806
36	A77058638	A80017806	A80052032	A80014648
37	A77054958	A80014648	A80038427	A79032442
38	C76024873	A79068724	A80019318	A79001149
39	A76071783	A79032442	A80017806	A78042585
40	C76022285	A79032441	A80014648	A77089088
41	C76019553	A79001149	A79050693	A77058638
42	yA76029876	A79001147	A79049054	C76019553
43	A76013039	A78077721	A79032442	A76013039
44	A75084886	A77072266	A79032441	A75084886
45	A75080660	A77058638	A79001149	A75080660
46	A75076294	A77054958	A79001147	A75076294
47	A75076254	A77011163	A78077721	
48	A75041264	A77011162	A78042585	
49	A74065639	C76024873	A77089088	
50	yA71044102	A76071783	A77058638	
51		C76022285	A77054958	
52		C76019553	C76024873	
53		A76013039	C76022285	
54		A75084886	C76019553	
55		A75080660	A76029876	
56		A75080620	A76013039	
57		A75076294	A75084886	
58		A75076254	A75080660	
59		A75041264	A75076294	
60		A75036793	A75076254	
61		A74065639	A74065639	
62		A73052460		

B.8 Overlap of retrieved documents in searches of case 62

The overlap of the documents in all the searches of pilot case 62 was measured by examining the retrieved sets in all searches, i.e. c62, r62rel, r62de, r62ftx. The overlap was calculated in absolute numbers without considering relevance (see chapter 7, page 129).

The table lists the overlap for each document which is represented by accession number (AN).

Overlap / AN	Overlap / AN	Overlap / AN
4 C76019553	3 A82055587	2 A75041264
4 A86126007	3 A82054487	1 B81039684
4 A86121911	3 A81060046	1 A87006180
4 A86085290	3 A80052032	1 A86121910
4 A86063402	3 A80019318	1 A86050205
4 A86050149	3 A80017806	1 A86044422
4 A86050145	3 A80014648	1 A86000830
4 A86030648	3 A79032441	1 A85089802
4 A86009121	3 A79001147	1 A85085090
4 A84049799	3 A78077721	1 A85006700
4 A79032442	3 A78042585	1 A85006698
4 A79001149	3 A77089088	1 A85001998
4 A77058638	3 A77054958	1 A83046517
4 A76013039	3 A75076254	1 A83002018
4 A75084886	3 A74065639	1 A82050311
4 A75080660	2 A87016142	1 A81104549
4 A75076294	2 A86063474	1 A81018027
3 C76024873	2 A85120154	1 A80106774
3 C76022285	2 A85100757	1 A80058967
3 A87037425	2 A85020951	1 A80033554
3 A87031792	2 A83079058	1 A79068724
3 A87026165	2 A81024781	1 A79050693
3 A86105154	2 A81019805	1 A79049054
3 A85059953	2 A81018028	1 A78092956
3 A85051876	2 A81018026	1 A77065730
3 A84112801	2 A80074824	1 A77011163
3 A84087797	2 A80069762	1 A77011162
3 A84049816	2 A80038427	1 A75080620
3 A84034095	2 A77072266	1 A75036793
3 A83113403	2 A76071783	1 A73052460
3 A82070895	2 A76029876	1 A71044102

B.9 Retrieved records for search 287

The accession numbers of the records retrieved by the searches c287, r287rel and r287de are given below (see chapter 7, page 134).

	c287	r287rel	r287de		r287rel	r287de
	AN	AN	AN		AN	AN
1	B86058480	B86034561	B86058480	33	B79023740	B78023269
2	A86055488	B86024042	B86034561	34	B79009757	B77035454
3	B86024041	B86024041	B86034585	35	B78043600	B77033240
4	B85044908	B86004808	A86055488	36	B78030350	A77038487
5	B85003628	B85037214	B86024042	37	B77037798	B77018073
6	B85003627	B85024983	B86024041	38	B77033240	B77003342
7	B84027108	B85003628	B85055398	39	B77018073	B77001399
8	B84027106	B85003627	B85044908	40	B77001399	B76029841
9	B84006886	B84060988	B85003628	41	B76048575	B76021889
10	B83012505	B84027108	B85003627	42	B76029841	B76014129
11	B82061099	B84027106	B84027108	43	B76021895	B76014128
12	B82061098	B84006886	B84027106	44	B76021889	B75043070
13	B82020016	B84002784	B84006886	45	B76014129	B75003725
14	B81003063	B84002783	B83049308	46	B76014128	B74039810
15	B80039576	B83014989	B83012505	47	B75043070	B74039806
16	B80025707	B83012505	B83012503	48	B75003727	B74036457
17	B78043600	B83009458	B83009456	49	B75003725	B74035981
18	B78030350	B82061099	B82061099	50	B74039810	B74016200
19	B77018073	B82061098	B82061098	51	B74039806	B74011683
20	B77001399	B82021926	B82059280	52	B74011683	B74003658
21	B76029841	B82021920	A82074955	53	B74003658	B73039372
22	B76021889	B82020016	B82020016	54	B73039372	B72014365
23	B76014129	B82011165	B81019645	55	B73017600	B71036286
24	B76014128	B81047767	B81003063	56	B72017437	B71034580
25	B74039806	B81008050	B81003062	57	B72014365	B71019419
26	B74016200	B81008045	B80039576	58	B71036286	A71011029
27	B74003658	B81003063	B80025707	59	B71019419	B70021888
28	B72014365	B80048182	B80013989	60	B71005462	B70020690
29	B71019419	B80039576	B79039888	61	B70005077	
30	A71011029	B80025707	B79021877	62	B96024456	
31	B70021888	B80013989	B78043600			
32	B70020690	B79030196	B78030350			
		continue	*continue*			

B.10 Overlap of retrieved documents in searches of case 287

The overlap of the documents in all the searches of pilot case 287 was measured by examining the retrieved sets in all searches, i.e. c287, r287rel, r287de. The overlap was calculated in absolute numbers without considering relevance (see chapter 7, page 134).

The table lists the overlap for each document which is represented by accession number (AN).

Overlap / AN	Overlap / AN	Overlap / AN
3 B86024041	2 B80013989	1 B82021926
3 B85003627	2 B77033240	1 B82059280
3 B85003628	2 B75003725	1 B81003062
3 B84006886	2 B75043070	1 B81008045
3 B84027106	2 B74011683	1 B81008050
3 B84027108	2 B74016200	1 B81019645
3 B83012505	2 B74039810	1 B81047767
3 B82020016	2 B73039372	1 B80048182
3 B82061098	2 B71036286	1 B79009757
3 B82061099	2 B70020690	1 B79021877
3 B81003063	2 B70021888	1 B79023740
3 B80025707	2 A86055488	1 B79030196
3 B80039576	2 A71011029	1 B79039888
3 B78030350	1 B96024456	1 B78023269
3 B78043600	1 B86004808	1 B77003342
3 B77001399	1 B86034585	1 B77035454
3 B77018073	1 B85024983	1 B77037798
3 B76014128	1 B85037214	1 B76021895
3 B76014129	1 B85055398	1 B76048575
3 B76021889	1 B84002783	1 B75003727
3 B76029841	1 B84002784	1 B74035981
3 B74003658	1 B84060988	1 B74036457
3 B74039806	1 B83009456	1 B73017600
3 B72014365	1 B83009458	1 B72017437
3 B71019419	1 B83012503	1 B71005462
2 B86024042	1 B83014989	1 B71034580
2 B86034561	1 B83049308	1 B70005077
2 B86058480	1 B82011165	1 A82074955
2 B85044908	1 B82021920	1 A77038487

Appendix C

Pilot 2

C.1 Search c68

Accession numbers of the 4 documents from the relevance judgements of search c68.

c81010510
a80072170
a78053406
a76042588

Ranked list of the 24 terms that were generated by the relevant document set.

weight	term
8.1	colour coding
6.2	colour vision
3.7	trichromatic colour scheme
3.7	tetrachromatic colour scheme
3.7	colour mixing equation
3.7	achromatic codes
3.5	medical machine
3.5	man machine system safety
3.4	man machine systems
3.3	rod involvement
3.3	contrast processes
3.1	perceived hue
3.1	hue matching
2.9	human engineering problems
2.6	physiological basis
2.4	background activity
2.1	visual displays
2.0	psychophysics
1.9	medical equipment
1.5	psychology
1.3	medicine
1.1	vision
0.8	safety
0.7	reviews

C.2 Search c291

Accession numbers of the 22 documents from the relevance judgements of search c291.

c86025007	c82011213	c83028111
c87012174	c86014153	c86049015
c86044627	c83039489	c86038429
c85046404	c83014234	c86038410
c84045686	c84022526	c86035070
c87002999	c83006280	c86034762
c84021329	c82023304	c86028705
	c81021086	

Ranked list of the 171 terms that were generated by the relevant document set.

weight	term	weight	term
4.2	functional data model	0.6	extended function
2.8	database management systems	0.6	entity relationship model
2.7	data structures	0.6	efdm
2.0	daplex	0.6	data types
1.8	adaplex	0.6	complex data types
1.5	database theory	0.6	c
1.4	high level languages	0.6	access path optimisation
1.3	relational databases	0.5	database systems
1.3	abstract data types	0.5	modularity
1.2	domain range constraints	0.5	hope
1.2	database storage functions	0.5	polymorphism
1.1	ada	0.5	types
1.1	database update operations	0.5	transaction
1.0	attribute types	0.5	functional language
0.9	entity sets	0.5	functional data model databases
0.9	semantic modeling	0.5	dbms
0.8	data language	0.5	algebraic language
0.7	prolog	0.5	coercion
0.7	integrity constraints	0.5	programming
0.6	codasyl	0.5	polymorphic types
0.6	query language	0.5	nonredundant data model
0.6	semantic data model	0.5	type constancy
0.6	run time site selection	0.5	maximal semantic content
0.6	odm	0.5	unified query by example
0.6	object orientation		information manipulation language
0.6	name equivalence	0.5	specification languages
0.6	metadata	0.5	relational database
0.6	medium sized software project	0.5	higher order functions
0.6	local views	0.5	functional databases
0.6	lambda calculus based model	0.5	functional entity relationship model
0.6	hybrid model	0.5	distributed processing
0.6	generalization hierarchies	0.5	continuous functions
0.6	functional data models	0.5	multiple inheritance
0.6	functional data model design	0.5	g whiz
0.6	ferm	0.5	formal languages
			continues...

Ranked list, continued

weight	term	weight	term
0.5	data entities	0.3	replacement
0.5	hierarchically organised cavity types	0.3	database application development environment
0.5	computer design aid	0.3	relational attributes
0.4	programming theory	0.3	information
0.4	programming languages	0.3	distributed databases
0.4	relational algebra	0.3	referential transparency
0.4	tasl	0.3	data conversion
0.4	integration	0.3	tool kit
0.4	ada compatible distributed database manager	0.3	data declaration
		0.3	data abstraction
0.4	failure conditions	0.3	parameterization
0.4	polymorphic language	0.3	subtyping
0.4	primitive elements	0.3	pdp 11 system
0.4	query languages	0.3	database management system
0.4	productivity	0.3	data structure
0.4	preci	0.3	design applications
0.4	data handling	0.2	recovery management
0.4	access language	0.2	data modelling
0.4	overloading	0.2	distributed heterogeneous databases
0.4	access structures	0.2	fdm databases
0.4	recursion	0.2	data values
0.4	equivalence mechanism	0.2	pattern matching
0.4	astrid relational algebra system	0.2	storage management
0.4	common data shapes	0.2	database programming
0.4	occurrence equivalence	0.2	relational dbms
0.4	retrieval	0.2	high level semantics
0.4	modeling facilities	0.2	storage structures
0.4	interface language	0.2	abstraction
0.4	database sublanguage daplex	0.2	data type genericity
0.4	procedural abstraction	0.2	transaction management
0.4	isomorphic equivalence	0.2	redundancies
0.4	highly parallel function network	0.2	declarative language
0.4	program patterns	0.2	ps algol
0.4	deletion	0.2	view integration
0.4	spatial applications	0.2	fquery
0.4	requirements	0.2	data model
0.4	user interfaces	0.2	visual interface
0.4	analysis	0.1	application program interface
0.4	n ary relational view	0.1	dynamic schema definition
0.4	requirement specification	0.1	integrated language
0.4	type equivalence	0.1	replicated data
0.3	computer graphics	0.1	program conversion
0.3	structural components	0.1	insertion
0.3	object oriented data model	0.1	riss relational system
0.3	fdm	0.1	canonical database
0.3	transaction conflicts	0.1	interactive terminal
0.3	data protection	0.0	programming environments
0.3	information retrieval	0.0	retracts
0.3	entity types	0.0	update operations
0.3	interactive user interface	0.0	information hiding
0.3	functional query language	0.0	cad
0.3	abstract data type		

Appendix D

The Experiment

The following are included in the appendices:

The project described below is a continuation of the City front-end projects (CIRT), which were funded by the British Library. Its aim is to establish ways that query expansion could be incorporated in front-ends. It is headed by Professor S.E. Robertson of the Department of Information Science at City University.

We are looking at the possibility of a system which would suggest words or phrases to the user, from which the user would select some for use during the search. We want to assess the value of such a method for real-life information seeking. For this experiment, we will use the INSPEC database (produced by IEEE) and the CIRT front-end system.

Your participation in the project will involve completing a questionnaire in three parts.

1. The first part will be given before the search indicating your expectations of the search. This will provide us with background information regarding your subject enquiry.
2. The second part will be given just after finishing the online search. This will give your assessment of certain aspects of the search.
3. The last part will involve evaluating a copy of the offline prints (references with abstracts) which could range up to a maximum of 50 citations. This is the most important stage of the experiment because it assesses the degree to which your initially expressed request has been satisfied.

The time required for a complete session is estimated to be approximately two hours. This may seem high; however, what is being offered here (online search and full bibliographic citations with abstracts) could cost well above £30 for almost the same amount of time. If it were done manually in the library using the printed index the time required would be a lot more than 2 hours.

All the information will be STRICTLY CONFIDENTIAL and used for no other purpose than the experiment. The data will be held on a computer only for the duration of the experiment and will be used for statistical processing. No individuals will be identified at any stage of the project. In fact, your name will appear only at the bottom of this letter and nowhere else.

If you are willing to participate would you please sign the bottom of this form.

Thank you for your cooperation.

Signed Date Query no:

qe-ltr.tex

QUESTIONNAIRE

Background Information & Context

Query no.:......
Date:...........

1. You are:
   (a) undergraduate
   (b) postgraduate (MA, MSc, Dip.)
   (c) doctoral student
   (d) faculty
   (e) researcher
   (f) other

2. How do you intend to use this information?
   (a) course related use:
   for:
   [ ] coursework/essay
   [ ] 3rd year project
   [ ] MA/MSc project
   [ ] other (please specify)
   (b) research/professional use:
   for:
   [ ] research project
   [ ] PhD dissertation
   [ ] teaching
   [ ] publication
   [ ] other (please specify)

---

3. Indicate your general assessment of the NATURE of your SUBJECT ENQUIRY.
   Precise or Accurate .... General .... Vague ....

4. How far along are you on the project or purpose given in your answer to question 2? (where 1 is beginning to think and 5 is end of project).
   1 .... 2 .... 3 .... 4 .... 5 ....

5. How much do you feel you knew about the project/reason that brought you here before you came here?
(where 1 is nothing and 5 is a lot).
1 .... 2 .... 3 .... 4 .... 5 ....

6. What type of search do you require?

[ ] BROAD — i.e. all the references on a subject including peripheral material.

[ ] NARROW — i.e. only very specific references.

7. Have you had online searches done for you before? YES .... NO ....
If YES about how many?

[ ] 1-3
[ ] 4-10
[ ] 11- 50

8. Have you done an online search on your own without an intermediary?
YES ...... NO ......
If YES about how many?

[ ] 1-3
[ ] 4-10
[ ] 11- 50

9. If you answered 'YES' in question 7 or 8 were any of the searches directly related to the present one? YES .... NO ....

10. Please describe below in your own words the topic/information you are seeking.

11. Please list any references relevant to your enquiry that you know of and that you would expect to find in this search.

QUESTIONNAIRE

Query Expansion

Query no.:.......

To answer the following questions please refer back to the ranked list of terms from which you chose new search terms.

12. Did you choose any terms from the list presented to you? Yes......... No.........

If NO, why?

[ ] couldn't find better term(s) to express the subject of the enquiry
[ ] other

If YES, did you select those terms because you thought of them as:
(Please indicate all which apply)

[ ] variant expressions or synonyms
[ ] alternative (related) terms
[ ] couldn't find better term(s) to express the subject of the enquiry
[ ] representing new ideas (i.e. not part of your original request)

13. For each one of the 5 best terms that you selected:

Does it correspond to an existing term?	1	2	3	4	5
NO					
YES					
If YES, is it broader?					
is it narrower?					
is it related?					

QUESTIONNAIRE

Post-search Assessment

Query no.:......

14. Indicate your SATISFACTION with the search on the basis of the scale below.

Excellent .... Good .... Satisfactory .... Poor .... Bad ....

15. What was your impression of the ease or difficulty of the search.

Easy .... Average .... Difficult ....

16. Generally speaking were the RESULTS of the search:

Excellent .... Good .... Satisfactory .... Poor .... Bad ....

17. How close was the online search to your original or intended enquiry?

Exact .... Fairly close .... Considerably altered ....

18. Did you GET the number of REFERENCES EXPECTED?

[ ] Less than expected
[ ] As expected
[ ] More than expected

QUESTIONNAIRE

Offline Print Evaluation

Query no.:........

19. Are you satisfied with the references? Yes .... No ....

20. After the event in retrospect are there other areas/concepts that you would like to search on?

Evaluation of Offline Prints

Relevance assessment — Instructions sheet

Query no.:.....
Date:...........

Instructions

Please answer the following questions on the copy of the offline prints. For each bibliographic reference use a combination of a letter and of a number from the ones below. A model answer is given at the bottom of this page.

1. From the information given, is the document an answer to, or about your subject enquiry?

   Y — yes
   P — partially
   N — no

2. Please indicate ONE of the following categories:

   (1) I have SEEN THE DOCUMENT itself before, and it WAS USEFUL.
   (2) I have SEEN THE DOCUMENT itself, but it was NOT USEFUL.
   (3) I have NOT SEEN the document represented by this reference, but I WOULD LIKE to see it.
   (4) I have NOT SEEN the document represented by this reference and I would NOT LIKE to see it.

Please respond in the left hand margin and circle your complete response, e.g.

```
      1
AN C89005059 8812.
AU Loushin-L-L.
TI First application of artificial intelligence for corrosion control in
   the petroleum industry.
SO Mater-Perform (USA), vol.27, no.6, p77-83, June 1988. 0 refs.
AB The  first  artificial intelligence programmed expert adviser for the
   corrosion  control  of  a refinery process unit has been installed at
   the  Sun  Refining & Marketing Co. Yabucoa Refinery. The 'MOR or LES'
   crude  unit  expert  adviser  is  a  resident  part  of the operating
   computer  system  and  is  available  on  a  call  basis to the chief
   operator  through his CRT computer console in the control room. Other
   operations  personnel  also  have  access  to  the adviser guidelines
   through  their  own  operating system terminals. The adviser provides
   real  time  analysis  of  the process conditions and is configured to
   become  an  automatic  process  control system. Details regarding the
   problem  definition,  system  logic,  and informational responses are
   presented.  An  overview  of  the  AI  programming  development  and
   conversion to the operational computer system is discussed in general
   terms.
```

QUESTIONNAIRE

Search Notes

Query no.:......
Date:...........
Start time:.....
Online from:.....
to:.....
Finish time:....

Search Terms

First iteration:

Query expansion terms:

ZOOM:

Process:
[] RUNC [1] — aof
[] login D-S
[] search — rels
[] LOGIN ESA
[] get rel. docs
[] combine
[] ZOOM
[] run Korn shell
[] qe terms
[] RUNC [1]
[] add qe terms
[] neww
[] search
[] print
[] OFFLINE Evaluation

N =

R =

AN of relevant documents

que-not-.tex

Table D.1: Term distribution of all terms chosen by the users as being potentially useful. Ranked-lists divided into 2 parts. List ranked with $w(p-q)$ algorithm.
(o2-wpq.tex)

User	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%†	Total	hb%†
101	62	10	7	70	3	30
102	38	9	6	67	3	33
103	137	11	7	64	4	36
105	77	14	6	43	8	57
108	33	8	6	75	2	25
110	61	10	6	60	4	40
111	48	31	15	48	16	52
112	93	6	6	100	0	0
113	65	24	16	67	8	33
114	62	15	10	67	5	33
115	42	9	6	67	3	33
116	77	3	3	100	0	0
117	64	25	14	56	11	44
118	55	32	20	63	12	38
119	117	34	22	65	12	35
120	44	9	7	78	2	22
121	61	17	16	94	1	6
122	60	13	8	62	5	38
123	61	12	7	58	5	42
124	80	27	13	48	14	52
125	41	21	11	52	10	48
126	39	7	6	86	1	14
127	62	11	9	82	2	18
128	113	98	52	53	46	47
129	34	8	5	63	3	38

†ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	67.52	15.44	3.09	43.00	100.00
hb%	25	32.56	15.47	3.09	0.00	57.00

Table D.2: Term distribution of all terms chosen by the users as being good terms to be included in the search. Ranked-lists divided into 3 parts. List ranked with the $w(p-q)$ algorithm.

(o3-wpq.tex)

User	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	62	10	5	50	4	40	1	10
102	38	9	5	56	1	11	3	33
103	137	11	6	55	4	36	1	9
105	77	14	3	21	5	36	6	43
108	33	8	4	50	2	25	2	25
110	61	10	5	50	3	30	2	20
111	48	31	10	32	10	32	11	35
112	93	6	5	83	1	17	0	0
113	65	24	12	50	7	29	5	21
114	62	15	8	53	5	33	2	13
115	42	9	5	56	3	33	1	11
116	77	3	0	0	3	100	0	0
117	64	25	12	48	8	32	5	20
118	55	32	16	50	9	28	7	22
119	117	34	16	47	12	35	6	18
120	44	9	5	56	3	33	1	11
121	61	17	13	76	3	18	1	6
122	60	13	7	54	3	23	3	23
123	61	12	5	42	3	25	4	33
124	80	27	10	37	7	26	10	37
125	41	21	6	29	10	48	5	24
126	39	7	5	71	1	14	1	14
127	62	11	9	82	1	9	1	9
128	113	98	35	36	33	34	30	31
129	34	8	4	50	3	38	1	13

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	49.36	18.17	3.63	0.00	83.00
am%	25	31.40	17.05	3.41	9.00	100.00
ab%	25	19.24	11.51	2.30	0.00	43.00

Table D.3: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List ranked with $w(p-q)$ algorithm.

(o5-wpq.tex)

User‡	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	5t%†	Total	5m%†	Total	5b%†
101	62	10	1	20	3	60	1	20
102	38	9	3	60	0	0	2	40
103	137	11	3	60	2	40	0	0
105	77	14	3	60	1	20	1	20
108	33	8	3	60	1	20	1	20
110	61	10	4	80	1	20	0	0
111	48	31	4	80	1	20	0	0
112	93	6	4	80	1	20	0	0
113	65	24	3	60	2	40	0	0
114	62	15	5	100	0	0	0	0
115	42	9	3	60	1	20	1	20
116	77	3	0	0	3	100	0	0
117	64	25	3	60	1	20	1	20
118	55	32	3	60	2	40	0	0
119	117	34	4	80	0	0	1	20
120	44	9	5	100	0	0	0	0
121	61	17	4	80	1	20	0	0
122	60	13	3	60	1	20	1	20
123	61	12	2	40	0	0	3	60
124	80	27	4	80	0	0	1	20
125	41	21	1	20	3	60	1	20
126	39	7	3	60	1	20	1	20
127	62	11	4	80	1	20	0	0
128	113	98	2	50	0	0	2	50
129	34	8	3	75	1	25	0	0

[†]‡Users 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. User 116 has apparently chosen only 3 terms in total.

[†]at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	62.60	23.59	4.72	0.00	100.00
5m%	25	23.40	23.57	4.71	0.00	100.00
5b%	25	14.00	16.83	3.37	0.00	60.00

Table D.4: Relevance assessments of offline prints

See section 10.2.4.2 and section 10.3.1.5.

search	y1	%	y2	%	y3	%	y4	%	p1	%	p2	%	p3	%	p4	%	n2	%	n3	%	n4	%	total
101	4	9	1	2	11	24	7	15					7	15	14	30					2	4	46
102					12	48	2	8					4	16	2	8					5	20	25
103					13	50	6	23					3	12	4	15							26
105			1	2	13	20					2	3	10	15	5	8			6	9	29	44	66
108					10	18							7	12							40	70	57
110					18	34							6	11					1	2	28	53	53
111					6	33							9	50	2	11					1	6	18
112	2	4			3	6							4	8					2	4	37	77	48
113					5	29							4	24							8	47	17
114					8	47							1	6	6	35					2	12	17
115	6	24			7	28	1	4	4	16			3	12	3	12	1	4					25
116	4	11			14	38									19	51							37
117					2	11							4	22	3	17					9	50	18
118	1	5			9	43							3	14			1	5			7	33	21
119	2	4			18	38			1	2			20	42	3	6					4	8	48
120													6	29	2	10			8	38	5	24	21
121	1	5			11	52							3	14	3	14					3	14	21
122	7	20			11	31							14	40							3	9	35
123	2	7			12	43	3	11	1	4			5	18	4	14					1	4	28
124	1	5			9	45							7	35							3	15	20
125	3	19	1	6	3	19							7	44							2	13	16
126	2	11			1	6							1	6					1	6	13	72	18
127					7	28							5	20							13	52	25
128	8	25			20	63	1	3													3	9	32
129	1	5			7	33							7	33					1	5	5	24	21

Summary statistics

	N	MEAN	MEDIAN	STDEV	SEMEAN	MIN	MAX
total	25	30.36	25.00	14.46	2.89	16.00	66.00

Table D.5: Relevance assessments and precision ratios for all searches

See section 10.2.3.1 and section 10.3.1.5.

Search	Rel1	Rel2	total	Prec1	Prec2
101	23	44	46	0.50	0.96
102	14	20	25	0.21	0.80
103	19	26	26	0.73	1.00
105	14	31	66	0.22	0.48
108	10	17	57	0.18	0.30
110	18	24	53	0.34	0.55
111	6	17	18	0.33	0.94
112	5	9	48	0.10	0.18
113	5	9	17	0.29	0.53
114	8	15	17	0.47	0.88
115	14	24	25	0.56	0.96
116	18	37	37	0.49	1.00
117	2	9	18	0.11	0.50
118	10	13	21	0.48	0.62
119	20	44	48	0.42	0.92
120	*	8	21	*	0.39
121	12	18	21	0.57	0.86
122	18	32	35	0.51	0.91
123	17	27	28	0.61	0.96
124	10	17	20	0.50	0.85
125	7	14	16	0.44	0.87
126	3	4	18	0.17	0.23
127	7	12	25	0.28	0.48
128	29	29	32	0.91	0.91
129	8	15	21	0.38	0.71

Table D.6: Correspondence of online to offline relevance jugdements

See section 10.2.4.2 and section 10.3.1.5.

search	y1	y1%	y3	y3%	p2	p2%	p3	p3%	p4	p4%	n3	n3%	n4	n4%	total
101	4	80					1	20							5
102			4	100											4
103			3	50			1	17	2	33					10
105					1	50							1	50	8
108			2	50									2	50	15
110			6	75			2	25							11
111			3	75			1	25							5
112	1	20	1	20			3	60							11
113			3	60			2	40							12
114			5	83					1	17					13
115	1	25	2	50			1	25							12
116			4	100											15
117			2	40			1	20					2	40	11
118	1	14	3	43			2	29					1	14	10
119							1	20	1	20	3	60			9
120			7	88			1	13							10
121	1	20	2	40			2	40							14
122	1	13	4	50			3	38							14
123	2	50	2	50											4
124			2	33			2	33					2	33	10
125							5	100							12
126	2	40	1	20			1	20					1	20	19
127			4	57			3	43							9
128	5	42	7	58											14
129			1	50			1	50							16

Appendix E

Evaluation of the six algorithms

Tables:

Table E.1: Term distribution of all terms chosen by the subjects as being potentially useful. Ranked-lists divided into 2 parts. List ranked with $w(p-q)$ algorithm.

Search	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%[†]	Total	hb%[†]
101	62	10	7	70	3	30
102	38	9	6	67	3	33
103	137	11	7	64	4	36
105	77	14	6	43	8	57
108	33	8	6	75	2	25
110	61	10	6	60	4	40
111	48	31	15	48	16	52
112	93	6	6	100	0	0
113	65	24	16	67	8	33
114	62	15	10	67	5	33
115	42	9	6	67	3	33
116	77	3	3	100	0	0
117	64	25	14	56	11	44
118	55	32	20	63	12	38
119	117	34	22	65	12	35
120	44	9	7	78	2	22
121	61	17	16	94	1	6
122	60	13	8	62	5	38
123	61	12	7	58	5	42
124	80	27	13	48	14	52
125	41	21	11	52	10	48
126	39	7	6	86	1	14
127	62	11	9	82	2	18
128	113	98	52	53	46	47
129	34	8	5	63	3	38

[†] ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	67.52	15.44	3.09	43.00	100.00
hb%	25	32.56	15.47	3.09	0.00	57.00

Table E.2: Term distribution of all terms chosen by the subjects as being potentially useful. Ranked-lists divided into 2 parts. List ranked with the EMIM algorithm.

Search	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%[†]	Total	hb%[†]
101	62	10	8	80	2	20
102	38	9	6	67	3	33
103	137	11	7	64	2	18
105	77	14	6	43	8	57
108	33	8	6	75	2	25
110	61	10	6	60	4	40
111	48	31	15	48	16	52
112	93	6	6	100	0	0
113	65	24	16	67	8	33
114	62	15	10	67	5	33
115	42	9	7	78	2	22
116	77	3	3	100	0	0
117	64	25	14	56	11	44
118	55	32	20	63	12	38
119	117	34	23	68	11	32
120	44	9	7	78	2	22
121	61	17	16	94	1	6
122	60	13	8	62	5	38
123	61	12	7	58	5	42
124	80	27	13	48	14	52
125	41	21	11	52	10	48
126	39	7	6	86	1	14
127	62	11	9	82	2	18
128	113	98	52	53	46	47
129	34	8	5	63	3	38

[†] ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	68.48	15.75	3.15	43.00	100.00
hb%	25	30.88	15.98	3.20	0.00	57.00

Table E.3: Term distribution of all terms chosen by the subjects as being potentially useful. Ranked-lists divided into 2 parts. List ranked with the F4 formula.

Search	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%†	Total	hb%†
101	62	10	8	80	2	20
102	38	9	5	56	4	44
103	137	11	8	73	3	27
105	77	14	5	36	9	64
108	33	8	5	63	3	38
110	61	10	7	70	3	30
111	48	31	14	45	17	55
112	93	6	6	100	0	0
113	65	24	15	63	9	38
114	62	15	10	67	5	33
115	42	9	6	67	3	33
116	77	3	3	100	0	0
117	64	25	15	60	10	40
118	55	32	16	50	16	50
119	117	34	25	74	9	26
120	44	9	8	89	1	11
121	61	17	15	88	2	12
122	60	13	7	54	6	46
123	61	12	5	42	7	58
124	80	27	11	41	16	59
125	41	21	13	62	8	38
126	39	7	6	86	1	14
127	62	11	4	36	7	64
128	113	98	51	52	47	48
129	34	8	5	63	3	38

†ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	64.68	18.51	3.70	36.00	100.00
hb%	25	35.44	18.53	3.71	0.00	64.00

Table E.4: Term distribution of all terms chosen by the subjects as being potentially useful. Ranked-lists divided into 2 parts. List ranked with the F4-modified formula.

Search	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%†	Total	hb%†
101	62	10	8	80	2	20
102	38	9	5	56	4	44
103	137	11	9	82	2	18
105	77	14	5	36	9	64
108	33	8	5	63	3	38
110	61	10	7	70	3	30
111	48	31	14	45	17	55
112	93	6	6	100	0	0
113	65	24	15	63	9	38
114	62	15	11	73	4	27
115	42	9	6	67	3	33
116	77	3	3	100	0	0
117	64	25	15	60	10	40
118	55	32	16	50	16	50
119	117	34	25	74	9	26
120	44	9	8	89	1	11
121	61	17	15	88	2	12
122	60	13	7	54	6	46
123	61	12	5	42	7	58
124	80	27	11	41	16	59
125	41	21	13	62	8	38
126	39	7	6	86	1	14
127	62	11	4	36	7	64
128	113	98	51	52	47	48
129	34	8	5	63	3	38

†ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	65.28	18.83	3.77	36.00	100.00
hb%	25	34.84	18.85	3.77	0.00	64.00

Table E.5: Term distribution of all terms chosen by the subjects as being potentially useful. Ranked-lists divided into 2 parts. List ranked using Porter's algorithm.

Search	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%†	Total	hb%†
101	62	10	8	80	2	20
102	38	9	6	67	3	33
103	137	11	7	64	4	36
105	77	14	6	43	8	57
108	33	8	6	75	2	25
110	61	10	6	60	4	40
111	48	31	15	48	16	52
112	93	6	6	100	0	0
113	65	24	16	67	8	33
114	62	15	10	67	4	27
115	42	9	6	67	3	33
116	77	3	3	100	0	0
117	64	25	14	56	11	44
118	55	32	20	63	12	38
119	117	34	23	68	11	32
120	44	9	7	78	2	22
121	61	17	16	94	1	6
122	60	13	8	62	5	38
123	61	12	7	58	5	42
124	80	27	13	48	14	52
125	41	21	11	52	10	48
126	39	7	6	86	1	14
127	62	11	9	82	2	18
128	113	98	53	54	45	46
129	34	8	5	63	3	38

†ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	68	16	3	43	100
hb%	25	32	16	3	0	57

Table E.6: Term distribution of all terms chosen by the subjects as being potentially useful. Ranked-lists divided into 2 parts. List as ranked by ZOOM.

Search	Terms	Terms	Top 1/2		Bottom 1/2	
	in list	chosen	Total	ht%†	Total	hb%†
101	66	10	6	60	4	40
102	39	9	5	56	4	44
103	140	11	4	36	6	55
105	78	14	5	36	8	57
108	34	8	6	75	2	25
110	62	10	4	40	6	60
111	49	31	16	52	15	48
112	97	6	3	50	3	50
113	65	24	15	63	9	38
114	65	15	5	33	10	67
115	47	9	6	67	3	33
116	80	3	3	100	0	0
117	69	25	16	64	9	36
118	62	32	19	59	13	41
119	139	34	15	44	19	56
120	50	9	5	56	4	44
121	72	17	9	53	8	47
122	71	13	10	77	3	23
123	76	12	8	67	4	33
124	85	27	17	63	10	37
125	45	21	11	52	10	48
126	54	7	3	43	4	57
127	75	11	11	100	0	0
128	135	98	51	52	47	48
129	46	8	2	25	6	75

†ht% = percentage of terms in the top half of the list
hb% = percentage of terms in the bottom half of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
ht%	25	56.92	18.30	3.66	25.00	100.00
hb%	25	42.48	17.65	3.53	0.00	75.00

Table E.7: Term distribution of all terms chosen by the subjects as being good terms to be included in the search. Ranked-lists divided into 3 parts. List ranked with the $w(p-q)$ algorithm.

Subject	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	62	10	5	50	4	40	1	10
102	38	9	5	56	1	11	3	33
103	137	11	6	55	4	36	1	9
105	77	14	3	21	5	36	6	43
108	33	8	4	50	2	25	2	25
110	61	10	5	50	3	30	2	20
111	48	31	10	32	10	32	11	35
112	93	6	5	83	1	17	0	0
113	65	24	12	50	7	29	5	21
114	62	15	8	53	5	33	2	13
115	42	9	5	56	3	33	1	11
116	77	3	0	0	3	100	0	0
117	64	25	12	48	8	32	5	20
118	55	32	16	50	9	28	7	22
119	117	34	16	47	12	35	6	18
120	44	9	5	56	3	33	1	11
121	61	17	13	76	3	18	1	6
122	60	13	7	54	3	23	3	23
123	61	12	5	42	3	25	4	33
124	80	27	10	37	7	26	10	37
125	41	21	6	29	10	48	5	24
126	39	7	5	71	1	14	1	14
127	62	11	9	82	1	9	1	9
128	113	98	35	36	33	34	30	31
129	34	8	4	50	3	38	1	13

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	49.36	18.17	3.63	0.00	83.00
am%	25	31.40	17.05	3.41	9.00	100.00
ab%	25	19.24	11.51	2.30	0.00	43.00

Table E.8: Term distribution of all terms chosen by the subjects as being good terms to be included in the search. Ranked-lists divided into 3 parts. List ranked using the EMIM algorithm.

Subject	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	62	10	5	50	4	40	1	10
102	38	9	5	56	1	11	3	33
103	137	11	6	55	4	36	1	9
105	77	14	3	21	5	36	6	43
108	33	8	4	50	2	25	2	25
110	61	10	5	50	3	30	2	20
111	48	31	11	35	9	29	11	35
112	93	6	5	83	1	17	0	0
113	65	24	12	50	7	29	5	21
114	62	15	8	53	5	33	2	13
115	42	9	5	56	3	33	1	11
116	77	3	0	0	3	100	0	0
117	64	25	11	44	9	36	5	20
118	55	32	15	47	10	31	7	22
119	117	34	16	47	11	32	7	21
120	44	9	5	56	3	33	1	11
121	61	17	13	76	3	18	1	6
122	60	13	7	54	3	23	3	23
123	61	12	4	33	4	33	4	33
124	80	27	10	37	7	26	10	37
125	41	21	6	29	10	48	5	24
126	39	7	6	86	0	0	1	14
127	62	11	9	82	1	9	1	9
128	113	98	35	36	33	34	30	31
129	34	8	4	50	3	38	1	13

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	49.44	19.31	3.86	0.00	86.00
am%	25	31.20	17.82	3.56	0.00	100.00
ab%	25	19.36	11.51	2.30	0.00	43.00

Table E.9: Term distribution of all terms chosen by the subjects as being good terms to be included in the search. Ranked-lists divided into 3 parts. List ranked with the F4 formula.

Subject	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	62	10	7	70	2	20	1	10
102	38	9	5	56	1	11	3	33
103	137	11	4	36	7	64	0	0
105	77	14	2	14	6	43	6	43
108	33	8	4	50	2	25	2	25
110	61	10	5	50	4	40	1	10
111	48	31	10	32	10	32	11	35
112	93	6	5	83	1	17	0	0
113	65	24	13	54	4	17	7	29
114	62	15	7	47	7	47	1	7
115	42	9	5	56	4	44	0	0
116	77	3	0	0	3	100	0	0
117	64	25	8	32	11	44	6	24
118	55	32	10	31	13	41	9	28
119	117	34	15	44	14	41	5	15
120	44	9	7	78	1	11	1	11
121	61	17	10	59	5	29	2	12
122	60	13	6	46	4	31	3	23
123	61	12	5	42	2	17	5	42
124	80	27	8	30	9	33	10	37
125	41	21	8	38	7	33	6	29
126	39	7	5	71	1	14	1	14
127	62	11	3	27	7	64	1	9
128	113	98	36	37	30	31	32	33
129	34	8	4	50	3	38	1	13

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	45.32	19.04	3.81	0.00	83.00
am%	25	35.48	19.74	3.95	11.00	100.00
ab%	25	19.28	13.65	2.73	0.00	43.00

Table E.10: Term distribution of all terms chosen by the subjects as being good terms to be included in the search. Ranked-lists divided into 3 parts. List ranked with the F4-modified formula.

Subject	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	62	10	7	70	2	20	1	10
102	38	9	5	56	1	11	3	33
103	137	11	5	45	6	55	0	0
105	77	14	2	14	6	43	6	43
108	33	8	4	50	2	25	2	25
110	61	10	5	50	4	40	1	10
111	48	31	10	32	10	32	11	35
112	93	6	5	83	1	17	0	0
113	65	24	13	54	4	17	7	29
114	62	15	7	47	7	47	1	7
115	42	9	5	56	4	44	0	0
116	77	3	0	0	3	100	0	0
117	64	25	8	32	11	44	6	24
118	55	32	10	31	14	44	8	25
119	117	34	15	44	13	38	6	18
120	44	9	7	78	1	11	1	11
121	61	17	10	59	5	29	2	12
122	60	13	6	46	4	31	3	23
123	61	12	5	42	2	17	5	42
124	80	27	8	30	9	33	10	37
125	41	21	8	38	7	33	6	29
126	39	7	6	86	0	0	1	14
127	62	11	3	27	7	64	1	9
128	113	98	36	37	30	31	32	33
129	34	8	4	50	3	38	1	13

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	46.28	19.99	4.00	0.00	86.00
am%	25	34.56	20.12	4.02	0.00	100.00
ab%	25	19.28	13.56	2.71	0.00	43.00

Table E.11: Term distribution of all terms chosen by the subjects as being good terms to be included in the search. Ranked-lists divided into 3 parts. List ranked with Porter's algorithm.

Subject	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	62	10	5	50	4	40	1	10
102	38	9	5	56	1	11	3	33
103	137	11	5	45	5	45	1	9
105	77	14	3	21	5	36	6	43
108	33	8	4	50	2	25	2	25
110	61	10	4	40	4	40	2	20
111	48	31	11	35	9	29	11	35
112	93	6	5	83	1	17	0	0
113	65	24	12	50	7	29	5	21
114	62	15	7	47	3	20	1	7
115	42	9	5	56	3	33	1	11
116	77	3	0	0	3	100	0	0
117	64	25	11	44	9	36	5	20
118	55	32	16	50	9	28	7	22
119	117	34	16	47	11	32	7	21
120	44	9	5	56	3	33	1	11
121	61	17	12	71	4	24	1	6
122	60	13	6	46	4	31	3	23
123	61	12	5	42	3	25	4	33
124	80	27	10	37	7	26	10	37
125	41	21	6	29	10	48	5	24
126	39	7	4	57	2	29	1	14
127	62	11	9	82	0	0	2	18
128	113	98	36	37	32	33	30	31
129	34	8	4	50	3	38	1	13

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	47	17	3	0.00	83
am%	25	32	17	3	0.00	100
ab%	25	19	12	2	0.00	43

Table E.12: Term distribution of all terms chosen by the subjects as being good terms to be included in the search. Ranked-lists divided into 3 parts. List as ranked by ZOOM.

Subject	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	at%†	Total	am%†	Total	ab%†
101	66	10	4	40	3	30	3	30
102	39	9	5	56	3	33	1	11
103	140	11	2	18	6	55	2	18
105	78	14	4	29	5	36	5	36
108	34	8	4	50	3	38	1	13
110	62	10	3	30	4	40	3	30
111	49	31	11	35	9	29	11	35
112	97	6	1	17	4	67	1	17
113	65	24	10	42	6	25	8	33
114	65	15	3	20	3	20	9	60
115	47	9	5	56	3	33	1	11
116	80	3	2	67	1	33	0	0
117	69	25	7	28	13	52	5	20
118	62	32	17	53	4	13	11	34
119	139	34	9	26	11	32	14	41
120	50	9	4	44	1	11	4	44
121	72	17	6	35	7	41	4	24
122	71	13	7	54	3	23	3	23
123	76	12	4	33	5	42	3	25
124	85	27	14	52	7	26	6	22
125	45	21	7	33	7	33	7	33
126	54	7	2	29	3	43	2	29
127	75	11	9	82	2	18	0	0
128	135	98	36	37	28	29	34	35
129	46	8	1	13	2	25	5	63

†at% = percentage of terms in the top third of the list
am% = percentage of terms in the middle third of the list
ab% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
at%	25	39.16	16.58	3.32	13.00	82.00
am%	25	33.08	12.79	2.56	11.00	67.00
ab%	25	27.48	15.37	3.07	0.00	63.00

Table E.13: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List ranked with $w(p-q)$ algorithm.

Subject‡	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	5t%†	Total	5m%†	Total	5b%†
101	62	10	1	20	3	60	1	20
102	38	9	3	60	0	0	2	40
103	137	11	3	60	2	40	0	0
105	77	14	3	60	1	20	1	20
108	33	8	3	60	1	20	1	20
110	61	10	4	80	1	20	0	0
111	48	31	4	80	1	20	0	0
112	93	6	4	80	1	20	0	0
113	65	24	3	60	2	40	0	0
114	62	15	5	100	0	0	0	0
115	42	9	3	60	1	20	1	20
116	77	3	0	0	3	100	0	0
117	64	25	3	60	1	20	1	20
118	55	32	3	60	2	40	0	0
119	117	34	4	80	0	0	1	20
120	44	9	5	100	0	0	0	0
121	61	17	4	80	1	20	0	0
122	60	13	3	60	1	20	1	20
123	61	12	2	40	0	0	3	60
124	80	27	4	80	0	0	1	20
125	41	21	1	20	3	60	1	20
126	39	7	3	60	1	20	1	20
127	62	11	4	80	1	20	0	0
128	113	98	2	50	0	0	2	50
129	34	8	3	75	1	25	0	0

†‡Subjects 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. Subject 116 has apparently chosen only 3 terms in total.

†at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	62.60	23.59	4.72	0.00	100.00
5m%	25	23.40	23.57	4.71	0.00	100.00
5b%	25	14.00	16.83	3.37	0.00	60.00

Table E.14: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List ranked with the EMIM algorithm.

Subject[1]‡	Terms in list	Terms chosen	Top 1/3		Middle 1/3		Bottom 1/3	
			Total	5t%†	Total	5m%†	Total	5b%†
101	62	10	1	20	3	60	1	20
102	38	9	3	60	0	0	2	40
103	137	11	3	60	2	40	0	0
105	77	14	3	60	1	20	1	20
108	33	8	3	60	1	20	1	20
110	61	10	4	80	1	20	0	0
111	48	31	5	100	0	0	0	0
112	93	6	4	80	1	20	0	0
113	65	24	3	60	2	40	0	0
114	62	15	5	100	0	0	0	0
115	42	9	3	60	1	20	1	20
116	77	3	0	0	3	100	0	0
117	64	25	3	60	1	20	1	20
118	55	32	3	60	2	40	0	0
119	117	34	4	80	0	0	1	20
120	44	9	5	100	0	0	0	0
121	61	17	4	80	1	20	0	0
122	60	13	3	60	1	20	1	20
123	61	12	2	40	0	0	3	60
124	80	27	4	80	0	0	1	20
125	41	21	1	20	3	60	1	20
126	39	7	4	80	0	0	1	20
127	62	11	4	80	1	20	0	0
128	113	98	2	50	0	0	2	50
129	34	8	3	75	1	25	0	0

†‡Subjects 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. Subject 116 has apparently chosen only 3 terms in total.

†at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	64.20	24.74	4.95	0.00	100.00
5m%	25	21.80	24.45	4.89	0.00	100.00
5b%	25	14.00	16.83	3.37	0.00	60.00

Table E.15: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List ranked with the F4 formula.

Subject‡	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	5t%†	Total	5m%†	Total	5b%†
101	62	10	4	80	0	0	1	20
102	38	9	2	40	1	20	2	40
103	137	11	2	40	3	60	0	0
105	77	14	2	40	2	40	1	20
108	33	8	2	40	1	20	2	40
110	61	10	3	60	2	40	0	0
111	48	31	3	60	1	20	1	20
112	93	6	4	80	1	20	0	0
113	65	24	5	100	0	0	0	0
114	62	15	2	40	3	60	0	0
115	42	9	3	60	2	40	0	0
116	77	3	0	0	3	100	0	0
117	64	25	3	60	2	40	0	0
118	55	32	3	60	1	20	1	20
119	117	34	4	80	0	0	1	20
120	44	9	5	100	0	0	0	0
121	61	17	3	60	2	40	0	0
122	60	13	3	60	1	20	1	20
123	61	12	1	20	1	20	3	60
124	80	27	2	40	2	40	1	20
125	41	21	1	20	3	60	1	20
126	39	7	3	60	1	20	1	20
127	62	11	2	40	2	40	1	20
128	113	98	1	25	1	25	2	50
129	34	8	3	75	1	25	0	0

†‡Subjects 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. Subject 116 has apparently chosen only 3 terms in total.

†at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	53.60	24.52	4.90	0.00	100.00
5m%	25	30.80	23.03	4.61	0.00	100.00
5b%	25	15.60	17.34	3.47	0.00	60.00

Table E.16: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List ranked with the F4-modified formula.

Subject‡	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	5t%†	Total	5m%†	Total	5b%†
101	62	10	4	80	0	0	1	20
102	38	9	2	40	1	20	2	40
103	137	11	2	40	3	60	0	0
105	77	14	2	40	2	40	1	20
108	33	8	2	40	1	20	2	40
110	61	10	3	60	2	40	0	0
111	48	31	3	60	1	20	1	20
112	93	6	4	80	1	20	0	0
113	65	24	5	100	0	0	0	0
114	62	15	2	40	3	60	0	0
115	42	9	3	60	2	40	0	0
116	77	3	0	0	3	100	0	0
117	64	25	3	60	2	40	0	0
118	55	32	3	60	1	20	1	20
119	117	34	4	80	0	0	1	20
120	44	9	5	100	0	0	0	0
121	61	17	3	60	2	40	0	0
122	60	13	3	60	1	20	1	20
123	61	12	1	20	1	20	3	60
124	80	27	2	40	2	40	1	20
125	41	21	1	20	3	60	1	20
126	39	7	4	80	0	0	1	20
127	62	11	2	40	2	40	1	20
128	113	98	1	25	1	25	2	50
129	34	8	3	75	1	25	0	0

†‡Subjects 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. Subject 116 has apparently chosen only 3 terms in total.

†at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	54.40	25.05	5.01	0.00	100.00
5m%	25	30.00	23.76	4.75	0.00	100.00
5b%	25	15.60	17.34	3.47	0.00	60.00

Table E.17: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List ranked with Porter's algorithm.

Subject‡	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	5t%†	Total	5m%†	Total	5b%†
101	62	10	1	20	3	60	1	20
102	38	9	3	60	0	0	2	40
103	137	11	2	40	3	60	0	0
105	77	14	3	60	1	20	1	20
108	33	8	3	60	1	20	1	20
110	61	10	4	80	1	20	0	0
111	48	31	4	80	1	20	0	0
112	93	6	4	80	1	20	0	0
113	65	24	3	60	2	40	0	0
114	62	15	5	100	0	0	0	0
115	42	9	3	60	1	20	1	20
116	77	3	0	0	3	100	0	0
117	64	25	2	40	2	40	1	20
118	55	32	3	60	2	40	0	0
119	117	34	4	80	0	0	1	20
120	44	9	5	100	0	0	0	0
121	61	17	4	80	1	20	0	0
122	60	13	2	40	2	40	1	20
123	61	12	2	40	0	0	3	60
124	80	27	4	80	0	0	1	20
125	41	21	1	20	3	60	1	20
126	39	7	3	60	1	20	1	20
127	62	11	4	80	0	0	1	20
128	113	98	2	50	0	0	2	50
129	34	8	3	75	1	25	0	0

†‡Subjects 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. Subject 116 has apparently chosen only 3 terms in total.

†at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	60	25	5	0	100
5m%	25	25	25	5	0	100
5b%	25	15	17	3	0	60

Table E.18: Term distribution of the 5 best terms. Ranked-lists divided into 3 parts. List as ranked by ZOOM.

Subject‡	Terms	Terms	Top 1/3		Middle 1/3		Bottom 1/3	
	in list	chosen	Total	5t%†	Total	5m%†	Total	5b%†
101	66	10	2	40	1	20	2	40
102	39	9	3	60	2	40	0	0
103	140	11	0	0	4	80	1	20
105	78	14	2	40	1	20	2	40
108	34	8	2	40	2	40	1	20
110	62	10	2	40	1	20	2	40
111	49	31	3	60	0	0	2	40
112	97	6	1	20	3	60	1	20
113	65	24	1	20	1	20	3	60
114	65	15	3	60	0	0	2	40
115	47	9	3	60	1	20	1	20
116	80	3	2	67	1	33	0	0
117	69	25	1	20	3	60	1	20
118	62	32	2	40	0	0	3	60
119	139	34	2	40	0	0	3	60
120	50	9	3	60	1	20	1	20
121	72	17	2	40	2	40	1	20
122	71	13	4	80	1	20	0	0
123	76	12	1	20	3	60	1	20
124	85	27	4	80	1	20	0	0
125	45	21	2	40	1	20	2	40
126	54	7	2	40	2	40	1	20
127	75	11	3	60	2	40	0	0
128	135	98	1	25	2	50	1	25
129	46	8	1	25	0	0	3	75

†‡Subjects 116, 128 and 129 selected as best terms only 3, 4 and 4 terms respectively. Subject 116 has apparently chosen only 3 terms in total.

†at% = percentage of terms in the top third of the list
5m% = percentage of terms in the middle third of the list
5b% = percentage of terms in the bottom third of the list
N.B. All percentages have been rounded to the nearest integer

	N	MEAN	STDEV	SEMEAN	MIN	MAX
5t%	25	43.08	20.15	4.03	0.00	80.00
5m%	25	28.92	21.87	4.37	0.00	80.00
5b%	25	28.00	21.07	4.21	0.00	75.00

Table E.19: Search 101: five top-ranked terms for each algorithm.

file: weigths.101

w(p-q)	r	weight	n	term
7.8	5	7.8	8951	high temperature superconductors
3.0	3	5.0	19115	ceramics
2.6	2	6.4	2375	brittleness
2.6	1	13.0	2	metal superconductor composites
2.6	1	13.0	2	electrical current transport

file: emim_1.101

weight	r	n	term
41.60	5	8951	high temperature superconductors
23.54	3	19115	ceramics
21.91	1	2	metal superconductor composites
21.91	1	2	electrical current transport
21.58	2	2375	brittleness

file: f4_1.101

weight	r	n	term
13.03	1	2	metal superconductor composites
13.03	1	2	electrical current transport
12.18	1	4	pulsed tokamaks
10.91	1	13	low resistivity contacts
9.90	1	35	surface resistivities

file: f4modified_1.101

weight	r	n	term
12.93	1	2	metal superconductor composites
12.93	1	2	electrical current transport
11.83	1	4	pulsed tokamaks
10.44	1	13	low resistivity contacts
9.40	1	35	surface resistivities

file: porter_1.101

weight	r	n	term
0.9956405868	5	8951	high temperature superconductors
0.5906904052	3	19115	ceramics
0.3988433017	2	2375	brittleness
0.3985846883	2	2906	electrical contacts
0.3981020408	2	3897	powder metallurgy

file: zoomlist.101

rank	r	term
1	6	high temperature superconductors
2	3	ceramics
3	2	barium compounds
4	2	brittleness
5	2	ceramic

Table E.20: Search 102: five top-ranked terms for each algorithm.

file: weigths.102

w(p-q)	r	weight	n	term
11.5	4	11.5	320	galileo
7.8	4	7.8	13075	database management systems
6.1	2	12.3	18	type hierarchies
5.5	2	11.0	57	abstraction mechanisms
5.3	3	7.1	6646	high level languages

file: emim_1.102

weight	r	n	term
51.46	4	320	galileo
35.13	2	18	type hierarchies
34.77	4	13075	database management systems
32.45	2	57	abstraction mechanisms
29.45	3	6646	high level languages

file: f4_1.102

weight	r	n	term
14.91	1	1	semantic data model features
14.91	1	1	interactive conceptual language
13.81	1	2	object oriented database language
13.81	1	2	modularization mechanism
13.81	1	2	dialogo

file: f4modified_1.102

weight	r	n	term
28.74	1	1	semantic data model features
28.74	1	1	interactive conceptual language
13.68	1	2	object oriented database language
13.68	1	2	modularization mechanism
13.68	1	2	dialogo

file: porter_1.102

weight	r	n	term
0.9999082000	4	320	galileo
0.9962491096	4	13075	database management systems
0.7480934289	3	6646	high level languages
0.4999948363	2	18	type hierarchies
0.4999836481	2	57	abstraction mechanisms

file: zoomlist.102

rank	r	term
1	4	database management systems
2	4	galileo
3	3	high level languages
4	2	abstraction mechanisms
5	2	data structures

Table E.21: Search 103: five top-ranked terms for each algorithm.

file: weigths.103

w(p-q)	r	weight	n	term
6.2	5	7.5	7343	waste disposal
3.5	3	6.9	3385	health hazards
3.3	2	9.8	114	polychlorinated biphenyls
2.5	3	5.1	20935	pollution
2.4	1	14.5	1	water agencies

file: emim_1.103

weight	r	n	term
45.84	5	7343	waste disposal
31.26	3	3385	health hazards
30.18	2	114	polychlorinated biphenyls
24.80	3	20935	pollution
22.77	1	1	water agencies

file: f4_1.103

weight	r	n	term
14.46	1	1	water agencies
14.46	1	1	waste acids
14.46	1	1	uncontrolled hazardous waste site
14.46	1	1	tsca wastes
14.46	1	1	supercritical water oxidation

file: f4modified_1.103

weight	r	n	term
28.34	1	1	water agencies
28.34	1	1	waste acids
28.34	1	1	uncontrolled hazardous waste site
28.34	1	1	tsca wastes
28.34	1	1	supercritical water oxidation

file: porter_1.103

weight	r	n	term
0.8312268103	5	7343	waste disposal
0.4990289282	3	3385	health hazards
0.4939942723	3	20935	pollution
0.3333006296	2	114	polychlorinated biphenyls
0.3323355821	2	3478	groundwater

file: zoomlist.103

rank	r	term
1	5	waste disposal
2	3	health hazards
3	3	pollution
4	2	combustion
5	2	disposal

Table E.22: Search 105: five top-ranked terms for each algorithm.

file: weigths.105

w(p-q)	r	weight	n	term
7.4	3	7.4	8370	operations research
4.9	1	14.7	1	tps development management
4.9	1	14.7	1	priority based interactive algorithm
4.9	1	14.7	1	manpower assignment
4.9	1	14.7	1	ion implanter diagnosis system

file: emim_1.105

weight	r	n	term
26.80	3	8370	operations research
22.66	1	1	tps development management
22.66	1	1	priority based interactive algorithm
22.66	1	1	manpower assignment
22.66	1	1	ion implanter diagnosis system

file: f4_1.105

weight	r	n	term
14.72	1	1	tps development management
14.72	1	1	priority based interactive algorithm
14.72	1	1	manpower assignment
14.72	1	1	ion implanter diagnosis system
14.72	1	1	hits environment

file: f4modified_1.105

weight	r	n	term
27.97	1	1	tps development management
27.97	1	1	priority based interactive algorithm
27.97	1	1	manpower assignment
27.97	1	1	ion implanter diagnosis system
27.97	1	1	hits environment

file: porter_1.105

weight	r	n	term
0.9959235517	3	8370	operations research
0.3333328463	1	1	tps development management
0.3333328463	1	1	priority based interactive algorithm
0.3333328463	1	1	manpower assignment
0.3333328463	1	1	ion implanter diagnosis system

file: zoomlist.105

rank	r	term
1	3	operations research
2	1	accounting
3	1	ai research
4	1	arbitration models
5	1	asynchronous process control

Table E.23: Search 108: five top-ranked terms for each algorithm.

file: weigths.108

w(p-q)	r	weight	n	term
3.7	1	14.9	1	trend detection method
3.7	1	14.9	1	emp filtering
3.7	1	14.9	1	electronic implantable devices
3.5	1	13.8	2	trigg s tracking signal
3.3	2	6.6	4808	time series

file: emim_1.108

weight	r	n	term
23.20	1	1	trend detection method
23.20	1	1	emp filtering
23.20	1	1	electronic implantable devices
22.94	1	2	trigg s tracking signal
21.48	1	9	clinical states

file: f4_1.108

weight	r	n	term
14.92	1	1	trend detection method
14.92	1	1	emp filtering
14.92	1	1	electronic implantable devices
13.82	1	2	trigg s tracking signal
12.08	1	9	clinical states

file: f4modified_1.108

weight	r	n	term
28.75	1	1	trend detection method
28.75	1	1	emp filtering
28.75	1	1	electronic implantable devices
13.68	1	2	trigg s tracking signal
11.60	1	9	clinical states

file: porter_1.108

weight	r	n	term
0.4986289186	2	4808	time series
0.4980785469	2	6738	biology
0.4976679069	2	8178	parameter estimation
0.2499997148	1	1	trend detection method
0.2499997148	1	1	emp filtering

file: zoomlist.108

rank	r	term
1	2	biology
2	2	parameter estimation
3	2	time series
4	1	adaptive forecasting
5	1	autoregression model

Table E.24: Search 110: five top-ranked terms for each algorithm.

file: weigths.110

w(p-q)	r	weight	n	term
6.8	7	7.8	6868	marketing
6.1	6	8.1	2687	research and development management
3.3	3	8.7	374	market research
2.3	3	6.0	5335	design engineering
1.8	1	14.2	1	specific aim importance

file: emim_1.110

weight	r	n	term
63.04	7	6868	marketing
60.69	6	2687	research and development management
38.39	3	374	market research
29.07	3	5335	design engineering
22.44	1	1	specific aim importance

file: f4_1.110

weight	r	n	term
14.15	1	1	specific aim importance
14.15	1	1	signal acquisition mode
14.15	1	1	r and d marketing co operation
14.15	1	1	product practicality
14.15	1	1	marketing task similarity

file: f4modified_1.110

weight	r	n	term
28.06	1	1	specific aim importance
28.06	1	1	signal acquisition mode
28.06	1	1	r and d marketing co operation
28.06	1	1	product practicality
28.06	1	1	marketing task similarity

file: porter_1.110

weight	r	n	term
0.8730414752	7	6868	marketing
0.7492337571	6	2687	research and development management
0.3748933477	3	374	market research
0.3734786357	3	5335	design engineering
0.2493766256	2	2186	project engineering

file: zoomlist.110

rank	r	term
1	7	marketing
2	6	research and development management
3	3	design engineering
4	3	market research
5	2	production

Table E.25: Search 111: five top-ranked terms for each algorithm.

file: weigths.111

w(p-q)	r	weight	n	term
7.9	4	7.9	12054	air pollution
5.0	2	10.1	149	human health
3.7	1	14.9	1	pb analysis techniques
3.7	1	14.9	1	automotive efficiency
3.5	3	4.8	67392	lead

file: emim_1.111

weight	r	n	term
35.24	4	12054	air pollution
30.16	2	149	human health
23.22	1	1	pb analysis techniques
23.22	1	1	automotive efficiency
22.96	1	2	wind direction effects

file: f4_1.111

weight	r	n	term
14.93	1	1	pb analysis techniques
14.93	1	1	automotive efficiency
13.83	1	2	wind direction effects
13.32	1	3	societal cost
12.98	1	4	pb transport

file: f4modified_1.111

weight	r	n	term
28.78	1	1	pb analysis techniques
28.78	1	1	automotive efficiency
13.70	1	2	wind direction effects
13.01	1	3	societal cost
12.60	1	4	pb transport

file: porter_1.111

weight	r	n	term
0.9966127785	4	12054	air pollution
0.7310625827	3	67392	lead
0.4999581304	2	149	human health
0.4978477908	2	7659	pollution detection and control
0.4931089404	2	24523	pb

file: zoomlist.111

rank	r	term
1	5	air pollution
2	3	lead
3	2	human health
4	2	pb
5	2	pollution detection and control

Table E.26: Search 112: five top-ranked terms for each algorithm.

file: weigths.112

w(p-q)	r	weight	n	term
6.7	4	8.4	2512	natural languages
6.2	2	15.4	2	self extending lexicon
5.2	2	12.9	8	rina
3.5	2	8.8	404	lexicon
3.0	2	7.4	1557	computational linguistics

file: emim_1.112

weight	r	n	term
42.44	4	2512	natural languages
38.07	2	2	self extending lexicon
36.54	2	8	rina
27.40	2	404	lexicon
24.03	2	1557	computational linguistics

file: f4_1.112

weight	r	n	term
15.44	2	2	self extending lexicon
14.68	1	1	vocabulary collection methods
14.68	1	1	text building blocks
14.68	1	1	russian homograph dictionary
14.68	1	1	rule based parsar

file: f4modified_1.112

weight	r	n	term
28.79	2	2	self extending lexicon
28.57	1	1	vocabulary collection methods
28.57	1	1	text building blocks
28.57	1	1	russian homograph dictionary
28.57	1	1	rule based parsar

file: porter 1.112

weight	r	n	term
0.7992963209	4	2512	natural languages
0.3999994397	2	2	self extending lexicon
0.3999977590	2	8	rina
0.3998868287	2	404	lexicon
0.3995638422	2	1557	computational linguistics

file: zoomlist.112

rank	r	term
1	4	natural languages
2	2	computational linguistics
3	2	error analysis
4	2	lexicon
5	2	rina

Table E.27: Search 113: five top-ranked terms for each algorithm.

file: weigths.113

w(p-q)	r	weight	n	term
3.5	3	5.9	13601	software packages
3.2	2	7.9	916	fuel consumption
3.0	2	7.5	1410	traffic computer control
2.9	1	14.7	1	vehicle s cumulative costs
2.9	1	14.7	1	vehicle management control

file: emim_1.113

weight	r	n	term
26.74	3	13601	software packages
25.36	2	916	fuel consumption
24.28	2	1410	traffic computer control
23.29	2	2093	finance
23.00	1	1	vehicle s cumulative costs

file: f4_1.113

weight	r	n	term
14.68	1	1	vehicle s cumulative costs
14.68	1	1	vehicle management control
14.68	1	1	registration renewals
14.68	1	1	personal property tax
14.68	1	1	personal finance programs

file: f4modified_1.113

weight	r	n	term
28.57	1	1	vehicle s cumulative costs
28.57	1	1	vehicle management control
28.57	1	1	registration renewals
28.57	1	1	personal property tax
28.57	1	1	personal finance programs

file: porter_1.113

weight	r	n	term
0.5961899922	3	13601	software packages
0.3997434036	2	916	fuel consumption
0.3996050209	2	1410	traffic computer control
0.3994136941	2	2093	finance
0.3971393427	2	10212	transportation

file: zoomlist.113

rank	r	term
1	3	software packages
2	2	finance
3	2	fuel consumption
4	2	traffic computer control
5	2	transportation

Table E.28: Search 114: five top-ranked terms for each algorithm.

file: weigths.114

w(p-q)	r	weight	n	term
9.0	6	9.1	5389	computer vision
4.2	4	6.4	11056	artificial intelligence
4.1	3	8.1	1053	machine vision
3.7	3	7.3	2297	computational geometry
3.5	2	10.4	59	geometric reasoning

file: emim_1.114

weight	r	n	term
56.03	6	5389	computer vision
35.53	4	11056	artificial intelligence
35.44	3	1053	machine vision
32.70	3	2297	computational geometry
31.85	2	59	geometric reasoning

file: f4_1.114

weight	r	n	term
14.48	1	1	shape analysis
14.48	1	1	photometric data
14.48	1	1	image understanding
14.48	1	1	geometric knowledge
14.48	1	1	2d images

file: f4modified_1.114

weight	r	n	term
28.38	1	1	shape analysis
28.38	1	1	photometric data
28.38	1	1	image understanding
28.38	1	1	geometric knowledge
28.38	1	1	2d images

file: porter_1.114

weight	r	n	term
0.9984903954	6	5389	computer vision
0.6635695822	4	11056	artificial intelligence
0.4997050262	3	1053	machine vision
0.4993565482	3	2297	computational geometry
0.4983693805	3	5821	computerised pattern recognition

file: zoomlist.114

rank	r	term
1	7	computer vision
2	4	artificial intelligence
3	3	computational geometry
4	3	computerised pattern recognition
5	3	machine vision

Table E.29: Search 115: five top-ranked terms for each algorithm.

file: weigths.115

w(p-q)	r	weight	n	term
9.3	4	9.3	2922	learning systems
3.7	1	14.9	1	replanner
3.7	1	14.9	1	intra example reasoning
3.7	1	14.9	1	inter example reasoning
3.7	1	14.9	1	explanation patterns

file: emim_1.115

weight	r	n	term
41.65	4	2922	learning systems
23.23	1	1	replanner
23.23	1	1	intra example reasoning
23.23	1	1	inter example reasoning
23.23	1	1	explanation patterns

file: f4_1.115

weight	r	n	term
14.93	1	1	replanner
14.93	1	1	intra example reasoning
14.93	1	1	inter example reasoning
14.93	1	1	explanation patterns
14.93	1	1	dynamic planning system

file: f4modified_1.115

weight	r	n	term
28.79	1	1	replanner
28.79	1	1	intra example reasoning
28.79	1	1	inter example reasoning
28.79	1	1	explanation patterns
28.79	1	1	dynamic planning system

file: porter_1.115

weight	r	n	term
0.9991814688	4	2922	learning systems
0.4969029155	2	11056	artificial intelligence
0.2499997199	1	1	replanner
0.2499997199	1	1	intra example reasoning
0.2499997199	1	1	inter example reasoning

file: zoomlist.115

rank	r	term
1	4	learning systems
2	2	artificial intelligence
3	1	abstract schemata
4	1	aerospace computing
5	1	case based explainer

Table E.30: Search 116: five top-ranked terms for each algorithm.

file: weigths.116

w(p-q)	r	weight	n	term
4.8	3	6.4	13948	inspection
3.7	1	14.9	1	system performance features
3.7	1	14.9	1	synchronized to product flow
3.7	1	14.9	1	solid state cid camera
3.7	1	14.9	1	rounded corner detection

file: emim_1.116

weight	r	n	term
26.92	3	13948	inspection
23.23	1	1	system performance features
23.23	1	1	synchronized to product flow
23.23	1	1	solid state cid camera
23.23	1	1	rounded corner detection

file: f4_1.116

weight	r	n	term
14.93	1	1	system performance features
14.93	1	1	synchronized to product flow
14.93	1	1	solid state cid camera
14.93	1	1	rounded corner detection
14.93	1	1	processes randomly oriented objects

file: f4modified_1.116

weight	r	n	term
28.79	1	1	system performance features
28.79	1	1	synchronized to product flow
28.79	1	1	solid state cid camera
28.79	1	1	rounded corner detection
28.79	1	1	processes randomly oriented objects

file: porter_1.116

weight	r	n	term
0.7460927882	3	13948	inspection
0.4984803109	2	5425	computer vision
0.4983620972	2	5847	computerised pattern recognition
0.4960165936	2	14220	computerised picture processing
0.4947946795	2	18582	pattern recognition

file: zoomlist.116

rank	r	term
1	3	inspection
2	2	computer vision
3	2	computerised pattern recognition
4	2	computerised picture processing
5	2	pattern recognition

Table E.31: Search 117: five top-ranked terms for each algorithm.

file: weigths.117

w(p-q)	r	weight	n	term
8.9	5	8.9	5425	computer vision
8.8	5	8.8	5847	computerised pattern recognition
8.1	5	8.2	11069	artificial intelligence
5.3	4	6.7	13582	expert systems
5.3	4	6.6	14220	computerised picture processing

file: emim_1.117

weight	r	n	term
47.43	5	5425	computer vision
47.02	5	5847	computerised pattern recognition
43.49	5	11069	artificial intelligence
34.80	4	13582	expert systems
34.60	4	14220	computerised picture processing

file: f4_1.117

weight	r	n	term
14.68	1	1	online scene interpretation
14.68	1	1	freely moving gold fish
14.68	1	1	distributed cooperative processes
14.68	1	1	distributed a1 environments
14.68	1	1	3d correspondence information

file: f4modified_1.117

weight	r	n	term
28.57	1	1	online scene interpretation
28.57	1	1	freely moving gold fish
28.57	1	1	distributed cooperative processes
28.57	1	1	distributed a1 environments
28.57	1	1	3d correspondence information

file: porter_1.117

weight	r	n	term
0.9984803109	5	5425	computer vision
0.9983620972	5	5847	computerised pattern recognition
0.9968992739	5	11069	artificial intelligence
0.7961953147	4	13582	expert systems
0.7960165936	4	14220	computerised picture processing

file: zoomlist.117

rank	r	term
1	7	computer vision
2	7	computerised pattern recognition
3	5	artificial intelligence
4	4	computerised picture processing
5	4	expert systems

Table E.32: Search 118: five top-ranked terms for each algorithm.

file: weigths.118

w(p-q)	r	weight	n	term
10.1	7	10.1	2239	sampled data systems
6.3	6	7.4	9625	control system synthesis
6.2	7	6.3	92484	stability
4.0	4	7.1	3852	discrete time systems
3.7	2	12.8	6	sampling time selection

file: emim_1.118

weight	r	n	term
71.18	7	2239	sampled data systems
52.56	6	9625	control system synthesis
42.87	7	92484	stability
40.02	4	3852	discrete time systems
36.50	2	6	sampling time selection

file: f4_1.118

weight	r	n	term
13.22	1	2	wideband frequency synthesizers
13.22	1	2	underdamped response
13.22	1	2	robust digital controllers
13.22	1	2	proportional integral plus control
12.80	2	6	sampling time selection

file: f4modified_1.118

weight	r	n	term
13.14	1	2	wideband frequency synthesizers
13.14	1	2	underdamped response
13.14	1	2	robust digital controllers
13.14	1	2	proportional integral plus control
12.60	2	6	sampling time selection

file: porter_1.118

weight	r	n	term
0.9993727956	7	2239	sampled data systems
0.9740927316	7	92484	stability
0.8544466345	6	9625	control system synthesis
0.5703495221	4	3852	discrete time systems
0.5674051069	4	14363	linear systems

file: zoomlist.118

rank	r	term
1	8	sampled data systems
2	8	stability
3	6	control system synthesis
4	4	discrete time systems
5	4	linear systems

Table E.33: Search 119: five top-ranked terms for each algorithm.

file: weigths.119

w(p-q)	r	weight	n	term
3.5	4	7.0	3366	programming environments
3.2	2	12.9	5	direct manipulation user interface
3.1	4	6.2	7181	user interfaces
2.4	2	9.6	93	visual programming
2.3	2	9.1	151	direct manipulation

file: emim_1.119

weight	r	n	term
40.32	4	3366	programming environments
36.89	4	7181	user interfaces
36.54	2	5	direct manipulation user interface
30.24	2	93	visual programming
29.05	2	151	direct manipulation

file: f4_1.119

weight	r	n	term
13.07	1	2	user librarian negotiations
13.07	1	2	object oriented interaction model
13.07	1	2	modular control structure
13.07	1	2	iconic programming language
13.07	1	2	event driven processes

file: f4modified_1.119

weight	r	n	term
13.01	1	2	user librarian negotiations
13.01	1	2	object oriented interaction model
13.01	1	2	modular control structure
13.01	1	2	iconic programming language
13.01	1	2	event driven processes

file: porter_1.119

weight	r	n	term
0.4990570924	4	3366	programming environments
0.4979884078	4	7181	user interfaces
0.3734497770	3	5534	interactive systems
0.3704851926	3	16117	computer graphics
0.2499985994	2	5	direct manipulation user interface

file: zoomlist.119

rank	r	term
1	4	programming environments
2	4	user interfaces
3	3	computer graphics
4	3	interactive systems
5	2	computer aided instruction

Table E.34: Search 120: five top-ranked terms for each algorithm.

file: weigths.120

w(p-q)	r	weight	n	term
8.5	5	8.5	7806	data structures
4.6	2	11.4	29	visual languages
4.3	3	7.1	4063	grammars
4.1	2	10.3	89	syntax directed editor
3.9	2	9.7	161	graph grammars

file: emim_1.120

weight	r	n	term
45.42	5	7806	data structures
33.81	2	29	visual languages
31.13	2	89	syntax directed editor
31.01	3	4063	grammars
29.68	2	161	graph grammars

file: f4_1.120

weight	r	n	term
13.07	1	3	attributed representation graph
12.74	1	4	specification paradigm
12.74	1	4	programmed attributed graph grammars
12.28	1	6	specification environments
12.28	1	6	diagram languages

file: f4modified_1.120

weight	r	n	term
12.79	1	3	attributed representation graph
12.38	1	4	specification paradigm
12.38	1	4	programmed attributed graph grammars
11.87	1	6	specification environments
11.87	1	6	diagram languages

file: porter_1.120

weight	r	n	term
0.9978133284	5	7806	data structures
0.5988618439	3	4063	grammars
0.5972452308	3	9834	graph theory
0.3999918763	2	29	visual languages
0.3999750687	2	89	syntax directed editor

file: zoomlist.120

rank	r	term
1	5	data structures
2	3	grammars
3	3	graph theory
4	2	formal specification
5	2	graph grammars

Table E.35: Search 121: five top-ranked terms for each algorithm.

file: weigths.121

w(p-q)	r	weight	n	term
9.3	5	9.3	3443	knowledge engineering
8.6	5	8.6	7181	user interfaces
4.5	3	7.5	2720	knowledge based systems
3.5	3	5.9	13582	expert systems
2.7	1	13.6	2	knowledge acquisition module

file: emim_1.121

weight	r	n	term
49.94	5	3443	knowledge engineering
45.88	5	7181	user interfaces
32.42	3	2720	knowledge based systems
26.75	3	13582	expert systems
22.74	1	2	knowledge acquisition module

file: f4_1.121

weight	r	n	term
13.58	1	2	knowledge acquisition module
13.58	1	2	kee
13.58	1	2	frame hierarchy
12.74	1	4	knowledge acquisition
12.12	1	7	semantic representation

file: f4modified_1.121

weight	r	n	term
13.48	1	2	knowledge acquisition module
13.48	1	2	kee
13.48	1	2	frame hierarchy
12.38	1	4	knowledge acquisition
11.69	1	7	semantic representation

file: porter_1.121

weight	r	n	term
0.9990355226	5	3443	knowledge engineering
0.9979884078	5	7181	user interfaces
0.5992380545	3	2720	knowledge based systems
0.5961953147	3	13582	expert systems
0.3954851926	2	16117	computer graphics

file: zoomlist.121

rank	r	term
1	5	knowledge engineering
2	5	user interfaces
3	3	expert systems
4	3	knowledge based systems
5	2	computer graphics

Table E.36: Search 122: five top-ranked terms for each algorithm.

file: weigths.122

w(p-q)	r	weight	n	term
7.5	6	10.1	406	load balancing
6.7	7	7.7	8417	distributed processing
3.1	4	6.3	6622	performance evaluation
3.1	3	8.2	644	resource sharing
2.5	3	6.8	2654	distributed systems

file: emim_1.122

weight	r	n	term
73.06	6	406	load balancing
61.67	7	8417	distributed processing
37.27	4	6622	performance evaluation
36.56	3	644	resource sharing
31.61	3	2654	distributed systems

file: f4_1.122

weight	r	n	term
13.08	1	2	multiple contention buses
13.08	1	2	heuristic multiwindow protocol
13.08	1	2	group load balancing system
13.08	1	2	distributed load balancing system
13.08	1	2	dfm ii

file: f4modified_1.122

weight	r	n	term
13.01	1	2	multiple contention buses
13.01	1	2	heuristic multiwindow protocol
13.01	1	2	group load balancing system
13.01	1	2	distributed load balancing system
13.01	1	2	dfm ii

file: porter_1.122

weight	r	n	term
0.8726484189	7	8417	distributed processing
0.7498865698	6	406	load balancing
0.4981499145	4	6622	performance evaluation
0.3748200762	3	644	resource sharing
0.3742585130	3	2654	distributed systems

file: zoomlist.122

rank	r	term
1	7	distributed processing
2	6	load balancing
3	4	performance evaluation
4	3	distributed systems
5	3	resource sharing

Table E.37: Search 123: five top-ranked terms for each algorithm.

file: weigths.123

w(p-q)	r	weight	n	term
11.0	4	11.0	536	transputers
8.7	4	8.7	5425	computer vision
7.3	2	14.7	3	road following algorithms
7.1	3	9.5	613	mobile robots
4.5	2	8.9	475	occam

file: emim_1.123

weight	r	n	term
49.28	4	536	transputers
38.87	4	5425	computer vision
38.23	2	3	road following algorithms
37.89	3	613	mobile robots
27.30	2	475	occam

file: f4_1.123

weight	r	n	term
14.69	2	3	road following algorithms
13.84	1	2	vme based framestore unit
13.84	1	2	transputer based machine
13.84	1	2	structured lighting vision system
13.84	1	2	alvey parsifal project

file: f4modified_1.123

weight	r	n	term
14.69	2	3	road following algorithms
13.70	1	2	vme based framestore unit
13.70	1	2	transputer based machine
13.70	1	2	structured lighting vision system
13.70	1	2	alvey parsifal project

file: porter_1.123

weight	r	n	term
0.9998502498	4	536	transputers
0.9984843380	4	5425	computer vision
0.7498287372	3	613	mobile robots
0.4999991618	2	3	road following algorithms
0.4998672923	2	475	occam

file: zoomlist.123

rank	r	term
1	5	computer vision
2	4	transputers
3	3	mobile robots
4	2	computerised picture processing
5	2	image processing

Table E.38: Search 124: five top-ranked terms for each algorithm.

file: weigths.124

w(p-q)	r	weight	n	term
7.7	6	7.7	20159	convection
3.9	2	11.8	16	square cavities
3.6	3	7.2	2598	buoyancy
3.4	2	10.1	82	rectangular cavities
2.9	2	8.7	324	confined flow

file: emim_1.124

weight	r	n	term
47.42	6	20159	convection
34.86	2	16	square cavities
32.28	3	2598	buoyancy
31.06	2	82	rectangular cavities
27.67	2	324	confined flow

file: f4_1.124

weight	r	n	term
13.39	1	2	msi algorithm
13.39	1	2	isothermal cubical cavities
12.88	1	3	solar thermal electric receivers
12.88	1	3	side facing apertures
12.88	1	3	axisymmetric regime

file: f4modified_1.124

weight	r	n	term
13.30	1	2	msi algorithm
13.30	1	2	isothermal cubical cavities
12.61	1	3	solar thermal electric receivers
12.61	1	3	side facing apertures
12.61	1	3	axisymmetric regime

file: porter_1.124

weight	r	n	term
0.9943678837	6	20159	convection
0.4992741585	3	2598	buoyancy
0.3333288632	2	16	square cavities
0.3333104238	2	82	rectangular cavities
0.3332428127	2	324	confined flow

file: zoomlist.124

rank	r	term
1	6	convection
2	3	buoyancy
3	2	confined flow
4	2	heat radiation
5	2	radiative transfer

Table E.39: Search 125: five top-ranked terms for each algorithm.

file: weigths.125

w(p-q)	r	weight	n	term
5.9	5	6.0	92743	stability
4.8	2	12.0	18	unstructured uncertainty
4.4	3	7.3	3243	robustness
3.3	3	5.6	18790	optimal control
3.3	2	8.3	635	two term control

file: emim_1.125

weight	r	n	term
34.89	2	18	unstructured uncertainty
31.81	3	3243	robustness
31.53	5	92743	stability
26.28	2	635	two term control
25.60	3	18790	optimal control

file: f4_1.125

weight	r	n	term
13.59	1	2	real parameter uncertainty
13.59	1	2	optimal model choice
13.08	1	3	pseudo output
11.98	1	8	robust stability condition
11.95	2	18	unstructured uncertainty

file: f4modified_1.125

weight	r	n	term
13.48	1	2	real parameter uncertainty
13.48	1	2	optimal model choice
12.79	1	3	pseudo output
11.62	2	18	unstructured uncertainty
11.54	1	8	robust stability condition

file: porter_1.125

weight	r	n	term
0.9740890243	5	92743	stability
0.5990939554	3	3243	robustness
0.5947503614	3	18790	optimal control
0.3999949711	2	18	unstructured uncertainty
0.3998225907	2	635	two term control

file: zoomlist.125

rank	r	term
1	5	stability
2	3	optimal control
3	3	robustness
4	2	adaptive control
5	2	parameter estimation

Table E.40: Search 126: five top-ranked terms for each algorithm.

file: weigths.126

w(p-q)	r	weight	n	term
6.2	2	15.4	2	2 iodonitrobenzene
4.2	2	10.5	70	organic synthesis
4.0	3	6.7	5939	ultrasound
3.6	4	4.7	94850	organic compounds
3.0	2	7.6	1249	ultrasonic effects

file: emim_1.126

weight	r	n	term
38.07	2	2	2 iodonitrobenzene
31.72	2	70	organic synthesis
29.68	3	5939	ultrasound
25.76	4	94850	organic compounds
24.59	2	1249	ultrasonic effects

file: f4_1.126

weight	r	n	term
15.45	2	2	2 iodonitrobenzene
13.08	1	3	carbon metal bond
12.29	1	6	nucleophilic substitutions
11.98	1	8	nucleophilic displacement
11.47	1	13	reacted species

file: f4modified_1.126

weight	r	n	term
28.80	2	2	2 iodonitrobenzene
12.79	1	3	carbon metal bond
11.87	1	6	nucleophilic substitutions
11.54	1	8	nucleophilic displacement
11.00	1	13	reacted species

file: porter_1.126

weight	r	n	term
0.7735003607	4	94850	organic compounds
0.5983407342	3	5939	ultrasound
0.3999994412	2	2	2 iodonitrobenzene
0.3999804431	2	70	organic synthesis
0.3996510485	2	1249	ultrasonic effects

file: zoomlist.126

rank	r	term
1	4	organic compounds
2	3	ultrasound
3	2	chemical reactions
4	2	cu
5	2	organic synthesis

Table E.41: Search 127: five top-ranked terms for each algorithm.

file: weigths.127

w(p-q)	r	weight	n	term
7.7	6	9.0	1867	air traffic control
7.4	5	10.4	246	air traffic computer control
6.2	6	7.2	11219	artificial intelligence
6.0	6	7.0	13896	expert systems
5.0	5	7.1	6715	expert system

file: emim_1.127

weight	r	n	term
64.48	5	246	air traffic computer control
63.27	6	1867	air traffic control
51.58	6	11219	artificial intelligence
50.17	6	13896	expert systems
46.34	5	6715	expert system

file: f4_1.127

weight	r	n	term
13.22	1	2	antico
13.22	1	2	aircraft trajectory generation
12.37	1	4	inference art
12.37	1	4	automated route planning
12.12	1	5	hybrid knowledge based system

file: f4modified_1.127

weight	r	n	term
13.14	1	2	antico
13.14	1	2	aircraft trajectory generation
12.05	1	4	inference art
12.05	1	4	automated route planning
11.76	1	5	hybrid knowledge based system

file: porter_1.127

weight	r	n	term
0.8566212459	6	1867	air traffic control
0.8540084401	6	11219	artificial intelligence
0.8532605273	6	13896	expert systems
0.7142169856	5	246	air traffic computer control
0.7124096460	5	6715	expert system

file: zoomlist.127

rank	r	term
1	6	air traffic control
2	6	artificial intelligence
3	6	expert systems
4	5	air traffic computer control
5	5	expert system

Table E.42: Search 128: five top-ranked terms for each algorithm.

file: weigths.128

w(p-q)	r	weight	n	term
7.4	11	8.1	8694	operating systems computers
6.2	10	7.5	8489	distributed processing
3.3	6	6.7	4424	fault tolerant computing
2.4	2	14.3	2	object thread model
2.4	2	14.3	2	clouds project

file: emim_1.128

weight	r	n	term
93.99	11	8694	operating systems computers
86.27	10	8489	distributed processing
56.48	6	4424	fault tolerant computing
36.50	2	2	object thread model
36.50	2	2	clouds project

file: f4_1.128

weight	r	n	term
14.35	2	2	object thread model
14.35	2	2	clouds project
14.35	2	2	clouds distributed operating system
14.35	2	2	aeolus programming language
13.25	2	3	isibas

file: f4modified_1.128

weight	r	n	term
27.78	2	2	object thread model
27.78	2	2	clouds project
27.78	2	2	clouds distributed operating system
27.78	2	2	aeolus programming language
13.39	2	3	isibas

file: porter_1.128

weight	r	n	term
0.9142376960	11	8694	operating systems computers
0.8309616366	10	8489	distributed processing
0.4987640021	6	4424	fault tolerant computing
0.3296518811	4	13177	clouds
0.2494934755	3	1813	fault tolerance

file: zoomlist.128

rank	r	term
1	11	operating systems computers
2	10	distributed processing
3	6	fault tolerant computing
4	4	clouds
5	3	fault tolerance

Table E.43: Search 129: five top-ranked terms for each algorithm.

file: weigths.129

w(p-q)	r	weight	n	term
8.9	2	8.9	2427	knowledge representation
6.9	1	13.8	4	life cycle paradigm
6.9	1	13.8	4	knowledge based software assistant
6.8	1	13.6	5	requirements decisions
6.8	1	13.6	5	requirements acquisition

file: emim_1.129

weight	r	n	term
23.67	2	2427	knowledge representation
22.97	1	4	life cycle paradigm
22.97	1	4	knowledge based software assistant
22.74	1	5	requirements decisions
22.74	1	5	requirements acquisition

file: f4_1.129

weight	r	n	term
13.84	1	4	life cycle paradigm
13.84	1	4	knowledge based software assistant
13.59	1	5	requirements decisions
13.59	1	5	requirements acquisition
12.49	1	14	system behaviors

file: f4modified_1.129

weight	r	n	term
13.30	1	4	life cycle paradigm
13.30	1	4	knowledge based software assistant
13.01	1	5	requirements decisions
13.01	1	5	requirements acquisition
11.83	1	14	system behaviors

file: porter_1.129

weight	r	n	term
0.9993219333	2	2427	knowledge representation
0.4999988825	1	4	life cycle paradigm
0.4999988825	1	4	knowledge based software assistant
0.4999986031	1	5	requirements decisions
0.4999986031	1	5	requirements acquisition

file: zoomlist.129

rank	r	term
1	2	knowledge representation
2	1	active reasoning agents
3	1	artificial intelligence
4	1	artificial intelligence techniques
5	1	automatic classification

References

Attar, L. and Fraenkel, A.S. (1977) Local feedback in full text retrieval systems. *Journal of the ACM,* 24, 397-417, 1977.

Attar, L. and Fraenkel, A.S. (1981) Experiments in local metrical feedback in full text retrieval systems. *Information Processing & Management,* 17, 115-126, 1981.

Bates, M.J. (1979a) Information search tactics. *Journal of the American Society for Information Science,* 30(4), 1979, pp.205-214.

Bates, M.J. (1979b) Idea tactics. *Journal of the American Society for Information Science,* 1979, 30(5), 280-289.

Bates, M.J. (1981) Search Techniques. In *Annual Review of Information Science and Technology,* Williams, M.E., editor. White Plains, N.Y.: Knowledge Industry Publications. 16, 1981, 139-169.

Bates, M.J. (1984) The fallacy of the perfect 30-item online search. *RQ,* 24(1): 43-50.

Bates, M.J. (1986) Terminological assistance for the online searcher. In *Proceedings of the 2nd Conference on Computer Interfaces and Intermediaries for Information Retrieval.* 1986 May 28-31, Boston, MA. Jacobson, C.E. and Witges, S.A. compilers. Report No. DTIC/TR-86/5, NTIS: AD A174-000-0. Alexandria, VA: Defence Technical Information Center; 1986, pp 285-293,

Bates, M.J. (1987) How to use information search tactics online. *Online,* 11(3), 1987, pp.47-54.

Belkin J.N., Seeger, T. and Wersig, G. (1983) Distributed expert problem treatment as a model for information system analysis and design. *Journal of Information Science,* 5:153-168.

Belkin, N.J. and Croft, W.B. (1987) Retrieval techniques. *Annual Review of Information Science and Technology,* Williams, M.E., ed. Amsterdam: Elsevier, vol. 22, 1987, pp.109-145.

Belkin, N.J. and Vickery, A. (1985) *Interaction in Information Systems: a review of research from document retrieval to knowledge-based systems.* Library and Information Research Report 35. London: British Library, 1985.

Belkin, N.J., Oddy, R.N. and Brooks, H.M. (1982) ASK for information retrieval: Part I. Background and theory. *Journal of Documentation,* 38(2), 61-71, 1982.

Bellardo, T. (1985) An investigation of online searcher traits and their relationship to search outcome. *Journal of the American Society for Information Science,* 36(4): 241-50

Bookstein, A. and Kraft, D. (1977) Operations research applied to document indexing and retrieval decisions. *Journal of the ACM,* 24:418-427.

Bookstein, A. and Swanson D. (1974) Probabilistic models for automatic indexing. *Journal of the American Society for Information Science*, 25:312-318.

Bookstein, A. and Swanson D. (1975) A decision theoretic foundation for indexing. *Journal of the American Society for Information Science*, 26:45-50.

Bookstein, A. (1985) Probability and fuzzy-set applications to information retrieval. In Williams, M.E. ed. *Annual Review of Information Science and Technology,* Knowledge Industries Publications, Inc. 1985, 117-151.

Borgman C.L. (1986) Why are online catalogs hard to use? Lessons learned from information retrieval studies. *Journal of the American Society for Information Science,* 37(6): 387-400.

Bovey, J.D. and Robertson, S.E. (1984) An algorithm for weighted searching on a Boolean system. *Information Technology,* 3(2), 1984, pp.84-87.

Brajnik, G., Guida, G. and Tasso, C. (1986) An expert interface for effective man-machine interaction. In: *Cooperative interfaces to information systems.* L. Bolc and M. Jarke, eds. Berlin: Springer-Verlag; 1986, pp.259-308.

Brookes, B.C. (1968) The measure of information retrieval effectiveness proposed by Swets. *Journal of Documentation,* 24, 1968, 41-54.

Brooks, H.M. (1986) *An intelligent interface for document retrieval systems: developing the problem description and retrieval strategy components.* Unpublished Ph.D. Thesis. London: Department of Information Science, City University, 1986.

Brooks, H.M., Daniels, P.J. and Belkin, N.J. (1986) Research on Information interaction and intelligent provision mechanisms. *Journal of Information Science,* 12, 1986, pp.37-44.

Burket, T.G., Emrath, P. and Kuck, D.J. (1979) The use of vocabulary files for online information retrieval. *Information Processing and Management,* 15, 1979, pp.281-289.

Chiaramella, Y. and Defude, B. (1987) A prototype of an Intelligent System for Information Retrieval: IOTA. *Information Processing & Management,* 23(4), 1987, pp.285-303.

Chow, C.K. & Liu, C.N. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory,* 1968, IT-14(3): 462-467.

Cleverdon, C.W. (1974) User evaluation of information retrieval systems. *Journal of Documentation,* 30:170-180.

Cleverdon, C.W. (1984) Optimizing convenient online access to bibliographic databases. *Information Services and Use,* 4:37-47.

Cooper, W.S. (1968) Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19(1): 30-41.

Cooper, W.S. (1977) The suboptimality of retrieval ranking based on the probability of usefulness, private communication to S.E. Robertson.

Croft, W.B. (1982) An overview of information systems. *Information Technology: Research and Development,* 1:73-96.

Croft, W.B. (1987) Approaches to Intelligent Information Retrieval. Information Processing and Management, Vol.23, No.4, 1987, pp. 249-254.

Croft, W.B. and Harper, D.J. (1979) Using probabilistic models of document retrieval without relevance information. *Journal of Documentation* 35 (4): 285-295; 1979.

Croft, W.B. and Thompson, R.H. (1987) I^3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science,* 38(6), 1987, pp.389-404.

Cuadra, C.A. and Katter, R.V. (1967) Opening the black box of relevance. *Journal of Documentation,* 23:251-303.

D'Elia, S. and Marchetti, P.G. (c1985) QUESTQUORUM: A new online search assistance tool from ESA-IRS. *Available online from ESA-IRS. c1985.*

Dillon, M. and Desper, J. (1980) The use of automatic relevance feedback in boolean retrieval systems. *Journal of Documentation,* 36(3), 1980, pp.197-208.

Dillon, M., Ulmschneider, J. and Desper, J. (1983) A prevalence formula for automatic relevance feedback in boolean systems. *Information Processing and Management,* 19(1), 1983, pp.27-36.

Doszkocs, T.E. (1978) AID - an associative interactive dictionary for on-line searching. *Online Review,* 2:163-173.

Doszkocs, T.E. (1978) An associative interactive dictionary (AID) for online bibliographic searching. In: *The Information Age in Perspective. Proceedings of the ASIS Annual Meeting 1978, New York, NY, USA, 13-17 Nov. 1978.* Knowledge Industry Publications Inc, White Plains, NY, USA, 1978. p.105-9.

Doszkocs, T.E. (1983) CITE NLM: Natural-language searching in an online catalog. *Information Technology and Libraries,* 1983, 2(4), 364-380.

Doszkocs, T.E. (1986) Natural Language Processing in Information Retrieval. Journal of the American Society for Information Science, Vol.37, No.4, 1986, pp. 191-196.

Doszkocs, T.E. and Rapp, B.A. (1979) Searching MEDLINE in English: A prototype user interface with natural language query, ranked output and relevance feedback. In: *Information Choices and Policies, Proceedings of the 42nd ASIS Annual meeting, Minneapolis, Minnesota, Oct. 14-18, 1979.* White Plains New York, Knowledge Industry Publications Inc, pp 131-139.

Draper, S.W. and Norman, D.A. (1985) Software engineering for user interfaces. In: (7th International Conference on Software Engineering, Orlando, FL, USA, 26-29 March 1984. Sponsored by IEEE, ACM, NBS.) *IEEE Transactions on Software Engineering,* SE-11, 1985, (3), 252-258.

Duda, R.O. & Hart, P.E. (1973) *Pattern classification and scene analysis.* New York, NY: Wiley-Interscience, 1973.

Efthimiadis, E.N. (1990) Online searching aids: a review of front-ends, gateways and other interfaces. *Journal of Documentation,* 1990, 46(3), 218-262.

Efthimiadis, E.N. and Robertson, S.E. (1989) Feedback and Interaction in Information Retrieval. In: *Perspectives in Information Management,* Oppenheim, C., ed. London: Butterworths, 1989, pp.257-272.

Ellis, D. (1984) Theory and explanation in information retrieval research, *Journal of Information Science,* 8: 25-38.

Eichman, T.L. (1978) The complex nature of opening reference questions. *RQ,* 1978, 17(3), 212-222.

Fenichel, C.H. (1981) Online searching: measures that discriminate among users with different types of experiences *Journal of the American Society for Information Science,* 32(1): 23-32

Fidel, R. (1985) Moves in online searching. *Online Review,* 9(1), 1985, pp.61-74.

Fidel, R. (1986) Towards expert systems for the selection of search keys. *Journal of the American Society for Information Science,* 37(1): 37-44.

Fidel, R. and Soergel, D. (1983) Factors affecting online bibliographic retrieval: a conceptual framework for research. *Journal of the American Society for Information Science,* 34(3): 163-180.

Fox, E.A. (1987) Development of the CODER system: a testbed for artificial intelligence methods in information retrieval. *Information Processing & Management,* 23(4), 1987, pp.341-366.

Fox, E.A. and Koll, M, B. (1988) Practical enhanced Boolean retrieval: experiences with the SMART and SIRE systems. *Information Processing & Management,* 24(3): 257-67.

Frei, H.P. and Jauslin, J.F. (1983) Graphical presentation of information and services: a user-oriented interface. *Information Technology: Research and Development,* 2: 23-42, (1983).

Freund, G.E. and Willett, P. (1982) Online identification of word variants and arbitrary truncation searching using a string similarity measure. *Information Technology: Research and Development,* 1982, 1(3), 177-187.

Fung, R.M., Crawford, L., Appelbaum, L.A, and Tong, R.M. (1990) An architecture for probabilistic concept, based information retrieval. *Proceedings of the 13th International Conference on Research and Development in Information Retrieval, Brussels, Belgium, 5-7 Sept. 1990,* Vidick, J.L. (ed.) ACM, New York, NY, USA, 1989, p.455, 68.

Gauch, S. and Smith, J.B. (1989) Query reformulation strategies for an intelligent search intermediary. In: *Proceedings of the annual AI systems in government conference,* IEEE-Cat.no.89CH2715-1, 1989, pp.65-71.

Goldsmith, G. and Williams, P.W. (1986) Online searching made simple: a microcomputer interface for inexperienced users. British Library, Library and Information Research Report No. 41, 108 pp.

Harman, D. (1988) Towards interactive query expansion. In: *11th International Conference on Research and Development in IR, SIGIR 1988, Grenoble, France,* Presses Universitaires de Grenoble, France, pp 321-331.

Harper, D.J. (1980) *Relevance Feedback in Document Retrieval Systems,* Ph.D. Thesis, Computer Laboratory, University of Cambridge.

Harper, D.J. and van Rijsbergen, C.J. (1978) An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation* 34 (3): 189-216; 1978.

Harter, S.P. (1975a) A probabilistic approach to automatic keyword indexing, Part I: On the distribution of specialty words in a technical literature. *Journal of the American Society for Information Science*, 26:197-206.

Harter, S.P. (1975b) A probabilistic approach to automatic keyword indexing, Part II: An algorithm for probabilistic indexing. *Journal of the American Society for Information Science*, 26:280-289.

Harter, S.P. (1986) *Online information retrieval: concepts, principles, and techniques.* Academic Press.

Harter, S.P. and Peters, A.R. (1985) Heuristics for online information retrieval: a typology and preliminary listing. *Online Review,* 9(5), 1985, pp.407-424.

Hartley R.J. *et al.* (1989) *Online searching: principles and practice.* London: Bowker-Sauer.

Hawkins, D.T. (1988) Applications of artificial intelligence (AI) and expert systems for online searching. *Online,* 12(1), 1988, pp.31-43.

Heine, M.H. (1973) The inverse relationship of precision and recall in terms of the Swets model. *Journal of Documentation,* 29(1):81-4.

Heine, M.H. (1982) A simple intelligent front-end for information retrieval systems using Boolean logic. *Information Technology: Research & Development,* 2, 247-260, 1982.

Heine, M.H. (1988) Logic assistant for the database searcher. *Information Processing & Management,* 24, 323-329, 1988.

Henry, M., Leigh, J., Tedd, L. and Williams, P. (1980) *Online Search: An Introduction.* London: Butterworths. 1980.

Hendry, I.G., Willett, P. and Wood, F.E. (1986) INSTRUCT: A teaching package for experimental methods in information retrieval. Part I. The users' view. *Program,* 1986, 20, 245-263.

Ingwersen, P. (1984) A cognitive view of three selected online search facilities. *Online Review,* 8(5), 1984, pp.465-492.

Jamieson, S.H. and Oddy, R.N. (1979) *Implementation and evaluation of interactive retrieval through an intelligent terminal.* A project proposal to the British Library Research and Development Department, 1979, unpublished.

Jamieson, S.H. (1979a) An intelligent terminal for information retrieval. *Journal of Informatics*, 3(1), April 1979, 51-56.

Jamieson, S.H. (1979b) The economic implementation of experimental retrieval techniques on a very large scale using an intelligent terminal. In: *Proceedings of the Second International Conference on Information Storage & Retrieval,* Dallas, TX, 1979. New York: Association of Computing Machinery, 45-51.

Kaye, D. (1973) A weighted rank correlation coefficient for the comparison of relevance judgements. *Journal of Documentation,* 29:380-389.

Katzer, J. (1982) A study of the overlap among document representations. *Information Technology: Research and Development,* 2:261-274.

Keen, E.M. (1971) Evaluation parameters. In: Salton G., ed. *The SMART retrieval system. Experiments in automatic document processing.* Englewood Cliffs, NJ: Prentice-Hall, pp 74-111.

Knapp, S.D. (1978) The reference interview in the computer-based setting. *RQ,* 1978, 17(4), 320-324.

Kraft, D.H. and Bookstein, A. (1978) Evaluation of information retrieval systems: a decision theory approach. *Journal of the American Society for Information Science*, 29:31-40.

Lamb, M.R., Auster, E.W. and Westel, E.R. (1985) A friendly front-end for bibliographic retrieval: the implementation of a flexible interface. In: *Proceedings of the 48th ASIS annual meeting,* vol. 22, Las Vegas, Oct 22-24, 1985. Parkhurst, C.A., ed. New York: Knowledge Industry Publications, 1985, pp.229-235.

Lancaster, F.W. (1979) *Information retrieval systems: characteristics, testing and evaluation.* 2nd edition. New York: John Wiley & Sons, 1979.

Lehmann, E.L. (1975) *Nonparametrics: statistical methods based on ranks.* Oakland, CA: Holden Hay.

Lesk, M.E. (1969) Word-word associations in document retrieval systems. *American Documentation,* 20:27-38.

Macaskill, M.J. (1987) Splitting CIRT into two processes. In: Robertson & Thompson, 1987, A1.1-A.3.

Marcus, R.S. (1983) Computer-assisted search planning and evaluation. In: *Proceedings of the 46th ASIS Annual Meeting.* vol.20, October 1983, pp.19-21.

Markey, K. (1981) Levels of question formulation in negotiation of information need during the online presearch interview: A proposal model. *Information Processing and Management,* 1982, 17(5), 215-225

Maron, M.E. and Kuhns, J.L. (1960) On relevance, probabilistic indexing, and information retrieval. *Journal of the ACM,* 7:216-244.

Martin, W.A. (1982) Helping the less experienced user. In: *6th International Online Meeting.* London, 7-9 December 1982. Oxford: Learned Information (Europe) Ltd. 1982, pp. 67-76.

McCune, B.P., Tong, R.M., Dean, J.S. and Shapiro, D.G. (1985) Rubric: A System for rule based Information Retrieval, IEEE Transactions on Software Engineering, Vol.SE-11, No.9, Sept.1985, pp. 939-944.

McGill, M., Koll, M. and Noreault, T. (1979) An evaluation of factors affecting document ranking by information retrieval systems. Technical report, Syracuse University, School of Information Studies, 1979.

McMath, C.F., Tamaru, R.S. and Rada, R. (1989) A graphical thesaurus-based information retrieval system. *International Journal of Man-Machine Studies,* 31, 1989, pp.121-147.

Meadow, C.T. and Cochrane, P.A. (1981) *Basics of Online Searching.* New York, N.Y.: John Wiley & Sons, 1981.

Meadow, C.T., Cerny, B.A., Borgman, C.L. and Case, D.O. (1989) Online Access to Knowledge: system design. *Journal of the American Society for Information Science,* 40(2), 1989, pp.86-98.

Meadow, C.T. (1979) The computer as a search intermediary. *Online*, 3(3): 54-59, (1979).

Minker, J., Wilson, G.A. and Zimmerman, B.H. (1972) An evaluation of query expansion, by the addition of clustered terms for a document retrieval system. Information Storage and Retrieval, Vol.8, 1972, pp. 329-348.

Mischo, B. (1986) End user searching in an online catalog environment. In: *Proceedings of the 2nd Conference on Computer Interfaces and Intermediaries for Information Retrieval.* 1986 May 28-31, Boston, MA. Jacobson, C.E. and Witges, S.A. compilers. Report No. DTIC/TR-86/5, Alexandria, VA: Defence Technical Information Center. 1986, pp.241-262.

Monarch, I and Carbonell, J. (1987) Coal SORT: a knowledge-based interface. *IEEE Expert,* 2: 39-53, (1987).

Morrissey, Joan. (1981) An intelligent terminal to access Euronet. Student project, Computer Science Department, University College Dublin, Ireland, unpublished.

Norman, D.A. (1984) Stages and levels in human-machine interaction. *International Journal of Man-Machine Studies.* 1984, 21(4), 365-375.

Nutter, J.T., Fox, E.A. and Evens, M.W. (1990) Building a lexicon from machine-readable dictionaries for improved information retrieval. *Literary and Linguistic Computing,* 5(2): 129-137.

Oddy, R.N. (1977) Information retrieval through man-machine dialogue. *Journal of Documentation,* 33(1), 1977, pp.1-14.

Peat, H.J., and Willett, P. (1991) The limitations of term co- occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42:378-383.

Pietilainen, P. (1983) Local feedback and intelligent automatic query expansion. *Information Processing and Management,* 1983, 19(1), 51-58.

Pollitt, A.S. (1981) An expert system as an online search intermediary. In *5th International Online Information Meeting.* London 8-10, December 1981, Oxford: Learned Information Ltd. 1981, pp. 25-32.

Pollitt, A.S. (1987) CANSEARCH: An expert systems approach to document retrieval. *Information Processing and Management,* 1987, 23(2), 119-138.

Pollitt, A.S. (1988) A common query interface using Men USE —A menu-based user interface search engine. In: *Proceedings of the 12th International Online Meeting.* London 6-8 December 1988. Oxford: Learned Information, 1988, vol. 2, pp.445-457.

Pollock, S.M. (1968) Measures for the comparison of information retrieval systems. *American Documentation,* 19:387-397.

Porter, M.F. (1982) Implementing a probabilistic information retrieval system. *Information Technology: Research and Development,* 1982, 1(2), 131-156.

Porter, M.F. and Galpin, V. (1988) Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute. *Program,* 1988, 22(1), 1-20.

Preece, S.E. (1980) An online associative query modification methodology. *Online Review,* 4(4): 375-82.

Rada, R. (1987) Knowledge-sparse and knowledge-rich learning in information retrieval. *Information Processing and Management,* 1987, 23(3), 195-210.

Radecki, T. (1988) Trends in research on information retrieval — the potential for improvements in conventional Boolean retrieval systems. *Information Processing & Management,* 24(3), 219-227, 1988.

Rees, A.M. and Schultz, D.G. (1967) A field experimental approach to the study of relevance assessments in relation to document searching. 2 Vols, Centre for Documentation and Communication Research, Case Western Reserve University.

Resnick, A. and Savage, T.R. (1964) The consistency of human judgements of relevance. *American Documentation,* 15:43-54.

Ro, J.S. (1988) Evaluation of the applicability of ranking algorithms, Pt. I and Pt. II. *Journal of the American Society for Information Science*, 39:73-78; 147-160.

Robertson, S.E. (1969) The parametric description of retrieval tests, Pt. I and Pt. II. *Journal of Documentation,* 25:1-27; 93- 107.

Robertson, S.E. (1974) Specificity and weighted retrieval. *Journal of Documentation,* 30(1): 41-46.

Robertson, S.E. (1977a) The probabilistic character of relevance. *Information Processing & Management*, 13(4): 247-51.

Robertson, S.E. (1977b) The probability ranking principle in IR. *Journal of Documentation*, 33, 1977, 294-304.

Robertson, S.E. (1978) Indexing theory and retrieval effectiveness. *Drexel Library Quarterly*, 14(2), 40-56, 1978

Robertson, S.E. (1986) On relevance weight estimation and query expansion. *Journal of Documentation* 42 (3): 182-188; 1986.

Robertson, S.E. (1990a) On sample sizes for non-matched-pair IR experiments. *Information Processing & Management*, 26(6), 739-753.

Robertson, S.E. (1990b) On term selection for query expansion. *Journal of Documentation*, 46(4), 359-364.

Robertson, S.E. and Belkin, N.J. (1978) Ranking in principle. *Journal of Documentation*, 34:93-100.

Robertson, S.E. and Bovey, J.D. (1983) *A front-end for IR experiments*. Final report to the British Library Research and Development Department, BLRDD Report No. 5807; 1983.

Robertson, S.E. and Sparck Jones, K. (1976) Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3), 1976, pp.129-146.

Robertson, S.E. and Thompson, C.L. (1987) *An operational evaluation of weighting, ranking and relevance feedback via a front-end system.* Final report to the British Library Research and Development Department, BLRDD Report No. 5949, SI/G 703, 1987.

Robertson, S.E. and Thompson, C.L. (1990) Weighted searching: the CIRT experiment. In: *Informatics 10: prospects for intelligent retrieval.* University of York, 21-23 March 1989. London: ASLIB, 1990, 153-166.

Robertson, S.E., Bovey, J.D., Thompson, C.L. and Macaskill, M.J. (1986) Weighting, ranking and relevance feedback in a front-end system. *Journal of Information Science*, 12, 1986, pp.71-75.

Robertson, S.E., Maron, M.E. and Cooper W.S. (1982) Probability of relevance: A unification of two competing models for document retrieval. *Information Technology: Research and Development*, 1:1-21.

Robertson, S.E., Maron, M.E. and Cooper W.S. (1983) The unified probabilistic model for IR. In: Salton, G. and Schneider, H-J, eds. *Research and Development in Information Retrieval: Proceedings of Conference; May 18-20, 1982; Berlin, West Germany.* Berlin: Springer-Verlag; pp 108-117.

Rocchio, Jr., R.R. (1966) *Document retrieval systems - Optimization and evaluation.* Doctoral thesis. Report IST-IO to the National Science Foundation, Harvard Computation Laboratory, Cambridge, MA.

Sager, W.K.H. and Lockemann, P.C. (1976) Classification of ranking algorithms. *International Forum for Information and Documentation*, 1:12-25.

Salton, G. (1968) *Automatic information organization and retrieval.* New York, NY: Mc Graw-Hill.

Salton G. (1970) Evaluation problems in interactive information retrieval. *Information Storage Retrieval,* 6:29-44.

Salton, G., ed. (1971) *The SMART retrieval system: Experiments in automatic document processing.* Englewood Cliffs, NJ: Prentice-Hall.

Salton, G. (1973) Comment on "an evaluation of query expansion by the addition of clustered terms for a document retrieval system". *Computing Reviews,* 14, 232.

Dynamic Information and Library Processing. Englewood Cliffs: Prentice-Hall.

Salton, G. (1988) A simple blueprint for automatic Boolean query processing. *Information Processing & Management,* 24(3), 269-280, 1988.

Salton, G. and Mc Gill, J. (1983) *Introduction to Modern Information Retrieval,* New York: Mc Graw Hill Inc. 1983.

Salton, G., Wu, H. and Yu, C.T. (1981) The measurement of term importance in automatic indexing. *Journal of the American Society for Information Science,* 32:175-186.

Salton, G., Fox, E.A. and Wu, H. (1983a) Extended Boolean information retrieval. *Communications of the ACM,* 26: 1022-1036.

Salton, G., Buckley, C. and Fox, E.A. (1983b) Automatic query formulation in Information Retrieval. Journal of the American Society for Information Science, Vol.34, No.4, 1983, pp. 262-286.

Salton, G. et al. (1984) A comparison of two methods for Boolean query relevance feedback. *Information Processing and Management,* 20:637-651.

Salton, G., Fox, E.A. and Voorhees, E. (1985) Advanced feedback methods in information retrieval. *Journal of the American Society for Information Science,* 36(3), 1985, pp.200-210.

Saracevic, T. (1975) Relevance: a review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science,* 26: 321-343.

Saracevic, T. and Kantor, P. (1988) A study of information seeking and retrieving. II. Users, questions, and effectiveness III. Searchers, searches, and overlap. *Journal of the American Society for Information Science,* 39(3): 177-96 and 197-216

Schamber, L., Eisenberg, M.B. and Nilan, M.S. (1990) A re-examination of relevance: toward a dynamic, situational definition. *Information Processing & Management,* 26(6), 755-776, 1990

Shneiderman, B. (1986) Designing menu selection systems. *Journal of the American Society for Information Science,* 1986, 37(2), 57-70.

Shoval, P. (1981) Expert/consultation system for a retrieval data-base with semantic network of concepts. *Proccedings of the ACM SIGIR Conference*, SIGIR Forum, 16: 145-149, (1981).

Shoval, P. (1985) Principles, procedures and rules in an expert system for information retrieval. *Information Processing and Management,* 21(6), 1985, pp.475-487.

Smeaton A.F. (1982) *The retrieval effects of query expansion on a feedback document retrieval system.* Master Thesis, Department of Computer Science, University College Dublin.

Smeaton, A.F. (1984) Relevance feedback and a fuzzy set of search terms in an information retrieval system. *Information Technology,* 3: 15-23.

Smeaton, A.F. and van Rijsbergen, C.J. (1981) The nearest neighbour problem in information retrieval. An algorithm using upper bounds. *Proccedings of the Fourth International Conference on Information Storage and Retrieval, Oakland, CA, USA, 31 May - 2 June 1981.* In: *SIGIR Forum,* 16(1): 83-87.

Smeaton, A.F. and van Rijsbergen, C.J. (1983) The retrieval effects of query expansion on a feedback document retrieval system. *The Computer Journal,* 26(3), 1983, pp.239-246.

Smith, J.B., Weiss, S.F. and Ferguson, G.J. (1987) MICROARRAS: An advanced full-text retrieval and analysis system. In: *Proceedings of the Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* van Risjbergen, C.J. and Yu, C.T., eds. ACM Press, 1987, pp.187-195.

Sparck Jones, K., ed. (1971) *Automatic keyword classification for information retrieval.* London:Butterworths.

Sparck Jones, K. (1972) A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation,* 28: 11-21.

Sparck Jones, K. (1973) Does indexing exhaustivity matter? *Journal of the American Society for Information Science*, 24:313-316.

Sparck Jones, K. (1975) A performance yardstick for test collections. *Journal of Documentation,* 31: 266-272.

Sparck Jones, K. (1978) Performance averaging for recall and precision. *Journal of Informatics,* 2:95-105.

Sparck Jones, K. (1979a) Search term relevance weighting given little relevance information. *Journal of Documentation* 35 (1): 30-48; 1979.

Sparck Jones, K. (1979b) Experiments in relevance weighting of search terms. *Information Processing and Management* 15 (3): 133-144; 1979.

Sparck Jones, K. (1980) Search term weighting: some recent results. *Journal of Information Science* 1 (6): 325-332; 1980.

Sparck Jones, K., ed. (1981) *Information Retrieval Experiment.* London:Butterworths.

Sparck Jones, K. (1988) A look back and a look forward. In: *Proceedings of the 11th International Conference on Research & Development in Information Retrieval. June 13-15, 1988, Grenoble, France.* Yves Chiaramella (ed.) ACM Press. 13-29.

Sparck Jones, K. and Barber, E.O. (1971) What makes an automatic keyword classification effective? *Journal of the American Society for Information Science,* 22(3): 166-75.

Sparck Jones, K. and Bates, R.G. (1977) Report on a design study for the "ideal" information retrieval test collection. British Library Research and Development Report 5428, Computer Laboratory, University of Cambridge.

Sparck Jones, K. and Jackson, D.M. (1970) The use of automatically-obtained keyword classifications for information retrieval. *Information Storage and Retrieval,* 5:175-201.

Sparck Jones, K. and van Rijsbergen, C.J. (1976) Information retrieval test collections. *Journal of Documentation,* 32:59-75.

Su, L. (1989) An investigation to find appropriate measures for evaluating interactive information retrieval. In: ASIS 1989, *Proceedings of the 52nd ASIS Annual Meeting, Washington, DC, October 30 - November 2, 1989.* Katzer, J. and Newby, G.B. (eds). vol. 26, pp 13-23.

Svenonius, E. (1986) Unanswered questions in the design of controlled vocabularies. *Journal of the American Society for Information Science,* 37:331-340.

Swets, J.A. (1963) Information retrieval systems. *Science,* 141, 1963, 245-250.

Swets, J.A. (1969) Effectiveness of information retrieval methods. *American Documentation,* 20:72-89.

Taylor, R. (1968) Question-negotiation and information seeking in libraries. *College and Research Libraries,* 1968, 29(3), 178-194.

Tenopir, C. (1988) An interface for self-service searching. *Library Journal,* September 1, 1988, pg.142-143.

Thompson, R.H. and Croft, W.B. (1989) Support for browsing in an intelligent text retrieval system. *International Journal of Man-Machine Studies,* 30, 1989, pp.639-668.

Trenner, L. (1989) A comparative survey of the friendliness of online "help" in interactive information retrieval systems. *Information Processing & Management,* 25(2), 1989, pp.119-136.

Turtle, H. and Croft.W.B. (1990) Inference networks for document retrieval. *Proceedings of the 13th International Conference on Research and Development in Information Retrieval, Brussels, Belgium, 5-7 Sept. 1990,* Vidick, J-L. (ed.) ACM, New York, NY, USA. p.124-129.

van Rijsbergen, C.J. (1974) Further experiments with hierarchic clustering in Document Retrieval. Information Storage and Retrieval, Vol.10, 1974, pp. 1-17.

van Rijsbergen, C.J. (1977) A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation,* 33:106-119.

van Rijsbergen, C.J. (1979) *Information Retrieval.* 2nd edition, London: Butterworth, 1979.

van Rijsbergen, C.J. (1986) A non-classical logic for Information Retrieval. Computer Journal, Vol.29, No.6, 1986, pp. 481-485.

van Rijsbergen, C.J., Harper, D.J. and Porter, M.F. (1981) The selection of good search terms. *Information Processing and Management,* 17(2), 1981, pp.77-91.

Vernimb, C. (1977) Automatic query adjustment in document retrieval. *Information Processing and Management,* 13:339-353.

Vickery, A. (1988) The experience of building expert search systems. In: *Proceedings of the 12th International Online Meeting.* London 6-8 December 1988. Oxford: Learned Information, 1988, vol. 1, pp.301-313.

Vickery, A., Brooks, H.M., Robinson, B. and Vickery, B.C. (1986) *Expert System for Referral,* Final Report, University of London, Central Information Service. 1986.

Vickery, A., Brooks, H.M., Robinson, B. and Vickery, B.C. (1987) A reference and referral system using expert system techniques. *Journal of Documentation,* 1987, 4, 198-203.

Wade, S.J. and Willett, P. (1988) INSTRUCT: a teaching package for experimental methods in information retrieval. Part III. Browsing, clustering and query expansion. *Program,* 1988, 22(1), 44-61.

Walker, S. (1989) The Okapi online catalogue research projects. In *The online catalogue,* Hildreth, C.R., ed. London, Library Association. 84-106, (1989).

Walker, S. and de Vere R. (1990) *Improving subject retrieval in online catalogues: 2. Relevance feedback and query expansion.* British Library Research Paper 72. London: British Library, 1990.

Walker, S. and Jones, R.M. (1987) *Improving subject retrieval in online catalogues: 1. Stemming, automatic spelling correction and cross-reference tables.* British Library Research Paper 24. London: British Library, 1987.

White, H.D. (1989) Toward automated search strategies. *Online Information 89. 13th International Online Information Meeting Proceedings, London, UK, 12-14 Dec. 1989.* Learned Information, Oxford, UK. p.33-48.

Willett, P., ed. (1988) *Document retrieval systems.* The Foundations of Information Science, Vol. 3, Taylor Graham and the Institute of Information Scientists.

Williams, P.W. (1983) A microcomputer system to improve the recall performance of skilled searchers. In: *Proceedings of the 4th National Online Meeting, New York, April 12-14 1983.* Medford, NJ: Learned Information. 1983, pp.581-590.

Williams, P.W. (1984) A model for an expert system for automated information retrieval. In: *8th International Online Information Meeting. London, 4-6 December 1984.* Oxford: Learned Information. 1984, pp.139-149.

Williams, P.W. (1985) The design of an expert system for access to information. In: *9th International Online Information Meeting, London 3-5 December 1985.* Oxford: Learned Information; 1985, pp.23-29.

Wu, H. and Salton, G. (1981) The estimation of term relevance weights using relevance feedback. *Journal of Documentation,* 37(4), 1981, pp.194-214.

Yip, M.K. (1981) An expert system for document retrieval. *M.S. Thesis,* Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA, USA, 1981.

Yu, C.T., Luk, W.S. and Siu, M.K. (1979) On models of information retrieval processes. *Information Systems,* 4(3): 205-218.

Yu, C.T., Buckley, C., Lam, K. and Salton, G. (1983) A generalized term dependence model in information retrieval. *Information Technology Research & Developmemnt,* 2(4): 129-154.