Abduction, explanation and relevance feedback

Ian Ruthven

Department of Computing Science

University of Glasgow

UNIVERSITY
of
GLASGOW

Submitted for the Degree of Doctor of Philosophy

at the University of Glasgow

31st October 2001

Volume 1

Declaration of originality

The material in this thesis is entirely the results of my own independent research under the supervision of Professor C. J. van Rijsbergen and Dr M. Lalmas, and is not the outcome of any collaborative work. All published or unpublished material used in this thesis has been given full acknowledgement.

Permission to copy

Acknowledgements

A PhD thesis has one name on the front. However behind this one person, who takes all the credit, are dozens of other people; cajoling, encouraging, criticising, stimulating and generally making sure that several years of alternating anguish and excitement are eventually transformed into several hundred pages of text. Many people contributed to this thesis, directly or indirectly, either by discussing the ideas contained within, reading huge chunks, or watching me endlessly pace the floor muttering 'This is never going to work' without flinging something heavy at my head. For this especially I thank you all.

Special thanks are due to certain special people. Firstly I have to thank Keith van Rijsbergen, my supervisor. He deserves thanks mainly for sticking by me in the difficult early days; he will claim he had faith, I think he just likes a long risk. He also deserves many thanks for his constant intellectual stimulation, a wonderful working environment and much support throughout my years at Glasgow. Keith, I never left your office without something to think about (and usually another thesis to read).

A huge big set of thanks go to Mounia Lalmas, my other supervisor. Mounia you were a joy to work with. Thank you (in no particular order): for reading everything I ever wrote, at least ten times[1], for providing so much *practical* advice and motivation, for making sure this thesis saw the light of day, for teaching my how to use a semi-colon, and for trusting me to get on with it. I realise this last part was not always easy ;-)

Mark Dunlop, my other, other supervisor, also deserves much credit for never saying no to the thankless task of reading this stuff, providing useful hints on evaluation and being such a calming influence.

The Glasgow IR group - past and present and honorary members - provided a wonderful 'family'. To Mirna, Robert, Di, Tassos, Marcos, Iadh, Jon, Naveed, Mark, Mark, Mark, Iain, Fabio, Martin, and Ryen. Thanks for all the good times!

I am much indebted to other friends, colleagues and partners in crime. These include Pia Borlund for many interesting discussions, some of which were about research, and, in particular, for many helpful comments on the evaluation of this research and making available data from her own experiments. Joemon Jose, for valiantly reading the whole thing and for

[1] And improving my writing no end, usually by covering my papers with comments like 'What?', 'Say this in English', 'What does this mean?', Hein?', 'I cannot parse this'.

talking sense about implementation when I wasn't. Peter Ingwersen for reading much of this work in paper form, for his kind encouragement and, especially, for *wanting* to read this thesis. Jane Reid for much common sense, laughs and muffins. Anne Sinclair for keeping everything ticking along nicely.

My parents and brother have been a huge source of emotional support for which I thank them greatly.

Last, but never least, John Rooney deserves the biggest thanks for all his encouragement and support before, during and after this thesis.

Abstract

Selecting good query terms to represent an information need is difficult. The complexity of verbalising an information need can increase when the need is vague, when the document collection is unfamiliar or when the searcher is inexperienced with information retrieval (IR) systems. It is much easier, however, for a user to assess which documents contain relevant information.

Relevance feedback (RF) techniques make use of this fact to automatically modify a query representation based on the documents a user considers relevant. RF has proved to be relatively successful at increasing the effectiveness of retrieval systems in certain types of search, and RF techniques have gradually appeared in operational systems and even some Web engines. However, the traditional approaches to RF do not consider the behavioural aspects of information seeking. The standard RF algorithms consider only what documents the user has marked as relevant; they do not consider how the user has assessed relevance. For RF to become an effective support to information seeking it is imperative to develop new models of RF that are capable of incorporating how users make relevance assessments.

In this thesis I view RF as a process of explanation. A RF theory should provide an explanation of why a document is relevant to an information need. Such an explanation can be based on how information is used within documents. I use abductive inference to provide a framework for an explanation-based account of RF. Abductive inference is specifically designed as a technique for generating explanations of complex events, and has been widely used in a range of diagnostic systems. Such a framework is capable of producing a set of possible explanations for why a user marked a number of documents relevant at the current search iteration.

The choice of which explanation to use is guided by information on how the user has interacted with the system – how many documents they have marked relevant, where in the document ranking the relevant documents occur and the relevance score given to a document by the user. This behavioural information is used to create explanations and to choose which type of explanation is required in the search. The explanation is then used as the basis of a modified query to be submitted to the system.

I also investigate how the notion of explanation can be used at the interface to encourage more use of RF by searchers.

Table of contents

Table of equations

Table of figures

Table of tables

Part I

Introduction

Chapter One

Introduction and background

1.1 Introduction

Information retrieval (IR) systems allow users to access large amounts of electronically stored information objects. A user submitting a request to an IR system will receive, in return, a number of objects that potentially provide information relating to her request. These objects may include images, pieces of text, web pages, segments of video or speech samples.

A number of features distinguish IR systems from other information access tools. For example, an IR system does not extract information from the objects that it accesses. Neither, typically, does it process information contained within these objects. This separates IR systems from knowledge based systems such as expert systems, conceptual graphs or semantic networks. These knowledge-based tools depend heavily on a pre-defined representation of a domain, such as medicine or law. This domain knowledge can be used to manipulate, infer or categorise information for a user. Instead, IR systems are used to direct the user to objects that may help satisfy a need for information.

The data accessed by IR systems is usually unstructured, or at best semi-structured. The requests submitted to IR systems are generally also unstructured. Whereas a database system will be used to answer requests such as "*How many female members of parliament are there in the British Parliament?"* or *"Which British MPs are women?",* IR systems will be used to answer requests such as *"What are the main causes of the poor representation of women in UK politics?"* or *"In what ways are the British political parties attempting to increase the number of female MPs".* IR systems are intended to deal with requests that do not necessarily specify a unique, objective answer.

The process of information retrieval is an inherently *uncertain* one. Searchers may not have a developed idea of what information they are searching for, they may not be able to express their conceptual idea of what information they want into a suitable query and they may not have a good idea of what information is available for retrieval.

Early in the field, researchers recognised that, although users had difficulty expressing exactly the information that they required, they could recognise useful information when they saw it. That is, although searchers may not be able to convert their need for information into a request, once the system had presented the user with an initial set of documents the user could indicate those documents that did contain useful information.

This lead to the notion of *relevance feedback* (RF) - users marking documents as *relevant* to their needs and presenting this information to the IR system. The system can then use this information quantitatively - retrieving more documents like the relevant documents - and qualitatively - retrieving documents similar to the relevant ones before other documents.

The process of RF is usually presented as a cycle of activity: an IR system presents a user with a set of retrieved documents, the user indicates those that are relevant and the system uses this information to produce a modified version of the query. The modified query is then used to retrieve a new set of documents for presentation to the user. This process is known as an *iteration* of RF.

The mechanism by which an IR system uses the relevance information given by the user is the main focus of this thesis. The thesis covers several aspects of RF: the representations used in RF, how these representations lead to deciding how to modify a query and the role of interaction in RF. Before I introduce the specific contributions of this thesis in Chapter Two, I shall use the remainder of Chapter One to outline the main approaches to RF within IR.

Section 1.2 presents a discussion of the retrieval process as a whole and outlines how RF has been incorporated into the major retrieval models. In section 1.3 I discuss extensions and modifications to the traditional models of RF and I summarise the discussion in section 1.4.

Historically, most RF approaches have been based on *automatic* techniques for modifying queries. More recently, a number of researchers have examined the role of the user in RF and have presented techniques designed to increase the interaction between the user and system in RF. These *interactive* techniques are the main topic in sections 1.5 and 1.6. In section 1.7 I examine some of the important aspects of user involvement that are important to RF, and I conclude this overview in section 1.8.

1.2 The information retrieval process

The IR process is composed of four main technical stages. The first stage, *indexing* the document collection, during which the documents are prepared for use by an IR system, is

discussed in section 1.2.1. Document *retrieval*, the process of selecting which documents to display to the user, is described in section 1.2.2. The *presentation* of retrieved documents and the *evaluation* of the retrieval results are discussed briefly in sections 1.2.3 and 1.2.4 respectively. In the section on retrieval I shall outline the basic approaches to RF in the major retrieval models. In section 1.2.5 I shall summarise the difference between these main approaches to RF.

1.2.1 Indexing

For small collections of documents it may be possible for an IR system to assess each document in turn, deciding whether or not it is likely to be relevant to a user's query. However, for larger collections, especially in interactive systems, this becomes impractical. Hence it is usually necessary to prepare the raw document collection into an easily accessible representation; one that can target those documents that are most likely to be relevant, for example those documents that contain at least one word that appears in the user's query.

This transformation from a document text to a *representation* of a text is known as *indexing* the documents. There are a variety of indexing techniques but the majority rely on selecting good document descriptors, such as keywords, or *terms*, to represent the information content of documents. A 'good' descriptor for IR is a term that helps *describe* the information content of the document but is also one that helps differentiate the document from other documents in the collection. A 'good' descriptor, then, has a certain *discriminatory* power[2]. This power of a term in discriminating documents can be used to differentiate between relevant and non-relevant documents, as will be discussed in the section on retrieval.

Figure 1.1 outlines the basic steps in transforming a document into an indexed form. The first stage is to convert the document text (**Document text**, Figure 1.1a) into a stream of terms, typically converting all the terms into lower case and removing punctuation characters (**Tokenisation**, Figure 1.1**b**).

Once the document text has been indexed it is necessary to decide which terms should be used to represent the documents. That is, we need to decide which descriptors are useful for the joint role of describing the document's content and discriminating the document from the other documents in the collection.

---

[2]See [VR79], Chapter 2, for a more detailed explanation of the trade-off between the descriptive and discriminatory power of terms.

Very high frequency terms, ones that appear in a high proportion of the documents in the collection, tend not to be effective either in discriminating between documents or in representing documents. There are two main reasons for this.

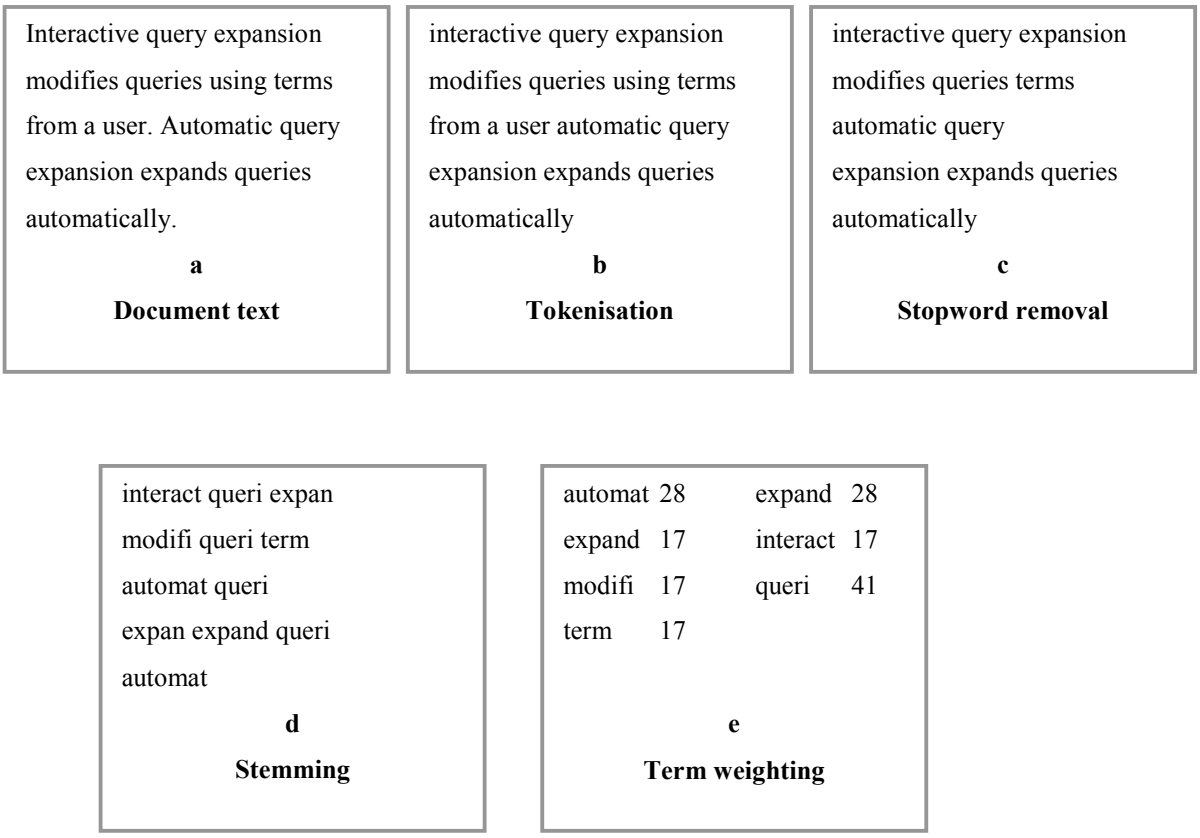

Figure 1.1: Indexing a document

The first is that, for the majority of realistic user queries, the number of documents that are *relevant* to a query is likely to be a small proportion of the collection. A term that will be effective in separating the relevant documents from the non-relevant documents, then, is likely to be a term that appears in a small number of documents. Therefore high frequency terms are likely to be poor at discriminating

The second reason is related to the notion of *information content*. A term that can appear in many contexts, such as prepositions, are not generally regarded as *content*-bearing words; they do not define a topic or sub-topic of a document. The more documents in which a term appears (the more contexts in which it is used) then the less likely it is to be a content-bearing term. Consequently it is less likely that the term is one of those terms that contributes to the user's relevance assessment. That is, terms that appear in many documents are less likely to be the ones used by a searcher to discriminate between relevant and non-relevant documents.

A common indexing stage is, then, to remove all terms which appear commonly in the document collection, and which will not aid retrieval of relevant material, (**Stopword removal**, Figure 1.1**c**). The list of terms to be removed is known as a *stop-list*; these can either be generic lists, ones that can be applied to most collections, e.g. [VR79], or lists that are specifically created for an individual collection. A term does not have to appear in the majority of documents to be considered a stop term. For example, in [CRS+95] the removal of all terms that appeared in more than 5% of documents did not significantly degrade retrieval performance in a standard IR system.

Terms may appear as linguistic variants of the same word, e.g. in the example in Figure 1.1, the terms *queries* and *query* are the plural and singular of the same object and the terms *expansion* and *expand* refer fundamentally to the same activity. As most IR systems rely on functions that *match* terms (see section 1.2.2) to retrieve documents, this variation in word use could cause problems for the user.

For example, if a user enters a query '*hill walks*' then an IR system will retrieve all documents that contain the term '*walks*' but not documents containing '*hill walking*', '*hill walk*' or '*hill walker*', any of which may contain relevant information. To avoid the user having to instantiate every possible variation of each query term, many indexing systems reduce terms to their root variant, a process known as *stemming*, [Por80] (**Stemming**, Figure 1.1**d**)[3].

The result of the indexing process, so far, is a list of low to medium frequency terms that represent the information content of the document and help discriminate the document from other documents. This information can be included in a file containing the information on all the document collection, known as an *inverted file*, Figure 1.2. In this file each line consists of information on one of the terms in the collection; in this example we have the term (*automat*), followed by a series of document identifiers.

automat	1	2	3 ....
expan	1	4	6 ....
expansion	1	17	46 ....
...			

Figure 1.2: Inverted file with no term weights

[3]I shall continue to refer to stemmed terms as terms for ease of description.

The final stage in most IR indexing applications is to weight each term according to its importance, either in the collection, in the individual documents or some combination of both, (**Term Weighting**, Figure 1.1**e**). Two common weighting measures are inverse document frequency (*idf*), [SJ72], and term frequency (*tf*), [Har92a]. *idf* (or as it is sometimes referred to, inverse collection frequency) weights a term according to the inverse of its frequency in the document collection: the more documents in which the term appears, the lower *idf* value it receives, Equation 1.1. The *idf* weighting function, then, assigns high weights to terms that have a high discriminatory power in the document collection.

$$idf(t) = \ln\frac{N}{n}$$

Equation 1.1: Inverse document frequency
where N = number of documents in the collection
n = number of documents containing the term t

Term frequency, or *tf*, measures (see [Har92a] for an overview) assign larger weights to terms that appear more frequently within an individual document. Unlike the *idf* value, the *tf* value of a term is dependent on the document in which it appears, Equation 1.2. The *tf* weighting function assigns high weights to terms that appear more frequently within a document.

$$tf_d(t) = \frac{\ln(occs_t)}{\ln(length_d)}$$

Equation 1.2: Term frequency
where $length_d$ = the number of terms in document d
$occs_t$ = number of occurrences of term t in document d

Term weighting information can be also be included in the inverted file; in Figure 1.3 we have the term (*automat*), its *idf* value (*36*), followed by a series of tuples of the form <document identifier, *tf* value>

automat	36	<1, 28> <2, 14> <3, 28> ....
expan	14	<1, 28> <4, 15> <6, 29> ....
expansion	11	<1, 17>...
...		

Figure 1.3: Inverted file with *idf* and *tf* weights

The inverted file is the main data structure of most IR systems and its use means that the IR system can easily detect which documents contain which query terms. Stopword removal and stemming reduce the size of the inverted file and increase the efficiency of the system.

Although indexing makes it possible to access information from very large document collections, the conversion from a document *text* to a list of weighted keywords does result in a loss of information. Writing a document is an intentional process; a document is intended to convey a message. The translation to a list of keywords retains the essential building blocks of the message, the terms themselves, but the message(s) that the author intended cannot be accessed by the retrieval mechanism. The effect of this loss of information may be ameliorated or deteriorated by the use of controlled vocabularies - pre-defined sets of indexing terms, [Ing92, Chap 3]. However, the fact remains that when we talk of representing the information content of documents we are only representing the *components* of the message, not the message itself.

The reduction of the document text into a series of keywords also transforms the task of an IR system from retrieving *information* to retrieving *objects* that contain information. Some authors argue that objects such as documents cannot be held to contain information as such, rather information is a change in a cognitive, or internal, state brought about by exposure to the contents of these objects. The following early quote by Maron, [Mar64], illustrates this concern,

> "..information is not a *stuff* contained in books as marbles might be contained in a bag - even though we sometimes speak of it in that way. It is, rather a *relationship*. The impact of a given message on an individual is *relative* to what he already knows, and of course, the same message could convey different amounts of information to different receivers, depending on each one's internal model or map."

The degradation of the document text, necessary for computation, and the subjectivity of relevance results in a layer of indirection between the user and the documents. The goal of the IR system is to bridge this gap between the user and potentially relevant material.

Indexing techniques identify and highlight potentially good indicators of relevant material, and retrieval techniques use these indicators of relevance to select which documents to present to the user. *How* individual retrieval systems use these indicators to retrieve documents is the topic of the next section.

1.2.2 Retrieval and feedback

Retrieval is the process of *matching* a representation of an information need, usually a user-supplied *query*, to an indexed document representation. Queries will be indexed in the same way as a document and compared with a document index to determine if a document is likely to be relevant to a query.

How the indexed query is compared with the indexed document differentiates the major retrieval models. In Appendix A I give a detailed discussion of the four main models of retrieval: *Boolean*, *vector-space*, *probabilistic*, and *logical*, and describe the basic approaches to RF in each of the models. In this section I shall summarise the major differences in retrieval and RF in the models.

i. Boolean model. The Boolean model, [FBK+92], is an exact match model: documents are only retrieved if they exactly match the user's query formula. For example the query '*information* AND *retrieval'* will only retrieve documents that contain both terms *indexing* and *retrieval.* Relevance feedback in Boolean models typically consists of suggesting new query terms to the user or altering the Boolean connectives, e.g. AND, in the query, [Har92a].

ii. Vector-space and probabilistic models. These models are best-match models: they provide the user with documents that best match the user's query. This means that the retrieval system may retrieve documents that only contain some of the user's query terms. Best-match models typically *rank* documents; they use term weighting schemes such as *tf* and *idf* to assign each document a retrieval score. This allows the system to present the user first with the documents most likely to be relevant to the user's information needs. RF in best-match models typically consists of two stages: adding new terms to the query (query expansion) and reweighting query terms. The second stage assigns new weights to each query term to reflect how good the term is at discriminating relevant and non-relevant documents. The new weights will be used in place of *tf* and *idf* to score documents for retrieval.

iii. Logical model. The logical model is also based on a best-match principle. In this case, however, the retrieval mechanism is one of inference: inferring how likely the information contained within the document is to be relevant to the query. RF in logical models can take many forms, Appendix A, some of these can involve changing the inference rules used by the system: changing *how* documents are retrieved rather than simply the content of the query.

1.2.3 Presentation of retrieved documents

A lengthy discussion of interfaces to IR systems will not be given at this point. Unless otherwise stated I shall assume that retrieved documents are presented either as a list (best-match) or set (exact-match). Hearst, [Hea99], discusses the wide range of graphical and visualisation techniques that have been suggested for IR systems. Interfaces designed specifically for RF will be discussed in more detail in section 1.6.

1.2.4 Evaluation of retrieval systems and relevance feedback

I will now discuss the evaluation of IR systems and RF. The most common evaluation tool for IR systems is a *test collection*. This is a set of documents, a set of queries and a list of which documents are considered relevant for each query. The list of documents assessed as being relevant for each query are known as the *relevance assessments*. Test collections are primarily used for comparative evaluation: comparing the performance of two systems, or two versions of the same system on the same set of queries.

Two standard evaluation measures are commonly used with test collections: *precision* and *recall*. Recall is measured as the ratio of relevant documents retrieved to the number of relevant documents in the collection. Precision is the ratio of relevant documents retrieved to the number of documents retrieved. In Appendix B I give a more detailed discussion of how recall and precision are used to evaluate IR systems and the specific modifications that are necessary to evaluate RF algorithms. For the majority of the results presented in this thesis I shall use the full-freezing method of evaluation, [CCR71], Appendix B. This is a means of using recall and precision to evaluate RF algorithms to allow comparative evaluation.

1.2.5 Summary of RF

In this section I shall summarise outline some of the major issues in the core RF models. In section 1.2.5.1 I shall summarise the comparison between Boolean and best-match models, in section 1.2.5.2 I shall compare the types of best-match model, and in section 1.2.5.3 I shall compare the two main components of RF – query term reweighting and query expansion.

1.2.5.1 Boolean vs Best-match

Although Boolean models are still popular and have strong advocates, e.g. [FST+99], in general there are many advantages to best-match models over exact-match models. The first advantage is that the user does not need to generate a query expression in the same way as with the Boolean model. Instead they can enter a natural language expression. This means that users can initiate retrieval sessions without knowledge of the collection, previous searching experience or experience in creating Boolean queries.

A second difference is that ranking documents allows the users to interact in a more meaningful fashion with the system, [Beau97]; documents are presented in order of match and documents are not excluded if they miss out elements of the query.

Thirdly the system can automatically alter a query through RF. The main strength of best-match models is that they allow for *iterative* improvement, often using similar techniques to retrieve documents as to modify queries. The strength of ranking models for RF is that, after initial querying, the user can interact without further *describing* the information for which they are searching. The RF algorithms discussed in the main body of this chapter deal almost exclusively with best-match algorithms. In the next section I shall look at the relative performance of the best-match models discussed previously.

1.2.5.2 Relative performance of best-match models

In [SB90] Salton and Buckley investigated the relative performance of 12 feedback algorithms on six standard test collections[4]. These algorithms were based on the vector space and probabilistic models for RF and are discussed in Appendix A.

Salton and Buckley found that, for all collections, except the NPL collection[5], the models performed fairly consistently with respect to each other, with the vector space Ide-dec-hi algorithm performing best overall. In general, although the probabilistic model performed well, it did not quite reach the performance level set by the vector space models. This was advantageous as the vector space Ide-dec-hi RF technique is computationally very efficient.

Salton and Buckley also provide some general guidelines based on predicting RF performance. For example, short queries, on the whole, do better with RF than longer queries. Longer queries, or those queries with more terms that appear in the relevant documents, will tend to achieve better initial rankings. This means that there is greater *potential* improvement to be gained from RF on short initial queries. For a similar reason queries that do poorly on initial runs tend to obtain greater improvements with RF than those with good initial retrieval runs

---

[4] CACM, CISI, Cranfield, Inspec, MEDLARS and NPL collections. These are relatively short document collections ranging from 1, 033 documents (MEDLARS) to 12, 684 documents (INSPEC).

[5]The NPL collection differed in a number of ways from the other collections investigated. It had much shorter query and document vectors, and lower term frequency. For this collection, although the same relative ordering was found between algorithms, binary document weighting was better than weighting document terms. This may result in the vector-space normalisation procedure being ineffective for this collection.

Finally, domain-specific collections also perform better with RF than domain-independent collections. This may be because it is easier to select good expansion terms from a domain-dependent collection, or because the ambiguity of search terms is less significant.

As well as considering variations on the probabilistic and vector space models Salton and Buckley investigated weighting document terms (as opposed to binary weighting based on term presence/absence in each document) and three variations on query expansion - no expansion (only reweighting), full expansion by all the terms in the relevant documents and partial expansion, adding only some of the relevant terms to the query. For all collections, again except the NPL, weighting document terms gives a considerable improvement in feedback, as does full expansion by all terms in the relevant set[6]. Queries should be expanded by those terms that appear with the highest frequency in the relevant documents rather than those with the highest feedback weight.

Rocchio's original formula vector-space RF algorithm and the Ide-dec-hi variant, perform the joint function of modifying query terms and query term weights. These and the other vector space RF techniques use the original document term weights to calculate the new term weights for query terms. The probabilistic-based F_4 weights, on the other hand, are derived directly from the feedback process itself. The traditional probabilistic version presented in Appendix A, section A.3 however, ignores the frequency with which a term appears in the query and in documents. This latter feature has been extended in [RW94].

Harman, [Har92b], section 1.2.5.3, and Salton and Buckley, [SB90], both showed that query expansion and query term reweighting are essential to RF.

Salton and Buckley's experiments were carried out in an experimental setting. In such a setting, especially with smaller test collections such as the CACM, Cranfield, and NPL, we can assume complete relevance information; that we know all the relevant documents for a query. However in a real information-seeking situation, users will not necessarily assess every retrieved document; often they may only assess a small number of documents, before trying RF. This could be significant as a standard assumption in operational systems is to assume all documents that are not explicitly marked relevant should be treated as non-relevant.

Sparck Jones, [SJ79], ran a set of experiments to test how well the probabilistic F_4 weighting scheme performed with little relevance information and demonstrated that even very few

[6]Although full expansion is preferable, partial expansion also gives good results and can be used to reduce storage.

relevance assessments, as few as one or two relevant documents can still improve a search over no term weighting.

1.2.5.3 Query expansion vs term reweighting

In [Har88, Har92b] Harman examined the relationship between query expansion and reweighting in the probabilistic model. As the original probabilistic model did not incorporate the addition of new terms to the query, it is important to make sure that best possible terms are added. One obvious solution is to add all terms in the relevant documents but Harman hypothesised that improved performance could be obtained by ranking these terms and adding only a number of them to the query. This raises two questions both examined in [Har88]: how to rank the terms, and how many terms to add to the query?

In [Har88] she examined six techniques for ranking terms, and demonstrated on the Cranfield 1400 test collection, that adding between 20 - 40 terms much improved performance over adding all terms with a peak at around 20 terms. The best technique for ranking the terms was one that combined *idf*-like information and frequency of term occurrences in relevant documents.

In [Har92b] she extended this work, on the same document collection, using a set of new algorithms for term ranking, and reinforced the suggestion of adding around 20 terms to the query[7]. She also explored the relationship between query expansion and term reweighting: query expansion *and* reweighting of query terms gave increased performance, with the major benefit coming from query expansion component rather than reweighting.

[Har92b] also explored a number of alternative methods for ranking terms. The details of these new algorithms are not significant here but what is important to note is that, although the improvements of certain of these techniques were similar, the terms they added to the query we not identical. This means that different algorithms may present different documents to the user based on the same relevance assessments. One possible way to exploit this is to combine methods for RF as in section 1.3.2. An alternative is to allow the user to make the choice of which terms to add to the query, which is discussed in section 1.5.

In this section I have outlined basic operations of IR systems and how RF is implemented in the major retrieval models. In the remainder of this chapter I shall discuss extensions to these models to incorporate aspects such as changing information needs (section 1.3). I shall

[7] Experiments carried out by Magennis and Van Rijsbergen [MVR97], and in this thesis, Chapter Nine, indicate that the optimal number of expansion terms for a test collection can vary between collections and query sets.

summarise the overall features of *automatic* RF in section 1.4 and turn to the interactive aspects of RF in sections 1.5 – 1.8.

1.3 Extensions to RF

The two sections that follow all extend, rather than challenge, the RF techniques discussed previously. In section 1.3.1 I describe how to incorporate the fact that what a user finds relevant may change over time and in section 1.3.2, I discuss combination of evidence in RF.

1.3.1 The dynamic nature of information seeking

Implicit to much of the early work on RF is the assumption that users have a fixed information need: that the information for which they are searching does not change over the course of a search. Whilst this may be true in certain cases, evidence from a range of studies on information seeking, e.g. [Kuh93, Ell89, SW99], show that information needs should be regarded as transient, developing entities rather than a fixed request.

The techniques discussed previously modify queries based on the difference between relevant and non-relevant documents but they do not consider *when* a document was marked relevant: a document marked relevant at the start of a search contributes as much to RF as a document marked relevant at the current iteration. If we assume that user's information needs are static then this is correct. However if the user's need is developing or changing throughout the search, then documents which were assessed as relevant early in the search may not be good examples of what the user *currently* regards as relevant. Campbell, in a series of papers on developing information needs, has addressed this issue through the notion of *Ostensive Relevance*, [Cam95, Cam99, CVR96].

The basic premise behind Ostensive Relevance, [Cam95], is that documents selected at the current iteration of RF are the best indicators of what the user finds relevant; documents assessed as relevant in previous iterations are decreasingly useful at describing a user's information need.

Relevant documents, then, are not seen as a set of *equally* important documents but sets of documents of *varying* importance. In [CVR96] Campbell and Van Rijsbergen produce an extension to the probabilistic model of retrieval that incorporates an 'ageing' component to term weighting. When calculating the weight of a term this ageing component incorporates when the documents containing the term were assessed relevant: if the documents were marked relevant at an early stage in the search then the term receives a lower weight than if the document was assessed relevant in recent iterations. The ageing component can be tuned

to differentiate more or less strongly between older and more recent documents. In [Cam99] a preliminary test of this approach indicated that ostensive weighting can improve searches in fewer search iterations that non-ostensive approaches.

Standard RF techniques, such as Rocchio, [Roc71], or F_4, [RSJ76], will also adapt to changing information needs but they will require more evidence to do so as they will require an accumulation of new evidence to outweigh the old evidence. Campbell's ageing component reduces this mass of evidence required to shift a query towards the new information need. Relevance information is used to alter the importance of the document descriptors. In particular recency information is used to increase the importance of recently visited descriptors and lower the importance of descriptors visited earlier in the search.

Dynamic information needs also present a new problem for evaluation. If we assume a changing information need we can no longer rely on existing test collection methods as they also rely on the notion of a fixed information need. The assessment of recall in an interactive situation is especially problematic, as the desired set of relevant documents[8] will change from one search iteration to another.

One further problem of RF evaluation in this context is what to measure: the quality of the feedback (how well does the system improve the user's query) or the quality of the adaptation to the information need (how well does the algorithm track how the query is changing)? These are not necessarily the same entity: potentially a RF algorithm could be good at describing the known relevant documents but poor at detecting how the user's relevance assessments are changing.

1.3.2 Combination of evidence in RF

Many of the RF and retrieval techniques described so far have utilised a single query representation compared against a series of single document representations, using one retrieval algorithm.

Many researchers have argued that better retrieval effectiveness may be gained by exploiting *multiple* query representations, retrieval algorithms or feedback techniques and *combining* the results of a varied set of techniques or representations. Several researchers have examined approaches to multiple query representation, [BKF+95, HC93], multiple retrieval algorithms, [Sim96, Sme98], and multiple feedback algorithms, [Lee98].

---

[8] That is the set of documents that the user would regard if shown them at the current iteration, not the set of relevant documents used for feedback.

Combination of evidence has the potential to be a powerful technique for RF. However, the majority of techniques attempted have shown that combination of evidence is a very *variable* technique. It will improve some queries but degrade the performance of others. In addition, it is also very difficult to predict what evidence to combine for different collections or queries.

1.4 Summary of automatic techniques for relevance feedback

In this section I summarise the work on automatic RF techniques. It is clear from the vast majority of work on automatic query modification that can prove an effective, practical solution for improving the quality of on-line searching and it has been demonstrated to work well under a number of conditions. In particular, it is a very useful technique for improving the performance of short queries or queries which provide poor initial rankings.

The basic approach of reweighting and expanding queries using terms drawn from the relevant documents works well with the major contribution often coming from the expansion component of the query modification, [SB90], although this may be collection dependent.

Although there has been a large volume of theoretical work on RF, in the foundations to the probabilistic model for example, there remains a number of basic questions for which there are only heuristic solutions. For example, if we choose to add only a number of terms to the query, how should we choose how many terms to add? Similarly, how should we rank terms to give an optimal list of expansion terms? Functions such as F_4 which order terms by their discriminatory power are typically used for this purpose but the actual performance given by these functions, and by query expansion in general, is variable and is affected by collection, query and retrieval system used. Although the probabilistic model, Appendix A section A.3, gives a strong theoretical basis for ranking documents after relevance information has been provided, there is a lack of theoretical evidence to predict what makes a good set of expansion terms for a given collection-query-system combination.

One way round this problem is to involve the user in the process of modifying the query. In section 1.1 I argued that one of the benefits of RF is that it requires minimal effort from the user - a user only has to *identify* relevant material not *describe* it. However we may gain a better representation of what material is likely to be relevant if we allow the user more control over the term selection process and also if we pay more attention to the tasks a user is trying to achieve with a system. These interactive aspects of RF are the topic of the next section.

1.5 Interactive query modification

All the methods for query modification described previously *automatically* extract terms from documents and add some or all of them to the query. A natural alternative is to allow *users* to select the terms to be added - *interactive query expansion* (IQE). The user, who has the best insight for determining relevance, then has more control over which terms are added to the query. The strength that is claimed for IQE is that the user can select better query expansion terms than the system.

In this section I shall look at the basic research on IQE, section 1.5.1, examining how terms should be ranked for presentation to the user, section 1.5.2, and the effectiveness of IQE against automatic query expansion (AQE), section 1.5.3.

1.5.1 Fundamentals of IQE

In addition to the ranking functions described in section 1.2.5, Harman, [Har88], investigated the possible effectiveness of an interactive approach to query expansion. The experiments she carried out were designed to test how effective query expansion *could* be if the user selected expansion terms from a list of terms that were pre-selected by the system.

She performed an initial experiment, on the Cranfield 1400 test collection, in which a variable number of possible expansion terms[9] were added to the query. This experiment gave two main conclusions. First, she found that different methods of sorting the expansion terms gave different performance: some methods for sorting terms were better than other methods. Second, and more importantly for IQE, the performance of query expansion varied according to how many terms were added to the query. For the Cranfield 1400 collection, expansion by 20 terms gave optimal effectiveness.

She performed a further experiment in which the system selected expansion terms from a list of those terms that occurred in at least one of the *unseen* relevant documents. This simulated a 'perfect' choice of expansion terms on behalf of the user - the system only added terms that would retrieve unseen relevant documents. This approach *(IQE-simulated*) was compared against the performance given by expansion using the top 20 expansion terms (*AQE*).

This IQE-simulated approach reduced the number of expansion terms from the 20 that were added in the AQE version to an average of 12 terms per query. Comparing AQE and IQE-simulated, Harman found that, although the AQE worked well and gave large overall

[9]With no reweighting of the query terms.

improvements in retrieval effectiveness, the IQE-simulated expansion was capable of improving these results further.

In addition, the IQE-simulated expansion was more consistent in improving performance. This latter finding was important: automatic query expansion (AQE) shows good overall performance when averaged over a set of queries but this performance increase is variable, some queries do very well with AQE others improve very little or suffer a degradation in performance. IQE as Harman deployed it, on the other hand, improves more of the queries.

Harman explored alternatives for obtaining terms for query expansion: query expansion by term variants, expansion by nearest neighbours. The first method - expanding the query by query term variant - showed little improvement when performed automatically, adding all variants of query terms. However using the 'perfect user' strategy Harman did obtain significant improvements. The second strategy - expansion by similar terms as given by co-occurrence information - also showed a drop in performance when performed automatically but an increase when performed in the simulation of a perfect user. Harman also demonstrated that *combining* query expansion techniques can further improve performance.

Harman's 1988 experiments only examined query expansion: the expansion terms were not weighted according to their utility in retrieving relevant documents. In [Har92b] she ran a series of experiments on the same collection as in [Har88], the Cranfield 1400 collection, to determine the relative effectiveness of expansion and reweighting. She showed that, on this collection at least, expanding the query is more important than only reweighting query terms. Combining both techniques will give best overall performance.

The relative merits of term reweighting and expansion may differ between collections and models but probably generally hold. She also demonstrated that multiple iterations of RF can increase performance over single iterations, so RF is useful over the course of a search.

The work on AQE demonstrated that, although RF can dramatically improve retrieval effectiveness, it is variable across queries: some queries do very well with relevant feedback, other can show degraded performance. In IQE it might be reasonable to assume that a user can improve this variability by selecting only good RF terms and ignoring the non-relevant ones. This potential benefit raises a number of questions regarding how good AQE methods are for IQE purposes. In the following sections I shall examine how ranking terms for IQE can affect performance, and the relative effectiveness of AQE and IQE.

1.5.2 Ranking expansion terms in IQE

It may be that the traditional term ranking algorithms used for AQE will perform differently when used by real subjects. That is, techniques that are successful in automatically selecting expansion terms are not suitable as a basis for a user selecting terms. One reason for this is that the reasons for a user selecting a term may not be based only on retrieval effectiveness. A user may, for example, choose fewer expansion terms due to the increased effort of term selection, or may choose terms that refine rather than modify a search topic.

$$E_{iq} = -\log\left(r_i N / R r_i\right) \bullet r_i - \log\left(\frac{(n_i - r_i)N}{(N-R)n_i}\right) \bullet (n_i - r_i)$$
$$-\log\left(\frac{(R-r_i)N}{(N-n_i)R}\right) \bullet (R-r_i) + \log\left(\frac{(N-n_i-R+r_i)N}{(N-n_i)(N-R)}\right) \bullet (N-n_i-R+r_i)$$

Equation 1.3: EMIM term weighting function
where r_i = number of relevant documents containing term *i*
R = number of relevant documents
n_i = number of documents containing term *i*
N= number of documents in the collection

Efthimiadis, [Efth93, Efth95], examined eight term ranking algorithms, and investigated their performance in an IQE environment, when users performing real searches were making the relevance assessments and term selection. Three of these algorithms (F_4, F_4.modified[10], and $w_i(p_i - q_i)$[11]) are discussed in Appendix A, section A.3. The fourth – EMIM, [VR79], incorporates term dependence information. Specifically the EMIM value assumes that index terms may not be distributed independently of each other, Equation 1.3.

The fifth - Porter's algorithm, [PG88], - is similar to the F_1 function – Appendix A, A.3, placing emphasis on frequently occurring terms in the relevant set. This is shown in Equation 1.4.

$$Porter_i = r_i / R - n_i / N$$

Equation 1.4: Porter term weighting function
where r_i = number of relevant documents containing term *i,* R = number of relevant documents, n_i = number of documents containing term *i*, N= number of documents in the collection

[10] F_4.modified is the version of the F_4 weighting function that adds 0.5 to each cell in the numerator and denominator to prevent 0 entries (Appendix A, A.3)

[11] Abbreviated, for convenience, to *wpq*, Appendix A, A.3.

The sixth algorithm - the ZOOM frequency measure, [Mar82], - ranks terms by their total frequency of occurrence in the retrieved set. All within document occurrences are also included so this measure ranks terms by the total frequency within a set of documents. Ties between equally frequent terms are resolved by ranking terms alphabetically.

The seventh algorithm, *r-lohi*, ranks terms according to their frequency of occurrence in the relevant set of documents, resolving ties by the *tf* value of the terms (low *tf* to high *tf*). The final algorithm, *r-hilo*, is identical to r-lohi except that it resolves ties by ranking from high *tf* to low *tf* value.

In the data collection section of these experiments, Efthimiadis's subjects were asked to mark all potentially useful expansion terms and the five best terms. The terms were selected from documents that the user had assessed as relevant during relevance feedback.

Efthimiadis evaluated the performance of the eight term ranking algorithms by comparing the rankings given for each query against the list generated by the users. For this, he used three criteria.

i. *comparing systems and user's ranking of term utility.* The first test looked at *where* the user-selected terms appeared in the system's ranking of terms (the top 25 terms give by EMIM, Porter, etc). Term ranking algorithms that have more user-selected terms further up the ranking are better than those algorithms that place user-selected terms further down the ranking of terms.

The most finely-grained test split the system generated list of terms into three sections (top, middle, bottom). The user-selected terms showed a distribution of 20%-30%-50% (20% of terms in bottom third of system ranking, 30% in middle third, 50% in top third) for all measures except ZOOM (with a distribution of 30%-30%-40%) and r-hilo(40%-30%-30%). The *wpq*, EMIM and r-lohi performed at very similar levels, followed by Porter, and, slightly behind, the two F_4 variants.

The same analysis was performed for the five best terms identified by the users, which showed similar results: *wpq*, EMIM and r-lohi performing best, followed by Porter, then the F_4 variants, and finally ZOOM and r-hilo.

ii. *examining top five ranked terms.* The second analysis examined the top five terms in each ranking to compare the *similarity* of the term rankings. The result showed that pairs of algorithms (*wpq* and EMIM, F_4 and F_4.modified, Porter and ZOOM) were very similar. The terms of r-lohi are similar to *wpq* and EMIM, whilst those of r-hilo are more close to those of ZOOM than anything else. In certain cases, e.g. *wpq* and EMIM, the top five terms are almost identical with only the ranking differing slightly. The major differences were between the F4 cases (mostly influenced by *n*) and the other algorithms (mostly influenced by *r* and only different is when *r* is tied).

iii. *mean of their rank position of user's five best terms.* The rank position of the users' five best terms were summed to determine which algorithms gave the best ranking of these important terms. The results (*wpq*, EMIM > r-lohi, Porter > F_4.modified >F_4 > ZOOM > r-hilo) also highlight differences between pairs of algorithms but there were no significant differences between the superior *wpq*, EMIM, r-lohi and Porter algorithms.

Each of these analyses were designed to test how good the algorithm was at ranking terms for IQE. In each case *wpq*, and EMIM performed best with Porter and the F_4 variants performing well. The ZOOM and r-hilo measures scored lowest in all cases.

These results substantiate the relative merit of the algorithms derived for AQE when used for IQE (*wpq* and F_4). They also highlight Robertson's original concern, [Rob90], Appendix A section A.3, that functions designed to measure discriminatory power of existing terms (F_4) were not necessarily the best to use in selecting new terms, as shown by the better performance of *wpq* over F_4.

1.5.3 Performance of IQE against AQE

Harman's original proposal for IQE was that user selection of expansion terms could give better performance than automatic expansion by the system. This may be true for a number of reasons. For example the system will typically base its estimate of term utility on very little relevance information which could lead to a poor set of expansion terms. A user, on the other hand, will be better able to filter out poor terms and only use those s/he feels are appropriate.

Harman, [Har88], demonstrated that selecting terms could improve retrieval effectiveness in a *simulated* case. Magennis and Van Rijsbergen, [MVR97], extended this study in two ways: by studying the *degree* to which IQE can theoretically improve performance over AQE and whether this theoretical improvement can be realised with actual users.

Magennis and Van Rijsbergen's experiments to determine the theoretical performance of IQE are based on Harman's [Har88] notion of a perfect user choice. The choice of a different test collection (the larger Wall Street Journal (WSJ) collection) necessitated repeating some of Harman's work. In particular they investigated how many terms to add[12]. They found that the range of terms, to *automatically* add to the query, to achieve optimal performance is closer to 0-10 for the WSJ than Harman's 20-40 terms for the Cranfield 1400. This shows the difficulty of predicting good estimates of numbers of expansion terms, in particular for different collections and different query sets.

Magennis and Van Rijsbergen repeated Harman's simulation experiment, which expanded the query using terms chosen from the unseen relevant documents. They ranked the 20 terms chosen from the unseen relevant documents, and added the top *n* terms. The cut-off value, *n*, was treated as an experimental variable with five values: 0 (no expansion) 3, 6, 10, and 20 (no selection of expansion terms).

For all queries, each *combination* of cut-offs was tried. AQE systems will generally expand every query by the same number of expansion terms. As a user may expand each query by a different number of expansion terms, combinations of cut-offs were used to establish the best cut-off for each query. For example, expand query one by 0 terms, expand query two by 10 terms, query three by six terms, etc. Combinations, therefore, allow the simulation of a user adding a variable number of expansion terms.

The experiment was run over four iterations of feedback and the best retrieval effectiveness was taken as the performance that could be expected by an experienced user.

The best retrieval effectiveness (precision over 100 documents retrieved) for the AQE case was achieved by adding the top 6 expansion terms. This method improved precision over automatic expansion by all 20 terms. The experienced user simulation outperformed both automatic expansion by the top 6 and by the top 20 terms. Moreover, the simulated experienced user selections improved the retrieval effectiveness for more queries: it was a more *stable* improvement over the AQE methods.

The experiment also compared the performance of the experienced user against Harman's original proposal, [Har88], of adding any term that appeared in a relevant, unseen, document. Harman's technique worked well against expansion by the top 20 terms, but only marginally better than automatic expansion by the top 6 terms, and less well than Magennis and Van

[12] Using the F_4 measure to rank terms.

Rijsbergen's approach. This supports Harman's 1992 conclusion, [Har92b], that term weighting (as was done in [MVR97] but not [Har88]) is important for query expansion.

A second experiment was run, using the same queries and same test collection, in which experimental subjects were asked to select expansion terms. This was designed to test the actual performance of IQE when relatively inexperienced users were making the term selection decisions.

The subjects could add up to 20 terms, (the default being no expansion) and were allowed four iterations of RF. The searchers were asked to assess relevance but the test collection relevance assessments[13] were used to generate expansion terms. This was to ensure that the terms used for expansion were the same for all users, and were the same as in the experienced user simulation. This aspect of the experiment was hidden from the searchers.

For all queries, the users failed to reach the potential effectiveness of the simulated user and on the whole failed even to reach the level of AQE. So although IQE *can* improve retrieval effectiveness and *can* demonstrate consistent improvement over a set of queries, the subjects in this set of experiments failed to demonstrate the ability to make good term selections. This is a vital point for IR: if IQE is to realise the experimental potential demonstrated in Harman's earlier experiments, it is necessary to facilitate the selection of good query terms.

How this process of iteratively developing a query can be made easier requires a more careful analysis of what processes users follow within IQE. I look at this in the next section.

1.5.4 Using IQE

In this section I present three investigations on user behaviour when interacting with an IQE system. The results from these investigations are not consistent. However the very lack of consistency across the experiments highlight important aspects of IQE and user interaction. They also highlight the fact that it is difficult to predict, or make assumptions, about what functionality users want from IQE or IR systems.

Beaulieu, [Beau97], as part of the ongoing work on the Okapi probabilistic system, carried out an investigation of three interfaces to IR systems. One of these only offered AQE, two offered IQE. The systems, unlike many query expansion systems, were not investigated

[13] These were the relevance assessments associated with the WSJ test collection, rather than the assessments given by the users in the course of the experiment.

through laboratory investigation but through operational investigation: the systems were used as an interface to a university library catalogue.

The first interface offered only AQE. The user was asked, for each document viewed, if the viewed document was similar to what documents s/he would like to retrieve. If the user's answer was yes, then they were offered the option of searching for similar documents. The query modification was hidden from the user; the users only saw the results of the new search. In operational trials, the uptake rate was around 33% percent (number of users trying the AQE option) and this led to retrieval of further relevant items in around 50% of the searches[14].

The first IQE system was based on a series of overlapping windows with separate windows for query, relevant titles, and the retrieved set of titles. The user was asked the same relevance question as in the AQE case ("Is this the sort of thing you are looking for? Y/N"). If the user answered yes, the document title was added to a list of titles of relevant documents. Users requested term suggestions by the use of an Expand Search button which caused the system to extract the top 20 expansion terms for display to the user. Users could then select those terms that they would like to use in a modified query.

Uptake on this system was only 11% and query expansion only led to the retrieval of further relevant documents in 31% of the searches in which users tried IQE.

The results are significant for a number of reasons, relating to both the performance and behaviour of the IQE system. The take-up rate (number of users using query expansion) and the increase in relevant documents found after query expansion were both lower in the IQE system than with AQE. Users tended to select terms very strictly, with 50% of users reporting that they found it difficult to select appropriate terms, and around 25% of users editing their original query rather than modifying their query through the IQE facility.

A third interface was developed to give the user more information on which to base their choice of term selection. A number of changes were made to the system design:

i. the overlapping windows design was replaced by a multiple pane single window design.

ii. an interactive thesaurus component was added which allowed the users to view terms related to the initial query terms.

[14] Measured by analysis of search logs.

iii. a separate working space was included to view the developing query. The source of query terms was also colour coded (initial query, IQE added query, user added query, etc.)

iv. each time the user made a relevant document selection the interface was dynamically updated to show the effect of choosing this document.

The premise behind this interface was that the user would gain more information on the effects of actions such as making relevance assessments. The uptake rate for this system was 19.5% and it led to the retrieval of further relevant items in 46% of the searches. This system had higher take-up and effectiveness rates than the first IQE interface but the figures are still lower than the AQE interface. The indication is that, although an improved interface can increase the level of use of IQE and the effectiveness of term selection, it remains an open problem how to get users to employ IQE in operational environments.

Beaulieu and Jones, [BJ98], extended this study by looking in more detail at three factors that affect interaction: functional visibility, cognitive load and balance of control between the user and system, specifically relating them to this set of experiments. The functional visibility - allowing the user more information on how the system works - is important at two levels. Not only must the user be aware of what options are available at any stage but they must also be aware of the *effect* of these options. For example, the initial IQE interface was more difficult for user as it separated the act of modifying the query and that of assessing relevance.

The cognitive load, or effort that a user must put into an action, may deter the user from trying an action that would be beneficial such as choosing more query terms. Cognitive load is also related to the notion of *control*: generally the more control the user has the higher the overall cognitive load is placed upon the user. Thus, as Bates, [Bat90], reported, the balance of control, between the system and a user, is a question not necessarily of how much control the user has but of what to give the user control over. In this context it may be preferable to use AQE as a default expansion technique, and to use IQE as an option for certain types of search or search stage, rather than use a single method of query expansion.

Fowkes and Beaulieu, [FB00], in a separate investigation, hypothesised that the complexity of the search may be an indicator of when to use AQE or IQE. Searches for which the desired information is clearly defined and for which the user can retrieve relevant information easily benefit more from AQE. Searches for vague information needs or in cases where little relevant information is being retrieved benefit more from IQE. In addition, users are more likely to employ IQE in a complex or difficult search. A related point is that users may

employ RF, either AQE or IQE, less often when the retrieval system is performing well – when it is easy to retrieve relevant information.

Belkin and Koenneman, [KB96], also investigated the use of IQE versus AQE. In this study they looked at the performance and behaviour of 64 novice users in the use of three different types of RF mechanism: completely automatic query expansion, automatic which showed the expanded query after retrieval, and interactive which allowed users to modify query before re-evaluation. They also had a no-feedback control and each user was trained on this baseline system. On the whole the findings were positive: the subjects who could control the expansion terms (the third, interactive, case) had better performance, and feedback itself gave better performance than no feedback. Users tended to choose semantically related feedback terms, and entered fewer terms manually than were suggested automatically.

This set of experiments demonstrated that interactive expansion could give positive results over automatic expansion. One particular feature of the experimental design may hold the key to the experiments' success. The task that users were given was to develop a good query for an information filtering system, 'good' in this sense meaning one which was good at retrieving relevant documents. The task the users were given, then, was one that concentrated the users' attention on the development of good queries, a situation that would lend itself to the use of techniques such as IQE. How to encourage users to develop good queries and develop more sophisticated queries does remain a difficult area as shown by Beaulieu et al.'s experiments.

Dennis et al, [DMB98], in a study looking at different types of query expansion techniques found that although users could successfully use novel expansion techniques and could be convinced of the benefits of these techniques in a laboratory or training environment, they often stopped using these techniques in operational environments. The question may be, then, can we design systems that will lead users into spending time developing queries through IQE.

1.5.5 Summary of interactive query expansion

In this section I summarise the case for IQE over AQE. The general intuition that some increased control for the user in selecting query expansion terms would be beneficial seems to be valid. Although systems have access to internal statistical information that allows them to select good discriminatory terms, users can make more informed *relevance* decision. The question is how this process of query modification should be constructed to translate the potential benefits of IQE into actual increases in retrieval performance.

There are several issues involved in this problem. The first is to decide what is the actual role of the user: should we ask the user to interactively create queries or perform an editing role on system-generated queries? How much of the query-generating process should be interactive and at what stages should we expect and desire user involvement?

Several of the reasons given by users for not using RF are also applicable to IQE, [BCK+96, RTJ01], e.g. these are time-consuming actions, the relation between cause and effect is not clear and on what principles the selection of terms should be made is not obvious.

The latter point – how terms should be chosen – is significant. It may be the case that users are better at eliminating potentially poor terms than they are at selecting good terms for query expansion. IR systems need to be able to help users make difficult decisions regarding term quality.

In the next section I shall describe interfaces that were specifically designed for RF. These interfaces are an attempt to overcome the user's reluctance to initiate RF. The success of interactive approaches to RF may, of course, not simply be a result of the interface or algorithms used by the system. For example the characteristics of the user, such as experience with on-line searching, and the search itself may affect the use and the success of more user-oriented methods of interaction. In section 1.7, I shall discuss some features of making relevant assessments that affect how people use RF in practice.

1.6 Interfaces and RF

The reluctance of users to engage in RF often comes from a poor understanding of why RF may be useful and how RF should be used in a search. This may be because RF is presented as a separate task to querying and to assessing retrieved documents. In the next two sub-sections I discuss two systems that attempt to incorporate RF as a seamless task – the process of RF is integrated into querying and assessment of documents.

The two approaches have a common underlying principle: each relevance assessment given by the user initiates a cycle of RF. The major difference between the two approaches – incremental feedback, section 1.6.1 and ostensive browsing, section 1.6.2 – is the interface design and principles.

1.6.1 Incremental feedback

Most RF systems treat the process of relevance assessment as a batch process: users are shown a set of documents and provide relevance assessments on a number of documents

before requesting RF. Aalsberg, [Aal92], proposed the alternative technique of *incremental* RF. Rather than asking a user to batch process relevance assessments by assessing a *number* of documents in a ranking, he suggests presenting only one document at a time. The user is asked to make an assessment on the displayed document before being shown the next document. With each relevance assessment made by the user, the query can be iteratively modified through feedback.

The formula used by Aalsberg simplifies the Rocchio, Ide-dec-hi and Ide-regular formulae[15] to the one shown in Equation 15.

$$Q_{i+1} = \begin{cases} \alpha.Q_i + \beta.D_j & if\ rel(D_j) \\ \alpha.Q_i - \gamma.D_j & if \neg rel(D_j) \end{cases}$$

Equation 1.5: Iterative RF

where Q_i = query for iteration i, $Q_i + 1$ = query for iteration $i + 1$,
α and γ are weights to bias retrieval in favour of the query or relevance information

This technique does not require the user to explicitly request RF, thus side-stepping the difficulty of getting users to interact. However it may not allow users to make *relative* relevance assessments, which has been shown to affect users assessments and method of making relevance assessments, e.g. [FM95, EB88]. The particular implementation also forced users to make a relevance decision. Users, however, may not always be able to decide on the relevance of a document at the time they view it.

The model was tested in [Aal92] against Rocchio's formula, the Ide-dec-hi and Ide-regular.

The model was also tested against Ide's *variable* RF, Appendix A, section A.2. This model forms a new query from the first relevant document and all preceding non-relevant documents. This is, then, analogous to the Ide-dec-hi that uses all relevant and the first, retrieved, non-relevant document, Appendix A, section A.2.

The test collection evaluation showed iterative RF can perform better than the Rocchio, and Ide-variants but performs roughly the same as variable RF.

In a separate experimental investigation Iwayama, [Iwa00], suggests that incremental relevance feedback of the form proposed by Aalsberg works better for well-specified topics. These are topics for which the set of relevant documents has a high similarity. This is because iterative feedback retrieves documents that are very similar to the ones used for feedback. It

[15] Appendix A, A.2.

does not, however, perform as well in retrieving relevant documents that cover a number of topics.

1.6.2 Ostensive browsing

Campbell's ostensive weighting technique, described in section 1.3.1, was combined in [Cam99] with a novel browsing interface, an example of which is shown in Figure 1.4.

This interface contains two features: paths and nodes. A node consists of a retrieved object. In Figure 1.3 these objects are images. Clicking on a node will cause the system to perform a RF iteration using all the objects in the path that contains the node. The top five retrieved objects are then displayed to the user, who may choose to continue the path by clicking a new object or return to a previously followed path. If a user selects more than one retrieved object, this corresponds to a diverging path: two paths with the same initial components.

Each selection of a node by a user is taken to be an implicit relevance assessment or expression of interest in the object by the user. No explicit request for RF is necessary by the user. The paths themselves correspond to multiple iterations of feedback; each object is the result of RF performed on the objects preceding it in the path. Objects may appear in different paths as the result of being retrieved in response to different RF-modified queries.

This is similar to an extent to the iterative method of RF described in the previous section in that only one additional document is added to the relevant set at each iteration. The major interface difference is that the user is not asked to make an explicit assessment of relevance or decision on the relevance of a document. The major implementational difference is that Campbell uses the ostensive weighting extension to the probabilistic model, described in Appendix A, section A.3.

The use of paths also means that RF decisions are reversible: the user can backtrack to a previously selected document at any point in the search.

One of the main aims of Campbell's work on ostension is to remove the need for a user to manipulate a query. However this also removes the *control* from the user in modifying the content of the query. A user cannot manually manipulate the query as is generally possible with the traditional RF systems. Whether or not this hiding of the IR system's functionality benefits the user or not requires further investigation.

In particular this need for further experimentation is necessary because the range of factors that lead to the success or failure of interaction with an IR system are very diverse. Many

researchers have argued that the process of retrieving relevant information is richer and more complex than the relatively simple model described so far [Bat90, Ing92, BCS+95]. In the next section I shall concentrate on one reason that IR interaction is complex: the process of making relevance assessments.

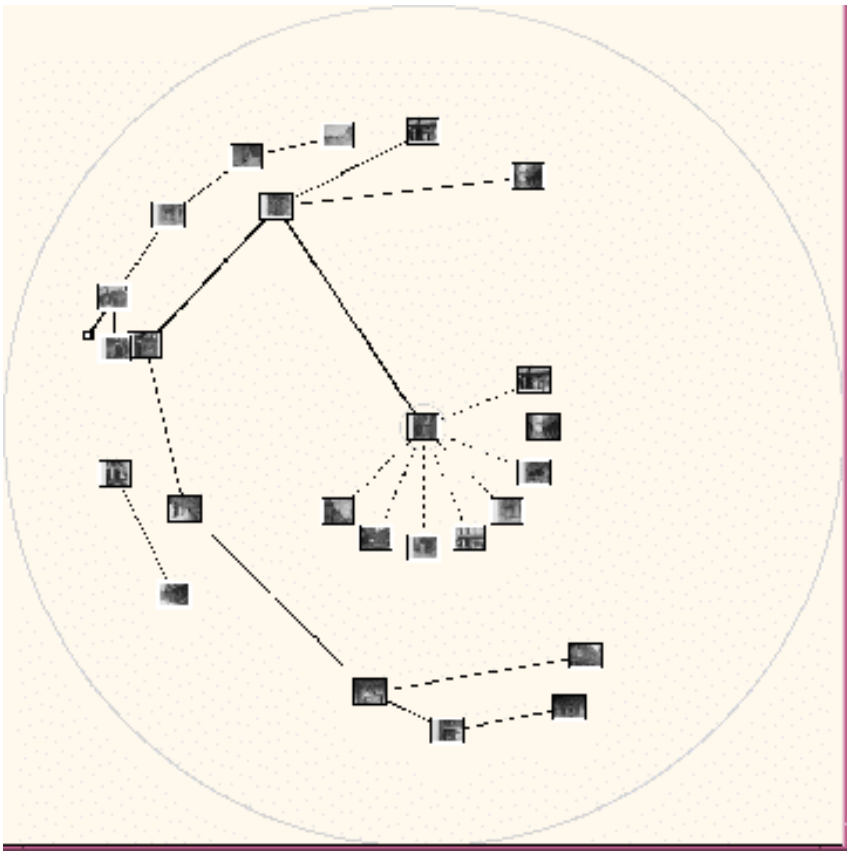

Figure 1.4: Ostensive browser interface, taken from [Cam99]

1.7 User issues

The final aspect of information-seeking I shall address, although briefly, is the process of making relevance assessments. RF algorithms require users to assess a sample of the retrieved documents but the criteria under which a user makes a relevance assessment can be subject to a number of factors. In this section, I shall introduce some of these factors.

One of the main factors is the *order* in which documents are shown to the user. Several studies, e.g. [FM95, EB88], point to the importance of the position of a document in a ranking when assessing the relevance of the document. Relevance assessments are relative: viewing one relevant document can change the user's perception of the relevance of subsequently viewed documents.

Tiamiyu and Ajiferuke, [TA88], also looked at the effect that the order in which relevance assessments are made can have on retrieval performance. They suggest three types of dependence that can exist in retrieval;

i. *independence*. Each document should be considered as an independent relevance assessments,

ii. *complementarity* relationship. The information contained within two documents sums to more than the sum of relevance ratings of each document together.

iii. *substitutability* relationship. The information in one document can substitute for the information in another document.

They show, theoretically, that the presence of different types of relationships can, although, giving same recall-precision results, give a very different result for user satisfaction. This also brings up the question of whether we should treat all relevance assessments as a single set of assessments. Draper, [Dra00], for example makes the point that users typically assess *individual* documents as relevant, not a group of documents, whereas RF systems as a set of related relevant items.

Janes, [JJ91], also demonstrates that different *representations* of documents (title, abstract, full-text) can affect relevance assessments, meaning how the document is presented can affect how likely it is to be assessed relevant.

Relevance assessments are often treated as *binary* assessments: a document is either relevant or not relevant. However, in practice, documents may be regarded as more or less relevant than each other: relevance assessments are often *partial* assessments[16].

Spink et al, [SGB98], examined relevance assessments from four separate studies of information seeking to examine the role of partial relevance assessments. In particular they looked at whether the use of partial relevance assessments correlated with other aspects of searching. The most conclusive finding was the number of partially relevant items was often positively correlated with a change in search topic or criteria for relevance: the more partial relevance assessments at a given stage in a search, the more uncertain is the user's current information need.

[16] In this context a partial assessment means a document is only somewhat relevant to the topic or the user is not sure of the document's relevance. This is distinguished from the situation where only part of the document is relevant.

This study concentrated mainly on users at the initial search stage, when information needs are more likely to be variable. However, partial relevance assessments as an indicator of search stage or search status may be useful in defining what type of documents should be retrieved. For example we may wish to increase retrieval of loosely-related material at certain stages, and suppress retrieval to only highly relevant material at other stages.

A further important factor in determining how users will make relevance assessments is the *task* the user is trying to complete. Users with different tasks will obviously mark different documents relevant, but a user with a long-running task may change their criteria for relevance over time.

Spink, [Spi96], for example, reports on a study of when and how academics use IR systems over the course of a research project. The majority of users search at the beginning of project and many search again throughout the project. One reason for searching at later stages of projects is to check new updated references - rerunning same searches against new data - but many users modify their search terms over time, either as their information problems change or they obtain information from new sources. Although the searches are similar and the basic topic of the searches are broadly the same, the reasons for searching and the type of information being sought is different leading to different relevance assessments.

Vakkari, [Vak00, Vak00b], also examined long-running searches to examine how relevance assessments changed over time. In his study he demonstrated that not only did subjects chose different documents at different stages in their task, they also used different search tactics and strategies when searching. Vakkari provided support for Spink's observation that high numbers of partial assessments correlates with a lack of ability to discriminate relevant and non-relevant. This may occur at the start of a search, for example. He also found evidence to indicate that when a user has a good idea of what constitutes relevant material he is less likely to make a high number of relevance assessments

These studies are important for RF because they point to the fact that not all relevance assessments are equal: users make assessments for different reasons and with different amounts of knowledge. A single RF approach may not be sufficient in all cases: we may need to develop RF techniques that adapt to the user's intentions.

1.8 Conclusion

RF has proved to be a useful and pragmatic solution to the uncertainty of describing an information need. It has further, in a test collection environment, been shown to be a

relatively stable procedure: it works in most cases, a wide range of algorithms give approximately the same performance and how the algorithmic parameters should be set are fairly well understood. Although I have not discussed non-text documents, such as images or speech, in this chapter the same basic principle of selecting good discriminators of relevance can be used for different media to implement RF functionality.

The conceptual simplicity of RF – users only have to recognise useful material, not describe it – neatly hides the complexity and variety of the query modification features behind the interface. However, there is a growing awareness that RF is not sufficient on its own to improve retrieval. RF is useful in that it is conceptually simple but it does not yet provide adequate support for the range of strategies and tactics demonstrated by the user in research such as [Bat90]. RF may only be part of the interaction process and will require integration with other functionalities.

Further, although RF is simple for the user to employ, the interaction decisions involved in RF can be obscure. That is, RF generally does not give the user enough context on which to based their relevance decisions, e.g. how many documents should be marked as relevant, how relevant should a document be before being marked as relevant, what does not relevant mean? Although RF research has answers to some of these questions (e.g. more relevance information is generally better), getting the user to provide the necessary input data is not easy, and making the process of assessing relevance more difficult may result in less interaction not more.

Chapter Two

Thesis outline

2.1 Introduction

In this chapter I will introduce the four main aspects of RF that form the basis of this thesis; discuss each of these in the light of the preceding discussion of IR and RF, and set out in more detail the novel contribution made to each of these areas.

The RF process can be viewed as a loop, as exemplified in the diagram in Figure 2.1.

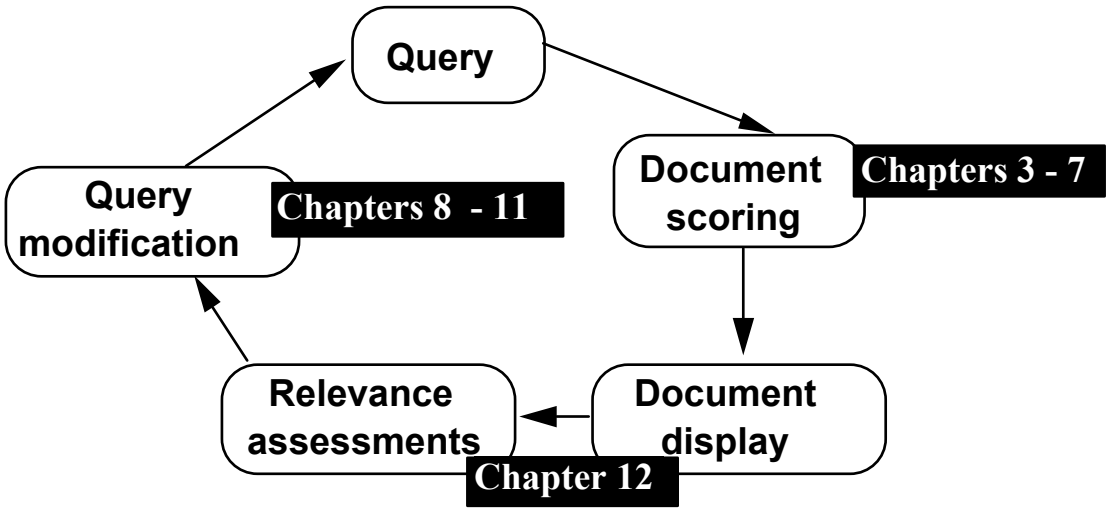

Figure 2.1: RF process

From the discussion in Chapter One, the two main tasks of a RF algorithm are the selection of good indicators of relevant material, such as indexing terms, and the appropriate weighting of these indicators to reflect their utility in attracting relevant material. For the purposes of the discussion in this chapter I will use *query modification* to refer to the process of altering the *content* of the query – the identification of good indicators of relevance, this will be discussed in section 2.4.

Document scoring will be used to refer to the process of ranking documents based on a query. This involves two sub-processes: deciding what information is used to score documents, e.g. which term weighting schemes, and deciding how to use the information to estimate the likely relevance of a document. The former process – selecting the information to be used in

document scoring – is discussed in section 2.2, the latter process – providing the document ranking – is discussed in section 2.3.

In section 2.5, I shall discuss the research completed on the presentation of RF. This aims to make RF a more accessible operation to potential users of IR systems by presenting more information on what RF decisions have been made by the system.

2.2 Representations

From the discussion in Chapter One, it can be seen that the majority of relevance feedback techniques, [Har92c, Spi96b] are based on the presence or absence of keywords in relevant documents. RF algorithms select which keywords are good at indicating relevant material – those that should be used in a new query. These algorithms also decide how important each of the keywords are in the new query. In this thesis I argue that the *presence* of a term, or indexing terms, within relevant documents is only one indication of the utility of the term. RF effectiveness can be increased by taking into account how terms are *used* within documents, rather than just their presence or absence.

This means expanding the representation of a term to allow the selection of indicators of relevance based on how a term is used. For example,

- there may be a large overlap in the content of marked relevant documents but not the structure of the documents. This may correspond to a search in which the user wants all possible information on a topic.

- there may be a high structural similarity in the relevant documents, e.g. some terms have to be the main topic of the document but may appear in a variety of contexts. This may be a search for information that the collection has a lot of information on (so the user only wants whole documents about the subject) or the user is finding his way around the collection.

- or there may be a high similarity in context, but not in content or topical relationships. This may corresponds to searches where information is only relevant in certain combinations, for example *'Information retrieval systems'* not *'Information retrieval'* or *' Information...systems'*.

By using a set of multiple term and document weighting schemes, each reflecting some aspect of a term's use or a document's structure, it is possible to *select* which weighting schemes are

good indicators of relevant material. That is we can discuss *why* a term is relevant in more detail than simply its (non-)occurrence within relevant material.

In Chapters Three and Four, I analyse this strategy. In Chapter Three I introduce the term and document weighting schemes used in this thesis. In Chapters Four and Five, by comparing the selection of weighting schemes using relevance information, I show that the selection process can give beneficial results over good combinations of weighting schemes, no feedback and alternative feedback weighting schemes. I also show that this approach is stable over user relevance assessments and simulated assessments from a test collection. In Chapter Six I show that the approach also holds when different methods of document scoring are used. In Chapter Seven I summarise the overall approach to selecting good weighting schemes for the original query terms- those chosen by a user. In Chapter Nine I show that the selection weighting schemes can also benefit terms that are chosen by the system.

2.3 Document scoring

Once the system has created a new query through RF, the query will be used to retrieve a new ranking of the document collection. If RF is operating effectively the new ranking should be better than the previous ranking; relevant documents should be placed higher in the ranking than before.

How the system uses the query to retrieve documents is important – an IR system should retrieve the documents most likely to satisfy the user's information need. In Chapter Six, I present a model for document retrieval, based on Dempster-Shafer's Theory of Evidence. Dempster-Shafer's Theory of Evidence provides a flexible framework for the representation and combination of uncertain evidence.

This model is designed to incorporate and manipulate many of the sources of uncertainty in the retrieval process. These include the degree of relevance of a document, the discriminatory power of a term and the quality of weighting schemes. The document retrieval model is expanded, in Chapter Six, to provide a RF model that also incorporates uncertain evidence.

2.4 Query modification

The process of query modification attempts to improve the user's query; either by eliminating poor query terms, or adding query terms that will assist the retrieval of more relevant material. How to select good terms for query expansion is a central aim for most of the algorithms described in Chapter One.

In this thesis I develop an approach to query modification that is based on *abductive reasoning*. This approach to query modification incorporates behavioural information, such as the degree to which the user thinks a document is relevant, or when in a search the document was assessed relevant, and uses this information to select expansion terms.

Abductive reasoning is based on the notion of *explanation*, where explanations are possible characterisations of a set of data. In this thesis, the query modification process seeks an explanation for why some documents are assessed as relevant. The process of selecting weighting schemes, outlined in section 2.2, is also an example of abductive reasoning.

In Chapter Eight I discuss the overall research goal in using abductive reasoning for IR and RF. In Chapters Nine and Ten I present an experimental investigation of the relative effectiveness of various *types* of explanation in RF and in Chapter Twelve I present the results of an investigation into the use of explanations in a user-oriented evaluation.

2.5 RF and interaction

RF *can* help a user find more relevant material but this is only a potential benefit. To realise this benefit a user must enter the feedback loop. Often, however, users do not engage in RF. There are many possible reasons for this, for example the user may not understand the purpose of RF, or the user may not know how to use RF mechanisms. In Chapter Twelve, I will present an experimental investigation of techniques to help users understand the effect of RF in a search. This is based on the abductive research described in Chapters Four, Nine and Ten, but also incorporates additional features to help introduce RF processes to the user.

2.6 Overall thesis layout

I have structured this thesis into five main sections:

Part I *Introduction*. This section is comprised of Chapters One and Two and serves as an introduction to RF and the thesis.

Part II *Information use*. This section is comprised of Chapter Three to Chapter Seven and examines the document scoring methods – using multiple term and document weighting schemes.

Part III *Abduction*. This section examines the query modification techniques and contains Chapters Eight, Nine, Ten and Eleven.

Part IV *User experiments*. This section describes the investigation of the RF techniques suggested in this thesis in a set of experiments with novice searchers. This section contains Chapter Twelve.

Part V *Conclusion*. This section contains the main conclusions of the thesis and is comprised of Chapter Thirteen.

Part II

Information Use

Chapter Three

Characteristics of information use

3.1 Introduction

Most Relevance Feedback (RF) algorithms attempt to bring a query closer to the user's information need by reweighting or modifying the terms in a query. The implicit assumption behind these algorithms is that we can find an optimal combination of weighted terms to represent the user's information need at the current stage in a search. This description of the information need is based on the indexing language(s) of the retrieval system and is intended to prioritise retrieval of those documents that are most likely to contain relevant information.

However relevance, as a user judgement, is not necessarily dictated only by the presence or absence of terms in a document. Rather relevance is a factor of what concepts the terms represent, the relations between these concepts, how users interpret the concepts and how they relate to the information in the document. From studies, such as those carried out by Barry and Schamber, [BS98], it is clear that current models of RF, although successful at improving recall-precision, are not very sophisticated in expressing what makes a document relevant to a user. Denos et al, [DBM97] for example, make the good point that although users *can* make explicit judgements on why documents are relevant, often systems cannot use this information to improve a search.

Not only are users' judgements affected by a variety of factors but they are based on the document *text*. RF algorithms, on the other hand, typically are based on a representation of a text and only consider frequency information or the presence or absence of terms in documents. These algorithms do not look deeper to see what it is about terms that indicate relevance; they ignore information on how terms are *used* within documents. For example, a document may only be relevant if the terms appear in a certain context, if certain combinations of terms occur, or if the main topic of the document is important. Extending feedback algorithms to incorporate the *usage* of a term within documents would not only allow more precise querying by the user but also allows RF algorithms to adapt more subtly to users' relevance judgements.

In this section of the thesis, Part II, I investigate how incorporating more information on the usage of terms can improve retrieval effectiveness. This investigation is based on a set of term and document weighting functions - *term* and *document characteristics* - each of which can be used to distinguish between terms or documents according to how information is used. The term characteristics are used to distinguish between *how* terms are used in collections or individual documents; the document characteristics are used to differentiate documents based on their information content and their structure. The experiments reported in Part II compare two uses of multiple term and document characteristics: combination of evidence, and *selective* combination of evidence.

In the combination of evidence experiments I examine how combining term and document characteristic information affects retrieval performance. Combination of evidence, as described in section 1.3.2, can give improved retrieval effectiveness over no combination of evidence. Selective combination of evidence, where relevance assessments are used to select which evidence to combine, is particularly designed for RF. In my experiments, selective combination of evidence is based around selecting good term and document characteristics for individual queries.

The following sections outline how I implemented term and document characteristics (section 3.2), and introduce the experiments reported in Part II.

3.2 Term and document characteristics

In this section I outline five ways of describing term importance in a document or collection - five *term characteristics*. Three of these are standard term weighting functions, *idf*, *tf* and *noise*; the other two are developed specifically for the research described in this thesis.

- *inverse document frequency*, based on how often a term appears within a collection, described in section 3.2.1
- *noise*, also based on how often a term appears within a collection but based on within-document frequency, section 3.2.2
- *term frequency*, based on how often a term appears within a document, section 3.2.3
- *thematic nature*, or *theme*, based on how a term is distributed within a document, section 3.2.4
- *context*, based on the proximity of one query term to another query term within the same document, section 3.2.5

In addition, I introduce two *document* characteristics. These describe some aspect of a document's content that differentiates it from other documents.

- *specificity*, based on how many unique terms appear in a document, section 3.2.6
- *information-noise*, based on the proportion of useful to non-useful content within a document, section 3.2.7

3.2.1 *idf*

Inverse document frequency, or *idf*, [SJ72], is a standard IR term weighting function that measures the infrequency, or rarity, of a term's occurrence within a document collection. The less likely a term is to appear in a collection the better is it likely to be at discriminating relevant from irrelevant documents. In these experiments I measure *idf* by the equation shown in Equation 3.1.

$$idf(t) = \log\left(\frac{N}{n} + 1\right)$$

Equation 3.1: inverse document frequency (*idf*)

where n is the number of documents containing the indexing term t
and N is the number of documents in the collection

3.2.2 *noise*

The second term characteristic I investigated was the *noise* characteristic discussed in [Sal83, Har86], Equation 3.2. The *noise* characteristic gives a measure of how important a term is within a collection but unlike *idf*, *noise* is based on within-document frequency.

$$noise(t) = \sum_{i=1}^{N} \frac{Freq_{it}}{TFreqt} \log \frac{TFreq_t}{Freq_{it}}$$

Equation 3.2: *noise*

where N = number of documents in the collection,
$Freq_{it}$ = the number of occurrences of term t in document i,
$TFreq_t$ = total occurrences of term t in the collection

The *noise* characteristic, taken from [Har86], shown in Equation 3.2 requires special processing for IR. An example of the calculation of term's noise values is shown in Table 3.1. This example shows the number of times a term appears within a collection, including

within-document occurrences (column 1), the number of documents in which the term appears (column 2) and the *noise* value as calculated by Equation 3.2 (column 3)[17].

total occurrences of term *t*	**number of documents containing *t***	***noise* value**	**normalised *noise* value**
100	1	0.00	23.03
100	25	2.31	20.72
100	50	7.49	15.54
100	75	14.57	8.46
100	100	23.03	0.00

Table 3.1: Calculation and normalisation of *noise* characteristic

If all the occurrences of term appear within one document, the term receives a *noise* score of zero (row 2). Terms that appear more commonly throughout a collection receive a higher *noise* value (rows 3 - 5). A term which has only one occurrence in each document in which it appears receives the highest *noise* value (row 6).

The *noise* value is then *inversely* proportional to its discriminatory power as it assigns high values to terms that have a low discriminatory power and low values to terms with a high discriminatory power. The *noise* characteristic as defined here therefore requires normalisation, [Har86], to ensure that the *noise* value of a term reflects its discriminatory power. To normalise the *noise* score, we subtracted the *noise* score of a term from the maximum *noise* score. The result of this is shown in Table 3.1, column 4, where all the values in column 3 have been subtracted from the maximum *noise* value for term *t* (23.03).

The normalised *noise* characteristic gives a maximum *noise* score to a term if all its occurrences[18] appear in one document and the lowest *noise* score if all occurrences of the term appear in different documents.

[17] For simplicity, this example assumes that the term occurrences are equally split between the documents in which a term appears. For example, if there 100 occurrences of a term and the term appears in 25 documents (Table 3.1, row 3) then I assume that the term has four occurrences in each of the 25 documents.

[18] Occurrences here refers to the tokens that represent a term, therefore a term appearing in two documents or a term appearing twice in the same document both give two occurrences of the term.

3.2.3 *tf*

Including information about how often a term occurs in a document - *term frequency* (*tf*) information - has often been shown to increase retrieval performance, e.g. [Har92a]. For these experiment I used the following formula,

$$tf_d(t) = \log(occs_t(d) + 1) / \log(occs_{total}(d))$$

Equation 3.3: term frequency (*tf*)
where $occs_t(d)$ is the number of occurrences of term *t* in document *d*,
$occs_{total}(d)$ is the total number of term occurrences in document *d*.

3.2.4 *theme*

Previous work by for example Hearst and Plaunt [HP93] and Paradis and Berrut, [PB96], demonstrate that taking into account the topical or thematic nature of documents can improve retrieval effectiveness. Hearst and Plaunt presented a method specifically for long documents, whereas Paradis and Berrut's method is based on a precise conceptual indexing of documents.

I present a simple term-based alternative based on the distribution of term occurrences within a document. This is based on the assumption that the less evenly distributed the occurrences of a term are within a document, then the more likely the term is to correspond to a localised discussion in the document, e.g. a topic in one section of the document only. Conversely, if the term's occurrences are more evenly spread throughout the document, then we may assume that the term is somehow related to the main topic of the document. Unlike Hearst and Plaunt I do not split the document into topics and assign a sub- or main-topic classification. Instead I define a *theme* value of a term, which is based on the likelihood of a term to be a main topic. The algorithm which I developed for this purpose is shown in Equation 3.4.

$$theme_d(t) = (length_d - difference_d(t)) / length_d$$

where

$$difference_d(t) = first_d(t) + last_d(t) + \sum_{i=2}^{occs_d(t)-1} |epos_i(t) - pos_i(t)|$$

$$\begin{aligned} first_d(t) &= 0, && if \quad pos_1(t) \le distr_d(t) \\ &= pos_1(t) - distr_d(t), && ow \end{aligned}$$

$$\begin{aligned} last_d(t) &= 0, && if \quad (length_d - pos_{occs_d(t)}(t) \le distr_d(t)) \\ &= (length_d - (pos_{occs_d(t)}(t) + distr_d(t)), && ow \end{aligned}$$

$$epos_i = pos_{i-1} + distr_d(t)$$

$$distr_d(t) = length_d / occs_d(t)$$

Equation 3.4: *theme* characteristic

where $distr_d(t)$ is the expected distribution of term *t* in document *d,* assuming all occurrences of *t* are equally distributed, $epos_i$ is the expected position of the *i*th occurrence of term *t*, pos_i is the actual position of the *i*th occurrence. $occs_d(t)$ is the number of occurrences of term *t* in document *d*.

The *theme* value is based on the difference between the position of each occurrence of a term and the *expected* positions. Table 3.2 gives a short example for a document containing 1000 terms and five occurrences of term *t*. First, I calculate whether the first occurrence of term *t* occurs further into the document that we would expect, based on the expected distribution ($first_d(t)$ - line three, Equation 3.4; Column 7, Table 3.2). Next we calculate whether the last occurrence of the term appears further from the end of the document than we would expect ($last_d(t)$ - line four, Equation 3.4; Column 8, Table 3.2). For the remainder of the terms we calculate the difference between the expected position of a term, based on the actual position of the last occurrence and the expected difference between two occurrences (– line two, Equation 3.4; Column 4-6, Table 3.2, $\sum_{i=2}^{n-1} |epos_i(t) - pos_i(t)|$).

length	occs	distr	epos	pos	diff	first	last	difference	*theme*
1000	5	200	-	100		0			
			300	500	200				
			700	551	349				
			751	553	547				
			753	700	600				
			900				100		
					600	0	100	700	0.3

Table 3.2: Example calculation of *theme* value for a term

I then sum these values to obtain a measure of the difference between the expected position of the term occurrences and their actual positions (-line two Equation 3.4; Column 3.2, Table 3.2). In the example this difference is 700, that is the sum of the difference between each occurrence of a term *should* appear, given an equal distribution of terms within a document, and where the terms *actually* appear. This value (700) is used to calculate the *theme* value. The greater the difference between where term occurrences appear and where we would expect them to appear, given an equal distribution of the term within the document, the smaller the *theme* value for the term. The smaller the difference, the larger the *theme* value for the term.

3.2.5 *context*

There are various ways in which one might incorporate information about the context of a query term. For example, we might rely on coocurrence information, [VRHP81], information about phrases, [Lew92], or information about the logical structures, e.g. sentences, in which the term appears, [TS98]. I defined the importance of context to a query term as being measured by its distance from the nearest query term, relative to the average expected distribution of all query terms in the document. This is shown in Equation 3.5.

$$\begin{aligned} context_d(t) &= (distr_d(q) - \min_d(t)) / distr_d(q) \\ \min_d(t) &= \min t \neq t' \left| (pos_d(t) - pos_d(t') \right| \\ distr_d(q) &= length_d / occs_d(q) \end{aligned}$$

Equation 3.5: *context* characteristic for term t in document d
where $distr_d(q)$ is the expected distribution of the query terms in the document, assuming terms are distributed equally, $pos_d(t)$ is the position of term t and $min_d(t)$ is the minimum difference from any occurrence of term t to another, different query term, $occs_d(q)$= the total occurrences of the query terms in the document

3.2.6 *specificity*

The first document characteristic I propose is the *specificity* characteristic which is related to *idf*. The *idf* characteristic measures the infrequency of a term's occurrence within a document collection; the less likely a term is to appear in a document the better is it likely to be at discriminating relevant from irrelevant documents. However, *idf* does not consider the relative discriminatory power of other terms in the document.

If a document contains a higher proportion of terms with a high *idf*, it may be more difficult to read, e.g. if it contains a lot of technical terms. On the other hand a document containing a lot of terms with very low *idf* values may contain too few information-bearing words. I propose the *specificity* characteristic as a measure of the technical complexity of the document. This is a very simple measure of technical complexity as it does not take into account the domain of the document or external knowledge sources. These would be used to represent the complexity of the document based on its *semantic* content. Rather I am attempting to define a relative notion of how specialised a document is compared to the other documents in the collection.

specificity is a document characteristic, giving a score to an entire document rather than individual terms. It is measured by the sum of the *idf* values of each term in the document, divided by the number of unique terms in the document, giving an average *idf* value for the document, Equation 3.6.

$$specificity(d) = \frac{\sum_{i \in d}^{n} idf(i)}{n}$$

Equation 3.6: *specificity* document characteristic of document *d*
where *n* = number of terms in document *d*

3.2.7 *information-to-noise*

The *specificity* characteristic measured the complexity of the document based on *idf* values. An alternative measure is the *information-to-noise* ratio, suggested by Zhu and Gauch, [ZG00], abbreviated to *info-noise*. This is calculated as the number of tokens after processing (stemming and stopping) of the document divided by the length of the document before stopping and stemming, Equation 3.7.

$$info\_noise(d) = \frac{processed\_length(d)}{length(d)}$$

Equation 3.7: *info_noise* document characteristic of document *d*
where *processed_length*(*d*) = number of terms in document *d* after stopping and stemming
length(*d*) = number of terms in document *d* before stopping and stemming

info_noise, as described in [ZG00], measures the proportion of useful to non-useful information content within a document.

3.2.8 Summary

The *idf* and *noise* characteristics give values to a term depending on its importance within a collection, the *tf* and *theme* characteristics give values depending on the term's importance within individual documents and *context* gives values based on the relative position of other query terms in the individual documents. The *specificity* and *info_noise* characteristics give values to individual documents based on their content.

Each of the term characteristics can be used to differentiate documents based on how a term is used within the documents and the document characteristics allow differentiation of documents based on their content. The document characteristics also allow retrieval algorithms to base retrieval decisions on the document taken as a whole, rather than only individual components of the document.

Each of the algorithms that calculate the characteristic values give scores in different ranges. In my experiments I scaled all values of the characteristics to fall within the same range, 0 - 50, to ensure that I was working with comparable values for each characteristic.

3.3 Outline of experiments

In this section I give a brief outline to the experimental investigation reported Part II. A more detailed introduction will be given at the start of each chapter.

Chapter Four examines the basic approach of combining term and document characteristic information. In particular I examine the reasons why combination may perform well and why it can be a technique that gives very variable performance. I also introduce the notion of selective combination of evidence: selecting which evidence to use for each query. This exploits the relevance assessments to make decisions on which evidence is appropriate for individual retrieval situations. All the experiments in Chapter Four are carried out on a set of standard IR test collections.

In Chapter Five I re-examine the findings from Chapter Four on a set of data derived from experiments ran by Borlund and Ingwersen [BI99]. The relevance assessments in this data were made by novice searchers rather than expert relevance assessors as would be the case in Chapter Four. This set of data allowed the examination of the role of task, partial relevance assessments and the user in the process of combining term and document characteristics.

In Chapter Six I present a more detailed examination of the uncertainty attached to the combination of term and document characteristics. Specifically, I present a model for retrieval and RF based on Dempster-Shafer's Theory of Evidence, [Dem68, Sha76]. This model is capable of incorporating aspects of combination, such as the quality or reliability of evidence, that are important for retrieval success.

In Chapter Seven I summarise the main findings of the experiments reported in Part II.

Chapter Four

Combining and selecting characteristics of information use

4.1 Introduction

In this chapter I shall describe two sets of experiments. In the first set of experiments, I examine how information on term use, the term and document characteristics, can be combined to increase retrieval effectiveness. In effect this means using more information on *why* a term may indicate relevance.

The second set of experiments examines the role of relevance assessments in the combination process – using the relevant documents to select which aspects of a term's use may indicate relevance.

In section 4.2 I describe the data I used in these experiments, in section 4.3 I present the main introduction to the experiments themselves. In sections 4.4 – 4.7 I present the results of the experiments and I summarise the main conclusions in section 4.8.

4.2 Data

For the experiments reported in this chapter I used two sets of collections. The first is a set of three small test collections (**CACM**, **CISI** and **MEDLARS** collections[19]), the second is a set of two larger collections (the Associated Press (1988) (**AP**) collection and the Wall Street Journal (1990-92) (**WSJ**)) collection from the TREC initiative [VH96]. Statistics of these collections are given in Table 4.1.

[19] *http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/*

	CACM	CISI	MEDLARS	AP	WSJ
Number of documents	3 204	1 460	1 033	79 919	74 520
Number of queries used[20]	52	76	30	48	45
Average document length[21]	47.36	75.4	89	284	326
Average words per query[22]	11.88	27.27	10.4	3.04	3.04
Average relevant documents per query	15.3	41	23	35	24
Number of unique terms in the collection	7 861	7 156	9 397	129 240	123 852

Table 4.1: Details of CACM, CISI, MEDLARS, AP and WSJ collections

The AP and WSJ test collections each come with fifty so-called TREC topics. Each topic describes an information need and those criteria that were used in assessing relevance when the test collection was created. A TREC topic has a number of sections, (see Figure 4.1 for an example of a topic). In my experiments I only used the short **Title** section from topics 251 – 300 as queries, as using any more of the topic description may be an unrealistic as a user query.

Number: 301
Title: International Organized Crime
Description:
Identify organisations that participate in international criminal activity, the activity, and, if possible, collaborating organisations and the countries involved.
Narrative:
A relevant document must as a minimum identify the organisation and the type of illegal activity (e.g., Columbian cartel exporting cocaine). Vague references to international drug trade without identification of the organisation(s) involved would not be relevant.

Figure 4.1: TREC topic 301

[20]Each collection comes with a number of queries. However, for some queries there are no relevant documents in the collection, i.e. none of the assessed documents were considered relevant. As these queries cannot be used to calculate recall-precision figures they are not used in these experiments. This row shows the number of queries, for each collection, for which there is at least one relevant document.

[21]After the application of stemming and stopword removal.

[22]This row shows the average length of the queries that were used in the experiments after the application of stopword removal and stemming.

Stopwords were removed, using the stopword list in [VR79], and the collections were stemmed using the Porter stemming algorithm, [Por80].

4.3 Outline of experiments

In this chapter I describe three sets of experiments:

i. *retrieval by single characteristic*. In section 4.4 I present results obtained by running each characteristic as a single retrieval function. In this section I examine the relative performance of each characteristics on the test collections, and discuss why some characteristics perform better than others as retrieval functions.

ii. *retrieval by combination of characteristics*. In section 4.5 I investigate whether combining characteristics can improve retrieval effectiveness over retrieval by single characteristic. I also discuss factors that affect the success of combination, such as the size of the combination and which characteristics are combined.

iii. *relevance feedback*. In section 4.6 I investigate how we can use relevance assessments to *select* good combinations of characteristics of terms and documents to use for RF. I describe several methods of selecting which characteristics are important for a query and compare these methods against methods that do not use selection of characteristics. The results from these experiments will be discussed in section 4.7.

4.4 Retrieval by single characteristic

In this section I examine the performance of running each characteristic (term and document characteristics) as a single retrieval function (retrieval by the sum of the *idf* value of each query term, retrieval by the sum of *tf* values of each query term, etc.). The results are presented in section 4.4.2 but before this, in section 4.4.1, I look at how document characteristics should be used to score documents.

4.4.1 Document characteristics - initial investigations

As the *specificity* and *info-noise* characteristics are document rather than term characteristics, they assign the same value to each document irrespective of which terms are in the query. However, the document characteristics can be used to produce different rankings based on two criteria:

i. *which documents receive a score*. Although all documents have a pre-calculated value for the *specificity* and *info-noise* characteristics, we may choose to score only those documents that contain at least one query term, as these documents are those that are the most likely to be relevant.

I assessed two methods of scoring documents - the *query dependent* - and the *query independent* strategies.

In the query independent strategy the retrieval score of a document is the characteristic score (*info_noise* or *specificity*). This method gives an identical ranking of documents for all queries. In the query dependent strategy the retrieval score of a document is also the characteristic score but this score is only assigned to those documents that contain at least one query term. If the document contains no query terms then the retrieval score is zero. In this method all documents that contain a query term are retrieved before the documents that contain no query terms, giving a different document ranking to each query.

ii. *how to order the documents*. The *specificity* characteristic gives high scores to more complex documents, whereas the *info_noise* characteristic gives high scores to documents that have a high proportion of useful information. This means that I am asserting that relevant documents are more likely to have a higher amount of useful information or a higher complexity. This requires testing. I tested two strategies - *standard* - in which documents are ranked in decreasing order of characteristic score and *reverse* - in documents are ranked in increasing order of characteristic score.

These two criteria give four combinations of strategy - query dependent/standard, query independent/standard, query independent/reverse, query dependent/reverse. Each of these strategies correspond to a different method of ranking documents.

The results of these ranking strategies are shown in Table 4.2 for the *specificity* characteristic. Also shown in Table 4.2, for comparison, are the results of two random retrieval runs on each collection. These are also based on a query dependent strategy (random order of all documents containing a query term, followed by random order of the remaining documents) and a query independent strategy (a completely random ordering of all documents).

	standard *specificity*		**reverse** *specificity*		**random**	
Collection	**query** ***dep***	**query** ***ind***	**query** ***dep***	**query** ***ind***	**query** ***dep***	**query** ***ind***
CACM	**1.19**	0.98	**1.19**	1.18	1.14	0.36
CISI	**10.55**	2.83	2.75	3.51	4.66	3.86
MEDLARS	4.62	3.33	4.62	4.48	**12.39**	4.82
AP	0.33	0.06	**0.47**	0.05	0.28	0.05
WSJ	0.42	0.10	**0.57**	0.02	0.35	0.04

Table 4.2: Average precision figures for *specificity* characteristic
dep = dependent strategy, *ind* = independent strategy
Highest average precision figures for each collection are shown in bold

The results were also tested for statistical significance using a paired *t*-test, $p < 0.05$, holding recall fixed and varying precision. The results of this are shown in Table 4.3. The results show that the query dependent random retrieval is a stricter baseline comparison: it gives better results than a completely random retrieval (Table 4.2) and this difference is statistically significant in all collections (Table 4.3, Column 4).

The query dependent method of scoring documents always gives significantly better retrieval effectiveness over the query independent method when documents are ranked in decreasing order of specificity score (*standard* method) (Table 4.2, Column 2 and Table 4.3 Columns 2 and 3). This does not hold so neatly when documents are ranked according to the reverse method. In this case the differences are only significant for three out of the five cases (Table 4.3, Column 3) and the independent strategy is better than the dependent strategy for the CISI collection.

Comparing the two methods of ranking documents (standard versus reverse, Table 4.3, Columns 5 and 6), the reverse strategy gives better results on the small collections when using a query independent method of scoring documents but the reverse holds for large collections (query dependent gives better results). The standard method of ranking documents gives better results on small collections but poorer results on the larger collections.

	standard dep vs ind	reverse dep vs ind	random dep vs ind	dep standard vs reverse	ind standard vs reverse
CACM	sig	not sig	sig	not sig	sig
CISI	sig	sig	sig	sig	sig
MEDLARS	sig	not sig	sig	not sig	sig
AP	sig	sig	sig	sig	sig
WSJ	sig	sig	sig	sig	sig

Table 4.3: Significance tests for the *specificity* document characteristic
where *sig* = statistically significant difference, *dep* = dependent strategy, *ind* = independent strategy, *standard* = documents ranked by decreasing characteristic score, *reverse* = documents ranked by decreasing characteristic score.

From Table 4.2 and Table 4.3 it is clear that overall the *specificity* characteristic performs quite poorly in that there is no clear method of applying it to all collections. However at least one of the combination of document scoring and ranking methods gives statistically significant increases in retrieval effectiveness over the query dependent random retrieval baseline. This is true for all collections except the MEDLINE collection. One possible reason for the poorer results on this collection is that the range of *specificity* characteristic values for this collection is not very wide. Consequently the characteristic does not provide enough information to discriminate between documents.

Overall the *specificity* characteristic is best applied using a query dependent strategy. Whether or not it is applied in decreasing order of characteristic value (standard), or increasing order of characteristic score (reverse) is collection dependent. However the overall preference is for the reverse strategy.

The results of using the *info-noise* characteristic is shown in Table 4.4. The same statistical tests were performed on the results from the *info_noise* rankings and are shown in Table 4.5. From Tables 4.4 and 4.5 the *info_noise* characteristic is best applied using the query-dependent standard strategy: ordering documents containing a query term and with the highest proportion of useful information at the top of the ranking.

	standard *info-noise*		reverse *info-noise*		random	
Collection	**query *dep***	**query *ind***	**query *dep***	**query *ind***	**query *dep***	**query *ind***
CACM	**1.67**	0.5	0.86	1.63	1.14	0.36
CISI	4.08	3.28	3.48	2.78	**4.66**	3.86
MEDLARS	8.67	2.56	8.25	2.98	**12.39**	4.82
AP	**0.44**	0.05	0.29	0.05	0.28	0.05
WSJ	**0.48**	0.03	0.32	0.03	0.35	0.04

Table 4.4: Average precision figures for *info_noise* characteristic
dep = dependent strategy, *ind* = independent strategy
Highest average precision figures for each collection are shown in bold

	standard dep vs ind	**reverse dep vs ind**	**random dep vs ind**	**dep standard vs reverse**	**ind standard vs reverse**
CACM	sig	sig	sig	sig	sig
CISI	sig	sig	sig	sig	sig
MEDLARS	sig	sig	sig	sig	sig
AP	sig	sig	sig	not sig	not sig
WSJ	sig	sig	sig	sig	sig

Table 4.5: Significance tests for the *info_noise* document characteristic
where *sig* = statistically significant difference, *dep* = dependent strategy, *ind* = independent strategy, *standard* = documents ranked by decreasing characteristic score, *reverse* = documents ranked by decreasing characteristic score.

Overall, on all collections, except the MEDLARS collection, at least one method of applying the *specificity* and *info-noise* characteristics gave better performance than random (query independent), and with the exception of MEDLARS and CISI also performed better than the query dependent random run. As stated before the poorer results on these collections may be caused by the small range of values given by the characteristics to the documents.

It is better to rank only those documents that contain a query term than all documents. This is not surprising as, using the query dependent strategy, we are in fact re-ranking the basic *idf* ranking for each query.

I shall discuss the relative performance of the document characteristics against the term characteristics in the next section. Although the document characteristics do not give better results than the term characteristics (see next section), they do generally give better results than the random retrieval runs. This means that they *can* be useful in retrieval if they are used appropriately. One method of using the document characteristics is in combination with other characteristics. This will be discussed in section 4.5.

4.4.2 Single retrieval on all characteristics

The results from running each characteristic as a single retrieval function are summarised in Table 4.6, measured against the query dependent random strategy. This is used as a baseline for this experiment as all the characteristics prioritise retrieval of documents that contain a query term over those documents that contain no query terms. Hence this method of running a random retrieval is more similar in nature to the term characteristics and, as it gives higher average precision, provides a stricter baseline measure for comparison.

Documents are scored by the sum of the characteristic values of each query term contained within the document, e.g. the sum of the *idf* values of all query terms, or the sum of the *tf* values of the query terms.

	Characteristic							
Collection	***idf***	***tf***	***theme***	***context***	***spec***	***noise***	***inf***	**random**
CACM	22.00	22.70	4.36	14.80	1.19	**24.15**	1.67	1.14
CISI	11.50	**12.50**	5.10	9.60	10.55	11.00	4.08	4.66
MEDLARS	43.10	43.70	11.10	36.10	4.60	**43.90**	8.80	12.39
AP	**10.10**	9.86	4.63	9.57	0.47	1.00	0.44	0.28
WSJ	**12.19**	7.39	1.00	0.04	0.42	1.05	0.48	0.38

Table 4.6: Average precision figures for term and document characteristics used as single retrieval functions
where *spec* = specificity, *inf* = info-noise
Highest average precision figures for each collection are shown in bold

The majority of characteristics outperform the query dependent random retrieval baseline. However some characteristics do perform more poorly than a random retrieval of the documents (*info_noise* on CISI, *theme*, *specificity* and *info_noise* on MEDLARS, *context* on WSJ)[23].

[23] All characteristics, for all collections except MEDLARS, outperformed a completely random retrieval.

The order in which the characteristics[24] performed is shown in Figure 4.2 where > indicates statistical significance and ≥ indicates non-statistical significance.[25]

CACM	*noise*	≥	*tf*	≥	*idf*	>	*con*	>	*theme*	≥	*inf*	>	*spec*	>	*rand*		
CISI	*tf*	>	*idf*	>	*noise*	≥	*spec*	>	*con*	>	*theme*	≥	*rand*	>	*inf*		
MEDLARS	*noise*	≥	*tf*	≥	*idf*	>	*con*	>	*rand*	>	*theme*	>	*inf*	≥	*spec*		
AP	*idf*	≥	*tf*	≥	*con*	>	*theme*	>	*noise*	>	*spec*	≥	*inf*	≥	*rand*		
WSJ	*idf*	>	*tf*	>	*noise*	≥	*theme*	≥	*inf*	>	*spec*	>	*rand*	>	*con*		

Figure 4.2: Statistical and non-statistical differences between characteristics on all collections where *spec* = specificity, *con* = context, *inf* = info_noise, *rand* = random

The document characteristics perform quite poorly as they are insensitive to query terms. That is, although, when using these characteristics we score only documents that contain a query term, the document characteristics do not distinguish between documents that contain good query terms and documents that contain poor query terms.

On nearly all collections the standard characteristics (*idf*, *tf*, *noise*[26]) outperformed the new characteristics. One possible reason for this is that, although, the new term characteristics (*theme*, *context*) give a weight to every term in a document, unlike the standard characteristics they do not always give a non-zero weight. The *context* characteristic, for example, will only assign a weight to a term if at least two query terms appear in the same document. In the case of the two larger collections we have relatively smaller queries. Hence the co-occurrence of query terms within a document may be low with the resulting effect that most terms have a zero weight for this characteristic. This, in turn, will lead to a poor retrieval result as the characteristic cannot distinguish well between relevant and non-relevant documents.

Similarly, the *theme* characteristic, as implemented here, will also lead a high proportion of terms being assigned a zero weight compared with the *tf* characteristic. One reason for this is that *theme* assigns a zero weight to a term if it only appears once within a document. A collection such as the MEDLARS collection, which has a high number of terms that only appear in one document may be more susceptible to this, as it contains a large number of unique terms.

---

[24] The query dependent standard strategy was used for the *specificity* and *info-noise* characteristics.

[25] Calculated using a paired *t*-test, $p < 0.05$, holding recall fixed and varying precision

[26] Harman's, [Har86], experimental investigation of the *noise* term weighting function on the Cranfield collection showed superior results for *noise* over *idf*. In these experiments, this held for the shorter CACM and MEDLARS collection. However in the larger collections, the *noise* characteristic performed relatively poorly.

The standard characteristics are also less *strict* algorithms: the information they represent, e.g. frequency of a term within a document, is more general than that represented by the new characteristics. This will mean that the standard characteristics will be useful for a wider range of queries. For example, *tf* will be a useful characteristic for most query terms as, generally, the more often a query term appears within a document, the more likely the document is to be relevant. The *theme* characteristic, on the other hand, will only be useful for those queries where the query terms are related to the main topic of the document. For queries where this condition is not met, the *theme* characteristic will not be useful.

Even though the new characteristics do not perform as well as the traditional weighting functions they do improve retrieval effectiveness over random retrieval. These algorithms are not intended as alternative weighting schemes but as additional ones: ones that provide additional methods of discriminating relevant from non-relevant material. In RF these additional characteristics will be used to score query terms if they are useful at indicating relevant documents for individual queries. That is, by providing evidence of different aspects of information use, they can be used to help retrieval performance in combination with other characteristics. This combination of evidence is the subject of the next section.

4.5 Retrieval by combination of characteristics

In the previous section I described the performance of each characteristic as individual retrieval algorithms. In this section I look at whether the retrieval effectiveness of characteristics will be improved if they are used in combination.

In this experiment I tested all possible combinations of the characteristics, running each possible combination as a retrieval algorithm. For each collection, I effectively run the powerset of combinations, each set comprising a different combination of characteristics. For each combination, the retrieval score of a document was given by sum of the score of each characteristic of each query term that occurred in the document. For example, for the combination of *tf* and *theme*, the score of a document was equal to the sum of the *tf* value of each query term plus the sum of the *theme* value of each query term that occurs in the document.

Two versions of this experiment were run, the first used the values of characteristics given at indexing time, the second treated the characteristics as being more or less important than each other. There are several reasons why one characteristic may be treated as more important than another characteristic. For example, some characteristics may reflect aspects of information

use that are more easily measured than another, some characteristics are better as retrieval functions and should be treated as being more important or some characteristics rely on more sophisticated implementations[27]. I attempt to reflect this by introducing a set of scaling weights (*idf* 1, *tf* 0.75, *theme* 0.15, *context* 0.5, *noise* 0.1, *specificity* and *information_noise* 0.1[28]) that are used to alter the weight given to a term at indexing time. Each indexing weight of a term characteristic is multiplied by the corresponding scaling weight, e.g. all *tf* values are multiplied by 0.75, all *theme* values by 0.15, etc.

This gives two conditions - *weighting* and *non-weighting* of characteristics - for each combination of characteristics.

The results of these experiments are summarised in Appendix C. Tables C.1 – C.10 show the ranking, by average precision, of the combinations on each collection. Some statistical testing was performed on the results to test how discrete the results were, i.e. how often combinations of characteristics gave results that were statistically significant from other combinations with similar average precision figures[29].

The results of statistical testing are indicated in Tables C.131 – C.140 where a dividing line separates statistically significant results. Table 4.7 shows a section of Table C.131 to illustrate this: the combination of *tf* and *noise* is significantly better than the combination of *idf*, *tf* and *noise,* which is better than the combination of *idf, tf*, *noise* and *info-noise*. The combination of *idf, tf*, *noise* and *info-noise* was better, although not significantly better, than the combination of *tf*, *specificity* and *noise* (no dividing line between entries).

tf + ***nse***	30.26
idf + ***tf*** + ***nse***	26.83
idf + ***tf*** + ***nse*** + ***inf***	25.74
tf + ***spec*** + ***nse***	25.41

Table 4.7: Snapshot of Table C.1

Only combinations that are adjacent in the combination ranking are tested for significance. That is the significance testing splits the rankings into groups of combinations that are not

[27] This will be discussed more fully in Chapter Six.

[28] These weights were derived from experiments using a sample of the data from each collection.

[29] The significance test was performed on the whole RP figures, not the average precision figure.

statistically significant from the preceding combination. This is intended to show how *distinct* are the differences between combinations.

The results vary across collections and weighting conditions. The major trend is that statistical testing tends to split the rankings into large groups of combinations. That is, although there is a large difference between good combinations and poor combinations, there are large groups of combinations that have very little performance difference. This is very noticeable, for example, in the CACM collection (with no weighting) where there are only five sets of adjacent combinations with statistically significant differences in precision. The remainder of the combinations differ only slightly from adjacent combinations.

One general conclusion from this analysis is some collections are more susceptible to changes in combination of characteristics or weighting the characteristics than others. For example, weighting characteristics creates more distinct groups of combinations on the CACM and CISI collection but removes these distinct groups on the MEDLARS, AP and WSJ collections. This is primarily because, on these three collections, strong individual characteristics dominate any combinations in which they appear and the results of combinations tend to produce clusters of similar results. This use of statistical testing produces an alternative view on the results.

In the following sections I shall summarise the findings of the combination experiment regarding three aspects: the effect on retrieval effectiveness of combining characteristics, the effect of weighting characteristics, and the effect of adding individual characteristics to other combinations. Each of these will be discussed in a separate section in sections 4.5.1 – 4.5.3. I shall summarise in section 4.5.4.

4.5.1 Effecting of combining characteristics

The experimental hypothesis is that combining characteristics can increase retrieval effectiveness over using individual characteristics. In section 4.5.3 I shall discuss how well the individual characteristics performed in combination. In this section I shall examine the basic hypothesis and discuss general findings.

In Table 4.8 I outline the effect on individual characteristic performance by the addition of other characteristics. Of the 127 possible combinations of characteristics for each collection, each characteristic appeared in 63[30] combinations. Each row is a count of how many of these 63 combinations containing each characteristic had higher average precision (*inc*) than the

[30] Not including the combination that contained only the single characteristic.

characteristic as a single retrieval function, lower average precision (*dec*), or no change in average precision (*none*). For example, how many combinations containing *idf* gave an average precision figure that was better, worse or identical to the average precision of *idf* alone?

The first general conclusion from Table 4.8 is that all characteristics can benefit from combination with another characteristic or set of characteristics. Furthermore, with the exception of the *noise* characteristic on the CACM, and the *tf* and *idf* characteristics on the CISI, any characteristic was more likely to benefit from combination than be harmed by it. This conclusion held under both the weighing and non-weighting conditions.

Collection	Condition	Change	*idf*	*tf*	*theme*	*context*	*spec*	*noise*	*inf*
CACM	NW	inc	**54**	**41**	**63**	**63**	**62**	15	**62**
		dec	9	22	0	0	0	**48**	0
		none	0	0	0	0	1	0	1
	W	inc	**50**	**42**	**63**	**63**	**62**	11	**62**
		dec	8	18	0	0	0	**52**	0
		none	5	3	0	0	1	0	1
CISI	NW	inc	27	1	**63**	**63**	**49**	**39**	**63**
		dec	**35**	**62**	0	0	14	24	0
		none	1	0	0	0	0	0	0
	W	inc	23	7	**63**	**63**	**52**	**40**	**63**
		dec	**34**	**53**	0	0	0	23	0
		none	6	3	0	0	11	0	0
MEDLARS	NW	inc	**47**	**44**	**63**	**63**	**63**	**43**	**63**
		dec	16	19	0	0	0	20	0
		none	0	0	0	0	0	0	0
	W	inc	**45**	**55**	**63**	**60**	**63**	**37**	**63**
		dec	18	8	0	3	0	26	0
		none	0	0	0	0	0	0	0
AP	NW	inc	**47**	**55**	**63**	**59**	**62**	**62**	**62**
		dec	16	8	0	4	1	1	1
		none	0	0	0	0	0	0	0
	W	inc	**54**	**60**	**62**	**61**	**63**	**60**	**63**
		dec	4	0	3	0	0	0	0
		none	5	3	0	2	0	3	0
WSJ	NW	inc	**40**	**63**	**63**	**63**	**63**	**63**	**63**
		dec	22	0	0	0	0	0	0
		none	1	0	0	0	0	0	0
	W	inc	**46**	**63**	**63**	**63**	**63**	**60**	**63**
		dec	8	0	0	0	0	3	0
		none	9	0	0	0	0	0	0

Table 4.8: Effect of combination on individual characteristics
where *inc* = increase in average precision when combined, *dec* = decrease in average precision when in combination, *none* = no difference in average precision when in combination, *NW* = non-weighting condition, *W* = weighting condition
Bold figures indicate the predominant effect of the characteristic in combination

The second general conclusion is that the performance of a characteristic as a single retrieval function (section 4.3.2) is a good indicator of how well the characteristic will perform in combination. The poorer the characteristic is at retrieving relevant documents the more likely it is to benefit from combination with another characteristic. For each collection, on the whole, the poorer characteristics[31] improve more often in combination with other characteristics. The reverse also holds: if a characteristic is good as a single retrieval function, then there is less chance that it will be improved in combination. For example the best characteristics in the small collections (*tf*, *idf* on CISI, and *noise* on CACM) showed the lowest overall improvement in combination. However the overall tendency is beneficial: combination benefits more characteristics than it harms.

In the remainder of this section I look at what affects the success of combination. In particular, I look examine the size of combinations and the components of combinations.

In Table 4.9 I analyse the success of combination by *size* of combination, that is how many characteristics were combined. For each condition, weighting and non-weighting, on each collection I ranked all combinations by average precision[32]. I then took the median[33] value and the size of the combinations that appeared above and below this point. In Table 4.9 bold figures indicate where most combinations, of a given size, appeared (above or below the median point).

In the majority of cases the larger combinations (combinations of 4-7 characteristics) performed better than the median value, and the smaller combinations (combinations of 1-3 characteristics) performed worse than the median. There was little difference between the weighting and non-weighting conditions.

One possible reason for the success of the larger combinations is that poor characteristics have a lower overall effect in a larger combination. That is, if we only combine two characteristics and one of these is a poor characteristic, then there is a greater chance that the combination will perform less well than the better individual characteristic. Conversely, if we combine a number of characteristics, and one is poorer than the rest, then this will not have such a great effect on the performance of the combination.

---

[31] These were the *theme*, *context*, *specificity* and *info_noise* for the CACM, CISI and MEDLARS collections and *theme*, *context*, *noise*, *specificity* and *info_noise* for the AP and WSJ collections.

[32] Tables C.1 – C.10.

[33] For each collection, in each condition, there were 127 possible combinations, the median point was taken to be the 64th combination in the ranking of all combinations.

A further reason for larger combinations performing more effectively is that they allow for a more *distinct* ranking. That is, the more methods we have of scoring documents, the less chance that documents will receive an equal retrieval score.

Collection	Position	Condition	1	2	3	4	5	6	7
CACM	Above	NW	2	5	12	**20**	**17**	**7**	**1**
		W	2	6	13	**21**	**15**	**6**	**1**
	Below	NW	5	16	23	15	4	0	0
		W	5	15	22	14	6	0	0
CISI	Above	NW	2	7	**19**	**21**	**15**	0	**1**
		W	2	9	17	**22**	**11**	**2**	**1**
	Below	NW	5	14	16	14	6	7	0
		W	5	12	18	13	10	5	0
MEDLARS	Above	NW	0	5	15	**24**	**13**	**6**	1
		W	0	7	18	13	**18**	**7**	**1**
	Below	NW	7	16	20	11	8	1	0
		W	7	14	18	22	3	0	0
AP	Above	NW	0	7	11	**20**	**18**	**7**	**1**
		W	0	3	11	**23**	**19**	**7**	**1**
	Below	NW	7	14	24	15	3	0	0
		W	7	18	24	12	2	0	0
WSJ	Above	NW	1	5	13	**21**	**17**	**7**	**1**
		W	0	3	12	**23**	**18**	**7**	**1**
	Below	NW	7	16	22	14	4	0	0
		W	7	18	23	12	3	0	0

Table 4.9: Distribution of combinations over ranking of median precision
where *Above* = combination falls above or at median point of ranking, *Below* = combination falls below median point of ranking, *NW* = non-weighting condition, *W* = weighting condition

Now I look at how the components of the combinations affect the success of combining characteristics. As stated before, each characteristic appeared in a total of 63 combinations. Table 4.10 presents how many of these combinations appeared above the median combination in the ranking of average precision, i.e. how many times a combination containing a characteristic performed better than the median combination. The better individual characteristics, e.g. *idf* and *tf*, appeared in more combinations above the median than below

for all collections. The poorer characteristics, e.g. *info_noise*, tended to appear in more combinations below the median than above.

This is not necessarily to say, however, that poor characteristics always decrease the performance of a combination. Often a characteristic that performs less well as a single characteristic can improve a combination. What is important is how well a combination of characteristics separates relevant from irrelevant documents for an individual query: a particular combination may work poorly on average but work well for certain queries. This is important for the RF experiments, in which I select which are good characteristics for individual queries, section 4.6.

	CACM	**CACM**	**CISI**	**CISI**	**MEDLARS**	**MEDLARS**	**AP**	**AP**	**WSJ**	*WSJ*
	NW	**W**	**NW**	**W**	**NW**	**W**	**NW**	**W**	**NW**	**W**
idf	**42** ***(67%)***	**41** ***(65%)***	**38** ***(60%)***	**43** ***(68%)***	**41** ***(65%)***	**40** ***(63%)***	**39** ***(62%)***	**43** ***(68%)***	**41** ***(65%)***	**46** ***(73%)***
tf	**47** ***(75%)***	**52** ***(83%)***	**41** ***(65%)***	**44** ***(70%)***	**42** ***(67%)***	**50** ***(79%)***	**51** ***(81%)***	**47** ***(75%)***	**52** ***(83%)***	**47** ***(75%)***
theme	**33** ***(52%)***	**32** ***(51%)***	**44** ***(70%)***	**38** ***(60%)***	**48** ***(76%)***	**42** ***(67%)***	30 *(48%)*	**41** ***(65%)***	**32** ***(51%)***	**41** ***(65%)***
con	29 *(46%)*	30 *(48%)*	20 *(32%)*	16 *(25%)*	28 *(44%)*	28 *(44%)*	**41** ***(65%)***	**45** ***(71%)***	**44** ***(70%)***	**42** ***(67%)***
spec	30 *(48%)*	**32** ***(51%)***	30 *(48%)*	**32** ***(51%)***	31 *(49%)*	**33** ***(52%)***	**37** ***(59%)***	**32** ***(51%)***	**32** ***(51%)***	**33** ***(52%)***
noise	**49** ***(78%)***	**50** **(79%)**	27 *(43%)*	29 *(46%)*	**41** ***(65%)***	**37** ***(59%)***	**36** ***(57%)***	**36** ***(57%)***	**32** ***(51%)***	**34** ***(54%)***
inf	**32** ***(51%)***	**32** ***(51%)***	**32** ***(51%)***	31 *(49%)*	28 *(44%)*	31 *(49%)*	**32** ***(51%)***	31 *(49%)*	**34** ***(54%)***	30 *(48%)*

Table 4.10: Number of appearances of a characteristic in a combination appearing above median combination
Bold figures indicate where the majority of the combinations containing an individual characteristic appeared above the median value.
con = context, *spec = specificity*, *inf = info-noise*.

To summarise the findings: combinations of characteristics, whether weighted or not, is beneficial for all characteristics on all collections tested. This benefit is greater when the characteristic is poor as a single retrieval function but the overall benefits of combination still holds for good characteristics. The larger combinations (4-7 characteristics) tend to be better than small (1-3 characteristics) as retrieval functions over the collections.

4.5.2 Effect of weighting characteristics

The basis behind weighting characteristics was that some characteristics may be better at indicating relevance than others. In Table 4.11, I summarise the effect of weighting on each collection, indicating the number of combinations that increased/decreased in average precision when using weighting. Overall, 47% of combinations improved using weighting on CACM collection, 61% on CISI, 60% MEDLARS, 69% on AP and 66% on WSJ.

As can be seen for all collections, except CACM, weighting was beneficial in that it improved the average precision of more combinations than it decreased. Generally these improvements were statistically significant.

	Increase		**Decrease**	
Collection	**Significant**	**Non-significant**	**Significant**	**Non-significant**
CACM	24 *20%*	32 *27%*	31 *26%*	**33** *28%*
CISI	**59** *49%*	14 *12%*	37 *31%*	10 *8%*
MEDLARS	**45** *38%*	27 *23%*	23 *19%*	25 *21%*
AP	**51** *43%*	32 *27%*	22 *18%*	15 *13%*
WSJ	**67** *56%*	12 *10%*	26 *22%*	15 *13%*

Table 4.11: Effect of weighting on combination performance
Significant = statistically significant change,
Non-significant = non statistically significant change
Bold figures indicate predominant effect of weighting on each collection

Table 4.12 breaks down these figures by size of combination, the number of characteristics in the combination. The combination that benefited most from weighting were also these tended to be the ones that performed best in combination, i.e. those combination of four or greater characteristics.

In Table 4.13, I analyse which characteristics appeared in the combinations that did better using weighting than no weighting. Generally, combinations containing *idf* and *tf* were helped by weighting across the collections and *theme* and *context* were helped in the larger collection. The only characteristic to be consistently harmed by weighting was the *noise* characteristic.

Collection	Change	2	3	4	5	6	7
CACM	**Increase**	8	14	17	**12**	**4**	0
	Decrease	13	21	18	9	3	**1**
CISI	**Increase**	9	**22**	**24**	**11**	**7**	**1**
	Decrease	12	13	11	10	0	0
MEDLARS	**Increase**	9	**19**	**23**	**14**	**6**	0
	Decrease	12	16	12	7	1	**1**
AP	**Increase**	8	**21**	**27**	7	1	**1**
	Decrease	13	14	8	19	6	0
WSJ	**Increase**	8	**19**	**25**	**19**	**7**	**1**
	Decrease	13	16	10	2	0	0

Table 4.12: Effect of weighting by size of combination
bold figures indicate predominant effect on each size of combination

	idf	*tf*	*theme*	*context*	*spec*	*noise*	*inf*
CACM	**36**	**42**	**34**	23	**33**	18	26
	64%	75%	61%	41%	59%	32%	46%
CISI	**46**	**49**	27	32	**42**	21	**38**
	63%	67%	37%	44%	58%	29%	52%
MEDLARS	**43**	**40**	29	35	**46**	9	**48**
	60%	56%	40%	49%	64%	13%	67%
AP	**52**	**46**	**55**	**45**	40	15	**48**
	63%	55%	66%	54%	48%	18%	58%
WSJ	**54**	**45**	**49**	**45**	39	20	39
	68%	57%	62%	57%	49%	25%	49%

Table 4.13: Appearance of individual characteristics in combinations that were improved by weighting
bold figures indicate those characteristics for which weighting was beneficial overall.

Weighting is generally beneficial but it is important to get good values for the characteristics. For example, both *idf* and *tf* were good individual retrieval algorithms and were highly weighted which helped their performance in combination as the combination was more heavily biased towards the ranking given by these characteristics.

noise, on the other hand, was a variable retrieval algorithm in that it performed well on some collections and more poorly on others. As it was weighted lowly the overall effect of *noise* in combination was lessened in the weighting condition. Consequently in cases where *noise*

would have been a good individual retrieval algorithm the combination did not perform as well as it might have without weighting.

A final observation is that although weighting did not generally improve the best combination for the collections[34], it did tend to improve the performance of the middle ranking combinations significantly. These were the combinations that appeared in the middle of the ranking of combinations described in section 4.5.1. Weighting then was a success in that it improved the performance of most combinations. However it achieved this by decreasing the performance of the poorer combinations and increasing the performance of the average combinations.

4.5.3 Effect of adding individual characteristics

In section 4.5.1, I gave general conclusions about the effect of combining characteristics. In this section I look more closely at the effect of combining individual characteristics and the effect of characteristics on the performance of a combination of characteristics. In Table 4.14 I summarise the effect of adding a characteristic to other combinations, e.g. adding *idf* to the 63 combinations that did not already contain *idf*.

I measure whether the new information causes an increase in average precision (adding *idf* improves retrieval), a decrease in average precision (adding *idf* worsens retrieval), or no change in average precision (adding *idf* gives the same retrieval effectiveness).

[34] Tables C.1 – C.10

		CACM		CISI		MEDLARS		AP		WSJ	
		No Wgt	**Wgt**	**No Wgt**	**Wgt**	**No Wgt**	**Wgt**	**No Wgt**	**Wgt**	**No Wgt**	**Wgt**
idf	Inc	**51**	**58**	**54**	**50**	**47**	**48**	**55**	**63**	**62**	**62**
	Same	0	1	0	0	0	0	0	0	0	0
	Dec	12	4	9	13	16	15	8	0	1	1
tf	Inc	**60**	**59**	**57**	**54**	**53**	**56**	**60**	**62**	**62**	**62**
	Same	1	3	1	1	1	1	1	1	1	1
	Dec	2	1	5	8	9	6	2	0	0	0
theme	Inc	**33**	**26**	**48**	**45**	**51**	**49**	22	**38**	26	**54**
	Same	2	6	3	2	1	1	1	2	2	2
	Dec	28	31	12	16	11	13	**40**	23	**35**	7
context	Inc	27	18	8	12	17	14	**56**	**63**	**59**	**48**
	Same	2	4	0	0	0	0	0	0	0	0
	Dec	**34**	**41**	**55**	**51**	**46**	**49**	7	0	4	15
spec	Inc	19	14	16	22	17	13	**46**	4	22	6
	Same	1	**36**	3	17	0	**35**	1	0	2	**54**
	Dec	**43**	13	**44**	**24**	**46**	15	14	**56**	**39**	3
noise	Inc	**60**	**50**	9	29	**51**	**53**	**48**	**57**	**52**	**48**
	Same	1	6	1	0	2	1	2	2	5	15
	Dec	2	7	**53**	**34**	10	9	13	4	6	0
info_ noise	Inc	**37**	18	**46**	18	18	16	**31**	5	**45**	5
	Same	0	**35**	1	16	0	**32**	1	**57**	0	**54**
	Dec	26	10	16	**29**	**45**	15	**31**	1	18	4

Table 4.14: Effect of the addition of a characteristic to combinations of characteristics
bold figures indicate predominant effect of each characteristic

I look first at the addition of individual characteristics to any combination of other characteristics.

On all collections the addition of *idf* or *tf* information to a combination of characteristics was beneficial. This was more pronounced in the larger AP and WSJ collections, and held under both the weighting and non-weighting conditions.

The addition of *theme* information improves the performance of other combinations for the smaller collections using either weighting or non-weighting. For the larger collections, the *theme* characteristic only improved performance under the weighting condition.

The addition of *context* characteristic performed poorly in the smaller collections, performing more poorly when using weighting. In the larger collections the majority of combinations improved after the addition of *context* information.

With exception of the CISI, the addition of the *noise* characteristic improves performance in both weighting and non-weighting conditions. This supports the earlier argument, that although a characteristic can perform poorly on its own, it can improve the performance of other characteristics when used in combination.

The two document characteristics – *specificity* and *info_noise* – are very susceptible to how they are treated. The *specificity* characteristic tends to decrease the effectiveness of a combination of characteristics if the characteristics are not weighted. If the characteristics are weighted, then addition of *specificity* information is neutral: the combination performs as well as without the *specificity* information. The WSJ collection is the exception to this general conclusion. For this collection, under no weighting, the addition of *specificity* increases the effectiveness of a combination. Under weighting *specificity* decreases the effectiveness of a combination.

The *info_noise* characteristic tends to improve the effectiveness of a combination when using no weighting and to be neutral with respect to weighting, i.e. it does not change the performance of the combination. The main exception to this is the MEDLARS collection in which *info_noise* tends to harm the performance of a combination when not using weighting.

Having considered which characteristics improved or worsened combinations, we now examine which combinations are affected by the addition of new information. In Tables C.11 – C.20, in the Appendix, I present a summary of how often individual characteristics will improve a combination containing another characteristic, e.g. how many combinations containing *idf* are improved by the addition of *tf*.

Under both the weighting and non-weighting conditions the following generally held:

- *idf* improved combinations containing *context* more than other characteristics and improved combinations containing *noise* least of all

- *tf* improved combinations containing *context* or *noise* more than other characteristics and *theme* least

- *theme* improved combinations containing *context* most and combinations containing *tf* least

- *context* improved combinations containing *noise* least

- *specificity* improved combinations that contained *theme* and *info_noise* more than combinations containing other characteristics

- for the *noise* characteristic there were no general findings except that combinations containing *idf* were usually less likely to be improved by the addition of *noise* information

- *info_noise* improved combinations containing *theme* and *specificity* most often.

The use of weighting slightly altered those combinations that performed well but the basic trends were the same across the conditions. On the larger collections, one effect of weighting was to reduce the effect of individual characteristics in that the effect of adding a characteristic was less likely to be dependent on which characteristics were already in the combination.

One further observation is that term weighting schemes that represent similar features (e.g. *idf* and *noise* which both represent global term statistics, and *tf*/*theme* which both represent within-document statistics) generally combine less well. That is combining these pairs of weights does not generally help retrieval as much as combining complementary weights, e.g. *idf* and *tf*, *idf* and *theme*, etc. Combining the two document characteristics, however, does seem to give better results.

4.5.4 Summary

The hypothesis was that combining evidence – combining characteristics of terms – can improve retrieval effectiveness over retrieval by single characteristics. In section 4.5, I demonstrated that this was generally the case: all characteristics could benefit from combination. However not all combinations are successful. Two aspects of combination that are likely to predict success are the nature of the characteristics– complementary functions combine better – and the success of the characteristic as a single retrieval function.

Weighting the characteristics to reflect the strength of each characteristic as a single retrieval function is also generally a good idea. However it can be difficult to set optimal weights for two reasons: firstly it is likely that good weights will be collection dependent as the individual characteristics have different levels of effectiveness on different collections. Secondly the weights should reflect the effectiveness of the characteristics relative to each other. However this becomes difficult to assess when we combine characteristics, as we have to measure the relative strength of each characteristic against a set of characteristics, e.g. the effectiveness of *idf* in combination with *tf* and *theme*. The performance of the characteristics as individual retrieval functions gives us some guidance on how to set weights but some experimentation is necessary to set useful values.

Smeaton, [Sme98], suggests that retrieval strategies which are conceptually independent should work better in combination, and that retrieval strategies that work to same general level of effectiveness should be suitable for conjunction. In his experiments Smeaton demonstrated that although this does generally hold it can be difficult to produce a good combination. I reinforce these findings in this paper and demonstrate how weighting the different retrieval functions – different characteristics – can help the combination process.

Collection and condition	**Best combination**	**Average precision of best combination**
CACM (NW)	***tf*** + ***noise***	30.26
CACM (W)	***idf*** + ***tf*** + ***noise***	25.68
CISI (NW)	***idf*** + ***tf***	12.87
CISI (W)	***idf*** + ***tf***	12.84
MEDLARS (NW)	***theme*** + ***noise***	48.64
MEDLARS (W)	***theme*** + ***noise***	47.29
AP (NW)	***idf*** + ***tf*** + ***context*** + ***noise***	15.31
AP (W)	***all***	14.09
WSJ (NW)	***idf*** + ***tf***	15.65
WSJ (W)	***all***	15.73

Table 4.15: Best combinations for each collection and condition
(**NW** = non-weighting condition, **W** = weighting condition)

In Table 4.15, I show the best combination of characteristics for each collection. As can be seen which set of characteristics constitutes the best combination differs over the collections. If we use weighting of characteristics, then the best combination for a collection may also

change, e.g. as is the case for the CACM, AP and WSJ collections. This is a further difficulty with a straightforward combination of evidence: it is difficult to derive a good set of characteristics that can be used on all collections. In the next section I propose a method to counter this difficulty: using the relevant documents to select a good set of characteristics for individual queries, irrespective of to which collection they are being applied.

4.6 Relevance feedback

The intention behind the set of experiments described in this chapter is twofold: first to demonstrate that taking into account how terms are used within documents can improve retrieval effectiveness; secondly that it is possible, for each query, to *select* an optimal set of characteristics for retrieval based on the relevance.

That is, I am not only asserting that considering how terms are used *can* improve retrieval, but that the characteristics that *will* improve retrieval will vary across queries and collections. For example, for some queries the context in which the query terms appear will be important, whereas for other queries it may be how often the query terms appear. For each query term, then, there will be a set of characteristics that will best indicate relevance. In the experiments described in the remainder of this chapter I test whether this hypothesis holds by investigating methods of *selecting* characteristics of query terms.

4.6.1 Methodology

In these experiments I performed a series of RF experiments, selecting characteristics to represent query terms based on the differences between the relevant and non-relevant documents.

The methodology was as follows:

- rank all documents in a collection using the combination of all the characteristics (*all* ranking)
- take the 30 top documents from the initial *all* ranking
- calculate for each query term the average score for each characteristic in the relevant and non-relevant set, e.g. the average *tf* value for query term 1 in relevant documents, the average *tf* value for query term 1 in non-relevant documents.
- select which characteristics of each query term to use to score documents and how the characteristics should be used. Four strategies were tried, each will be discussed separately in sections 4.6.3.1-4.6.3.4. Each strategy constructs a modified query containing characteristics of terms.
- re-rank the remaining retrieved documents

• calculate recall-precision values using a full-freezing ranking scheme, section 1.2.4, [CCR71] to ensure that we are only comparing the effect of each technique on the unretrieved, relevant documents.
• compare the results given, over the same set of documents, by doing no RF, the results obtained from the best combination of characteristics (section 4.6.4, Table 4.12) and an alternative RF algorithm, the F_4 method (section 4.6.2).

This set of experiments was designed to test the hypothesis that some queries or documents will be more suited to certain combinations of characteristics and that we can select these characteristics automatically.

Before I discuss the results of the experiments, I shall discuss the baseline measures, section 4.6.2, and the three methods of selecting characteristics of query terms, section 4.6.3.

4.6.2 Baseline measures

4.6.2.1 No feedback

The first baseline is the no feedback case: all documents are ranked by the combination of all term and document characteristics. This baseline is used to test which baselines and feedback techniques are better than the default ranking of documents.

4.6.2.2 Best combination

The second baseline is the combination of characteristics that gave the best performance in the combination of evidence experiments, section 4.5.4, Table 4.15. The Best Combination baseline is used to decide whether selecting characteristics for each query term is better than using a single good set of characteristics for all query terms.

4.6.2.3 F_4

The RF techniques that will be proposed in section 4.6.3, require comparison against another RF algorithm. For this I chose the F_4 weighting algorithm, [RSJ76], Equation 4.1, which assigns a new weight to a term based on relevance information. This technique for reweighting query terms was chosen partly because it has been shown to give good results but also because it does not add any new terms to the query. As my technique also does not add any new terms to the query but only modifies the existing query, I felt this is a fair comparison with which to test my techniques.

$$w_q(t) = \log\left(\frac{(r_t + 0.5)(N - n_t - R + r_t + 0.5)}{(n_t - r_t + 0.5)(R - r_t + 0.5)}\right)$$

Equation 4.1: F_4 function, which assigns a weight to term t for a given query. r_t = the number of relevant documents containing the term t, n_t = the number of documents containing t, R = the number of relevant documents for query q, and N = number of documents in the collection

4.6.3 Feedback strategies

In this section I propose four RF strategies all of which are based on selecting characteristics.

4.6.3.1 Feedback strategy one

In this method I select for each query which characteristics to use for each query term based on their average values in the relevant and non-relevant documents, described in section 4.6.1. For example, if the average *context* value for a query term was greater in the relevant documents than in the non-relevant documents, then the *context* value of the term was taken to be a better indicator of relevance than non-relevance and so was included in the new query. The modified query is a set of characteristics of the query terms. This is shown in Figure 4.3.

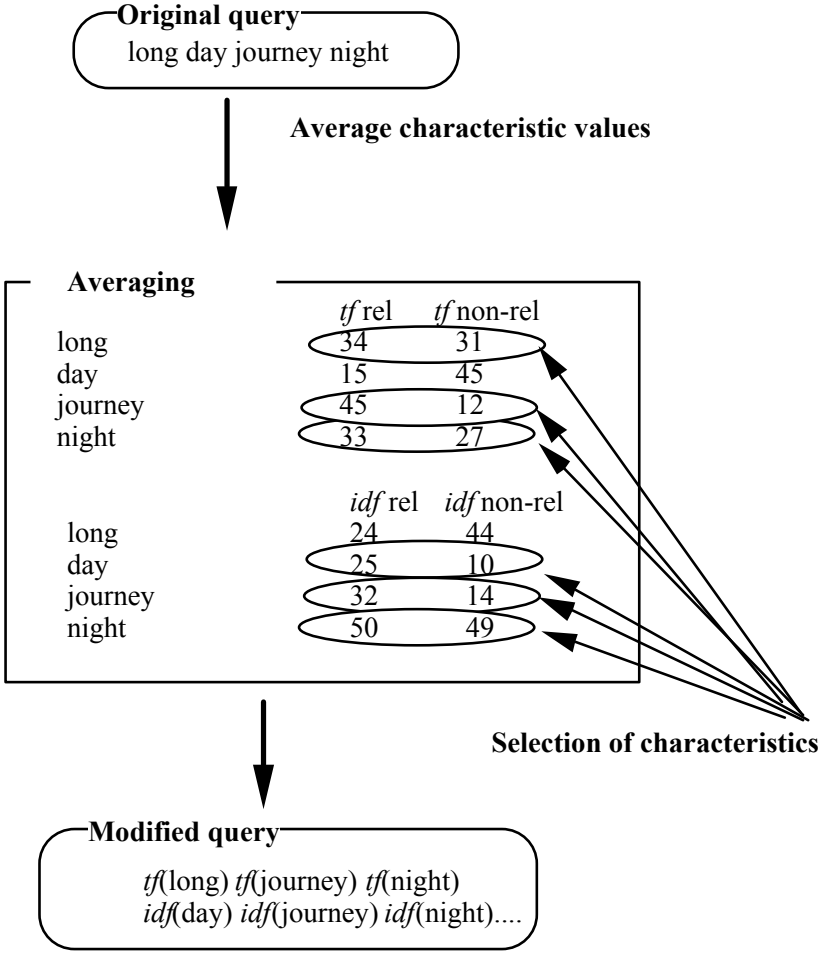

Figure 4.3: Feedback strategy one

The document characteristics are also chosen using relevance information: if the document characteristic score is higher in the relevant documents than in the non-relevant documents then the characteristic is used to score the document.

4.6.3.2 Feedback strategy two

Feedback strategy one (Feedback 1) selectively combined evidence on a query-to-query basis, ranking all documents based on the same set of query term characteristics. Feedback strategy two (Feedback 2) starts with the set of characteristics produced by Feedback 1, then selects which of these characteristics to use on a document-to-document basis. The result of this is that I first select a set of characteristics based on the set of relevant documents and then decide which of these characteristics to use to score each document.

The intuition behind this is: if a characteristic is indicated as a good indicator of relevance then we should not only bias retrieval of documents which demonstrate this characteristic but suppress retrieval of documents which do not. For example, if a query term must appear often in a document – high *tf* value – to be relevant, then documents that only contain a few occurrences of the query term – low *tf* value - should not be considered.

I use the same averaging technique as in the previous strategy to construct a modified query. Then, for each document I compare the characteristic score of each query term in the document against the average score. If the characteristic score is greater than the average then it is counted as part of the document score; if not the evidence is ignored. This experiment is, then, a more strict case of Feedback 1. Feedback 1 selected characteristics with which to rank all documents, whereas this experiment selects characteristics for a query and then uses them selectively across documents.

4.6.3.3 Feedback strategy three

This third experiment is also a refinement of Feedback 1. In Feedback 1 I included a characteristic of a term in a query if it was better at indicating relevance than non-relevance. In this experiment I also take into account how *well* a characteristic indicates relevance. I first select a set of characteristics as in Feedback 1, then weight each term by the ratio of the average characteristic value in the relevant to the non-relevant documents. This ratio is taken to be an indication of how well a characteristic indicates relevance and is used to weight characteristics.

The contribution of a characteristic of a term to the retrieval score of a document is the ratio multiplied by the weight of the characteristic of the term in the document. This combined

weight is a measure of the discrimination power of a characteristic of a term (the ratio) and its indexing strength (the indexing weight in the document). In the weighting condition (described in section 4.5) a third weight is given by the characteristic weight.

The intuition behind this is that if a characteristic does not discriminate well over the relevant and non-relevant set then we should not prioritise this information.

4.6.3.4 Feedback strategy five

The final feedback technique does not select characteristics but instead uses only the discrimination power of a characteristic of a term (the ratio). This will be known as Feedback strategy 5[35]. This technique is used to compare the effect of the discriminatory power of term characteristics against the selection of characteristics. That is, the performance of Feedback 1 against Feedback 3 tests the value of using the discriminatory power of characteristics and the performance of Feedback 5 against Feedback 3 tests the utility of selecting characteristics.

To summarise: Feedback 1 selects characteristics for each query term, Feedback 2 selects characteristics for each query term relative to each document, Feedback 5 does not select characteristics – it uses all characteristics – but it weights the characteristics according to how well they distinguish relevant material, Feedback 3 selects and weights the characteristics.

4.7 Results

In this section I examine three sets of results, to test different aspects of the feedback techniques.

i. the results from running the feedback strategies as *predictive* strategies. This is the methodology outlined above and is designed to test whether the feedback techniques help retrieve more relevant documents based on an initial sample of relevant documents. Results from this test will be discussed in section 4.7.1.

ii. the results from running the strategies as *retrospective* strategies. In this case I use the strategies to form modified queries based on knowledge of all the relevant documents. This success of a feedback strategy in retrospective feedback is measured by how well it ranks all the relevant documents, rather than by how well it improves the retrieval of new relevant documents. This technique, then should give the upper performance of a feedback strategy and is discussed in section 4.7.2.

---

[35] To differentiate it from the selection strategies Feedback 1 – Feedback 3 and the baseline F_4 strategy.

iii. the characteristics used in the feedback strategies. In section 4.7.3 I examine which characteristics were used in the feedback strategies. I do this to draw conclusions about the performance of the feedback strategies based on which characteristics were selected to describe query terms.

4.7.1 Predictive feedback

Table 4.16 presents the results of the predictive experiments. Each row shows the average precision after four iterations of feedback plus the percentage increase in average precision over no feedback (Table 4.16, column 3).

There are several conclusions from the predictive feedback experiments.

Collection/ Condition	No feedback	Best Comb	F_4	Fback 1	Fback 2	Fback 3	Fback 5
CACM	25.28	**30.26**	26.58	27.38	23.28	27.62	27.45
NW		**19.70%**	5.14%	8.31%	-7.91%	9.26%	8.58%
CACM	24.34	25.68	25.51	25.98	21.79	**26.44**	26.39
W		5.51%	4.81%	6.74%	-10.48%	**8.63%**	8.43%
CISI	11.66	12.87	14.05	14.1	13.73	**15.11**	14.89
NW		10.38%	20.50%	20.93%	17.75%	**29.59%**	27.73%
CISI	12.02	12.84	14.2	14.55	14.21	**15.57**	15.09
W		6.82%	18.14%	21.05%	18.22%	**29.53%**	25.48%
MEDLARS	45.92	48.64	47.93	48.69	48.23	**49.41**	49.27
NW		5.92%	4.38%	6.03%	5.03%	**7.60%**	7.31%
MEDLARS	45.29	47.29	47.61	48.14	47.61	**48.90**	48.49
W		4.42%	5.12%	6.29%	5.12%	**7.97%**	7.08%
AP	12.04	**15.31**	12.46	13.15	12.09	13.19	12.81
NW		**27.16%**	3.49%	9.22%	0.42%	9.55%	6.38%
AP	14.09	14.09	14.58	14.88	14.51	**15.01**	14.69
W		0.00%	3.48%	5.61%	2.98%	**6.53%**	4.25%
WSJ	13.33	**15.65**	13.53	14.4	13.96	14.47	14.22
NW		**17.40%**	1.50%	8.03%	4.73%	8.55%	6.71%
WSJ	15.73	15.73	15.89	16.37	15.86	**16.47**	16.20
W		0.00%	1.02%	4.07%	0.83%	**4.70%**	2.94%

Table 4.16: Summary of predictive RF experiments
Figures in bold represent the highest increase in average precision for each case
(**NW** = non-weighting condition, **W** = weighting condition)

Firstly, the selective feedback strategies (Feedback 1 – Feedback 3) do perform well. On the weighting condition at least one of the Feedback methods outperformed the No Feedback and Best Combination methods. However, if we did not use weighting then the Best Combination method outperformed the Feedback strategies on the AP, CACM and WSJ collections. Out of the ten tests (five collections, weighting and non-weighting conditions), seven achieved best overall performance with a Feedback strategy[36]. This latter finding demonstrates that selecting a good combination of characteristics for each query is better than using the best combination of characteristics for a set of queries. In addition, on all cases, the Feedback 1 and Feedback 3 strategies outperform the F_4 baseline.

Secondly, comparing the weighting and non-weighting conditions: the better the initial ranking, the better the feedback performance. That is, whichever condition gave the better average precision for the initial ranking (No feedback column) also gave the better average precision after four iterations of feedback. However, the conditions that gave the poorer initial average precision gave the higher improvement after feedback measured as a percentage increase. Thus, good initial rankings give better feedback in the sense that they retrieve relevant documents better but feedback improves a poor ranking more than a good ranking.

This latter conclusion possibly, in part, arises because there is greater improvement to be gained from a poor initial ranking than a good initial ranking. Weighting, however, does not change the relative performance of the feedback algorithms: if one feedback strategy performs better than another under the non-weighting condition, it will also perform better under the weighting condition.

Thirdly, there is a marked preference for the Feedback 3 strategy. This strategy selects term characteristics for each query term and also uses the discrimination power of a characteristic of a term to score documents. The extra information given by the discrimination power between relevant and non-relevant documents is the cause of the better performance of Feedback 3 over the other feedback strategies.

---

[36] The results were also tested for statistical significance. There were seven cases where the Feedback 3 strategy performed best. For three cases where the Feedback 3 strategy performed best, the difference between the Feedback 3 strategy and the next best technique was statistically significant (CISI W, MEDLARS W and AP W). For the three cases where the Best Combination performed best, there was no statistical significance between the Best Combination and Feedback 3. In addition, the best performing technique in each case was statistically better than no feedback.

On the larger collections (AP and WSJ), those collections that also have the shorter queries, the highest average precision was given by the Feedback 3 strategy using weighting of characteristics. This method uses the most evidence to score documents: evidence on the quality of the characteristics through the use of weighting, selection of good term characteristics and the weighting given by the discrimination between relevant and non-relevant documents.

Comparing the three selective strategies, on all the collections the Feedback 3 strategy outperformed the Feedback 1 strategy which outperformed the Feedback 2 strategy. The Feedback 2 and 3 strategies are both refinements of the basic Feedback 1 strategy and both use additional evidence to make a retrieval decision. In the case of Feedback 2 this additional information comes in the form of the index scores of the query term characteristics in individual documents and in the Feedback 3 strategy it comes from the discrimination power of a query term characteristic over the set of relevant and non-relevant documents. The consistency of the performance of the Feedback 3 strategy over the Feedback 2 strategy suggests discriminatory power is a better source of additional evidence.

The Feedback 5 strategy, which did not select characteristics of terms, performed best on the smaller collections (CACM, CISI and MEDLARS) where it always outperformed the selective Feedback 1 strategy. However on the larger collections (AP and WSJ) the Feedback 1 strategy outperformed the Feedback 5 strategy. Therefore the discriminatory power of term characteristics alone (Feedback 5) seems to be more important for small collections where we have smaller ranges of values for the term characteristics, whereas on larger collections selecting which characteristics to use is more important (Feedback 1). However the combination of selection and discrimination power (Feedback 3) always gives better results than simply selecting characteristics (Feedback 1) or assigning discriminatory weights to characteristics (Feedback 5).

4.7.2 Retrospective feedback

In Table 4.17 I present the results of the retrospective feedback experiments. These experiments use all the relevant documents to modify the query and this extra evidence should give better performance in RF. The first observation is that, for all collections and conditions, a feedback method does give best overall results and selection methods of feedback do give consistent increases in retrieval effectiveness. The selection methods all give better results than the retrospective F_4 baseline. The best performing technique for each collection and condition was statistically better than the next best performing technique for the CACM, MEDLARS and AP (NW) collections.

For all collections, weighting gives better overall performance than no weighting.

The most unusual case is the performance of the Feedback 3 strategy, when using weighting. This test not only performed more poorly than the Feedback 2 and Feedback 3 strategies but also performed more poorly when used retrospectively than predictively.

The Feedback 3 strategy uses three types of weights: index weights attached to terms, RF weights derived from analysing the relevant documents and weights use to reflect the relative importance of the characteristics. The index weights and characteristics weights are identical in the predictive and retrospective strategies, and the RF weights do give an increase in the non-weighting condition, so it appears that some interaction of the three are responsible. A deeper analysis is necessary to uncover the underlying problem.

Collection/ Condition	**No feedback**	**Best Comb**	**F_4**	**Fback 1**	**Fback 2**	**Fback 3**	**Fback 5**
CACM	25.28	30.26	27.02	39.9	39.68	37.65	**44.38**
NW		19.70%	6.88%	57.83%	56.96%	48.93%	**75.60%**
CACM	24.34	25.68	25.67	39.28	39.27	38.01	**43.76**
W		5.51%	5.46%	61.38%	61.34%	56.16%	**79.81**
CISI	11.66	12.87	13.21	19.48	19.68	20.3	**21.75**
NW		10.38%	13.29%	67.07%	68.78%	74.10%	**86.61%**
CISI	12.02	12.84	13.56	20.06	20.52	20.83	**22.13**
W		6.82%	12.81%	66.89%	70.72%	73.29%	**84.03%**
MEDLARS	45.92	48.64	47.87	52.59	51.68	56.13	**60.05**
NW		5.92%	4.25%	14.53%	12.54%	22.23%	**30.78%**
MEDLARS	45.29	47.29	47.28	51.67	50.43	56.66	**60.11**
W		4.42%	4.39%	14.09%	11.35%	25.10%	**32.72%**
AP	12.04	15.31	12.64	17	16.53	**18.61**	18.28
NW		27.16%	4.98%	41.20%	37.29%	**54.57%**	51.81%
AP	14.09	14.09	14.16	19.01	18.4	**19.91**	19.52
W		0.00%	0.50%	34.92%	30.59%	**41.31%**	40.55%
WSJ	13.33	15.65	13.73	15.13	**17.35**	15.57	16.54
NW		17.40%	3.00%	13.50%	**30.16%**	16.80%	24.06%
WSJ	15.73	15.73	15.88	16.66	17.9	15.95	**17.99**
W		0.00%	0.95%	5.91%	13.80%	1.40%	**14.33%**

Table 4.17: Summary of retrospective RF experiments
Figures in bold represent the highest increase in average precision for each case

For the smaller collections (CACM, CISI and MEDLARS) the Feedback 5 strategy was again the best technique, for the AP collection Feedback 3 was the best technique and for the WSJ either Feedback2 (NW) or Feedback 5 (W) was the best technique. This result suggests that when we have complete relevance information we can assign better discriminatory weights to the combination of term and characteristics. Selection of characteristics in this case may become unnecessary due to the better information we have on the quality of the characteristics of the query terms. However this holds less well for larger collections (AP and WSJ) where some kind of selection seems to be important.

4.7.3 Characteristics used in feedback

In this section I examine the characteristics that were selected in each of the selection feedback algorithms. In particular I concentrate on the Feedback 1 strategy, which selects characteristics for query terms and the Feedback 2 strategy, which then selects terms across documents. This is intended to analyse the performances of the feedback algorithms by which characteristics they selected in the feedback runs. Table 4.18 summarises the characteristics used in the Feedback 1 strategy (in which characteristics are selected for the query) and Table 4.19 summarises the characteristics used in the Feedback 2 strategy (in which characteristics are also selected for each document). The Feedback 3 strategy is basically the same as Feedback 1, the only difference being the addition of the discriminatory weights. As such I concentrate only on the difference between selecting term and document characteristics for the query (Feedback 1) and for the documents (Feedback 2).

The predictive cases (Columns 3 and 4) are averaged over four iterations of feedback. As the use of weighting changes the ranking of documents at each iteration, different relevant documents will be used for feedback in the weighting and non-weighting conditions. Consequently the figures for the two conditions are different. The retrospective case is measured over all the relevant documents and so the results of the selection procedures are identical for the non-weighting and weighting conditions (Column 5).

For the Feedback 1 strategy, the selection of characteristics tended to follow the quality of the characteristics as retrieval algorithms: characteristics that performed well as a retrieval function tended to be selected more often in RF. This seems intuitively correct: the characteristics that are better indicators of relevant are more likely to be selected.

There was very little difference between the characteristics selected in the weighting and non-weighting characteristics for the Feedback 1 strategy. The only exception to this was the CACM collection. For this collection the non-weighting condition showed a much higher percentage of characteristics were chosen across the query terms. This high use of

characteristics does not, however, appear to have improved retrieval effectiveness as the Feedback strategies performed worse than the Best Combination method for the non-weighting condition on the CACM (Table 4.18). The use of fewer characteristics in the weighting condition did help the retrieval effectiveness of the Feedback strategy.

Over all the collections there was a greater use of characteristics (more characteristics were selected for each query term) in the retrospective strategy than in the predictive strategy. The retrospective techniques base their selection on the difference between the relevant documents and the rest of the document collection, whereas the predictive strategies base the selection decision on the difference between the relevant and non-relevant on a sample of the top-ranked retrieved documents. As the latter set of documents may be relatively similar, the averaging procedure used to decide which characteristics are selected may not be able to differentiate good characteristics as well in the predictive as in the retrospective case.

Table 4.19 analyses the usage of characteristics in the Feedback 2 strategy. I shall recap this strategy with an example: if the *tf* value of query term *t* is selected to form part of the query – is a good indicator of relevance - I first calculate the average *tf* value of *t* in the relevant documents. This average value is compared with the value of *t* in each remaining document in the collection that contains *t.* If the value of *t* in document *d* is greater than the average then we use the *tf* value of *t* to give a retrieval score to *d.*

Table 4.19 displays the percentage of documents that received a score using this strategy, e.g. on average, for the CACM collection, only 6% of the documents containing a query term, had a *tf* value for the term that was greater than the average relevant *tf*.

The *idf* and *noise* characteristics were used to score each of the remaining documents. These characteristics are based on global information and give the same value to a term in each document in which the term occurs. Consequently they cannot be used to differentiate between documents. The *idf* or *noise* characteristic of a term will always be greater than or equal to the average *noise* or *idf* value in the relevant documents and so the term will always be chosen to score documents in the Feedback 2 strategy. What differs in this strategy is the use of the document characteristics and the document-dependent term characteristics: *tf*, *theme*, and *context*.

As in the Feedback 1 strategy there was roughly a similar percentage of usage of characteristics in the weighting and non-weighting strategies. Comparing the predictive and retrospective strategies, there was a greater use of the term characteristics and less use of the document characteristics for the same reasons as for the Feedback 1 strategy.

The Feedback 2 strategy works better retrospectively than predictively, usually because it eliminates more poor characteristics and uses a higher proportion of better ones. However, the Feedback 2 strategy performed less well than the Feedback 1 strategy overall. This suggests that Feedback 2 method of eliminating weak evidence is not useful for RF.

4.7.4 Summary

The main findings from the feedback experiments are that selecting characteristics of query terms provides better retrieval effectiveness than re-weighting the terms (F_4) or selecting a good combination of terms for all queries. In addition, using some measure of the discrimination power of a term (Feedback 3) improves the performance over simple selection (Feedback 1) in predictive feedback. In addition, weighting the characteristics at indexing also improves the effectiveness of the query term characteristics.

Collection	Characteristics	Predictive no weighting	Predictive weighting	Retrospective weighting
CACM	*idf*	41	37	**60**
	tf	39	35	**60**
	theme	48	30	46
	context	**69**	24	38
	specificity	45	48	43
	noise	**61**	31	38
	info_noise	**55**	**60**	7
CISI	*idf*	33	33	**54**
	tf	32	31	**53**
	theme	22	22	38
	context	33	33	**57**
	specificity	48	43	32
	noise	34	34	**56**
	info_noise	**54**	**55**	**70**
MEDLARS	*idf*	**53**	**53**	**74**
	tf	**52**	**53**	**73**
	theme	**51**	**53**	**70**
	context	49	49	**72**
	specificity	37	43	43
	noise	**54**	**54**	**73**
	info_noise	40	39	40
AP	*idf*	**61**	**61**	**82**
	tf	**55**	**55**	**82**
	theme	42	42	**73**
	context	**55**	**55**	**75**
	specificity	39	39	**67**
	noise	19	19	16
	info_noise	39	39	25
WSJ	*idf*	**62**	**62**	**85**
	tf	**51**	**51**	**83**
	theme	43	40	**72**
	context	**54**	**53**	**77**
	specificity	42	39	**96**
	noise	12	12	8
	info_noise	21	22	7

Table 4.18: Characteristics used in Feedback 1 strategy.
bold figures indicate that a characteristic was used for the majority of terms

Collection	Characteristics	Predictive no weighting	Predictive weighting	Retrospective weighting
CACM	*idf*	**100**	**100**	**100**
	tf	24	29	**83**
	theme	21	20	34
	context	20	18	41
	specificity	45	38	17
	noise	**100**	**100**	**100**
	info_noise	**100**	**100**	**100**
CISI	*idf*	**100**	**100**	**100**
	tf	**65**	**67**	**90**
	theme	34	36	39
	context	**66**	**67**	**85**
	specificity	41	39	30
	noise	**100**	**100**	**100**
	info_noise	**100**	**100**	32
MEDLARS	*idf*	**100**	**100**	**100**
	tf	**55**	**55**	**87**
	theme	**52**	**53**	**64**
	context	**53**	**56**	**52**
	specificity	48	48	15
	noise	**100**	**100**	**100**
	info_noise	46	49	16
AP	*idf*	**100**	**100**	**100**
	tf	18	19	**54**
	theme	26	29	37
	context	5	6	17
	specificity	39	34	7
	noise	**100**	**100**	**100**
	info_noise	27	27	8
WSJ	*idf*	**100**	**100**	**100**
	tf	20	18	**51**
	theme	23	30	38
	context	4	5	18
	specificity	11	17	6
	noise	**100**	**100**	**100**
	info_noise	20	24	0

Table 4.19: Characteristics used in Feedback 2 strategy
bold figures indicate that a characteristic was used for the majority of terms

4.8 Conclusion

In this chapter I investigated three areas:

i. the performance of new term and document characteristics. These characteristics showed variable performance as retrieval functions. Characteristics that only weighted documents, and did not weight terms, performed relatively poorly as they are unable to distinguish potentially relevant from irrelevant documents. Even when

only ranking documents that contain a query term, the document characteristics still did not perform as well as term characteristics. The standard IR term weighting functions *idf* and *tf* performed well over all the collections tested.

ii. the performance of characteristics in combination. Combining characteristics to form a joint retrieval function was shown to be a good idea overall. Combination is successful for most characteristics but I have only outlined general indications of what makes a good combination of characteristics. It still remains difficult to predict more precisely how characteristics will perform in combination.

iii. the performance of characteristics in RF. Although it is difficult to predict how characteristics will perform in combination, the relevance assessments for a query can be used, predictively and retrospectively, to select a good set of characteristics for each query term. This method of feedback, generally, works better than choosing a single good set of characteristics to be used for all query terms and can work better than a single good discriminatory weighting function.

The work outlined in this chapter describes an analysis of term and document weighting in combination and in RF. A deeper analysis of what factors influence the success of each weighting scheme will require taking into account factors such as length of document, number of unique terms per document, number of relevant documents per query, etc. Such an analysis could be used to improve the selection procedure. Even though I have presented only general conclusions here, I believe that the main conclusions demonstrate that taking into account how terms are used can, and should, be considered further in document ranking. In particular the use of RF techniques for selecting which aspects of a term's use is appropriate for scoring documents, appears to be a useful approach for increasing the effectiveness of interactive IR systems.

The following chapter extends this analysis, using data derived from real user searches. In particular I aim to elicit information about the role of the *user* in making relevance assessments.

Chapter Five

Information use and relevance assessments

5.1 Introduction

In the previous chapter I demonstrated that it was possible to use multiple weighting schemes to incorporate information on how terms are used within documents. The use of these weighting schemes, *term* and *document characteristics,* can lead to significant improvements in retrieval effectiveness across collections. I also demonstrated, experimentally, that different combinations of characteristics are more suitable for different queries. In other words, different combinations of characteristics are better at detecting relevance for individual queries.

The proposed solution was to use relevance information to select which characteristics to use for each query term. This technique - *selective relevance feedback* - not only performed well but outperformed standard RF algorithms such as the F4 term weighting scheme, [RSJ76], when applied to data obtained from the TREC initiative, [VH96].

The work described in the previous chapter gave a broad outline for how information on term use could improve retrieval effectiveness but the data I used was limited in one important way: it lacked information on the *user* in the process of making relevance assessments. In this chapter I am interested in investigating factors that may affect how users make relevance assessments and how these relate to combination of evidence. Specifically I investigate the use of partial, or non-binary relevance assessments, and the effect of different search tasks on the success of combination.

In this chapter I present a separate analysis of the approach to combination of evidence described in the previous chapter. This analysis is based on queries and relevance assessments obtained from non-expert searchers searching on a mixture of genuine search tasks and artificially created tasks. The experiments were carried out after an initial pilot test of the previous experiments[37] and before the large-scale experiments in Chapter Four. Consequently

[37] Reported in [RL99].

only a subset of the characteristics used in Chapter Four – *idf*, *tf*, *theme* and *context* – were used in this chapter.

In section 5.2 I shall first describe how the data used in these experiments differ from that used in Chapter Four. In section 5.3 I shall describe the data in more detail and discuss why the differences between the two sets of data are important. In section 5.4 I shall discuss how the data was used in this chapter. This is necessary as certain assumptions that can be made about test collections do not hold for this data. In section 5.5 I shall examine combination of evidence and in section 5.6 I shall present the results of selective combination of evidence in RF. I shall summarise the main findings in section 5.7.

5.2 Background

The test collections used for the combination of evidence investigation in Chapter Four were of two kinds – small collections (CACM, CISI, MEDLARS) and larger collections (AP and WSJ). The smaller collections contain small numbers of documents and we can assume relatively complete relevance information – we know which are the relevant documents and which are not relevant to individual queries.

The relevance assessments for the larger TREC test collections used in Chapter Four are made only on a sample of the documents retrieved by a number of retrieval systems [VH96]. The documents are retrieved and assessed using relatively detailed descriptions of what constitutes a relevant documents, e.g. Figure 5.1 for an example of such a *topic*[38].

> **Number**: 301
> **Title**: International Organized Crime
> **Description**:
> Identify organisations that participate in international criminal activity, the activity, and, if possible, collaborating organisations and the countries involved.
> **Narrative**:
> A relevant document must as a minimum identify the organisation and the type of illegal activity (e.g., Columbian cartel exporting cocaine). Vague references to international drug trade without identification of the organisation(s) involved would not be relevant.

Figure 5.1: TREC topic 301

[38] Which sections of the topic were used for retrieval varies according to the test collection

These topics are created by the assessors who will make the final relevance assessments, i.e. the same people who will decide which documents are relevant to the topic. The topics are intended to reflect personal 'user needs', [VH00]. Although it is the same people who create the topics and make the relevance assessments, there are differences between the TREC method of creating relevance assessments and when users assess documents. For example, there is a time delay in making the TREC assessments: the topics are created some three months before the assessments are made. This means that the situational and dynamic aspect of making relevance assessments that may be important for users is lacking. The TREC assessors are also given instructions on how to determine whether a document is relevant, e.g. a document is relevant if at least one sentence is relevant. These criteria may be very different from how users assess relevance relative to an individual information need.

To validate the techniques investigated in Chapter Four as an *interactive* technique it is necessary to assess them within a more realistic searching environment. This chapter describes such an investigation. In the following section I shall discuss the data used in the experiments contained within this chapter.

5.3 Data

The data (documents, queries, relevance assessments) I used in these experiments came from a previous set of experiments carried out by Borlund and Ingwersen [BI99]. In sections 5.3.1 and 5.3.2, I give a brief description of the document collection and experimental setting used in [BI99] to generate the queries and relevance assessments. In section 5.3.3, I discuss the queries and relevance assessments that I used for the experiments described in this chapter. In section 5.3.4, I summarise the important aspects of the data.

5.3.1 Document collection

The data used in these experiments came from a combination of the Financial Times (FT) and Herald Collections. The Herald collection consists of 135,477 full-length newspaper articles from January 1995 to May 1997. The Financial Times consists of 174,075 full-length newspaper articles and covers the period from May 1991 to September 1994[39].

5.3.2 Experimental setting

The relevance assessments and queries used in the experiments presented in this chapter were obtained from a series of experiments using a full-text on-line system with an underlying

[39]Borlund and Ingwersen were forced to exclude part of the Financial Times data (from the period October 1994 to December 1994) due to limits on the amount of data their system could index.

probabilistic-based retrieval engine, [Cam90]. 23 university students volunteered as subjects for the experiments. The subjects (19 male students, 4 female students) were from various academic fields and educational levels, e.g. computing, mathematics, geography, biochemistry, language, English history, psychology etc., and were a mixture of graduate and undergraduate students. The subjects had varying experience of IR systems but most could be regarded as novice users for the purpose of the study.

Each subject was asked to search on 6 search *topics*; one training topic to familiarise them with the system being used, four simulated topics created by Borlund and Ingwersen and one topic which the subjects were asked to create themselves. The simulated topics consisted of two parts: a simulated work task situation, a description of a situation which may promote an information need, and an indicative request, a suggestion to the subject of how a search may be initiated. A subject was either given only the simulated work task or both the simulated work task and indicative request. The task given to the subjects was to find useful information for each topic[40]. Figure 5.2 shows an example of a simulated topic.

> **Simulated work task situation:** After your graduation you will be looking for a job in industry. You want information to help you focus your future job seeking. You know it pays to know the market. You would like to find some information about employment patterns in industry and what kind of qualifications employers will be looking for from future employees.
> **Indicative request:** Find for instance something about future employment trends in industry, i.e. areas of growth and decline.

Figure 5.2: Example simulated topic

The subjects were presented each topic, in permutated order, and were given complete freedom regarding how they searched and how they generated query terms to put to the IR system. For each query 39 documents were retrieved and presented to the subject in groups of 12: the user could move between sets of 12 retrieved documents at will. The users were not asked to assess all retrieved documents or to assess a minimum number of retrieved documents.

One feature of these experiments was the use of *partial relevance assessments*. The subjects indicated the relevance scores by use of a slider, Figure 5.3, that was incorporated into the

[40]The simulated work task situations and indicative requests, used in [BI99], are shown in Appendix F. All subjects were shown the simulated work task situation, whether they were shown the indicative request was an experimental variable.

interface, and shown at the screen next to the title field and the field viewing the full-text documents. The subjects based their relevance assessments on either the title or the full-text of the document, and could indicate the *degree* of relevance of the assessed documents according to the relevance categories of: low, medium, and high relevance. Internally, the categories corresponded to 11 relevance levels: integer values 0 - 10, with 0 as the default relevance score signifying non-relevance.

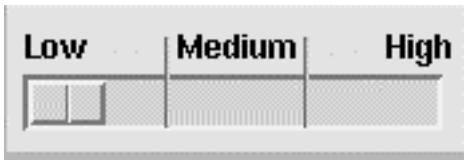

Figure 5.3: Slider used to assess relevance of documents

The search activities were logged, including the subject's relevance scores for the retrieved documents.

5.3.3 Queries and relevance assessments

Each subject was asked to supply at least one query for each topic. The subject could refine and resubmit the query, or use a new query as often as necessary or as desired in the course of the search. In Table 5.1, I present the total number of queries given for each topic. I excluded any query formulation for which no relevant documents were found.

The total number of queries given by all subjects for all topics is 246, an average of 1.8 queries per topic. This total assumes that any document to which a subject has assigned a relevance score of 1 or above - *relevance level* 1 - counts as relevant. As this threshold is increased, by asserting, for example, that a document must receive a relevance score of 5 to count as relevant then the number of valid queries changes. At relevance level 7, for example, we only have 161 queries. This means that only 161 out of the original 246 queries have at least one document assessed at relevance level 7. Thus, as seen in Table 5.1, the number of queries changes at each relevance level, the number of queries decreasing as the level of relevance increases.

Cross-comparing the number of queries for each topic (using a paired *t*-test, holding relevance level constant and varying number of queries, $p < 0.05$) we find that the training topic (TR) had significantly fewer valid queries than the other topics, while topic D, had significantly more valid queries. This difference in numbers of valid queries per topic suggests that some topics may have been more difficult than others.

Relevance Level	Topics							
	A	**B**	**C**	**D**	**Own**	**TR**	**Total**	**Average**
1	44	41	43	53	39	26	**246**	41
2	44	40	43	52	38	26	**243**	41
3	43	39	42	48	36	25	**233**	39
4	39	37	39	45	36	23	**219**	37
5	36	36	34	41	33	22	**202**	34
6	29	32	30	36	30	18	**175**	29
7	26	29	28	30	30	18	**161**	27
8	22	27	21	24	22	18	**134**	22
9	17	25	17	17	20	17	**113**	19
10	8	17	13	11	15	8	**72**	12

Table 5.1: Numbers of queries for each task at each of the ten relevance levels

In Table 5.1, Topics A - D are based on simulated information needs[41], Topic Own is based on the subject's own information need and Topic TR is the training topic.

The average number of terms per query was four terms, averaged over all topics, whereas the average number of query terms for the subjects' own topic was three query terms. These values are similar to the average query length of the larger TREC collections I used in Chapter Four, Table 4.2. This value is also consistent with the number of query terms typically entered to web search engines, [JAS+00].

5.3.4 Summary

This data is quite different from the data used in the previous chapter in a number of ways:

• the experimental subjects relatively inexperienced at making relevance assessments and were given no criteria as to how to assess relevance. This is in contrast to the TREC topics where detailed information was supplied on how relevance was to be decided.

• the subjects can give partial relevance assessments rather than a binary, relevant or not relevant, assessment. This is contrast to the collections used in Chapter Four, where only binary relevance assessments were considered.

[41] I shall use the more general term *simulated information needs* to refer to the simulated work tasks introduced in section 5.3.2.

• the search topics consist of a mixture of simulated and original information needs. The TREC topics can be considered as genuine information needs as they are written by the same people who assess relevance.

In the remainder of this chapter I re-examine the findings from Chapter Four on this new data, by running a series of similar experiments to those in Chapter Four. In particular I will examine three research questions.

i. how the results from TREC relevance data differed from that derived from non-expert assessors. In the data used in this chapter relevance was assessed by subjects searching for information using potentially developing information needs with no given criteria for assessing relevance. In this case, do the previous results, obtained using relatively fixed information needs, hold?

ii. is there any difference between assessments made on real and *simulated* information needs? The data I used in these experiments came from subjects performing searches on both real and simulated information needs. When a subject is assessing relevance for a *given* (simulated) information need do they use different criteria from when they are searching for a *personal* information need?

iii. how partial relevance affected the results. I am interested in how the use of non-binary relevance assessments affected the results of my approach.

In the next section I shall describe how I prepared the data for the experiments.

5.4 Preparation of data

In the data I used in this chapter all the relevant documents were retrieved by the system. This is because only documents that were *retrieved* by the system could be *assessed*. Therefore, in this set of experiments I only aim at a form of *precision enhancement*. Instead of trying to retrieve *more* relevant document I am only attempting to improve the order in which the user-selected relevant documents were retrieved.

The documents in [BI99] were retrieved using the *idf* function. A good precision enhancement algorithm should therefore re-rank the documents retrieved by the user's query in a better order than the default *idf* function.

A natural approach would be to re-score all the documents in the collection or just the documents retrieved by the user's query. However, as I shall argue in the remainder of this section, this is not appropriate and only the documents the user *assessed* should be considered. The result is that the precision enhancement functions are only re-ranking a subset of the retrieved documents.

In section 5.4.1, I shall present the results of ranking the retrieved documents using each of the four term characteristics. In section 5.4.2, I shall show the results of these retrievals change when we use different subsets of the retrieved documents. In section 5.4.3, I shall summarise the differences between these results, why these differences occur and which set of documents I shall use for the remainder of this chapter.

5.4.1 Retrieval by single characteristic

In this experiment I carried out a retrieval using each characteristic as a single retrieval function (ranking the retrieved documents only by *idf* score of each query term, ranking the retrieved documents only by *tf* score, etc). I treated all the queries as a single set of queries, regardless of which experimental subject issued the query. The overall performance of each characteristic is measured by the average precision of the characteristic on all queries. The average precision was calculated at each *relevance level*, e.g. at relevance level 1 all documents which a relevance score[42] of at least 1 counted as relevant, at relevance level 2 a document must have received a score of 2 to count as relevant.

Table 5.2 summarises the results for each of the four characteristics at each relevance level. For the *theme* and *context* characteristics there is a steady drop in average precision from relevance level 1 to relevance level 10. For the *idf* and *tf* characteristics there is also a steady drop until relevance level 8 when the average precision starts to increase again. Only at relevance levels 9 and 10 does any characteristic outperform the original *idf* ranking. For the majority of relevance levels, then, *idf* is the optimal ranking. Using a paired *t*-test, holding relevance level constant and varying average precision ($p < 0.05$) the difference between each pair of characteristics is statistically significant, i.e. *idf* significantly better than *tf*, which is significantly better than *theme* which is better than *context*.

[42] This was the score assigned to the document by the experimental subject.

Level	*idf*	*tf*	*theme*	*context*
1	**52.30**	35.70	32.20	30.60
2	**47.00**	32.90	28.80	27.50
3	**43.70**	31.20	26.50	25.40
4	**41.20**	29.80	24.60	22.80
5	**38.90**	28.40	23.30	21.70
6	**34.10**	27.60	21.20	19.00
7	**32.10**	26.00	20.00	17.90
8	**32.30**	31.10	20.50	18.00
9	29.70	**32.60**	19.90	16.50
10	32.20	**33.00**	19.80	14.60

Table 5.2: Average precision values for each of the four characteristics at each relevance level

In the next section I show how these results can change when using different subsets of the data.

5.4.2 Effects of the default ranking

The default ranking, the one that ordered the documents for presentation to the experimental subjects, was the *idf* ranking. From Table 5.2, it would appear that for the majority of relevance levels *idf* is the optimum weighting scheme. However, the fact that *idf* was the default ranking could bias retrieval performance in favour of *idf*. This bias could result from two sources:

i. The user is not forced to assess or even view all the retrieved documents for a query. Unlike the TREC experiments, [VH96], and other Cranfield[43]-like test collections, the subjects were not asked to assess a complete set or subset of the documents. Most subjects started assessing documents at the first document and worked their way down the list, assessing the full text of some documents, or assessing/viewing the title of others. If the subject stops assessing documents part of the way down the ranking, e.g. they have found enough relevant information, or they stop when they view the first non-relevant document, then the relevant documents[44] will only appear at or above the last assessed/viewed document. The rank position of the last

[43] This label describes the model of test collections described in Chapter One, and refers to early work on test collections carried out as part of the Cranfield Research Project [CK66].

[44]These are the documents assessed as being relevant, and does not include those documents that would have been assessed relevant if viewed or assessed by the subject.

relevant document then becomes a threshold - all the relevant documents appear at, or above, this rank position and all the documents below it are considered irrelevant.

When re-ranking the documents by characteristics other than *idf*, documents below this threshold position can be placed above the threshold. This means, in short, that the relevant documents for the *idf* ranking will only appear between rank position 1 and the threshold position, whereas for the other rankings the relevant documents may appear at any rank position in the retrieved document set. We are not then dealing with identical sets of documents for the different characteristics. There is, then, an inherent bias in favour of ranking by *idf* due to the way the documents were presented for assessment.

This is important as the analysis in this chapter is intended to investigate how term and document characteristics would perform when a real-life searcher was making the relevance assessments. If I do not adequately replicate this real-life search then any conclusions may be faulty.

I re-ran the experiment trying to estimate the effects of experimental subjects only assessing part of the ranking in two ways. In section 5.4.2.1, I excluded all queries in which all the relevant documents appeared consecutively at the top of the *idf* ranking, i.e. the user stopped assessing relevance at the first irrelevant document, or no relevant document appeared below an irrelevant document. In section 5.4.2.2, I only consider the part of the ranking that I know the subjects have at least viewed - from the first document to the last marked relevant document.

ii. The order in which the documents are *assessed* is important. Authors such as [FM95, EB88] point to the importance of the position of a document in a ranking when assessing the relevance of the document. Florance and Marchionini, [FM95], for example discuss how relevance assessment scores can change in the light of seeing new documents. In section 5.6.4, I looked at how the order in which the documents were presented and assessed could affect which characteristics performed well as a single retrieval algorithm.

5.4.2.1 Retrieval by single characteristic - excluding queries with consecutively relevant documents

The results from excluding all queries in which all the relevant documents are at the top of the ranking are shown in Table 5.3. The general trend is that the average precision figures for *idf* fall, whereas the average precision figures for the rest of the characteristics increase. The results from comparing the average precision figures at each relevance level for each characteristics show that the difference between *idf* (Tables 5.2 and 5.3) and *tf* under both conditions are statistically significant - removing these queries does have a significant effect on the

performance of these characteristics. However, *idf* still gives the best average precision for the majority of the relevance levels, with *tf* outperforming *idf* at relevance levels 9 and 10. Using a paired *t*-test, holding relevance level constant and varying average precision ($p < 0.05$), we have the same ordering of significance, *idf* significantly better than the other three characteristics, *tf* better than *theme* and *context*, and *theme* better than *context*.

Level	*idf*	*tf*	*theme*	*context*
1	**48.90**	36.30	33.10	31.10
2	**43.30**	33.80	29.90	28.00
3	**39.90**	31.60	26.80	25.10
4	**37.70**	31.50	24.60	22.40
5	**35.30**	29.50	22.80	21.00
6	**31.20**	28.30	20.60	18.20
7	**29.80**	27.20	19.70	17.00
8	**32.30**	31.50	20.60	17.90
9	29.60	**32.60**	20.00	15.90
10	31.40	**33.10**	20.00	14.70

Table 5.3: Average precision values for each of the four characteristics at ten relevance levels, ignoring rankings in which all relevant documents are at the top of the ranking

5.4.2.2 Retrieval by single characteristic - excluding non-viewed documents

The results from only considering the documents retrieved at or above the last marked relevant document are shown in Table 5.4. In this version of the experiment, at all relevance levels *tf* is the optimal characteristic. In addition, with the exception of relevance level one, *tf* performance is followed by *theme*, *context* and finally *idf*. This means that if I only consider the set of documents that I believe the user has assessed then *idf* is actually the poorest characteristic to rank documents, contradictory to the findings in Table 5.2. Using the test of statistical significance as before, *tf* is better than *theme*, which is better than *context*, which is better than *idf*.

It can, then, be argued that ranking all the retrieved documents does introduce a bias into the experiments in favour of the default *idf* ranking.

Level	*idf*	*tf*	*theme*	*context*
1	52.66	**56.82**	55.11	50.45
2	52.66	**54.13**	51.41	47.38
3	44.49	**51.62**	48.90	45.33
4	41.69	**49.60**	46.36	42.15
5	38.96	**46.85**	43.98	40.18
6	34.12	**44.23**	40.37	36.56
7	32.19	**42.80**	38.91	35.05
8	32.41	**48.08**	40.17	36.76
9	29.80	**49.88**	39.91	36.59
10	32.33	**41.77**	40.14	32.90

Table 5.4: Average precision values for each of the four characteristics at ten levels of relevance, ranking only the first document to the last relevant document.

It can, then, be argued that ranking all the retrieved documents does introduce a bias into the experiments in favour of the default *idf* ranking. Therefore, for the experiments in the remainder of the chapter, I had to decide whether I base the calculations on either:

i. *all* retrieved documents retrieved for a query and retaining the default ranking bias, or

ii. only the *subset* of documents that I assume the user has assessed and possibly not considering documents that the user has assessed as being not relevant (ones that appear below the last relevant document). This also has the effect of cutting the number of documents ranked for each query.

5.4.3 Summary

The difference between retrieved and assessed is given by the difference between Table 5.2 and Table 5.4. If all the retrieved documents are considered, Table 5.2, then *idf* is generally better at retrieving relevant documents first – it gives better average precision. This means that *idf* is the best characteristic for differentiating between the relevant documents and the retrieved documents. However if only the assessed documents are considered, Table 5.4, then *tf* is a better characteristic in that it discriminates better between the *assessed relevant* documents and the *assessed non-relevant* documents. As *tf* gives the best average precision in discriminating between the assessed documents, it is plausible to argue that *tf* is the aspect that the user employed to differentiate between documents in this particular experiment.

I opted for position **ii**, that is I only consider the assessed documents. This is because the experiments are designed to test whether different rankings *would* have been better if shown to the user. This can only be based on the documents the user assessed.

From this point, for clarity, I shall refer to the subset of documents (from the first document to the last relevant document) as the *assessed documents*. I am aware that not all the documents in this set will have been assessed by the experimental subject but I can guarantee that the subject has at least *seen* the title of the document and has made some implicit judgement on the relevance of the document.

In the remainder of this chapter I shall repeat the main experiment from the previous chapter. I shall examine the findings under four main conditions: performance at different relevance levels, the tasks set to the user, the order in which relevant documents were retrieved, and the performance of the combination strategies for users.

I shall discuss retrieval by single characteristic in section 5.5, combination of characteristics in section 5.6 and selective combination of characteristics in section 5.7.

5.5 Experiment one – retrieval by single characteristic

I have presented the averaged results of retrieving documents by each characteristic in Table 5.4. This, in effect, meant running each characteristic as a precision enhancement function: re-ranking the assessed documents using a different characteristic to the default *idf* retrieval function. In section 5.5.1, I shall look at how the characteristics perform when the relevance level changes. In section 5.5.2, I shall discuss how well the characteristics order the relevant documents. In section 5.5.3, I shall summarise how the characteristics perform for individual users and in section 5.5.4, I shall examine the effectiveness of the characteristics for the different search tasks.

5.5.1 Relevance level

From Table 5.4, is can be seen that, as the relevance level increases the power of all characteristics to differentiate relevant material falls. That is, the characteristics are less good at ranking documents when the threshold for relevance is high. However not all characteristics perform as poorly as each other. The power of *idf* to discriminate relevant material at relevance level 10 is around 62% of its power at relevance level 1[45], compared to 73% for *tf*, 73% for *theme* and 65% for *context*. *idf* is then less *stable* at identifying relevant material across the

[45] Percentage of average precision at relevance level 10 compared to average precision at relevance level 1.

relevance levels: the other characteristics not only perform better at high relevance levels but also at a higher percentage of their maximum performance.

5.5.2 'Perfect' rankings

In sections 5.4.2.1 and 5.4.2.2 I treated each relevance level as a filter: all documents with a relevance score less than the level being tested is regarded as being non-relevant. However, with partial relevance assessments, the quality of a retrieval algorithm is not only given by how many relevant documents it retrieves but also by how it *orders* the relevant documents: a good retrieval algorithm should place the most relevant documents at higher rank positions than less relevant documents. I carried out a new experiment to assess each characteristic as to how well it ranked the assessed documents and ordered the assessed documents

To calculate how good a ranking was in terms of how it ordered relevant documents I defined a function, *ranking_score*, Equation 5.1, to give a value to a document ranking based on a set of relevance assessments.

$$ranking\_score(ranking) = \sum_{i}^{N} s\_score(d_i) * {}^{1}\!/_{rank\_pos(d_i)}$$

Equation 5.1: *ranking_score* function
where N = number of assessed documents, $s\_score(d_i)$ is the relevance score given to document d_i by the user, and $rank\_pos(d_i)$ is the position of the document (d_i) in the ranking.

This equation gives higher values to rankings in which the documents with the highest relevance scores are further up the ranking and documents with lower relevance scores appear below highly relevant documents. The equation implicitly gives more importance to documents that appear higher in the ranking, that is the relative order of relevant documents is more important at the top of the ranking.

The strategy to test the different rankings given by the four characteristics (*idf*, *tf*, *theme*, *context*), for each set of assessed documents for a query, was as follows:

i. rank the documents in order of relevance score given by the subjects to achieve the 'perfect' ranking. This ranking has all the relevant documents consecutively at the top of the ranking, in decreasing order of relevance score given by the user,

ii. calculate the *ranking_score* value for the 'perfect' ranking to obtain the optimal score

iii. calculate the *ranking_score* value for each of the rankings given by the four characteristics and compare this with the *ranking_score* for the 'perfect' ranking.

Table 5.5 shows the results of this[46], and shows that *idf* ranked the relevant documents in a better order than any of the other characteristics. At all relevance levels, *theme* outperformed *context*, followed finally by *tf*.

From Table 5.5, it would appear that the *idf* function ranks documents in a better order than any of the other functions: the documents with higher relevant scores appear further up the ranking and those with lower scores appear further down the ranking. The *idf* function also improves most across relevance levels - the difference between the perfect ranking and the *idf* ranking at relevance level 1 is greater than at relevance level 10, compared to the other characteristics. This means that *idf* is better at ordering documents at higher than at lower levels of relevance.

Level	Perfect	*idf*	*tf*	*theme*	*context*
1	14.65	**7.09**	4.6	5.83	4.88
2	14.72	**6.96**	4.49	5.68	4.79
3	14.89	**6.86**	4.41	5.64	4.68
4	15.17	**6.8**	4.35	5.61	4.6
5	15.54	**6.7**	4.34	5.58	4.6
6	15.86	**6.34**	4.12	5.56	4.5
7	15.62	**5.96**	3.8	5.16	4.24
8	15.68	**5.82**	3.86	5.18	4.18
9	14.86	**4.99**	3.31	4.81	3.73
10	16.02	**5.64**	3.79	5.41	4.17

Table 5.5: Ordering performance of each single characteristic measured against 'perfect' ordering of relevant documents within the assessed set.
Highest individual characteristic performance in bold.

This could either be a factor of the different characteristics (*idf* retrieves documents that are more relevant in a better order) or a factor of the way users assess relevance (they are more likely to assess later documents relative to first documents). From the data available I cannot distinguish between these two cases, nevertheless there is a consistent difference in which characteristics ordered the relevant documents. I shall return to this point in section 5.6.3.

[46]The closer the *ranking_score* of a characteristic is to the 'perfect' ranking value, the better the characteristic is as ranking relevant documents.

A final observation is that, although *tf* is better at retrieving relevant documents, it does not appear to rank the highly relevant documents better. It may be then, that *tf* is successful in retrieving *likely* relevant documents but other characteristics are better at indicating *high* relevance.

5.5.3 Characteristics for individual subjects

In Experiment One, so far, I have treated all the queries as a single set, regardless of which subject issued the query. Table 5.6 outlines for how many subjects each characteristic was the optimum characteristic for that subject's queries i.e. comparing average precision for the set of queries issued by a subject. As can be seen the *tf* characteristic was optimal for the majority of subjects, followed by *theme*, *context* and *idf*.

However, at most only two thirds of the subjects[47] had *tf* as the optimal characteristic, other characteristics were better, on average, at retrieving relevant documents for the queries issued by the remaining third of the subjects. That is, *tf* is optimal overall but sub-optimal for a number of users.

	idf	***tf***	***theme***	***context***
1	5	**12**	6	0
2	6	**10**	5	2
3	1	**11**	7	4
4	0	**14**	5	4
5	1	**14**	4	4
6	1	**12**	6	4
7	1	**13**	5	4
8	0	**15**	4	4
9	0	**12**	9	3
10	2	7	**9**	3
total	17	**120**	60	32

Table 5.6: Numbers of users, at each relevance level, whose queries had highest average precision by different characteristics
bold figures indicate highest number of users

[47] That is the total for *tf* divided by the sum of each row in Table 5.6.

5.5.4 Performance by topic

In section 5.5.2 I analysed the performance of the characteristics as a single set of queries, in section 5.5.3 I analysed them by which characteristics performed best for each users, in this section I analyse the results by *topic*. Table 5.7 shows the average precision figures for each topic at each level of relevance.

		Relevance level									
Topic	**Char**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	***idf***	51.16	45.74	42.08	41.65	37.49	33.62	35.47	33.34	27.24	25.83
	tf	**55.28**	**51.65**	**49.53**	**48.25**	**45.94**	**42.32**	**41.95**	**43.03**	**49.62**	**38.07**
	theme	52.10	47.48	47.14	44.42	43.32	39.41	39.44	38.53	37.08	37.21
	con	46.43	42.75	41.88	41.78	40.82	36.16	39.40	41.64	38.48	36.31
B	***idf***	47.47	46.05	43.55	43.57	40.18	36.39	31.46	31.03	30.61	31.36
	tf	**51.37**	**49.41**	46.56	47.04	**43.75**	38.43	36.28	**42.74**	**39.47**	**33.91**
	theme	50.32	49.09	**47.02**	**47.25**	43.68	38.71	**36.35**	36.57	34.25	29.99
	con	48.64	47.28	43.96	44.98	42.16	**39.01**	33.44	33.80	32.68	29.84
C	***idf***	52.45	49.83	48.64	43.77	40.54	30.53	30.41	28.98	27.26	26.58
	tf	**59.21**	**59.66**	**57.55**	**53.81**	**51.56**	**44.53**	41.38	**41.19**	**47.48**	33.68
	theme	59.08	56.15	53.78	48.91	47.56	42.30	**42.78**	40.09	41.69	**36.30**
	con	49.13	47.09	48.09	42.91	41.40	32.91	34.72	36.80	41.30	34.27
D	***idf***	50.34	44.24	39.48	37.23	35.51	33.86	33.90	37.67	38.35	35.00
	tf	**57.86**	**54.16**	**49.78**	**50.00**	**44.87**	**43.41**	**45.43**	**44.75**	42.96	41.68
	theme	55.26	51.39	45.83	45.58	42.95	41.71	44.07	44.18	**48.49**	**56.05**
	con	50.49	47.64	43.78	40.59	38.22	35.04	32.67	36.54	36.73	33.54
Own	***idf***	55.23	46.60	42.28	37.68	34.22	32.74	30.49	30.66	26.44	26.30
	tf	**57.21**	**51.99**	**49.94**	45.94	**44.20**	**43.91**	**41.38**	**40.24**	31.89	**30.84**
	theme	56.61	51.50	47.99	**45.97**	42.86	39.49	36.64	36.21	**33.49**	30.51
	con	52.27	47.31	42.38	38.83	37.99	38.67	35.34	33.82	30.08	28.95
TR	***idf***	55.01	49.37	44.75	42.96	42.60	40.07	36.45	32.55	23.94	30.54
	tf	**58.60**	**56.42**	**52.56**	**52.09**	**52.00**	**58.13**	**55.00**	**55.53**	**53.19**	**57.06**
	theme	57.89	53.30	49.07	45.93	45.74	45.89	42.56	41.81	37.17	44.75
	con	54.85	52.14	50.37	45.16	44.10	42.05	38.60	34.81	28.32	29.50

Table 5.7: Average precision figures for single characteristics across topics.
Highest value shown in bold. *con* = context

The most common trend arising from this table is that *tf* gives consistently good results across relevance levels and topics. The *theme* characteristic also performed well generally giving the second highest average precision figures across the topics. There was no noticeable difference between the simulated topics and the topics created by the users, nor was there a noticeable difference between the performance of characteristics at the different relevance levels.

5.5.5 Summary of experiment one

In this section I summarise the results from Experiment One as I have constructed it: re-ranking the documents I assume the user has viewed or assessed by the individual characteristics. It is clear that the *tf* characteristic outperforms the other three characteristics in a number of ways: it gives better average precision figures (section 5.4.2.2) across the set of subjects' queries, it gives better performance across the topics (section 5.5.3) and it gives better results for the majority of subjects (section 5.5.4).

Although *tf*, followed by *theme*, does give better results under these conditions, for a significant number of subjects, Table 5.7, a different characteristic gives better results for their queries. So, using only *tf* for all query terms is better than any other individual characteristic but is not guaranteed to be optimal for all queries and for all users. This conclusion leads to the first hypothesis:

hypothesis one - combination of evidence: adding more information on how terms are used within documents will improves retrieval performance. That is, the more information we have on a term's usage, the more precisely we are able to detect what indicates relevance.

If no single retrieval function can be guaranteed to give optimal results for all users, then perhaps *combining* different retrieval functions can give better results for more subjects. I look at this in section 5.6.

5.6 Experiment Two - retrieval by combination of characteristics

In Experiment Two, as in Chapter Four, I tested if retrieval performance would increase by adding more information on term usage: scoring documents by more than one characteristic of each query term.

In sections 5.6.1.1 – 5.6.1.3. I discuss the results of combining sets of 2 characteristics, 3 characteristics and all 4 characteristics. In section 5.6.2 I describe the variant of this experiment that treats the characteristics as being of varying importance. In these sections I shall simply

present the results and the main findings from the combination of evidence. In sections 5.6.3 – 5.6.6 I shall examine the results for the effects of task, relevance level, and user. I shall summarise the overall combination of evidence experiments in sections 5.6.7 and 5.6.8.

5.6.1 Retrieval by addition of characteristic scores

In this experiment, I followed the methodology for Experiment One, and re-ranked the assessed documents by the sum of each 2-way, 3-way and 4-way combinations of characteristics. Documents were again scored by the sum of the characteristic weights, e.g. the sum of the *theme* weights for each query term in the document plus the sum of the *tf* weights.

5.6.1.1 Retrieval by combination of two characteristics

Table 5.8 shows the results of combining each combination of two characteristics of query terms, compared against *tf* - the best overall single characteristic. The main result is that no combination of two characteristics gives better average precision than *tf* at any relevance level, although the combination of *tf* and *idf* comes very close to *tf* performance.

Level	*tf*	*idf* + *tf*	*idf* + *theme*	*idf* + *context*	*tf* + *theme*	*tf* + *context*	*theme* + *context*
1	**56.82**	54.36	52.70	48.22	52.74	50.39	50.32
2	**54.13**	51.75	49.14	45.19	49.30	46.75	46.95
3	**51.62**	49.36	46.75	43.19	46.88	45.02	45.00
4	**49.60**	47.42	44.33	40.12	44.40	42.34	42.36
5	**46.85**	44.84	42.06	38.14	42.12	40.13	40.77
6	**44.23**	42.38	38.65	34.72	38.84	37.44	36.60
7	**42.80**	41.04	37.25	33.26	37.87	36.06	36.53
8	**48.08**	45.95	38.38	34.84	39.56	39.26	37.74
9	**49.88**	47.60	38.22	34.73	40.35	41.24	36.87
10	**41.77**	41.54	39.98	32.46	41.17	36.24	37.27

Table 5.8: Average precision figures for retrieval using combinations of two characteristics
Highest value shown in bold.

How the performance of individual characteristics changed when evidence, in the form of another characteristic was added varied across the characteristics, generally;

- *idf* performance increased with the addition of any new evidence
- *tf* performance decreases with the addition of new evidence

• *theme* performance decreases by the addition of any new evidence except *tf* at high relevance levels

• *context* performance decreases by the addition of *idf*. The addition of *tf* or *theme* decreases performance at low relevance levels (1 – 5) but increases performance at high relevance levels (6 – 10). This means when recall-precision figures are based only on those documents that have been judged as highly relevant, *context* alone is generally poorer than a combination of context and *tf* or *theme*.

Combination was not effective at increasing the best overall precision when combining two characteristics. However the conclusion from Chapter Four, that poorer characteristics will benefit most from combination still seems to hold: *idf* benefits from any combination, *context* can benefit from combination in certain circumstances, and the best characteristics (*tf* and *theme*) do not benefit from combination.

5.6.1.2 Retrieval by combination of three characteristics

Table 5.9 shows the results of combining each combination of three characteristics of query terms, compared against *tf* - the best overall single characteristic.

Level	*tf*	*tf* + *idf* + *theme*	*tf* + *idf* + *context*	*tf* + *theme* + *context*	*idf* + *theme* + *context*
1	**56.82**	56.61	56.45	51.73	56.75
2	54.13	53.90	53.74	48.14	**54.18**
3	51.62	51.52	51.28	46.14	**51.70**
4	**49.60**	49.42	48.88	43.57	49.53
5	46.85	46.63	46.04	41.80	**46.86**
6	44.23	44.16	43.67	38.23	**44.31**
7	42.80	42.86	42.26	37.16	**43.00**
8	48.08	48.09	47.42	38.29	**48.14**
9	49.88	**49.91**	49.55	37.50	49.90
10	**41.77**	39.21	38.96	33.04	39.54

Table 5.9: Average precision figures for retrieval using combinations of three characteristics. Highest value shown in bold.

In this experiment a combination of three characteristics outperformed the *tf* ranking at 7 of the 10 relevance levels. These differences were only marginal. However[48] at relevance levels 1-9, the combinations of three characteristics outperformed all combinations of two characteristics and the other three single characteristics – *idf*, *theme* and *context* – as retrieval functions. This does indicate that combination can prove effective although its power to increase retrieval effectiveness does seem limited.

5.6.1.3 Retrieval by combination of four characteristics

Table 5.10 shows the results of combining all characteristics of each query term, compared against *tf*. The combination of all four characteristics performed worse at each relevance level than the *tf* ranking but better than most of the combinations of two characteristics and the single characteristics. Combination of characteristics can work – as in the case of combining three characteristics – but combining as much information as possible is generally not a good strategy.

Level	*tf*	all
1	**56.82**	54.34
2	**54.13**	50.83
3	**51.62**	48.38
4	**49.60**	45.36
5	**46.85**	43.16
6	**44.23**	39.95
7	**42.80**	40.01
8	**48.08**	43.06
9	**49.88**	43.42
10	**41.77**	39.19

Table 5.10: Average precision figures for retrieval using combinations of four characteristics. Highest value shown in bold.

5.6.2 Varying importance of characteristics

In Experiment Two I have, so far, treated each characteristic as being equally important. In the indexing, each characteristic is scaled so as all values fall between 0 and 50 to ensure that we are dealing with values in the same range. This means, for example when scoring documents by a combination of *tf* and *theme*, that a query term with a maximum *tf* value contributes as much to a document as a term with a maximum *context* value, or a maximum *idf* value. However, as

[48] With the exception of the combination of *tf*, *theme* and *context.*

demonstrated in Chapter Four, it may be appropriate to treat different characteristics as being more or less important than each other.

To test this, I re-ran Experiment Two, varying the weights assigned to term by each characteristic, e.g. halving all the *tf* values or doubling the *context* values. I tried a number of these *scaling factors* with three general conclusions[49]:

i. that varying the scaling factors assigned to the term characteristics could change retrieval effectiveness for combinations of characteristics.

For example, Table D.3 (Appendix D) shows the result of varying the importance of the characteristics when combining sets of three characteristics. At all relevance levels the average precision for the combination of *tf*, *theme* and *context* gave better results when using scaling factors than without. Conversely the combination of *idf*, *tf* and *context* gave worse results when using scaling than without. The difference in this result, over the results in Table 5.10, demonstrates the treating the characteristics as variably important does have an effect on the results of combining characteristics.

ii. it is difficult to derive a set of scaling factors for each characteristics that is guaranteed to increase the retrieval effectiveness of *all* combinations. For example, we can find good scaling factors for combining *tf* relative to *context* but these are not necessarily the best for combining *tf* and *context* relative to *theme*. For each combination of characteristics we must derive a different scaling factor for each characteristic or select an optimal set of scaling factors and accept that this will harm some combinations. This reinforces one of the conclusions from Chapter Four: weighting can prove effective but not always.

iii. scaling factors gives better performance. Weighting the characteristics is important as it *generally* improves retrieval performance. For the combination of two characteristics 57 of the 60[50] cases gave better results with scaling factors. For the combination of three characteristics, only 21 out of 40 cases gave better results with scaling but a combination was better than *tf* at eight relevance levels. This improvement over *tf* was not achieved without scaling. Finally for the combination of all characteristics, at all relevance levels the combination was better with scaling and again, at eight relevance levels, the combination was better than *tf*. The weighting of characteristics using scaling factors, then, is important for good retrieval results.

---

[49] Tables D.1 – D.3 gives the results of using scaling factors for combinations of two, three and four characteristics. These tables are complementary to Tables 4.9 – 4.11.

[50] Six combinations at ten relevance levels.

The remainder of the experiments in section 5.6 use the scaling factors; *tf* is weighted higher than all characteristics, with *idf* higher than *theme* and *context*, and *context* higher than *theme* (the method used to obtain the results in Tables D.1 - D.3). The actual values used were – *tf* 1.25, *theme* 0.4, *context* 0.75, *idf* 1. These values are different from those used in Chapter Four but follow the same principle of weighting characteristics roughly according to their quality as individual weighting schemes and, as in Chapter Four, were based on experiments on samples of the data.

5.6.3 Relevance level

Treating the relevance score given by the experimental subjects as a threshold has two main affects on the combination of characteristics. Firstly, as noted in section 5.5.1, as the relevance level increases the overall performance – average precision – tends to decrease. This generally holds over all the combinations and for the best individual characteristic, *tf*.

Level	***tf***	***idf* + *tf***	***idf* + *theme***	***idf* + *context***	***tf* + *theme***	***tf* + *context***	***theme* + *context***
1	100%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
2	95%	**95%**	93%	94%	**95%**	**95%**	93%
3	91%	**91%**	89%	89%	90%	**91%**	89%
4	87%	**87%**	84%	83%	85%	**87%**	84%
5	82%	**82%**	80%	79%	80%	**82%**	81%
6	78%	**78%**	74%	72%	74%	**78%**	74%
7	75%	**75%**	71%	69%	72%	**76%**	72%
8	85%	**85%**	73%	72%	82%	**85%**	74%
9	88%	**88%**	73%	72%	87%	**88%**	72%
10	74%	73%	73%	65%	73%	73%	**109%**

Level	tf	*tf + idf + context*	*tf + idf + theme*	*tf + theme + context*	*idf + theme + context*	*all*
1	100%	**100%**	**100%**	**100%**	**100%**	**100%**
2	95%	94%	**95%**	**95%**	93%	**95%**
3	91%	89%	**91%**	**91%**	89%	**91%**
4	87%	84%	**87%**	**87%**	84%	**87%**
5	82%	80%	82%	**83%**	81%	**83%**
6	78%	77%	77%	**78%**	74%	**78%**
7	75%	75%	74%	**76%**	72%	**76%**
8	85%	83%	84%	**85%**	74%	**85%**
9	88%	87%	87%	**88%**	72%	**88%**
10	74%	**75%**	73%	74%	68%	**78%**

Table 5.11: Stability measures for combination of characteristics.
Values higher than, or equal to, *tf* are shown in bold.

Secondly, the stability of the combinations over the relevance levels is variable. In section 5.5.1, I showed that the performance of *tf* at relevance level 10 was approximately 73% of its performance at relevance level 1. This was taken to be a measure of how stable the *tf* characteristic was when we changed the relevance level. For the combinations of characteristics the general trend was for small combinations, e.g. combinations of two characteristics, to be less stable whereas larger combinations were slightly more stable. This is shown in Table 5.11.

5.6.4 'Perfect' rankings

I repeated the experiment in section 5.4.2.3 - assessing the combinations of characteristics according to how well they ordered the assessed documents. Table 5.12 shows the results from this measured against the optimal or perfect ranking and the *idf* ranking which was the optimal single characteristic for ordering documents.

The *idf* ranking continues to be the best retrieval function for ordering the assessed documents. In the combination of two characteristics it outperforms all combinations, with the combination of *idf* and *idf* at relevance level nine. In the combination of three characteristics, *idf* is optimal except for the combination of *tf*, *theme* and *context* at relevance level nine. Although combination does not improve over *idf* in this case, it does generally improve the order over the single characteristics (Table 5.5 for comparison). This effect is

most defined at higher levels of relevance where combinations tend to give better rankings than individual characteristics.

Level	Perfect	*idf*	*idf* + *context*	*tf* + *context*	*idf* + *theme*	*tf* + *theme*	*tf* + *idf*	*theme* + *context*
1	14.65	**7.09**	4.83	5.69	5.3	5.62	5.72	5.09
2	14.72	**6.96**	4.69	5.55	5.14	5.46	5.57	4.93
3	14.89	**6.86**	4.60	5.53	5.04	5.41	5.55	4.86
4	15.17	**6.80**	4.46	5.57	5.05	5.36	5.58	4.78
5	15.54	**6.70**	4.58	5.69	5.24	5.46	5.73	4.98
6	15.86	**6.34**	4.27	5.65	5.02	5.23	5.67	4.68
7	15.62	**5.96**	3.96	5.42	4.76	5.01	5.39	4.46
8	15.68	**5.82**	4.03	5.35	4.64	5.08	5.34	4.28
9	14.86	4.99	4.55	5.05	4.22	4.57	**5.06**	3.83
10	16.02	**5.64**	4.13	5.43	5.25	5.11	5.46	4.58

Level	Perfect	*idf*	*tf* + *idf* + *theme*	*tf* + *idf* + *context*	*tf* + *theme* + *context*	*idf* + *theme* + *context*	*all*
1	14.65	**7.09**	5.70	5.57	5.70	5.07	5.70
2	14.72	**6.96**	5.54	5.44	5.55	4.91	5.57
3	14.89	**6.86**	5.53	5.47	5.54	4.83	5.59
4	15.17	**6.80**	5.57	5.43	5.58	4.76	5.58
5	15.54	**6.70**	5.75	5.60	5.72	4.94	5.73
6	15.86	**6.34**	5.65	5.35	5.66	4.65	5.63
7	15.62	**5.96**	5.36	5.07	5.42	4.42	5.36
8	15.68	**5.82**	5.30	5.06	5.35	4.27	5.35
9	14.86	4.99	5.00	4.64	**5.06**	3.82	4.97
10	16.02	**5.64**	5.44	5.20	5.43	4.58	5.27

Table 5.12: Ordering performance of combinations of two, three and four characteristics measured against 'perfect' ordering of relevant documents within the assessed set and *idf* ordering.

In section 5.4.2.3 I mentioned that the success of *idf* in ordering the documents was either due to *idf* discriminating better between highly relevant and less relevant documents or due to the way the subjects used partial relevance assessments. In particular, if a user does not revise the

relevance scores they give to the documents the later relevance assessments are not necessarily indicative of the user's final assessment of a document's relevance.

The analysis of combinations does not really help in eliciting why *idf* gives a better ordering than the other methods investigated and this remains an open question. However the combination of characteristics often improves the ordering of relevant documents over the characteristics that are poor at retrieving highly relevant documents. Combination, therefore, may have the potential to be effective at ranking highly relevant documents in a better order.

5.6.5 Performance by subject

In this section I assess how many users would have received better overall performance if the system had retrieved documents using a combination of characteristics rather than *tf*. Table 5.13 lists the number of users whose queries had greatest average precision using a combination of characteristics.

From Table 5.13 (combination of two characteristics - top, combination of three characteristics – middle, combination of all characteristics - bottom), at most relevance levels, most users would seem to have better performance using some other retrieval algorithm than *tf*. This demonstrates one of the problems of combination: overall combination gives consistent results in that it is better for more users but no single combination outperforms the other combinations to a significant degree.

On one hand this is good news as it shows that combination is preferable to no combination, on the other hand it makes it difficult to select one combination to use for all users and all queries.

Level	*tf*	*idf* + *context*	*tf* + *context*	*idf* + *theme*	*tf* + *theme*	*tf* + *idf*	*theme* + *context*
1	0	0	5	5	4	**7**	2
2	0	3	**7**	4	5	4	0
3	0	2	**6**	5	3	5	2
4	0	2	5	3	3	**7**	3
5	0	1	5	3	4	**7**	3
6	0	2	**8**	4	2	5	2
7	0	2	**6**	5	2	5	3
8	0	2	**8**	2	4	5	2
9	1	2	**6**	3	5	2	4
10	2	1	**4**	**4**	**4**	1	1
Total	3	17	**60**	38	36	48	22

Level	*tf*	*tf* + *idf* + *theme*	*tf* + *idf* + *context*	*tf* + *theme* + *context*	*idf* + *theme* + *context*	*all*
1	0	**9**	5	5	4	**23**
2	0	4	6	**11**	2	**23**
3	0	7	4	**9**	3	**23**
4	0	6	2	**10**	5	**23**
5	0	7	1	**9**	6	**23**
6	0	**8**	5	6	4	**23**
7	0	**7**	5	**7**	4	**23**
8	0	**7**	**7**	5	4	**22**
9	1	5	**7**	5	5	**21**
10	2	4	3	3	**5**	**15**
Total	3	64	45	**70**	42	**219**

Table 5.13: Numbers of users, at each relevance level, whose queries had highest average precision by different combinations of characteristics measured against *tf*.
The largest number of users at each relevance level is shown in bold.

5.6.6 Performance by topic

In Appendix D, Tables D.4 - D.6, I outline the performance of the various combinations on each topic set to the user. The results can be summarised as follows:

• **Topic A.** There was no consistent trend for this topic although at lower relevance levels *tf* information (either singly or in combination gave good results), at higher relevance levels *context* information (in combination with either *tf* and *theme*) gave good results. Overall the higher the relevance level, better results were obtained through larger combinations.

• **Topic B.** For this topic, at all relevance levels *tf* was the optimal retrieval function to use.

• **Topic C.** At lower relevance levels larger combinations (of *tf, theme* and *context* or *all* characteristics) gave optimal results, whereas at higher relevance levels smaller combinations, either *tf* and *idf* or some combination of *context* was better.

• **Topic D.** For most relevance levels some combination of *tf* and *theme* with either *context* or *idf* was the best combination.

• The users' own information need (**Own**) was very variable: either a combination of *tf* and another characteristic or *tf* singly was best for the most relevance levels but *context* was also important.

• In the training topic (**TR**) – the first topic each subject ran a combination of *tf* and *theme* either run together or in combination with another characteristic showed good performance overall.

Overall there was a marked lack of consistency in what combination of characteristics are good for retrieving relevant documents across the topics, except that the characteristics that were good as single retrieval algorithms – *tf* and *theme* – always seemed important. The lack of consistency means that the topic is having an affect on which characteristics are good at indicating relevance.

5.6.7 Summary of Experiment Two

The results from Experiment Two can be summarised as follows: simple combinations of characteristic scores can give some improvement in retrieval effectiveness if the

characteristics are treated as being of variable importance. Combination of characteristic information does not improve the order in which relevant documents are retrieved over the best single characteristic. However, it does increase the number of users whose queries improved. There is also little consistency across topics as to what improves search effectiveness.

5.7 Summary of combination experiments

The initial hypothesis was that adding more information about term use in documents would increase retrieval effectiveness. In section 5.6.2 I showed that marginal improvements could be made using a simple combination approach. However from section 5.6.4 it indicates that different combinations will improve retrieval effectiveness for different topics and for different users. Therefore it would seem that although combinations are useful it is difficult to predict what combinations are going to be good for all searchers and all topics.

Without any information on how to select characteristics for initial query terms, we could select an optimal combination to use for all queries and all users but this would be disadvantageous to a significant number of users and queries. This is similar to the conclusions of the combination experiments in Chapter Four: combination can be effective but it is difficult to predict good combinations. With the user data used in this chapter this conclusion is more pronounced: good combinations are less effective as single retrieval strategies. If we are to make use of the potential benefit of term use information we need some way to detect good indicators of relevance. This leads to hypothesis two:

hypothesis two - selective combination of evidence or selective relevance feedback: the relevance assessments given by a user could allow us to *select* which characteristics of a term's use should be used in RF. By analysing the documents a subject has assessed as being relevant we can select characteristics that are best to retrieve more relevant documents

Analysing this hypothesis is the subject of sections 5.8 and 5.9.

5.8 Relevance feedback

In this section I investigate hypothesis two. In section 5.8.1, I describe the experimental methodology, in section 5.8.2 I introduce the baseline measures I used to compare the approach to RF, in sections 5.8.3 – 5.8.5 I describe three types of RF I used in this set of experiments and in sections 5.8.6 – 5.8.9 I analyse the results. I summarise the findings in section 5.8.10. The experiments described in this section are analogous to those presented in Chapter Four, section 4.7.

5.8.1 Methodology

This set of experiments was designed to test the hypothesis that some queries or documents will be more suited to certain combinations of characteristics. In these experiments I performed a series of RF experiments, selecting which characteristics to use based on the differences between the relevant and non-relevant documents.

The methodology was as follows:

• take the 12 top documents from the initial *idf* ranking. This was the first screen of data that the users were presented with after submitting their query.

• calculate for each term the average score of each characteristic in the relevant and non-relevant set, e.g. the average *tf* for query term 1 in the relevant documents, the average *tf* of query term 1 in the non-relevant documents. This is identical to the averaging procedure in Chapter Four.

• select characteristics based on the relative averages. Various selection methods were tried, each will be discussed separately in sections 5.8.2 and 5.8.3.

• re-rank the remaining retrieved documents according to the characteristics selected in the previous step.

• calculate recall-precision values using the freezing evaluation scheme, [CCR71].

• compare the results, over the same set of documents, against three baselines figures.

5.8.2 Baselines and feedback strategies

The results were compared against the same baselines as in the previous chapter: no feedback, F_4 and the best combination of characteristics. The Best Combination baseline was the combination of all characteristics using the scaling factors. The same four feedback strategies were tested.

5.8.3 Feedback strategies

In this section I summarise the feedback strategies used in these experiments, Table 5.14.

Feedback 1 selects characteristics for each query term. The same query term characteristics are used to score all documents and there is no use of additional information on the discriminatory power of a term characteristic.

Feedback 2 selects characteristics for each query term and also does not use any additional information on the discriminatory power of a term characteristic. Feedback 2, however, uses different subsets of the query to score each document.

Feedback 3 selects characteristics and uses the same set of characteristics to score each document but uses information on the discriminatory power of term characteristics in scoring documents for ranking.

Feedback 5 is a version of the Feedback 3 strategy but does not use any selection of characteristics. All documents are score by the sum of the characteristic scores of all query terms, multiplied by the discriminatory power of each query term characteristic. This strategy differs from Feedback 3 only in the lack of selection.

	Selection performed	**Selection performed on**	**Discrimination factors used**
Feedback 1	Yes	query	no
Feedback 2	Yes	query and document	no
Feedback 3	Yes	query	yes
Feedback 5	No	-	yes

Table 5.14: Summary of feedback strategies

5.8.4 Results

In Table 5.14 the performance of the three baseline measures are contained in the three rightmost columns, the four Feedback strategies are in columns 2 - 5.

There are three main findings:

i. Feedback is generally better than the default ranking. The default ranking (*idf*, column 7) gives the lowest results at all relevance levels, except relevance level 1.

ii. Selection is somewhat more important than discrimination. On this set of data, at least, the selection of characteristics seemed to be more important than the discriminatory power of the characteristics. This is shown by the superior results of

Feedback 1 (selection only) over Feedback 5 and F_4, both of which perform only discriminatory weighting.

iii. A good combination of characteristics was better than feedback. At all relevance levels the best combination of characteristics was better than all feedback strategies and the default ranking. This difference was also found to be statistically significant at all relevance levels.

	Feedback techniques				**Baselines**		
Level	**1**	**2**	**3**	**5**	***idf***	**Best Combination**	**F_4**
1	52.77	52.77	52.60	52.31	52.30	**56.75**	52.24
2	47.81	47.94	47.73	47.51	47.00	**54.18**	47.51
3	44.88	44.97	44.84	44.67	43.70	**51.70**	44.57
4	42.07	42.19	42.06	41.94	41.20	**49.53**	41.78
5	39.34	39.44	39.35	39.19	38.90	**46.86**	39.04
6	34.65	34.69	35.64	34.47	34.10	**44.31**	34.25
7	32.61	32.64	32.65	32.54	32.10	**43.00**	32.28
8	32.71	32.88	32.72	32.7	32.30	**48.14**	32.53
9	30.05	30.25	30.13	30.12	29.70	**49.90**	29.86
10	32.58	32.91	32.58	32.55	32.20	**41.86**	32.43

Table 5.15: Average precision figures for feedback techniques compared with *idf* ranking and ranking obtained from the optimal combination (**Best combination**).
Highest values shown in bold.

The main reason for the poor performance of the feedback strategies against the combination strategy may be due to the small amount of data that was being used. This would potentially harm the Feedback strategies 1-3, F_4 and Feedback 5 as they do not have enough information upon which to base good estimates of which characteristics are useful and the discriminatory power of these characteristics.

The Feedback 2 strategy performed better than the Feedback 1 strategy throughout. A strategy such as Feedback 2 which performed less well in Chapter Four may be more suited to situations such as this, with less information upon which to base relevance decisions, as it makes more precise retrieval decisions.

Feedback 3 (selection and discrimination) performed better than just discrimination (Feedback 5) at all relevance levels and better than just selection (Feedback 1) only at higher levels, reiterating point **ii.** that in this data selection was more effective than discrimination. Again, this may be due to the small data samples I was using.

In a set of informal experiments, not reported in this chapter, I found that altering the scaling factors used in the feedback and combination strategies affected the average precision at different relevance levels. That is, a good set of weighting values could improve different techniques at different levels. An improved version of the feedback strategies could perhaps exploit this aspect of relevance assessments.

5.8.5 Relevance level

	Feedback techniques				**Baselines**		
Level	**1**	**2**	**3**	**5**	***idf***	**Best Combination**	**F_4**
1	**100%**	**100%**	**100%**	**100%**	100%	**100%**	**100%**
2	**91%**	**91%**	**91%**	**91%**	90%	**95%**	**91%**
3	**85%**	**85%**	**85%**	**85%**	84%	**91%**	**85%**
4	**80%**	**80%**	**80%**	**80%**	79%	**87%**	**80%**
5	**75%**	**75%**	**75%**	**75%**	74%	**83%**	**75%**
6	**66%**	**66%**	**68%**	**66%**	65%	**78%**	**66%**
7	**62%**	**62%**	**62%**	**62%**	61%	**76%**	**62%**
8	**62%**	**62%**	**62%**	**63%**	62%	**85%**	**62%**
9	**57%**	**57%**	**57%**	**58%**	57%	**88%**	**57%**
10	**62%**	**62%**	**62%**	**62%**	62%	**74%**	**62%**

Table 5.16: Stability of feedback techniques.
Values greater than, or equal to, the default *idf* ranking are shown in bold.

As with the combination of evidence experiments the performance of each feedback strategy fell as the relevance level increased. The stability of each technique is compared in Table 5.16. All feedback techniques are slightly more stable than *idf* but are less stable than *tf*, section 5.5.1. In addition all the feedback techniques show relatively similar level of stability: none are noticeably less stable than the others.

5.8.6 Performance by subject

	Feedback techniques				Baselines		
Level	**1**	**2**	**3**	**5**	***idf***	**Best Combination**	**F_4**
1	1	2	0	0	4	**16**	0
2	1	0	0	1	2	**19**	0
3	2	1	0	2	1	**17**	0
4	1	2	0	1	0	**19**	0
5	0	2	1	2	0	**18**	0
6	1	1	1	2	0	**18**	0
7	1	2	0	1	0	**19**	0
8	1	2	1	0	0	**19**	0
9	1	2	1	1	0	**18**	0
10	3	2	0	0	0	**12**	0
Total	12	16	4	10	7	**175**	0

Table 5.17: Average precision figures for feedback techniques compared with *tf* ranking

Analysing the performance of the feedback strategies and baselines against how many users have optimal performance in Table 5.17, we can see that for all relevance levels the Best Combination strategy was optimal for most users. Only a small number of users had best overall performance with the feedback strategies (except F_4 which was optimal for no users) or the default ranking.

5.8.7 Performance by topic

Table D.7 (Appendix D) outlines the performance of each Feedback strategy and the baselines by topic. For all topics and all relevance levels the Best Combination of characteristics gave the best average precision.

5.8.8 Characteristics used in feedback

	Feedback 1							
	Possible	***tf***	***theme***	***context***	***tf + theme***	***tf + context***	***theme + context***	***tf + theme + context***
1	912	56	150	54	39	143	76	153
2	903	56	144	58	37	135	84	129
3	872	51	143	61	43	134	65	113
4	823	45	137	55	46	132	61	96
5	772	43	135	42	37	109	61	94
6	663	31	120	35	28	94	48	75
7	619	33	122	33	29	80	41	79
8	512	21	98	27	25	66	34	67
9	438	21	76	21	19	56	27	53
10	263	10	41	12	11	36	17	34

Table 5.18: Number of times each characteristic was used in modified query for each relevance level.
Possible is the number of times a characteristic could have been used.

In this section I analyse which characteristics are used in the two selection feedback strategies, Feedback 1 and 2, across the relevance levels. From Table D.18[51] we can see that for the Feedback 1 strategy the *tf*, *theme* and *context* characteristics were used to describe about 35-40 of the query terms. Most commonly used was the *theme* characteristic alone (17 of query terms), followed by *tf* + *context* (14) and *tf* + *theme* + *context* (12). The total number and percentage of characteristics used to described query terms dropped as the relevance level increased.

From Table 5.19[52] we can see that for the Feedback 2 strategy the *tf*, *theme* and *context* characteristics were used to describe about 60-70 of the query terms. Most commonly used was the *tf* +*theme* + *context* combination (35 of query terms), followed by *theme* (16) and *tf* + *context* (16). The total number of characteristics increased as the relevance level increased. *tf*

[51]Tables D.7 - D.8 give these figures as percentages.

[52]The figures in this table are higher as we are selecting term characteristics for each document rather than just for the query.

was used more commonly in this strategy and *context* rather less compared with the previous one.

	Feedback 2							
	Possible	***tf***	***theme***	***context***	***tf*** + ***theme***	***tf*** + ***context***	***theme*** + ***context***	***tf*** + ***theme*** + ***context***
1	11186	1158	2064	284	405	1221	408	3097
2	10578	1048	1880	253	438	1133	348	3221
3	9722	905	1676	230	403	1007	305	3171
4	8847	821	1474	217	381	1000	276	2868
5	8142	760	1279	203	358	925	248	2815
6	7077	684	1097	164	299	810	186	2623
7	6044	670	967	156	266	676	150	2195
8	4767	535	683	112	161	558	105	2012
9	3495	322	430	78	99	380	79	1709
10	2549	248	297	35	65	244	63	1281

Table 5.19: Number of times each characteristic was used in modified query for each relevance level.
Possible is the number of times a characteristic could have been used.

There are three differences in this set of results:

i. different characteristics are selected in the two methods. In both cases the highest use of characteristics were *theme*, *theme* + *tf* + *context* and *tf* + *context*, with a similar percentages of use for the *theme* and *tf* + *context* combinations. However the combination of *tf* + *theme* + *context* was higher in Feedback 2 (35) than Feedback 1 (12). This may be because in the Feedback 1 strategy the overall averages dictate which characteristics are used; in the Feedback 2 strategy, characteristics that may not have been selected using Feedback 1 can still be used to score individual documents. The combination of *tf*, *theme* and *context* was more successful (Appendix D, Table D.2) than either *theme* or the combination of *theme* or *context*, which may explain the relative success of Feedback 2 over Feedback 1: it was selecting better sets of characteristics.

ii. change in frequency of use with the change in relevance level. In Feedback 1 the use of term characteristics dropped as the relevance level increased, in Feedback 2 the reverse occurred. That is, analysing which characteristics to use on a document-document

basis results in more characteristics being used overall. The bulk of this increase in use, however, came from the increased use of the combination of *tf* + *theme* + *context*.

iii. change in frequency of use of individual characteristics. In Feedback 2 *tf* was used rather more often and *context* rather less than in the Feedback 1 strategy. This may be due to larger variations in values of *tf* compared with *context* in the relevant documents; large variations in the value of a characteristic in one set of documents (relevant or irrelevant) will bring the average value down. This may stop a characteristic being used in the Feedback 1 strategy but not in the Feedback 2 one.

The Feedback 2 strategy seems to be doing what we would expect and from Table 5.14, this seems to be correct - at higher levels of relevance we want to use more information in assessing relevance. One could argue that at higher levels of relevance we are dealing with fewer document so we have less evidence to create patterns of term characteristic selection but this is the situation we would be dealing with if a user was employing a higher criteria for relevance and marking fewer documents relevant.

5.8.9 Summary of Feedback Experiments

From the results in sections 8.4 - 8.7, I can summarise that the feedback approaches (Feedback 1 – 3, Feedback 5 and F_4) are not as effective overall as would have been hoped from the results in Chapter Four. Although some feedback strategies do perform credibly overall, the use of more information (the Best Combination method) is better in most cases. However, as shown in section 5.8.7, there is evidence that different strategies perform better for different topics, and for many users some form of feedback gave optimal results, section 5.8.4. It may be that we want not just to select term characteristics but also to select how these should be chosen.

5.9 Predictive versus retrospective query modification

In [RSJ76], Robertson and Sparck Jones differentiated between two types of query modification: *predictive* and *retrospective* modification. In the predictive case only a subset of relevant documents are used to modify the query, with the intention of improving retrieval of the remainder of the relevant documents. In the retrospective case the aim of the feedback is to develop a query to retrieve the documents already seen.

In the previous section I examined predictive RF as the aim of RF is generally the predictive case. In this section I examine retrospective RF to see if there is any difference between

queries produced from the users initial 12 relevance assessments and queries produced when we have complete knowledge of what the user found to be relevant.

In Table 5.20 I present the results of this experiment (this table is analogous to Table 5.14). The results show that when we have complete relevance information the Feedback 3 strategy is optimal at all relevance levels, the Feedback 1 strategy was second optimal at all levels, and at all levels the Best Combination outperformed F_4 which tends to outperform the default *idf* ranking.

Table D.10 (Appendix D) indicates how well each strategy performs for different search topics. For all topics at all relevance levels the Feedback 3 strategy is optimal, and in the majority of cases Feedback 1 was second optimal with Feedback 5 third. The Feedback 3 performs better than the other techniques and the this difference is statistically significant.

	Feedback techniques				**Baselines**		
Level	**1**	**2**	**3**	**5**	***idf***	**Best Combination**	**F_4**
1	65.70	57.95	**69.48**	58.96	52.30	56.75	53.83
2	62.98	54.73	**67.96**	56.26	47.00	54.18	49.40
3	60.89	51.85	**67.18**	55.42	43.70	51.70	46.63
4	59.02	49.98	**66.15**	53.34	41.20	49.53	43.81
5	57.51	47.49	**65.70**	52.29	38.90	46.86	40.97
6	56.48	46.69	**63.97**	49.69	34.10	44.31	36.25
7	56.11	45.81	**65.14**	49.91	32.10	43.00	34.30
8	60.37	52.20	**69.64**	50.85	32.30	48.14	33.77
9	60.52	55.13	**71.71**	53.82	29.70	49.90	30.83
10	58.38	46.34	**64.22**	50.1	32.20	41.86	30.26

Table 5.20: Average precision figures for retrospective feedback techniques compared with *idf* ranking and ranking obtained from the optimal combination (**Best combination**). Highest values shown in bold.

So there is, then, a preference for the Feedback 3 strategy. The preference for the Feedback 3 strategy under retrospective modification is also shown when we compare how many users had optimal performance with each strategy (Table D.11). At all relevance levels almost all users had the best performance with Feedback 3, with only a few users having optimal performance an alternative feedback strategy.

The stability of the feedback techniques under retrospective feedback is shown in Table 5.21. This table shows that the feedback techniques 1 – 5 are not only more stable than the default *idf*, the best individual single characteristic – *tf* and the Best Combination, they are also more stable than the baseline F_4 measure. The most stable algorithm is Feedback 3.

The higher level of stability for Feedback 3 means that it will perform better with less relevance information and this may be useful in situations where searchers are using a strict threshold for relevance, i.e. only marking highly relevant documents as relevant.

	Feedback techniques				**Baselines**		
Level	**1**	**2**	**3**	**5**	***idf***	**Best Combination**	**F_4**
1	**100%**	**100%**	**100%**	**100%**	100%	100%	100%
2	**96%**	**94%**	**98%**	**95%**	90%	95%	92%
3	**93%**	**89%**	**97%**	**94%**	84%	91%	87%
4	**90%**	**86%**	**95%**	**90%**	79%	87%	81%
5	**88%**	**82%**	**95%**	**89%**	74%	83%	76%
6	**86%**	**81%**	**92%**	**84%**	65%	78%	67%
7	**85%**	**79%**	**94%**	**85%**	61%	76%	64%
8	**92%**	**90%**	**100%**	**86%**	62%	85%	63%
9	**92%**	**95%**	**103%**	**91%**	57%	88%	57%
10	**89%**	**80%**	**92%**	**85%**	62%	74%	56%

Table 5.21: Stability values for retrospective feedback techniques compared with *idf* ranking and ranking obtained from the optimal combination (**Best combination**).
Highest values shown in bold.

The pattern of term characteristics use in the selection strategies (Feedback 1 and 2) was similar in the retrospective as predictive case: the percentage of term use dropped as relevance level increase when using Feedback 1 and increased when using Feedback 2. Similar combinations of terms were effective in both predictive and retrospective RF.

For the retrospective case, it seems to be effective to use feedback over combination, suggesting that better feedback techniques for predictive RF are required when using small data samples. Naturally we expect to feedback to work well in this case as we are modifying the query to preferentially retrieve the relevant documents. However, the combination of evidence still works better than F_4. The success of the retrospective feedback experiments

suggest that the overall aim of selecting important features of term use has potential to work for user data.

5.10 Relevance feedback summary

The second hypothesis I suggested was that user's relevance assessments could allow us to select good characteristics to terms to score documents in RF. The results from the predictive RF experiments were inconclusive. Although feedback was preferable to no feedback and the term characteristic feedback strategies slightly outperformed the baseline F_4 measure, the Best Combination baseline gave better results. The results of the Best Combination were also statistically better than the best (Feedback 2) of the feedback strategies[53].

The Best Combination approach gave better results when compared against the feedback strategies when the feedback algorithms were used predictively. However, when used retrospectively the feedback results are better than the best combination, and this difference is significant[54]. As indicated before, the predictive results are based on very little data, the retrospective results, being based on a slightly larger sample, show that the feedback techniques can achieve better performance than a good baseline combination of characteristics. One encouraging aspect of the retrospective results is that the feedback strategies (Feedback 3 especially) are very consistent. That is they work better across topics, users and relevance levels, thus reducing the variability inherent in the combination approach.

5.11 Conclusions

In section 5.3.4, I proposed three research question which I shall now attempt to answer.

i. the first question I looked at was whether the results I obtained from the earlier experiments on TREC data would hold on data derived from non-expert assessors, who were given no specific instructions on how to make relevance assessments. The earlier results in Chapter Four indicated that, although combination of evidence, in the form of term characteristics, could improve retrieval effectiveness, it was difficult to predict good combination of term characteristics that would work over a range of collections. The results in this chapter confirm this finding: combination of evidence can improve retrieval effectiveness (section 5.6), but this improvement does not hold for the majority of users (section 5.6.4) or queries (section 5.6.5). This also reinforced the point that evidence should be treated as of varying importance.

---

[53] Using a paired *t*-test, holding relevance level constant and varying average precision ($p < 0.05$), *t*-value 6.49.

[54] *t*-value 16.16

In the earlier experiments on RF I found that the feedback strategy (Feedback 3) outperformed other feedback techniques (Feedback 1, Feedback 2 and F_4) and a good combination of term characteristics. In these experiments I found this only to hold when I used retrospective feedback techniques.

ii. The second area of investigation was to discover whether subjects assessed documents differently if they were making assessments on their own information needs or given information needs. In the data I used, five out of the six search topics used were constructed by the experimental designers and the sixth was created by the subjects making the assessments, [BI99], section 5.3.

I found very little comprehensive difference regarding which combination of characteristics, or feedback method (retrospective or predictive) worked well between the users information needs and the given ones. There were differences between topics but not ones that distinguished between *sources* of information need. This is in line with Borlund and Ingwersen's findings that users behave in a similar manner making assessments on simulated and real information needs, [BI99].

iii. The final area of analysis looked at the use of partial or non-binary relevance assessments. The TREC data contained binary relevance assessments, the data I used in these experiments had relevance assessments ranging from 0-10. This factor seemed to be the most important variable in the experiments. The choice of which relevance level was taken as a threshold for relevance was important in two ways. First, different levels of relevance gave different results: higher levels of relevance gave lower average precision. Second, different combinations of characteristics gave different relative levels of effectiveness, e.g. different combinations gave better results for some topics depending on which relevant level was chosen to indicate relevance.

The results in this chapter are preliminary in that the experiments have several limitations: I only consider one iteration of RF, the number of documents I am dealing with is relatively small and I lack qualitative information from the users on their *reasons* for assessing relevance. Nevertheless I have pointed to certain important aspects of utilising relevance assessments in RF.

In summary, combination of evidence can be a useful and effective procedure for retrieval. However it is a variable technique: the actual combination of evidence that will increase

retrieval effectiveness varies across user, topic, and relevance level resulting the fact that any single combination is likely to be sub-optimal.

In the retrospective feedback situation, the selection procedure of Feedback 3, combined with the scaling and discrimination factors, proved to be consistently better for task, user and relevance level. That is it has the potential to be a more *consistent* technique for retrieval in the sense that it evens out the variability present with simple combination of evidence approaches.

I have demonstrated in the previous two chapters that selecting evidence can give better, and more consistent, improvements in retrieval effectiveness. In the next chapter I analyse how the evidence should be used once it has been selected.

Chapter Six

Using Dempster-Shafer's Theory of Evidence to combine aspects of information use

6.1 Introduction

In the previous two chapters I demonstrated that incorporating information on how words are *used* within documents – *term* and *document characteristics*, in a RF situation, can lead to significant improvements in retrieval effectiveness across collections. I also showed, experimentally, that the best performance came from *selecting* which set of characteristics, for each query term, best indicated relevant material.

The technique that gave the best overall results – Feedback 3 - was one that incorporated qualitative aspects regarding a term characteristic's importance or utility in describing an information need. In particular it specified three types of information:

i. *characteristic index weights*. These are the weights that are assigned to all characteristics of each term at indexing time, e.g. *idf* weight or *tf* weight of a query term.

ii. *characteristic utility weights* or *scaling factors*. In both Chapters Four and Five I demonstrated that treating some characteristics as being more important than important than others often gave better performance than treating all characteristics as being of equal importance. This was shown in the difference between the weighting (**W**) condition and non-weighting (**NW**) conditions.

iii. *feedback weights or discriminatory power*. In RF, it is possible to derive information on how well a combination of a term and characteristic discriminates relevant from non-relevant documents, e.g. we can estimate that the *idf* weight of query term *t* is a better indicator of relevance compared to the *tf* value of query term *t*.

These weights are applied to different retrieval components, e.g. scaling factors are assigned to a characteristic, such as *idf*, independent of which term the characteristic is describing, whereas the discriminatory weights are assigned to the *combination* of a term and characteristic, e.g. the *idf* weight of query term *t*.

These weights can be regarded as reflecting the *uncertainty* involved in the IR and RF processes. Each weight is used to estimate the quality, or certainty, regarding the evidence it supplies. There are many other aspects of uncertainty that we might want to incorporate within the retrieval and feedback procedures. However, the more sources of uncertainty that are considered, the more difficult it becomes to manage them.

It this chapter I propose a model for managing the uncertainty involved in combining evidence about which terms and term characteristics are good at retrieving relevant documents. This model is an attempt to *formally* model the uncertainty of RF.

The model is based on a widely-used system for combining multiple sources of evidence, namely Dempster-Shafer's Theory of Evidence (DS), [Dem68, Sha76]. The attraction for this theory over other formal techniques, such as inference networks, is that it allows us to explicitly represent and manipulate the uncertainty attached to the evidence combination process. As I shall describe later, DS theory is a powerful and coherent way of representing aspects of combination such as the quality of evidential sources, the user's assessments of evidence, and the reliability of evidence.

The chapter is structured as follow. In section 6.2, I give a working example that I shall use to illustrate the approach and highlight the salient modelling issues. In section 6.3 I give a brief introduction to DS theory and motivate the suitability of this theory in modelling RF. In section 6.4 I discuss the combination of evidence without relevance information - ranking the documents after the user has submitted a query but not yet assessed any documents. This models the situation in which the user has submitted a new query to the system and stands in contrast to the combination of evidence experiments described in the two previous chapters. The difference between the two sets of experiments – those in this chapter and those in Chapters Four and Five – is how the evidence is used to score documents. In sections 6.5 and 6.6, I deal with combination of evidence when the user has assessed some documents as relevant. This is the RF situation. In section 6.6 I present experimental results and discuss the main results. I summarise the overall research study in section 6.7.

I should note here that this model does not depend on a particular definition of relevance nor is it concerned with the actual mechanisms by which the user makes a relevance assessment (the details of the IR system interface). A user may assess a document as relevant for many reasons, the assessment of relevance may change over time (section 6.5.1), and some documents may be considered to be more relevant than others (section 6.5.1). What I do claim for the relevance assessments is that by a user assessing a document as relevant, s/he is indicating that the document contains information of the kind s/he is looking for at the current point in the search.

6.2 Working example

The discussion in the rest of the chapter will be illustrated by examples based on a simple document representation.

Document	Term	*theme*	*tf*
d_1	t_1	50	30
	t_2	25	15
	t_3	45	20
d_2	t_4	30	10
	t_5	10	10
	t_6	30	15
d_3	t_3	15	50
	t_4	25	30
	t_5	0	30
d_4	t_1	10	45
	t_3	0	30
	t_5	0	30
d_5	t_2	10	10
	t_4	50	20
	t_6	0	0

Table 6.1: Example document representations

Consider five documents each containing three terms: $d_1\{t_1, t_2, t_3\}$, $d_2\{t_4, t_5, t_6\}$, $d_3\{t_3, t_4, t_5\}$, $d_4\{t_1, t_3, t_5\}$, and $d_5\{t_2, t_4, t_6\}$. Table 6.1 shows the values for two characteristics of the terms used in the documents. All characteristics scores for terms that do not occur in a document are taken to be zero. Note that the *context* relation, as defined at present is query

dependent as well as document dependent as it is measured by the proximity of two query terms. Values for this characteristic will be defined further in the examples.

6.3 Dempster-Shafer's Theory of Evidence

My interest is in investigating the effect of combining evidence from different characteristics of term use in documents. There are a variety of formal theories I could use for this purpose. I have chosen *Dempster-Shafer's (**DS**) Theory of Evidence* as it is a well-understood, formal framework for combining sources of evidence. The mathematical connection between IR and DS Theory was suggested by Van Rijsbergen, [VR92], although this work concentrated on retrieval functions in general rather than specifically on RF. A continuing stream of research has investigated how theories based on DS can be used to model various aspects of the IR process, e.g. [TdSLM93, SH93, LR98].

DS is a theory of uncertainty, [Saf87], that was first developed by Dempster, [Dem68], and extended by Shafer, [Sha76]. Its main difference to probability theory, which is treated as a special case, is that it allows the explicit representation of ignorance and combination of evidence. This explicit representation of ignorance, or the imprecision of evidence, makes the use of the DS theory particularly attractive for modelling complex systems. The combination of evidence is expressed by *Dempster's combination rule*, which allows the expression of aggregation necessary in a model using multiple sources of evidence. In no other theory of uncertainty is the combination of evidence explicitly captured as a fundamental property.

In this section I describe the main concepts of DS theory, based on the description given in [Sha76], presented within the context of my work.

6.3.1 Frame of discernment

The DS framework is based on the view whereby propositions are represented as subsets of a given set. Suppose that we are concerned with the value of some quantity *u*, and the set of its possible values is *U*. The set *U* is called a *frame of discernment*. An example of a proposition is "the value of *u* is in *A*" for some $A \subseteq U$. Thus, the propositions of interest are in a one-to-one correspondence with the subsets of *U*. The proposition $A = \{a\}$ for $a \in U$ constitutes a basic proposition "the value of *u* is *a*". In my approach the frame of discernment is taken to be the set of available documents, which in the example is the set $\{d_1, .., d_6\}$.

6.3.2 Basic probability assignment

Beliefs can be assigned to propositions to express their uncertainty. The beliefs are usually computed based on a density function $m:\wp(U)\rightarrow[0,1]$ called a *basic probability assignment* (*bpa*) or *mass* function:

$$m(\varnothing)=0 \text{ and } \sum_{A\subseteq U} m(A)=1$$

Equation 6.1: Basic probability assignment

$m(A)$ represents the belief exactly committed to A, that is the exact evidence that the value of u is in A. If there is positive evidence for the value of u being in A then $m(A) > 0$, and A is called a *focal element.* The proposition A is said to be *discerned*. No belief can ever be assigned to the false proposition (represented as $\varnothing$). The focal elements and the associated *bpa* define a *body of evidence*.

In my approach, term characteristics, which assign mass only to singleton sets, act as a body of evidence assigning mass values to individual documents[55]. Each term characteristic acts as *bpa*. My approach is slightly different from most DS applications as I have, *a priori*, fixed the maximum mass value that can be assigned to a set. The maximum value that can be attached to a document is 50[56], which is the maximum value that can be attached to a term characteristic (section 1.3). The focal elements are then the documents that have a positive mass value assigned to them, i.e. display the term characteristic.

From the definition of the *bpa*, in Equation 6.1, the sum of the non-null *bpas* must equate to 1, i.e. each body of evidence must assign the same amount of evidence to the frame of discernment. In the working example, each term characteristic assigns a total evidence of 250 (5 documents * maximum characteristic value of 50). The total evidence can be scaled to fall between 0 and 1.

---

[55]The user's relevance assessments, which can assign mass values to singleton sets or sets with multiple elements also act as a *bpa*. This will be discussed separately in section 6.5.

[56] As in Chapters Four and Five, all characteristic values are scaled to fall within the range 0-50.

6.3.3 Belief function

Given a body of evidence with *bpa m*, we can compute the total belief provided by the body of evidence for a proposition. This is done with a *belief function* $Bel: \wp(U) \rightarrow [0,1]$ defined upon *m* as follows:

$$Bel(A) = \sum_{B \subseteq A} m(B)$$

Equation 6.2: Belief function

*Bel(*A*) is* the total belief committed to *A*, that is, the mass of *A* itself plus the mass attached to all subsets of *A*. *Bel(*A) is then the total positive effect the body of evidence has on the value of *u* being in *A*.

6.3.4 Plausibility function

A particular characteristic of the DS framework (one which makes it different from probability theory) is that if *Bel(A)*<1, then the remaining evidence 1-*Bel(A)* needs not necessarily refute *A* (i.e., supports its negation $\bar{A}$). That is we do not have the so-called *additivity rule* $Bel(A) + Bel(\bar{A}) = 1$. Some of the remaining evidence may be assigned to propositions which are not disjoint from *A*, and hence could be plausibly transferable to *A* in light of new information. This is formally represented by a plausibility function $Pl : \wp(U) \rightarrow [0,1]$ defined upon a *bpa, m,* as follows:

$$Pl(A) = \sum_{A \cap B \neq \varnothing} m(B)$$

Equation 6.3: Plausibility function

Pl(*A*) is the mass of *A* and the mass of all sets that intersect with *A*, i.e those that could transfer their mass to *A* or a subset of *A*. *Pl*(*A*) is the extent to which the available evidence fails to refute *A*.

6.3.5 Dempster's combination rule

DS theory has an operation, *Dempster's rule of combination*, for the pooling of evidence from a variety of sources. This rule aggregates two *independent* bodies of evidence defined within the same frame of discernment into one body of evidence. Let m_1 and m_2 be the *bpas* associated to two independent bodies of evidence defined in a frame of discernment *U*. The new body of evidence is defined by a *bpa m* on the same frame *U*:

$$m(A) = m_1 \otimes m_2(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{\sum_{B \cap C \neq \varnothing} m_1(B) m_2(C)}$$

Equation 6.4: Dempster's combination rule

Dempster's combination rule, then, computes a measure of *agreement* between two bodies of evidence concerning various propositions discerned from a common frame of discernment. The rule focuses only on those propositions that both bodies of evidence support. The new *bpa* takes into account the *bpa* associated to the propositions in both bodies that yield the propositions of the combined body. The denominator of the above equation is a normalisation factor that ensures that *m* is a *bpa*. In my approach, I use the combination rule to combine the *bpa*s from the term characteristics. This combination produces a single *bpa* over the documents in the collection derived from the combination of the individual term characteristic information.

6.3.6 Uncommitted belief

From the definition of the *bpa*, each body of evidence must assign the same total amount of belief to the frame of discernment *U*. The total amount of evidence that can be assigned to the documents is N*50 (where N is the number of documents in the collection and 50 is the maximum mass value that can be assigned to each document, see section 6.3.2). However, the maximum mass value will not be assigned to all documents, as each term does not appear in every document. Consequently there will be evidence which is unassigned, violating the definition of the *bpa*.

There are three possible ways to avoid this violation:

i. normalise the *bpa* values assigned to the focal elements such that each *bpa* sums to the same value.

ii. assign the remainder of the belief equally to the documents in the collection that do not display the characteristic

iii. treat the remainder of the belief as *uncommitted belief*.

In the first approach - normalisation - the *bpas* are scaled for each body of evidence such that the sum of the evidence attached to the focal elements sum to the same amount. Consider the example of two bodies of evidence with the *theme* values for terms t_1 and t_5, shown in Table 6.2. The total amount of evidence to be assigned is 250. The mass values for each term are then scaled so that they sum to 250 (column 4, Table 6.2). However as the only evidence

assigned by t_5 is to document d_2, then all the evidence is assigned to this document, irrespective of how well the document reflects the *theme* characteristic. Worse, the mass value assigned to d_1 by term t_1 is lower than that assigned to document d_2 by t_5 after normalisation, even though before normalisation it had a higher value. Normalisation, then, can give counter-intuitive results, changing the relative amount of evidence assigned to documents without justification.

Term	**Document**	**Mass**	**Normalised mass**
t_1	d_1	50	208.3
	d_2	0	0
	d_3	0	0
	d_4	10	41.7
	d_5	0	0
t_5	d_1	0	0
	d_2	10	250
	d_3	0	0
	d_4	0	0
	d_5	0	0

Table 6.2: Normalising mass values for theme characteristics (terms t_1 and t_5)

The second approach, taken by probability theory, assumes that any evidence that does not support a proposition is evidence against that proposition, i.e. $P(A) = 1 - P(\overline{A})$. DS theory views this as untenable, as any evidence that is not assigned to a proposition could turn out to support the proposition. It is merely evidence that has not been assigned. This leads to the notion of *uncommitted belief*, which is specific to the DS approach.

In my approach the uncommitted belief is the evidence not directly assigned by a term characteristic to a focal element (a document or a set of documents), and is given by,

$$ub = N * 50 - \sum_{i=1}^{n} m(\{d_i\})$$

Equation 6.5: Uncommitted belief

Equation 6.5 calculates the uncommitted belief for a term characteristic *bpa*, where *n*= number of documents in a collection, d_i is the *i*th document in the collection, and $m(d_i)$ is the mass assigned to document d_i for that term.

This equation gives a direct calculation of the uncommitted belief, based on the mass values assigned to the focal elements. However, we can further utilise the uncommitted belief by treating it as a measure of the *quality* of the evidence supplied by the term characteristic. This means using the uncommitted belief as a regulating device, controlling how much of the value of the characteristics are converted into the mass function. Take the example of the *tf* values for term t_5 (shown in Table 6.3, column 3). If the *tf* measure is unreliable, or is less accurate at measuring the term frequency than another algorithm, we could increase the measure of uncommitted belief and rescale the mass values accordingly (Table 6.3, column 4). The rescaling is based on a constant factor given by,

$$m'(\{d_i\}) = m(\{d_i\}) \Big/ \sum_{i=1}^{n} m(\{d_i\}) \times ((n \times 50) - ub')$$

Equation 6.6: Rescaling calculation

Equation 6.6 defines rescaling the mass for a term characteristic, where $m(d_i)$ is the original mass assigned to document d_i, $m'(d_i)$ is the new mass value. *n* is the number of documents in the collection, *ub'* is the value of the uncommitted belief in the new *bpa* . $\sum_{i=1}^{n} m(\{d_i\})$ is the amount of evidence assigned to the focal elements of the original *bpa*.

This differs from the normalisation approach in two ways: firstly, the mass values for each focal element are still within the same range, 0-50, as normalisation only ever decrease the mass values. Secondly *all* the *bpas* for each characteristic are scaled so the values are not affected by how many focal elements (documents displaying the characteristic) are present for each *bpa*. I am only recalculating the mass values for a term characteristic - asserting that a characteristic as a whole is better or worse than another characteristic.

Document	Term	Mass m	Mass m'
d_1	t_5	0	0
d_2	t_5	10	7.14
d_3	t_5	30	21.43
d_4	t_5	30	21.43
d_5	t_5	0	0
$\sum_{i=1}^{5} m(d_i)$		70	50
uncommitted belief		180	200

Table 6.3: Using uncommitted belief to reflect the quality of a term characteristic

Using the uncommitted belief in this fashion it is possible to reflect a number of aspects of a term characteristics:

i. the *uncertainty* of the characteristic. Some characteristics may reflect aspects of the document's information content that are more easily measurable. For instance the term frequency, *tf*, is an easier characteristic to provide an algorithm for, as it is more objective in nature than measuring the topical nature of the document, which is dependent on the interpretation of what constitutes the topical nature of the document. This aspect distinguishes between different characteristics of the same term.

ii. the *imprecision* of the characteristic. One algorithm may be more accurate at describing a characteristic than another. For example, there are several ways to calculate the term frequency (*tf*) in a document[57], some of which are more effective on different collections or for different types of documents but which require more or less computation. So we may choose a less precise (less effective) algorithm that has better computational properties. This aspect distinguishes between different versions of the same characteristic, e.g. two versions of *theme*.

iii. the *quality* of the characteristic. Some characteristics may be better at indicating relevant material than others. The focus of my work is to select *which* characteristics best indicate relevance at a particular point in a search. As this may change over time, as the user

[57]See Harman [Har92] for an overview of term frequency measures.

refines what they are looking for, or as the information need changes, the characteristic may become better/worse at discriminating relevant material.

For example the *theme* characteristic may be very good at indicating relevance at the start of the search (looking for documents about a particular topic) but later in the search the *context* may become more important (looking for documents in which a term appears only in a particular context). The uncommitted belief can then be used to reflect the changing importance of each term characteristic at different points in the search. Evidence supporting changes in users' criteria of this kind has been shown by, for example, [Vak00, Ell89], and other studies that show that relevance, and the process of making relevance assessments, are dynamic processes.

In Chapters Four and Five I incorporated feedback weights – the discriminatory weights derived from analysing the values of the term characteristics in the relevant and non-relevant documents to reflect this aspect of uncertainty.

iv. the *strength* of the characteristic. Some characteristics should be considered to be more important than others independent of any other information. For example in Chapter Four I showed that certain characteristics worked better on different collections independent of any other evidence. This may be due to the idiosyncrasies of individual collections and queries but means that some characteristics may need to be treated as more/less important than others, regardless of the user's relevance assessments. The *strength* of the characteristic reflects the difference in quality of term characteristics reflecting different aspects of information use (*tf* as opposed to *theme*) rather than different implementation of the same characteristic (given by the *imprecision* of the characteristic).

This aspect reflects, in part, the suitability of a characteristic for a collection. For example, the *theme* characteristic is unlikely to show good performance for collections such as MEDLARS (Chapter Four, Table 3.2) which have short documents. This is because *theme*, as I have devised it, relies on multiple occurrences of a term within a document to derive *theme* values. Short documents are less likely to contain multiple occurrences of terms than long documents. Hence *theme* is probably more suited to collections with longer documents. The strength aspect is intended to reflect the *appropriateness* of a characteristic for a given collection or type of collection. The *strength* differs from the uncertainty as the strength values is based on the actual implementation whereas the uncertainty value is based on a conceptual view of what information the characteristic represents.

v. the *importance* of the term. The uncommitted belief can also be used to represent information that is not document or query dependent. For example, I use the *idf* as a characteristic which forms a *bpa* but I could have used the *idf* to calculate the uncommitted belief by increasing the uncommitted belief of terms that have a *low* idf. Also, some terms may be better at retrieving relevant documents than others or we may be more certain of their utility, e.g. query terms. So it may be appropriate to treat the evidence regarding these terms as more certain.

The first four uses of uncommitted belief, **i.-iv.**, describe various aspects of term characteristics as a whole. These four values may be combined to a single value of the overall uncommitted belief for each term characteristic. The fifth use can be used to modify the evidence supplied by any characteristic of an individual term. In this chapter I do not discuss how values for all these aspects can be obtained but, in a practical implementation, this will probably rely on experimentation.

6.3.7 Conclusion

DS is a suitable framework for integrating term characteristic information into the RF process for three reasons:

i. combination of evidence: Evidence in a RF situation comes from two sets of evidence - evidence derived from algorithms describing how words are used within documents, Chapter Three, and evidence from the user in the form of relevance assessments, section 6.5. The combination of evidence in DS is not only conceptually simple but it is easily implemented. DS then provides a formal but manageable method of combining evidence from a variety of sources.

ii. representation of imprecision: All evidence is not equal, especially in RF, where the reasons for relevance may change over a search. So we need to be able to represent the quality of evidence. DS provides this with the notion of uncommitted belief.

iii. functions to score documents: As will be discussed in sections 6.4.2. and 6.5.2 I show that we do not always want to score documents based on the same evidence at every stage in the search. The three DS functions - mass, belief and plausibility functions - provide alternative methods for different retrieval situations.

My main interest is in providing a model for RF. This is accomplished in two stages. The first stage is to develop a method of retrieving documents when we have no relevance information from the user. This provides an initial set of documents that the user can assess for relevance.

In the next section I describe how I use DS in combining evidence from term characteristics to provide such a retrieval function.

The second stage is to combine the retrieval function for retrieving documents with information from the users' relevance assessments, the RF situation. The feedback model is described in section 6.5 and is an extension of the initial retrieval model.

6.4 Initial document retrieval

IR systems normally present a ranking of documents to the user: the documents are ranked in decreasing order of retrieval score. There are two sources of evidence we can employ to decide on the score of a document: - the evidence given by the term characteristics and the evidence given by the user's relevance assessments. For initial retrievals we have no evidence from the user (no relevance assessments) and can only use term characteristic information, sections 6.4.1 and 6.4.2. With relevance information we can use both sources; this is described in sections 6.5 and 6.6.

6.4.1 Combining term characteristic information

The evidence given by the term characteristics is assigned to individual documents (singleton sets) with each characteristic of a term describing a mass function. This mass function will assign zero mass to each non-singleton set[58] and a non-zero score to each document that contains a positive score for a term characteristic. I use the combination rule to calculate the score of each document, thus taking into account all the term characteristics of a term.

Example one:
Suppose we only consider the single word query t_3. The combination of two characteristics - *theme* and *tf* - for this term allow us to score the documents in order of estimated relevance based on how this term is used in the documents, as shown in Table 6.4, Column 4.

In this example I have calculated the uncommitted belief according to equation 6.5. If the uncommitted belief for the *theme* characteristic is increased from 190 to 210 and for *tf* is increased from 150 to 210, then we get the scores in Table 6.4, Column 5.

The mass function is then altered by the uncommitted belief. The combination with unaltered uncommitted belief assigns most evidence to d_3, followed by d_1, d_4, and none to d_2 or d_5. Treating the *tf* characteristic as less reliable than *theme*, by assigning a greater degree of

[58]With the exception of the frame of discernment itself.

uncommitted belief, changes the mass function to assigning most evidence to d_1, then d_3, d_4 and none to d_2 or d_5. Thus the use of the uncommitted belief can shift the emphasis of the combined mass function in the direction of one or other sources of evidence.

Documents	***theme***	***tf***	**Combined score initial *ub***	**Combined score altered *ub***
d_1	45	20	55	35
d_2	0	0	0	0
d_3	15	50	60	28
d_4	0	30	27	11
d_5	0	0	0	0
ub	190	150	108	176

Table 6.4: Mass function gained by combining two characteristics of term t_3
where *ub* = uncommitted belief

As noted in section 6.3.2, the maximum mass that can be assigned to a document by a term characteristic is 50 but a term can receive a higher mass as the result of combination. This is not a problem as the total evidence (total mass function) still sums to 250, i.e. the combination does not alter the total evidence over the frame of discernment.

Example two:
As Dempster's rule is associative and commutative we can combine multiple characteristics of multiple terms. If we consider a two-term query, say t_3 and t_4 we obtain Table 6.5. We then obtain a ranking that takes into account how the terms are used in the different documents.

Documents	t_3			t_4			**Combined score**
	theme	***tf***	***context***	***theme***	***tf***	***context***	
d_1	45	20	0	0	0	0	48
d_2	0	0	0	30	10	0	17
d_3	15	50	25	25	30	25	128
d_4	0	30	0	0	0	0	19
d_5	0	0	0	50	20	0	32

Table 6.5: Mass function gained by combining three characteristics of terms t_3 and t_4

6.4.2 Ranking and retrieval

Given a mass function over the documents in the collection, how should the documents be ranked for presentation to the user? DS provides three functions for scoring documents: mass, belief and plausibility functions. In this case, as all the evidence is divided between the frame of discernment (the uncommitted belief) and the singleton sets the belief function equates to the mass function. So the choice is then between the mass/belief functions and the plausibility function.

In this situation the plausibility is equal to the sum of the mass assigned to the document and the uncommitted belief. As the uncommitted belief is the same for each document, i.e. not document dependent, then the plausibility and mass functions will give identical rankings although different scores.

As I am only interested in ranking the documents I choose the mass function, as the simplest of the three available functions, to rank documents. In example two, the documents would then be presented to the user in the following order: d_3, d_1, d_5, d_4, and finally d_2. d_3, the only document that contains both query terms (t_3 and t_4) is retrieved first, all the other documents only contains one query term each.

In the next section, I describe an experiment to test the effectiveness of the DS retrieval model for ranking documents.

6.4.3 Experiment

In this section I shall first describe the data I used for this experiment, section 6.4.3.1, a baseline and then results of combining evidence from the term characteristics, section 6.4.3.2.

6.4.3.1 Experimental setup

In these experiments I used the Wall Street Journal (1990-92) (**WSJ**) and the Associated Press (1988) (**AP**) test collections from the TREC-5 set of collections, [VH96]. The details of these collections are summarised in Table 6.6. I applied common IR indexing steps such as the removal of highly frequent terms and the reduction of terms to their root variant, [VR79]. These collections were also used in Chapter Four. As in Chapter Five I only use the *idf*, *tf*, *theme* and *context* characteristics as these experiments were completed before the ones presented in Chapter Four[59].

[59] Small implementation differences such as the sorting algorithm used to rank documents and rounding of retrieval scores give slightly different average precision results for the combination experiments presented in this

Collection	AP	WSJ
Number of documents	79 919	74 580
Number of queries used	48	45
Average words per query	3	3
Number of unique terms in collection	129 240	123 852

Table 6.6: Details of collections used

6.4.3.2 Retrieval by combination of evidence

In this experiment I compared the performance of using each combination of characteristics as a retrieval function. I compared two methods of combination; Dempster's combination rule and a simple summation method that consisted of summing the characteristic scores for each query term in a document. This latter method was the one used in Chapters Four and Five.

The results, then, compare the methods from Chapter Four (*simple* method) against a new method (*DS* method) of scoring documents using a combination of evidence. Table 6.7 (columns 2 and 3) shows the average precision for this experiment (full tables are in Appendix E, Tables E.1 - E.8)[60].

As indicated in sections 6.1 and 6.3.6 it may not be appropriate to treat each characteristic as equally important in retrieving relevant documents. Consequently we also tried weighting each characteristic with different values to investigate the effect of different uncommitted beliefs on the combination. The results from this experiment are shown in Table 6.7 (columns 4 and 5).

---

chapter and in Chapter Four. I have given full recall-precision tables for the experiments in this chapter in Appendix E.

[60]As I lacked a formal theory to decide how to select good values to alter the uncommitted belief for characteristics, I weighted each characteristics in an ad-hoc manner with the following values: *idf* -1, *tf* - 0.75, *theme* - 0.15, *context* - 0.5. These were identical to those used in Chapter Three for the same collections. Different weights give different results, as indicated in Table E.16, for the combination of all characteristics on the CISI collection.

AP				
Combination	**simple, no weighting**	**DS, no weighting**	**simple, weighting**	**DS, weighting**
all	11.2	8.5	13.3	**16.5**
context	**9.6**	**9.6**	**9.6**	**9.6**
idf	**10.1**	**10.1**	**10.1**	**10.1**
idf + ***context***	10.4	**12.6**	10.2	12.5
idf + ***tf***	12.9	6.6	**13.1**	13
idf + ***tf*** + ***context***	**13.8**	13	13.4	2.2
idf + ***tf*** + ***theme***	9.9	1.9	13.1	**14.8**
idf + ***theme***	5.1	**14.2**	10.5	12.2
idf + ***theme*** + ***context***	9.9	**16.6**	11.5	12.9
tf	**9.9**	**9.9**	**9.9**	**9.9**
tf + ***context***	12.3	5.4	**12.4**	2.9
tf + ***theme***	8.8	7.4	**10.2**	7.7
tf + ***theme*** + ***context***	10.8	3.5	**12.5**	3.1
theme	**4.6**	**4.6**	**4.6**	**4.6**
theme + ***context***	9.4	8.9	**10.6**	9.9

WSJ				
Combination	**simple, no weighting**	**DS, no weighting**	**simple, weighting**	**DS, weighting**
all	12.7	14.7	**15.1**	14.2
context	**0**	**0**	**0**	**0**
idf	**12.2**	**12.2**	**12.2**	**12.2**
idf + ***context***	11	5.8	11.5	**12**
idf + ***tf***	15.2	15.6	15.4	**15.8**
idf + ***tf*** + ***context***	15	15.1	**15.2**	13.8
idf + ***tf*** + ***theme***	12.6	**19.9**	14.4	15.3
idf + ***theme***	11.2	11.2	**13.1**	12.6
idf + ***theme*** + ***context***	11.6	13.5	13.3	**14.8**
tf	**7.4**	**7.4**	**7.4**	**7.4**
tf + ***context***	14.3	**15.2**	14.2	**15.2**
tf + ***theme***	9.3	**15.8**	10.3	0.6
tf + ***theme*** + ***context***	12.4	9.5	**14.5**	1
theme	**1**	**1**	**1**	**1**
theme + ***context***	11	**14.6**	12.2	14

Table 6.7: Summarised results of combining characteristics

Table 6.7 shows the results of using Dempster's combination rule (**DS**), simply summing characteristic scores (**simple**), and either weighting the characteristic scores (**weighting**) or treating characteristics as equally important (**no weighting**). ***all*** is the combination of all characteristics. The highest average precision value for each combination is shown in bold.

AP				WSJ			
	No weighting	**Weighting**	**Total**		**No weighting**	**Weighting**	**Total**
simple	1	5	**6**	**simple**	0	4	**4**
DS	3	1	**4**	**DS**	3	4	**7**
Total	**4**	**6**		**Total**	**3**	**8**	

Table 6.8: Number of times each combination strategies gave highest average precision

Table 6.8 summarises how often each strategy obtained the highest average precision for a given combination, excluding single characteristics. This compares combining characteristics

using Dempster's combination rule (**DS**), summing characteristic scores (**simple**), either weighting the characteristic scores (**weighting**) or treating characteristics as equally important (**no weighting**). This count omits the single characteristic combinations as these are unaffected by the combination strategy or weighting.

The results can be compared under two conditions: the different combination methods and the effect of weighting the importance of the characteristics relative to each other.

i. Method of combination. From Tables 6.7 and 6.8 it can be seen that the method of combining the characteristic information does not have a big effect on how successful the strategies were overall. That is, using Dempster's combination rule instead of simply summing the characteristic scores did not significantly increase the number of combinations that gave higher average precision. This is not surprising as the way I have used the DS theory so far is basically also a summation method.

However, from Tables E.9 - E.10 in the Appendix, it is clear that the combination rule is having an effect. In particular, the different combination methods change the relative ordering of which combination of characteristics give better results, i.e. some combinations perform better using Dempster's combination rule and some perform better using the simple addition method. The combinations that involve a combination of *tf* and another characteristic tend to perform worse with the DS method than the simple method, whereas methods that combine *idf* do better with the DS method.

One possible cause of this effect is due to the way I assign the mass function. Although I manipulate the amount of mass assigned to each document by varying the uncommitted belief function, each characteristic will assign mass to a different *number* of focal elements. For example, the *idf* characteristic of a term will assign evidence to every document that contains the term; the other characteristics will only assign evidence to documents for which the characteristic has a non-zero value. As the values of *theme*, *context*, and *tf* may be zero for a number of documents in each case, it is likely that each of these characteristics will not only assign different values to each document, but also assign values to a variable number of documents.

In the DS method this will have the effect of increasing the uncommitted belief for the characteristics which assign a mass value to fewer focal elements. Thus the characteristics that assign mass to the fewest number of focal elements will have the least effect on scoring the documents. The DS method, then, biases retrieval in favour of characteristics that assign evidence to more characteristics. In our case this is *idf* so the results of a combination of *idf*

will be closer to the results given by *idf* alone. As *idf* is the best single retrieval function, DS generally gives better results for combinations with *idf*. The different characteristics also assign values to different numbers of characteristics using the simple method. However as the combination in the simple method is not affected by the total mass assigned to the documents (as is the case in the DS method, through the uncommitted belief) this bias does not occur.

ii. Weighting of characteristics. Although the method of combination did not produce any significant effects, treating different characteristics with varying importance to other characteristics did produce better overall results than treating all characteristics as equally important. Weighting of characteristics not only increases the average precision of most combinations of characteristics, it also modifies which combinations give better results in both methods of combination. For example, in Table 6.7 (**AP**), the combination of all characteristics performs better than the combination of *idf* and *tf* information, if weighting is used and poorer if weighting is not used.

In both collections the combination of DS and weighting can improve retrieval effectiveness although only slightly. Although I have not shown a clear advantage in using the DS combination rule in combining evidence from characteristics, I believe that the flexibility of the uncommitted belief in representing the various forms of uncertainty discussed above hold the potential for improved results. In particular the use of DS potentially allows the derivation of better weights for representing the importance of the different characteristics. This is because we can formally examine the effect of the uncommitted belief on retrieval effectiveness. For example, we could examine methods of weighting proportionally to the number of focal elements assigned a mass value by a term characteristic or how mass is distributed between focal elements. These two aspects, and others related to how the mass is assigned by a characteristic, may be important for the undercovering the reasons for the success or failure of a term characteristic in retrieval.

6.4.4 Summary

Sections 6.4.1 and 6.4.2 described how to score and rank documents using term characteristics. I have demonstrated, in section 6.4.3, that combining characteristics of information use under two methods (DS and simple) can increase average precision. I have also shown that Dempster's combination rule performs in the same range as a standard method of scoring documents and that characteristics should be treated as of varying importance.

I now turn to RF. My approach is to treat the relevance information from the user - the list of documents they regard as containing relevant information - as an additional source of

evidence to be combined. The RF model is an extension of the model outlined in the previous section but extended to incorporate RF information.

6.5 Relevance feedback

In a RF situation we want to extrapolate from the information in the relevant documents to facilitate the retrieval of more relevant documents. That is we want to use the information in the documents the user has marked relevant to help retrieve documents that the user may also consider relevant. In this section I suggest how this might be achieved in my combination model, section 6.4.1, and how documents should be ranked when we have RF information. In section 6.6 I describe a set of experiments designed to test the effectiveness of this approach.

6.5.1 Combination of characteristics with relevance information

When we have relevance information from the user, we have two sources of evidence to rank documents: the term characteristic information and the relevant documents. I have described how I use the term characteristic information to rank documents in section 6.4.2. The question now is how to use the term characteristic information in relevant and non-relevant documents? That is, how do we integrate evidence from the user with our DS model to define a *bpa* over the frame of discernment? There are a number of options:

i. we can treat the *value* of a term characteristic as important. In the working example the *theme* value of term t_3 in document d_1 is 45. If d_1 is relevant then we could say that a value of 45 for this characteristic of this term is a good indicator of relevance. However it cannot be claimed, with any credibility, that individual numerical values of a term characteristic leads to relevance; it is only possible to say that a thematic relation for a term indicates relevance better than no thematic relation.

ii. we can treat the values for individual documents as a range, e.g. the *theme* value of term t_3 in document d_1 is 45 and in document d_3 it is 15. If both these documents, and no others, have so far been assessed relevant then it may be assumed that only documents which have t_3 *theme* values in the range 15-45 should be considered. However the users may make few relevant judgements and it cannot be asserted for certain that one particular characteristic is the one that defines relevance. Also it cannot be guaranteed that users will have seen or assessed documents with *theme* values outside this range so we have no certainty that this range is significant.

iii. we can treat the evidence more generally by asserting that the *value* of particular term characteristics do not define which values are important, as in **i.** and **ii**., but instead define how well the characteristic predicts relevance based on its appearance in the relevant and non-relevant documents. Let us assume that the query contains one term, t_4, and documents d_2 and d_5 have been marked relevant. For each term characteristics there are four cases to consider, based on the presence/absence of the term t_4 in the relevant and non-relevant documents. These are outlined in Table 6.9.

	t_4 theme characteristic	
Relevance	**Present**	**Absent**
Relevant	$\{d_2, d_5\}$	$\{\}$
Non-relevant	$\{d_3\}$	$\{d_1, d_4\}$

Table 6.9: Contingency table based on the presence/absence of the *theme* characteristic of t_4 in the relevant and non-relevant documents

The first set of documents contain those that are relevant and display the term characteristic ($\{d_2, d_5\}$), the second contain the non-relevant documents that display the term characteristic ($\{d_3\}$). It is possible to derive values for each of the cells that display the term characteristic by simply averaging the characteristic value of the term in each document in the cell. In the example the average *theme* score for query term t_4 is 20[61] in the relevant set displaying the characteristic and 25 in the non-relevant set displaying the characteristic so we assign a mass of 20 to the set $\{d_2, d_5\}$ and 25 to the set $\{d_3\}$ shown in Table 6.9. The uncommitted belief is 205 (250-(25+20)).

The other two cells (right hand column of Table 6.9) contain the sets that do not display the term characteristic and are either relevant or not-relevant. As the term characteristic of a term that does not appear in a document is automatically 0, the mass assigned to these sets is 0. In this way, we only consider the cells that indicate presence of a term[62].

[61] Calculated from the values given in Table 6.1.

[62] D-S expressly forbids the use of negative evidence (something that does not happen) being used to assign evidence. In this situation DS differs from the F_4 weighting scheme, [RSJ76], which uses statistical information and a similar contingency table to derive weights that incorporate information on the absence of a term in a relevant/non-relevant document.

Repeating this for the *tf* characteristic would give a mass of 15 to the set $\{d_2, d_5\}$ and 30 to the set $\{d_3\}$ with an uncommitted belief of 210. These two mass functions can be combined using Dempster's combination rule to provide a single mass function based on the two term characteristics as demonstrated in example two.

I demonstrate the full model of RF incorporating user's relevance assessments and term characteristics in Example three.

Example three:
The simplest case is to consider RF with one relevant document. Assume that the user has issued a query, has marked document d_3 as relevant and has made no relevance decision on the other four documents[63]. For each query term in document d_3 there is some indication of how useful the term may be in detecting relevance[64].

	t_4		t_5	
	set	**mass**	**set**	**mass**
theme				
relevant	$\{d_3\}$	25	$\{d_3\}$	0
non-relevant	$\{d_2, d_5\}$	20	$\{d_2, d_4\}$	5
context				
relevant	$\{d_3\}$	15	$\{d_3\}$	40
non-relevant	$\{d_2, d_5\}$	20	$\{d_5\}$	20
tf				
relevant	$\{d_3\}$	30	$\{d_3\}$	30
non-relevant	$\{d_2, d_5\}$	15	$\{d_2, d_4\}$	20

Table 6.10: Mass functions based on relevance assessments

[63]It is customary in IR to assume that the documents that have not been marked explicitly as relevant or non-relevant can be assumed non-relevant, although they in all likelihood will contain a number of relevant documents that have either not been retrieved by the system or not been assessed by the user.

[64]Of course, it may be that a characteristic only appears by chance, and relevance is better described by another characteristic. By taking into account the characteristics of terms in non-relevant documents I can limit this to a certain extent - by only considering characteristics that better describe relevant documents than non-relevant documents.

The current query is composed of the terms t_4, and t_5. In Table 6.10 I show the various sets that are assigned a mass value based on this document selection. Also I have filled in values for the *context* characteristic.

Dempster's combination rule can then be used to obtain a single mass function based on the mass functions from t_4, and t_5, Table 6.11(**a**). All other subsets of the frame of discernment are assumed to have zero mass. The evidence from the relevance assessments can be combined with the evidence from term characteristics for t_4, and t_5, Table 6.11(**b**), to form a single mass function, Table 6.11(**c**). In none of the mass functions in Table 6.11 do I assign all the possible evidence - there is uncommitted belief at each stage.

Set	**mass**
$\{d_1\}$	0
$\{d_2\}$	7
$\{d_2, d_4\}$	11
$\{d_2, d_5\}$	40
$\{d_3\}$	86
$\{d_4\}$	0
$\{d_5\}$	10
ub	96

a

Set	**mass**
$\{d_1\}$	0
$\{d_2\}$	70
$\{d_2, d_4\}$	0
$\{d_2, d_5\}$	0
$\{d_3\}$	43
$\{d_4\}$	37
$\{d_5\}$	32
ub	68

b

Set	**mass**
$\{d_1\}$	0
$\{d_2\}$	48
$\{d_2, d_4\}$	1
$\{d_2, d_5\}$	18
$\{d_3\}$	73
$\{d_4\}$	23
$\{d_5\}$	26
ub	61

c

Table 6.11: Combination of evidence from multiple sources

a. mass function from combing relevance information only

b. mass function from combining term characteristic information only

c. mass function from combining relevance information and term characteristic information

The results of the final combination, Table 6.11(**c**), is represented diagramatically in Figure 6.1.

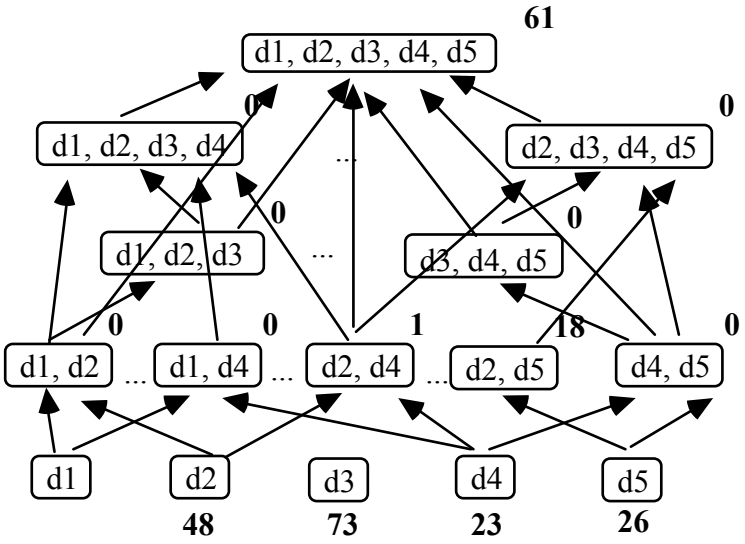

Figure 6.1: Diagrammatic representation of the combination of characteristics in a RF situation.
→ represents subset relation. Figures indicate mass values

In section 6.3.6 I enumerated a number of uses for the uncommitted belief (four of which reflected the quality of term characteristics, one which reflected the quality of individual terms). There are three further uses for the uncommitted belief when we have relevance information:

i. *partial* relevance assessments. Most IR systems only allow users to mark a document as relevant or not-relevant. However, researchers such as Borlund and Ingwersen, [BI97], have investigated the use of partial relevance assessments: asking users to give a numerical value describing the relevance of a document. I can use this information to modify the uncommitted belief of a term according to whether it appears in a highly-relevant or slightly-relevant document.

ii. *source of evidence* - biasing evidence between relevance assessments and query. Evidence from research such as Salton and Buckley, [SB90], indicates that relevance information and query information should not always be treated as being equally important. Furthermore, Haines and Croft, [HC93], showed that this is collection dependent; in some collections, better retrieval effectiveness is achieved by treating query terms as more important, and in other collections we should treat user relevance as being more important. The uncommitted belief, then, may be used to bias retrieval in favour of term characteristics appearing in the original query or those added from the user-selected relevant documents. If I extend my approach to include query term expansion, e.g. [Roc71], I could also bias evidence between

the original query characteristics of terms and characteristics of new query terms suggested by the system.

iii. *time of evidence*. In section 6.1 **iii.**, I argued the characteristics of a term that best indicate relevance can change over time. One reason for this is that a user may change her criteria for assessing relevance in the light of the relevant material. Typically RF algorithms do not consider time in deciding how to modify queries: each relevant document is considered to be an equal contributor to RF regardless of when in the search a document was assessed relevant. New relevance assessments can gradually change the system's view of which characteristics indicate relevance but a better way of handling the order in which assessments are made is by the use of ostensive weighting, suggested by Campbell and Van Rijsbergen, [CVR96].

Ostensive weighting of evidence, in a RF context, means treating the most recent relevance assessments as the best source of evidence regarding what the user regards as relevant material. Relevance assessments made early in the search, on the other hand, should be regarded as poorer indications of relevance. We can use the uncommitted belief to reflect this. If a term only appears in documents assessed early in the search, we should increase our uncertainty (uncommitted belief) regarding the term's utility for RF; if a term appears in the most recent relevant documents, they should be regarded as better evidence for RF and have a lower uncommitted belief.

6.5.2 Ranking and retrieval with relevance information

To re-rank documents after RF I need to obtain a score for each document; the characteristics give us a score for each document (section 6.2) and the relevance assessments can be used to give us a score for sets that represent the useful characteristics (section 6.5.1). I have three ways to score a document: mass, belief and plausibility functions, which we discuss in turn below.

i. mass function. The mass function considers the score for each set, and only that score. Intuitively this is not what we want as the characteristic evidence only gives a score to singleton sets and the RF evidence will tend to give evidence to non-singleton sets. We want a method that will score the documents on all the evidence available.

ii. belief function. The belief function measures the total evidence supporting a set, based on the mass assigned to itself and its subsets. If I was working on a model for calculating the score of a set of documents, e.g. in a clustering model, then this is exactly what I would want because it would calculate the score of all the sets including the non-singleton sets. However I am at the moment only interested in ranking the singleton sets (individual documents) so the

belief function is the exact opposite of what I require because it uses the evidence of the singleton sets to score the non-singleton sets, rather than the other way round.

iii. plausibility function. The plausibility function considers the total plausible evidence for a set. This is the mass for a set and all the sets with which it intersects. This is then what is required - a function that combines the evidence from the characteristics (attached mainly to the singleton sets) and for the usefulness of the characteristics (attached to the non-singleton sets). This method will score all sets (the singleton document sets and those sets containing more than one document). However when ranking the documents we need consider the singleton document sets as the user will only be presented with a list of ranked documents.

Document d_i	***Pl*(d_i)**
$\{d_1\}$	61
$\{d_2\}$	128
$\{d_3\}$	134
$\{d_4\}$	85
$\{d_5\}$	105

Table 6.12: Documents scored by plausibility function

Scoring the documents from Example 3, Table 6.11(**c**), according to the plausibility function, we arrive at the scores in Table 6.12 for the singleton document sets. In this case we would retrieve the documents in the order d_3 then d_2, d_5, d_4 and finally d_1. As d_3 is the only document marked relevant by the user, we should expect this to come at the top of the retrieved documents. d_2 is retrieved second as it contains both query terms and both query terms display the term characteristics. Documents d_5 and d_4 which both contain one query term appear next. d_5 is retrieved ahead of d_4 as the one query term it contains better displays the *theme* and *tf* characteristics than the query term contained within d_4. d_1 correctly appears at the bottom of the ranking as it does not contain either query term.

6.6 Experiments on RF

I now describe the experiments on RF. In these experiment I investigate the use of term characteristics and DS in the context of RF. I introduce the data I used in these experiments in section 6.1, the baseline comparison measures in section 6.6.2, the methodology in section 6.6.3 and the results of the experiments in sections 6.6.4 - 6.6.6. I summarise the results in section 6.7.

6.6.1 Data

In this experiment I used a different collection from the experiments in section 6.4.3, as my particular implementation of the model is computationally expensive. The collection I used is the CISI collection, details of which are given in Table 6.13. This collection contains fewer and shorter documents than either the AP or WSJ collection making it an easier collection upon which to experiment. This collection has much higher number of query terms per query, although the average query term count is skewed somewhat by some very long queries.

Collection	**CISI**
Number of documents	1 460
Number of queries used	76
Average words per query	27.3
Number of unique terms in the collection	7 156

Table 6.13: Details of CISI collection

I carried out identical combination experiments to those described in section 6.4.3 for the CISI collection. These are reported in Appendix E, Tables E.11 - E.16. The results I have previously obtained hold: combining information can improve retrieval effectiveness, weighting characteristics often improves retrieval effectiveness and DS and the simple combination method perform approximately as well as each other. The main differences between the two collections used previously and the CISI collection is that *tf* is a better single retrieval function than *idf*, and that *theme* and *context* give higher average precision when used as a single retrieval function than on the AP and WSJ collections.

6.6.2 Baseline measures

In sections 6.6.2.1 - 6.6.2.3 I introduce the three baseline measures I used to compare the RF method. These are the same baselines as used in Chapters Four and Five.

6.6.2.1 No feedback

The first baseline is the retrieval results obtained from doing no RF. For the CISI collection this is the combination of all characteristics combined using Dempster's combination rule. The characteristics were weighted as follows: *idf* - 1, *tf* - 0.75, *theme* - 0.15, *context* - 0.25.

6.6.2.2 Best combination

It may be that a better retrieval result could be obtained by using a good combination of characteristics rather than using RF. That is, we want to test whether the quality of the retrieval function is more important than the quality of the query: is developing a good query

(through RF) more important than developing a good retrieval function (selecting the best overall combination of characteristics)? To test this, the second baseline is the best combination of characteristics from the experiments on combination of evidence. This is a combination of *tf* and *idf* for the CISI collection, Table E.12.

6.6.2.3 F4

As in Chapters Four and Five I used the F_4 term reweighting scheme as a baseline RF measure.

6.6.3 Methodology

I carried out three experiments to test the performance of three aspects of the overall approach outlined in Chapters Four and Five; weighting of characteristics, selecting characteristics of terms and method of combination of characteristic information. I isolate these three stages to allow me to investigate what aspects of the general approach are successful and the relative effectiveness of the approaches when using DS. The first two experiments are similar to those described in Chapters Four and Five but are examined in more detail. I will briefly outline the methodology used then introduce each of the experiments in sections 6.6.4 – 6.6.6.

In each of the three experiments I used the following methodology:

i. documents were ranked using the combination of all characteristics, combined using Dempster's combination rule. This is the same ranking function as the first baseline.

ii. a cut-off was applied at rank position 30. Documents at or above this rank position were used to modify the query.

iii. documents in positions 30 - *N* (where *N* is the number of documents in the collection) were rescored by one of the methods described in sections 6.6.4 - 6.6.6. Each method corresponds to one of the experiments outlined above.

iv. recall-precision figures were calculated over the whole document ranking using a freezing method of evaluation.

These steps were applied for 4 iterations, or cycles, of RF (steps **i.** - **iv.** were followed for a cut-off at 30 documents, then steps **ii.** - **iv.** were followed for a cut-off at 60 documents, a cut-off at 90 documents, etc). This resulted in five document rankings. Results will be presented as the average precision of each ranking. Full RP tables are given in Appendix E, Tables E.18 – E.31.

6.6.4 Experiment one - RF using derived weighting factors

In Experiment One I repeat the Feedback 5 strategy from Chapter Four. This Feedback strategy assigns discriminatory weights to the combination of a term and characteristic based

on the average value of the term and characteristic in the relevant documents and in the non-relevant documents. I am then considering how good a characteristic of a term is at discriminating relevance.

Four versions of this approach were considered to test the effectiveness of incorporating more aspects of uncertainty into the combination process:

i. *Feedback 5.1*. This version performs an initial ranking using the no feedback baseline (section 6.6.2.1). In RF all characteristics of each query term are weighted by the Feedback 5 strategy. This method only uses the indexing weights of a term characteristic and the discriminatory power of a term characteristic to score documents.

ii. *Feedback 5.2*. This version is identical to Feedback 5.1 except that characteristics are not weighted for the initial ranking. The comparison of Feedback 5.1 and Feedback 5.2 indicates how important are the scaling factors that represent how good the characteristic is at retrieval.

iii. *Feedback 5*.3. This version is also identical to Feedback 5.1 except that the initial ranking is performed by the *idf* characteristic alone. The difference between Feedback 5.1 – 5.3 reflects the importance of the initial ranking in overall performance.

iv. *Feedback 5.4*. The final version of 5.1 uses the indexing weights, the derived discriminatory weights and the scaling weights. That is it uses the weights assigned by the term characteristics, the scaling factors that determine how good are the characteristics and weights that represent how well the term and characteristic differentiate relevance. The scaling factors weight each characteristics to reflect its *strength* (section 6.3.6), in Feedback 5.1 – 5.3 we weight each characteristic according to its *quality*. Feedback 5.4 combines these two attributes.

Table 6.14 summarises the information used by each Feedback 5 strategy for initial ranking and RF. In Table 6.14 I outline which characteristics are used to provide the initial ranking and feedback rankings (columns 3 and 5) and which sources of uncertainty are used to calculate the uncommitted belief (columns 2 and 4).

Feedback strategy	Initial ranking uncommitted belief calculated by	Initial ranking characteristics used	RF uncommitted belief calculated by	RF characteristics used
5.1	**i.** indexing weights **ii.** characteristic *strength* (scaling factors)	*idf*, *tf*, *theme*, *context*	**i.** indexing weights **ii.** characteristic *quality* (discriminatory power)	*idf*, *tf*, *theme*, *context*
5.2	**i.** indexing weights	*idf*, *tf*, *theme*, *context*	**i.** indexing weights **ii.** characteristic *quality*	*idf*, *tf*, *theme*, *context*
5.3	**i.** indexing weights	*idf*	**i.** indexing weights **ii.** characteristic *quality*	*idf*, *tf*, *theme*, *context*
5.4	**i.** indexing weights **ii.** characteristic *strength*	*idf*, *tf*, *theme*, *context*	**i.** indexing weights **ii.** characteristic *quality* **iii.** characteristic *strength*	*idf*, *tf*, *theme*, *context*

Table 6.14: Sources of evidence for Feedback 5 methods.

Table 6.15 gives the results of the four versions of Feedback 5. Comparing the Feedback 5.1 against Feedback 5.2 it can be seen that Feedback 5.1 (using scaling factors for the initial ranking) gave higher overall average precision. Not using any scaling factors gave a slightly greater percentage increase probably as the poorer initial ranking meant that an increase in performance was easier to obtain.

	CISI						
Iteration	**No feedback**	**Best combination**	**F_4**	**5.1**	**5.2**	**5.3**	**5.4**
0	11.7	**12.9**	11.7	11.7	9.4	11.5	11.7
1	11.7	12.9	14.0	14.4	11.5	14.0	**14.6**
2	11.7	12.9	13.9	14.4	11.9	14.4	**14.6**
3	11.7	12.9	13.9	14.8	12.0	14.3	**14.9**
4	11.7	12.9	13.8	14.9	12.1	14.5	**15.0**
%increase	**-**	**0.0**	**17.9**	**27.4**	**28.7**	**26.1**	**28.2**

Table 6.15: Results of Feedback 5 methods.
Highest value for each iteration is shown in bold. %age increase = percentage increase over no feedback

Measuring Feedback 5.1, Feedback 5.2 and Feedback 5.3, which only differed in their initial rankings, it is clear that better initial rankings give better end results: feedback will improve

good initial results and the end results will still be better than those achieved on the poorer initial ranking. After sufficient iterations of feedback, all techniques will retrieve all the relevant documents but the point here is that better initial rankings will help retrieve the relevant documents more quickly.

Feedback 5.1 – 5.3 all used discriminatory weights that reflect how well the term and characteristics discriminate relevance. All three strategies gave an increase in performance over no feedback, demonstrating that weighting characteristics by how well they discriminate can improve feedback without any other query modification.

In addition Feedback 5.1 and 5.4 both of which used scaling factors – the *strength* of the characteristics - for the initial ranking outperformed the Best Combination and F_4 baselines which also used scaling factors. This demonstrates that good initial rankings are important. The most successful approach, 5.4, used all three sources of uncertainty. This again shows that we need methods to model the uncertainty involved in combining evidence.

6.6.5 Experiment two - RF using selective combination of evidence

In Chapters Four and Five I demonstrated that selecting characteristics could give better performance over no selection of characteristics. I now investigate this when using DS scoring technique.

In this experiments I explored two cases investigating two parameters: the selection of characteristics alone and affect of weighting of characteristics on the success of selection, The results from these cases are discussed in section 6.6.5.1 (comparing selection against no selection of characteristics) and section 6.6.5.2 (comparing different weighting methods with selection).

6.6.5.1 Selecting characteristics

In Table 6.16, I compare the selection of characteristics against no selection. I examine four cases to compare the effect of weighting characteristics (by scaling factors) against the selection of characteristics. This compares whether the weighting of characteristics is more effective than the selection of characteristics.

CISI					
Iteration	**F_4 + Scaling factors**	**No Scaling factors + No Selection**	**No Scaling factors + Selection**	**Scaling factors + No Selection**	**Scaling factors + Selection**
0	11.7	9.4	9.4	**11.7**	**11.7**
1	14.0	9.4	10.9	11.7	13.1
2	13.9	9.4	11.3	11.7	13.3
3	13.9	9.4	11.3	11.7	13.4
4	13.8	9.4	11.3	11.7	13.5

Table 6.16: Average precision figures for initial rankings experiments. Highest values at each iteration shown in bold.

Comparing the two cases where selection of characteristics is performed (columns 4 and 6) it can be seen that selection does give substantial improvements over no selection (comparing column 3 against column 4 and column 5 against column 6). Selection of characteristics performs better than no feedback – it works as a RF technique and also performs better than the Best Combination of characteristics. However selection on its own[65] does not the level of performance of the F_4 RF technique on this collection. Selection the characteristics is more effective than simply weighting the characteristics.

6.6.5.2 Weighting and selection

In section 6.6.4, I demonstrated that more evidence as to the uncertainty of the characteristics gave better results. In this section I demonstrate that this is also true when we select the characteristics to be used in feedback. In Table 6.17 I present the results of four feedback trials; each use the same selection of characteristics but have different information on which to score documents.

As outlined in section 6.6.4 we have three sources of uncertainty: the indexing weights, the strength of the characteristics at retrieving relevant information (the scaling factors) and the quality of the characteristics at discriminating relevant from non-relevant material (the discriminatory weights). These can be used to give different methods of scoring documents.

[65] Only selection of characteristics, with no use of the discriminatory power of characteristics, corresponds to the Feedback 1 strategy from Chapters Four and Five.

The first selection feedback trial is selection of characteristics using only the indexing weights to score documents (Table 6.17 column 3). This performs most poorly and does not perform as well as the baseline feedback F_4 measure.

The second selection trial combines the indexing weights and the characteristic strength weights (Table 6.17 column 4). This trial performs better than only using indexing weights but not quite as well as the F_4 measure.

The third trial combines the indexing weights and the characteristic quality weights (Table 6.17 column 5). This trial uses the *discriminatory* power of a characteristic of a term, whereas the second trial only uses the power of a term at retrieving relevant information. This is equivalent to using selection and the Feedback 5.1 method of scoring. This performs well, outperforming the two selection feedback trials, the Best Combination and no feedback baselines and the F_4 baseline.

The final trial combines all three sources of uncertainty (Table 6.17 column 6) and uses the Feedback 5.4 method of scoring and selection of characteristics. This is the most successful of the four selection trials leading to the conclusion that the more evidence we have on how to *use* characteristics of terms the better. This version also performs better than all the baselines including the F_4.

Selecting term characteristics on a query-query basis, then, can improve retrieval effectiveness over what we can achieve from weighting alone, and over the best individual combination of characteristics. The addition of more information on how to weight characteristics of terms can give increased performance.

Iteration	F_4 Weighting	No Weighting Selection	Weighting Selection	Feedback 5.1 Selection	Feedback 5.4 Selection
0	**11.7**	9.3	**11.7**	**11.7**	**11.7**
1	14.0	10.9	13.1	14.4	**14.8**
2	13.9	11.3	13.3	14.5	**15.2**
3	13.9	11.3	13.4	14.9	**15.4**
4	13.8	11.3	13.5	15.1	**15.5**

Table 6.17: Average precision figures for selection experiments.
Highest values at each iteration shown in bold.

6.6.6 Experiment three - RF based on full model

The final experiment explores the method of combination of evidence; either only using values of characteristics derived from indexing (as in section 6.4) or combining these values according to the RF model outlined in section 6.5.

In this experiment I compare selection with four uses of weighting (selection using only the index weights, Table 6.18 column 5; selection and weighting by characteristic strength, column 6; selection and weighting by characteristic quality, column 7; selection and weighting by index weight, characteristic strength and characteristic quality, column 8). The baselines are shown in Table 6.18, columns 2 - 4.

CISI							
Iteration	**No feedback**	**Best combination**	**$F_{4.5}$**	**DS index selection**	**DS index strength selection**	**DS index quality selection**	**DS index strength quality selection**
0	11.7	**12.9**	11.7	9.4	11.7	11.7	11.7
1	11.7	12.9	14.0	10.8	**13.3**	10.9	11.6
2	11.7	12.9	**13.9**	10.9	13.4	12.3	**13.2**
3	11.7	12.9	13.9	11.2	13.7	13.0	**14.1**
4	11.7	12.9	13.8	11.3	13.7	13.2	**14.2**

Table 6.18: Results of using full DS model.
Highest average precision figures are shown in bold.

The results of the model of RF again show the merits of weighting and selecting characteristics of terms, with the biggest increase in average precision given by the combination of weighting and selection. Comparing these results against those obtained in sections 6.4 and 6.5 we see that this model slightly decreases performance in the cases where we use the quality (discriminatory power) of the characteristics[66]. If we use the index weights and selection only, then we achieve the same performance after four iterations. If we use the strength of the characteristic (scaling factors) then we do achieve an increase in performance. Only one of the four versions (column 8) outperforms the Best Combination baseline. The

[66] Table 6.18 column 5 should be compared with Table 6.16 column 3, Table 6.16 columns 6, 7 and 8 should be compared with Table 6.17 columns 4,5 and 6 respectively.

method that uses selection and all three sources of uncertainty (indexing weights, strength and quality) performs best (column 8). However this performs less well than the comparable method that does not use the feedback model suggested in section 6.5.

6.6.7 Summary

In this section I summarise the results of these experiments under three conditions:

i. *weighting of characteristics.* Incorporating evidence on the relative importance of terms is important for two reasons. Firstly, it will generally improve initial rankings, bringing more relevant documents higher up the ranking. This means that more relevant documents are likely to come into the documents we use for query modification and so increase the evidence we have to differentiate relevant documents from irrelevant ones. Secondly, as shown in section 6.4 we can use the discriminatory power of a term in discriminating relevant and non-relevant documents to weight characteristics to give improved retrieval of relevant documents. Combining more than one source of uncertainty of term characteristics can improve retrieval effectiveness even more than when only using one source.

This latter finding is significant as it demonstrates that incorporating information on the various sources of uncertainty in the retrieval process can improve retrieval effectiveness. This combination of uncertainty is an important aspect of our DS model, and the use of a formal model, such as DS, means that we can start isolating exactly how the different sources of uncertainty affect retrieval effectiveness.

ii. *selection of characteristics.* Selecting good characteristics of terms - those that are more likely to retrieve relevant documents than irrelevant ones also improves retrieval effectiveness, section 6.5. Combining this information with weighting can improve retrieval effectiveness even more than either technique alone. The weighting of characteristics incorporates the uncertainty regarding the evidence we use in combination, the selection procedure dictates to what evidence the combination is applied. This reflects back to the work described in section 1.1 by Belkin et al, [BKF+95], who suggest evidence combination should be tailored to individual queries. This is one aspect of such a tailoring process.

iii. *method of combining evidence.* The final experiment compared the effect of treating relevance information from the user as an additional source of evidence, as outlined in section 6.5, against query modification alone. The results from this experiment were not as effective as I hoped, in that incorporating RF information in the way I implemented it, tended to decrease performance. This may be because the model is not yet sophisticated enough in the manner in which it handles user relevance information. However the particular model I

outlined in section 6.5 is only one method of exploiting RF information, and the general approach to RF is still valid. The use of such a formal model allows us, however, to analyse where and in what way individual interpretations of this model are successful. This is the subject of ongoing research.

6.7 Conclusion

In this chapter I have proposed a model for RF that allows the integration of how terms are used within documents into the RF process. The core of this approach is the combination of evidence from algorithms describing the information use of terms and relevance information from users. This model is based on Dempster-Shafer's Theory of Evidence which allows flexibility in how we combine this evidence: it allows us to include the quality of evidence (via the uncommitted belief), whilst providing a uniform framework for combining evidence. It also allows the use of information in different ways to retrieve documents; so we retrieve documents using different scoring functions in the presence/absence of RF information (when we have no relevance information we use the mass function, and when we have relevance information from the user we use the plausibility function).

I also showed how the notion of uncommitted belief can be used to represent and combine various sources of uncertainty in the RF process. These aspects are described in sections 6.3.6 and 6.5.1, and are summarised in Table 6.19.

Characteristic	Term	Document
uncertainty	importance	partial relevance assessment
imprecision	source	
quality		time of assessment
strength		

Table 6.19: Sources of uncertainty that can be incorporated via the uncommitted belief of a mass function

These sources of uncertainty arise from different parts of the retrieval process: indexing the documents, retrieval of documents, RF and how the user assesses documents. In this model, these can be incorporated into a unified framework.

The results from this chapter are similar to the simple summation model presented in Chapters Four and Five. This demonstrates the overall stability of selecting good characteristics of a term, as the selection method is successful when using a different method of manipulating the term characteristic information. However the use of DS, as indicated in

section 6.4.3.2, is not intended simply as an alternative ranking method but as a formal tool for investigating the retrieval and feedback processes in terms of the evidence they use and how the evidence should be handled.

I have shown that the Dempster-Shafer approach can capture many important aspects of this combination, in particular the representation and manipulation of the uncertainty involved in RF. This representation of uncertainty is important to fully understand why some techniques work and others do not, and to provide a framework for future investigation.

Chapter Seven

Summary of combining term use in retrieval and relevance feedback

7.1 Introduction

In this chapter I shall summarise the main findings from my investigation on combining information on how terms are used within documents. I shall discuss four main aspects: selecting characteristics, section 7.2, weighting characteristics, section 7.3, using characteristics to score documents, section 7.4, and the characteristics themselves, section 7.5.

7.2 Selecting characteristics

The major argument presented in this part of the thesis[67], Part II, was that incorporating more information on how terms are used within documents can improve retrieval performance. This was converted into two sets of experiments: combination of evidence and selective combination of evidence.

The combination of evidence experiments combined characteristics of terms (information on term use) and characteristics of documents (information on the content of documents). Each specific combination of characteristics acted as a single retrieval function that was used to retrieve and rank documents.

The general approach of combining characteristics has the potential to improve retrieval effectiveness but it was shown to be difficult to *predict* which specific combinations will be effective for all queries and collections. This means that, although there may be a specific combination of characteristics that is effective for a specific query, selecting one combination to use for all queries[68] is usually not possible.

---

[67] Chapters Three – Six.

[68] i.e. choosing a fixed combination of characteristics that will be used as the default ranking and retrieval algorithm for an IR system.

I have not investigated this fully but I suggest that one method of selecting characteristics for initial queries may be to analyse the *types* of words that are used in the query. Based on a preliminary analysis of which characteristics were chosen to represent query terms in the user data from section 5.8.7, I believe some types of words are better suited to different characteristics. For example a better initial retrieval may be achieved if we used *tf* or *theme* to describe nouns, *context* when describing adjectives or nouns used as adjectives in the query, and any characteristic to describe an infrequent term in the collection.

In the absence of better methods of selecting characteristics for individual retrievals, there are some heuristics to help select good combinations of characteristics. For example, larger combinations are generally better. However combination of evidence remains a technique that gives variable performance.

The principle reason for this variation in performance is that relevance assessments themselves are variable: all relevant documents are not necessarily relevant for the same reasons. In addition, relevant documents for one query may display different attributes than relevant documents for a different query. Combining evidence can help retrieval by providing more ways of retrieving and ranking documents but, often, different combinations are necessary for individual queries. That is, combination of evidence is useful overall but individual combinations may not be useful for all queries.

Also, the reasons why a document may be marked relevant are not dependent on the representation of the document. This was a point made early in Chapter One, section 1.2.1 – users assess document *texts* not the *representation* of the documents. This means that the particular document representations used to retrieve documents may be more or less suitable for detecting the reasons why a document has been marked relevant.

The solution suggested is to *select,* from the set of possible characteristics, those characteristics that indicate relevance and to use only these characteristics in combination. This approach – selective combination of evidence – selects which aspects of a term's use are important for individual query terms. For example, the relevant documents may contain higher than average *theme* values for the term *macbeth*. We can then assume that the *theme* value of *macbeth* in relevant documents was one of the reasons why the document was marked relevant and use this information in a new query.

This is only an assumption as we cannot always assert that users make relevance assessments based on features of individual terms but the overall approach – selecting good characteristics

of terms and documents – proved successful over a wide range of tests: it gave consistent improvements in retrieval effectiveness.

7.3 Weighting characteristics

The characteristics give weights to terms – indexing weights – that represent how well the characteristic is reflected within a document or collection, e.g. high *tf* value reflects a high use of the term within a document.

Treating the characteristics as being of varying importance, i.e. asserting that some characteristics are more important than others, was useful in increasing retrieval effectiveness. These weights can be derived from running sample queries on the collection, and weighting successful characteristics more highly than less successful ones. The weights can also be estimated from the *type* of document considered in the collection. For example, characteristics that are based on within-document information such as *tf* or *theme* are unlikely to perform well on very short documents as these documents tend to have fewer within-document occurrences of terms.

A further reason that individual characteristics may perform at varying levels of effectiveness is that the characteristics themselves reflect aspects of term use that are more or less precise. For example, *tf* reflects occurrences of a term within a document, whereas *theme* reflects occurrences and position of a term. *theme*, therefore, measures an aspect of a term that is more specific. It is probably the case that the more specific characteristics perform less well because they are too specific for some queries. These characteristics are probably better suited to combination with more general characteristics.

Combination of characteristics and terms can also be weighted to reflect how well they discriminate between relevant and non-relevant material. For example the *tf* value of a particular term may be a better indicator of relevance than the *theme* value of the term. Using the discriminatory power of a characteristic of a term gave good performance, especially in combination with the scaling factors.

I listed several sources, in Chapter Six, for uncertainty in the combination process. This uncertainty arises from the fact that term and document characteristics are only *indications* of information use, not exact representations of information use. I investigated two main sources of uncertainty – scaling factors (strength of characteristic) and discriminatory power (quality of characteristic) – demonstrating that incorporating more information on the uncertainty of combination usually gives better results.

7.4 Scoring documents

Once we have selected characteristics we can use them to score, and hence to retrieve and rank, documents for presentation to the user. In Chapters Four and Five I used a simple method of scoring documents which consisted of summing the characteristic score of each query term that appeared in a document. This means that the indexing weights of the selected characteristics (multiplied by scaling factors and discriminatory weights) were simply added together to score the document.

In Chapter Six I presented an alternative method based on Dempster-Shafer's Theory of evidence. This model was intended to provide a more formal model, than the one used in Chapters Four and Five, of managing the uncertainty involved in combination of evidence, In the Dempster-Shafer model the selection and weighting of characteristics gave consistently better results than no selection or no weighting. This demonstrated that the selection and weighting methods are not dependent on a particular document scoring technique. That is, selecting good sources of evidence for relevance, and weighting them appropriately, are important however we retrieve the documents.

7.5 Characteristics

In Part II, I examined two types of characteristics: term characteristics – reflecting aspects of a term's use within documents or collections – and document characteristics – reflecting some aspect of the content of documents.

Although I have concentrated on characteristics that primarily reflect information content, the same approach could be used to reflect aspects of relevance assessments that are not based on content. For example. Barry and Schamber, [Bar94, BS98, Sch91], both list criteria that affect users' relevance assessments on bibliographic data. These criteria include ones such as accessibility (is the document available, is the document free of charge) or currency (is the document recent). Attributes of documents such as these can be used to infer information about why a user has marked a document relevant, and to prioritise the retrieval of documents that display similar attributes. This means that non-content aspects of relevance assessments can be incorporated into searching if we include this information into the representation of the document.

Not all aspects of relevance assessments can be incorporated into searching. For example Barry and Schamber also list criteria such as the validity of the information in a document, e.g. the information contained within the document is correct. It may not be possible to

capture these subjective aspects of making relevance assessments within a document description.

The overall conclusion is that the approach described Part II can widen the representations used in RF, although we may not be able to capture all aspects of *why* relevance assessments were made.

7.6 Summary

In Part II, I demonstrated that selecting and weighting evidence on term use can give significant and consistent increases in retrieval effectiveness. So far, this has only been demonstrated for query terms that form part of the original query. In Chapter Nine I will complete this investigation by assessing how well these techniques perform for terms suggested by the system: the process of *query expansion*.

Prior to this, in Chapter Eight, I shall present an overall model of RF, based on abductive reasoning, which will present the experimental work described so far in a theoretical setting. This model modifies the existing query by adding or removing terms from the query and then selects how each query terms should be used to retrieve and rank a new set of documents. The process of selecting how query terms should be used corresponds to the methods outlined in Part II.

Part III

Abduction

Chapter Eight

Abduction, explanation and relevance feedback

8.1 Introduction

In Part II, I presented a model of selecting those aspects of a term's use that indicated relevant material. This was an attempt to *explain* why a term might indicate relevance: term and document characteristics that discriminate relevant from non-relevant material help explain why a document is relevant. In Part III, I outline a model of RF that is explicitly based on the notion of explanation. The model completes the investigation in Part II by considering which terms should be used to explain relevance documents. This model is based specifically on the theory of *abductive reasoning* or *abductive inference*, [Wir98].

The process of abductive inference, or *abduction*, has been applied to a wide range of tasks including diagnosis, [JJ94b], text understanding, [NM90], word sense disambiguation, [Zad94], and natural language processing, [OR94]. The characteristic feature of abductive systems is that they provide possible reasons, causes or justifications for known events. For example in [JJ94b], Josephson et al. use abduction to detect which antibodies *cause* a particular immune response, Leake, [Lea94], uses abductive approaches to help *understand* anomalous events in news stories, and O'Rorke, [OR94], uses abduction to *interpret* ultrasonic waves in signal detection.

This notion of cause, understanding or interpretation, is often subsumed under the more general notion of *explanation*: abductive inferences drawn from an event are potential explanations of that event. Not all possible explanations of an event are equally likely, equally valid or equally useful. Hence, it is usually an important task of an abductive system to select the *best* explanation of an event from the set of possible explanations. As I shall discuss later, what constitutes the best explanation depends on criteria such as the task for which an explanation is necessary, what evidence supports each explanation and the relative quality of each explanation.

The main tasks of abductive systems are, then, to provide possible explanations of an event and to evaluate these explanations to select the most likely explanation(s). Given an event, or more simply a set of data, *D*, and a possible explanation, *H*, the abductive problem can be represented in the following way, [JJ94b]:

D is a collection of data (facts, observations, givens)
H explains *D* (would, if true, explain *D*)
No other hypothesis can explain *D* as well as *H* does
Therefore, *H* is probably true

Figure 8.1: Abductive process

This simple view encapsulates both functions of an abductive system: *explanation*, (hypothesis *H* explains *D*), and *evaluation*, (No other hypothesis can explain *D* as well as *H* does). This view of abduction is also commonly known as the process of making an *inference to the best explanation*, [Har65].

The process of RF outlined in the previous section - detecting which characteristics of terms and documents best distinguish relevant from irrelevant documents - can be viewed as an abductive process. In this view the term and document characteristics that are more likely to be scored highly in relevant than non-relevant document are good possible explanations of why the relevant documents were assessed as relevant. These explanations were used to modify the existing query in order to improve the retrieval of relevant documents.

In this chapter I propose a broader framework of RF based on abductive principles. One of the main aims of this approach is to incorporate behavioural information, information on how users have made relevance assessments, into the query modification process. This means that RF considers not only what the user has assessed as relevant (the content of the relevant documents) but also how a user has presented their relevance assessments. This will mean dealing with evidence such as the order of relevance assessments, degree of relevance, or number of assessments in a search.

For example, when creating an explanation we should take into account how relevant a document is, where in the ranking it appears, its similarity to other relevant documents and other features of how a user made the assessments. That is, we can gain useful insights into relevance by examining the process of making relevance assessments as well as what is marked relevant.

In Part III I, then, distinguish between *relevant documents* – the representation of the documents the user has marked relevant – and *relevance assessments*. I regard the relevance assessments as including the documents themselves and also information on the assessment such as when the assessment was made, the score given to a document by a user and the number of other documents marked relevant. In an abductive interpretation of RF, I attempt to *explain* the user's relevance assessments rather than simply the relevant documents. There are four main sources of evidence that can be considered: the documents marked relevant at current iteration, the documents marked relevant at the previous iterations (this corresponds to the context of the search), *how* users marked documents relevant (the user's behaviour) and the information both in the collection of documents and the non-relevant set of documents. I am, therefore, attempting to explain the current relevance assessments in the light of context (previous relevance assessments), content (relevant documents) and behaviour.

In the next section, section 8.2, I shall give a brief introduction to abductive reasoning, considering the two main approaches to abduction: logical and non-logical.

Abduction is a widely-used tool but the process of making abductive inferences, as I shall outline, can be difficult for a number of reasons. For example, the data to be explained may be complex, the relations between the data and the causes of the data may be unclear and the process itself may be complex or time-limited. In section 8.3, I shall examine some of the factors of constructing explanations that are important in abduction. In this section I shall also start to outline the components of the RF model.

In section 8.4, I present the problem of RF as an abductive process and introduce some definitions and notation that will be used in section 8.5 in which I present the abductive representation of RF.

Providing an explanation can be a complex process. In section 8.6 I consider the computational complexity of creating explanations. This is important as RF is intended to be an interactive technique. Therefore methods of creating new queries that are too complex will not be suitable for query modification in real-time systems.

I conclude with an overall discussion in section 8.7.

8.2. Approaches to abductive reasoning

Abduction and abductive systems can be divided into two broad groups: logical-based and non-logical approaches. In section 8.2.1, I concentrate on the logical approaches to abduction,

distinguishing the process of abductive reasoning from that of the other classical forms of inference: deduction and induction. In section 8.2.2, I examine statistical and knowledge-based approaches. In section 8.2.3, I analyse the appropriateness of these two alternatives for the research in this thesis.

8.2.1 Logical approaches to abductive reasoning

The major early philosophical work on abduction was due to Peirce, [Pei58, Pei98]. He attempted to distinguish between the three types of logical reasoning - *deduction*, *induction* and *abduction* - using arguments based on syllogisms.

The syllogism in Figure 8.2 is an example of deductive reasoning - a specific instance (*case*) of a general rule (*rule*) leads to a specific conclusion (*result*).

All documents that contain the term `donkey` *are relevant (rule)*
This document contains the term `donkey` *(case)*
Therefore, this document is relevant (result)
Figure 8.2: Deductive syllogism

If the result from Figure 8.2 is exchanged with the rule, as in Figure 8.3, we have an example of inductive generalisation - or induction: a general rule (*rule*) being formed from the combination of specific pieces of evidence (*case* and *result*). The rule that is obtained from induction may or may not be deductively valid: it may not be true for every case.

This document contains the term `donkey` *(case)*
This document is relevant (result)
Therefore, all documents that contain the term `donkey` are relevant (rule)
Figure 8.3: Inductive syllogism

If the result had been exchanged with the case, as in Figure 8.4, we have an example of an abductive syllogism: a general rule (*rule*) and a piece of evidence (*result*) leading to a new piece of evidence (*case*).

All documents that contain the term `donkey` *are relevant (rule)*
This document is relevant (result)
Therefore, this document contains the term `donkey` *(case)*
Figure 8.4: Abductive syllogism

As with induction the conclusion may not be true in every case. In the case above, Figure 8.4, the fact that all terms containing the term `donkey` are relevant does not infer that all relevant documents will contain the term `donkey`. However, the result of abduction can be viewed as providing possible *explanations*; in this example the case statement is a possible explanation of the result statement in light of the general rule (the documents are relevant, *possibly*, because they contain the term `donkey`). However, both induction and abduction are making predictions; they expand our knowledge of the problem.

Peirce later, [Pei31] compared the three different forms of reasoning in terms of the function they play in the role of scientific discovery. He outlined three stages,

i. formulating a hypothesis - this stage corresponds to abduction. If we are seeking an explanation for a discovery, in the RF situation this is a document, *d*, being marked relevant then we would ask what are the possible causes, or explanations, for *d* being relevant. We *abduce* possible explanations for the relevance of *d*. In the previous section of the thesis, the model abduced those characteristics of terms and documents that are possible explanations for relevance.

ii. drawing predictions from the hypothesis. If term *t* is a possible cause of *d*'s relevance, we may ask what other events would we expect to occur as a result of *t*? This is usually modelled by deductive reasoning, we are interested in known conclusions of *t* such as the relevance or non-relevance of other documents containing *t*.

iii. evaluating these predictions. To assess the worth of *t* as a cause of *d*'s relevance, we must evaluate the predictions obtained in step **ii**. This is done by induction. We induce confidence levels for *t* as a possible explanation for the relevance of *d*. In Part II this was modelled by the discriminatory power of the term and document characteristics.

Peirce's later formalisation of abduction emphasises a functional difference between abduction and induction. Abduction infers the *causes* of an event; induction infers the *consequences* of event. The distinction between the two approaches is blurred and opinions vary as to whether induction and abduction should be seen as separate processes and how they are related, [FlKa97]. Some authors see induction as a special case of abduction, others view abduction as a particular type of induction. Although we can abduce rules or theories, in the general case we abduce ground facts of a theory. This is a further difference between induction, in which we generally induce rules rather than facts.

There are also differences of opinion in the current literature as to how abduction itself should be treated. Flach and Kakas, [FlKa97], report on the results of a poll carried out on active researchers in abductive reasoning in which two-thirds of the respondents viewed abduction as inference to the best explanation whilst one-third favoured the Peircean definition of abduction as hypothesis formation. The poll also showed many differences as to the form of hypotheses that are inferred, the utility of these inferred hypotheses, the consequence relations involved and the computational methods used in abductive theory.

A further difference is what underlying phenomenon abduction reflects. Peirce's notion of abduction can be defined using deduction: given a theory *T*, *A* is an abductive explanation of event *C* if the combination of *A* and *T* deductively entails *C*. This definition of abduction assumes that *T* alone does not entail *C,* [Alis96]; we require the (additional) knowledge of A to conclude *C*.[69] The choice of which explanation, which *A*, to use to expand *T* within the logical models has tended to be guided by simplicity criteria, [Pau93]; *T* is expanded by the explanation that forces the minimal change in *T*.

However not all deductive proofs are explanations and not all explanations are deductive proofs, [JJ94b], leading some authors to consider the notion of explanation as one which represents *causality* relationships.

The logical approach has been used previously in IR by e.g. Miyata et al., [MFU99], to select concepts for query expansion. Concepts in this case are sets of terms drawn from a thesaurus. Müller and Thiel, [MT94], use the logical approach to abduction to select which rules should be used to interpret a query in a logical IR system.

8.2.2 Non-logical approaches to abductive reasoning

The alternative approach is to use non-logical methods to derive explanations. These methods do not use the notion of deductive entailment to define explanations but may still use formal structures to derive possible explanations. Charniak and Goldman, for example, [CG91], use Bayesian networks to construct and evaluate explanations for a set of observed actions in plan recognition. Leake, [Lea95], uses a case-based reasoning approach to generate explanations within story understanding and Obradovich et al., [OSG+96], use expert system technology for antibody identification.

[69] *A* alone should not entail *C* either. It is the combination of *A* and *T* that allows us to conclude *C*.

The techniques utilised by non-logical approaches are as diverse as the logical approaches for arriving at a set of explanations. The non-logical approaches also employ a wide range of methods for detecting which explanation is the best one, e.g. [TRG91, Lea94].

8.2.3 Discussion

Both the logical and non-logical approaches to abduction have a core aim: to create explanations for known sets of data. These explanations serve to bridge the gap between what we already know and what we have just observed. The better the explanation is at bridging this gap, the more likely it is to be correct (or useful, depending on the problem).

Some authors, e.g. [Seb83], see abduction as a predictive device: an explanation allows us to make predictions (as in stage **ii.** of Peirce's theory of scientific discovery). Other authors, e.g. [JJ94b], see abduction not as a matter of deductive entailment but one of *causality* in which the purpose of abduction is explain known events not to predict unknown ones. Abductions, then, are not predictions and predictions are not abductions. However, even if we assume that an explanation for an event cannot help predict a further event, the process of abduction itself can help uncover causal relationships that may be used predictively. This is because abduction relies on discovering patterns within data. These patterns then can be used to help predict new events.

My use of abduction follows the inference to best explanation approach[70]. Much writing on this form of explanation creation, e.g. [Lip97], has sought to produce a definitive notion of explanation, or a set of criteria to use for all problems. I seek a more functional definition. My interest is not in one *true* account of what constitutes a best explanation but to develop a model of *types* of explanation. That is I seek to develop individual types of explanation for individual situations in RF. I will discuss this in more detail in section 8.5.3.

In my use of abduction, to model RF, I use the known relevant documents to modify a query, and use this modified query to retrieve a new set of document. The aim here is to facilitate the retrieval of unseen relevant documents. The assumption is that the information in known relevant documents is somehow representative of class of relevant documents. The fact that I am treating the known relevant documents as representative of future relevant documents means that I aim to use the relevance documents to uncover causal patterns within the set of relevant documents. These patterns, in turn, will be used predictively to retrieve more documents.

---

[70]The particular form of abduction I use - inference to the best explanation - treats induction as one type of abduction.

My model of abduction does not use the logical reasoning approach, instead I take a non-logical approach to RF. The main reason for this is to avoid over-formalising my model too soon. Within logical approaches it is necessary to define sets of relations, concepts and rules, such as those outlined in Appendix A, section A.4. These are used to specify how retrieval is performed, and how information is represented. At this stage of research I am not able to formalise the concepts and rules necessary for such a model. For example, I cannot tell which kind of information requires to be modelled, i.e. what factors will affect the choice of appropriate explanation. The non-logical approach allows a more flexible, statistical investigation that can be used as the basis for later logical modelling. That is, the work described in this part of the thesis serves as the investigative framework which is used to uncover what aspects of an abductive account of RF require modelling.

The modified query generated in my model, section 8.5, is created by a process of explanation: I seek to create an explanation for why some documents are relevant and others are not. This will be guided by information on *how* users assess documents. Before I discuss the model, I shall discuss some of the salient features of explanations. This gives a broad outline to the use of explanations.

8.3 Nature of explanations

In this section I look at some important general aspects of explanations that should be considered in an abductive model. For each of these aspects, I shall highlight its relation to RF.

8.3.1 Explanation and cause

Abduction is strongly related to the notion of *cause*. Explanations provide possible causes for observed events. However, the choice of which cause, or causes, are responsible for an event is heavily dependent on contextual factors such as what knowledge is available, the quality of knowledge and what purpose an explanation is intended to fulfil.

For example, in The Comedy of Errors, [Shak88], the slave Dromio of Ephesus tries to explain to his master, Antipholus of Ephesus, why Antipholus's wife is angry

"...
She is so hot because the meat is cold,
The meat is cold because you come not home,
You come not home because you have no stomach,
You have no stomach, having broke your fast;
..."
[Shak88 Act 1 Scene 2 Lines 47 - 50]

Dromio highlights four important aspects of causes in relation to explanations, which I shall discuss below.

8.3.1.1 Not all causes of an event are available for explanation

Dromio's explanation for his mistress's rage is based on the evidence of which he is *aware*. There may be other possible causes for his mistress's rage that he is unaware of, e.g. that Antipholus is perennially late for his meals, that his wife's arthritis is playing up or she has failed, yet again, to win the Ephesian Good Housekeeping Competition. These additional, or alternative, reasons could also explain her anger but Dromio cannot draw on these causes to construct his explanation because they have not been made known to him.

The basis of explaining an event is primarily a matter of selecting likely causes and rejecting unlikely causes but we must be aware that we cannot always operate on all possible evidence for an event, only that evidence which is available. Often evidence may be *implicit*, for example Antipholus's wife may be internationally renowned for her temper and Dromio need not explain to Antipholus the consequences of not returning home immediately.

In section 8.1, I described *H* as the best explanation, if 'no other hypothesis can explain *D* as well as *H* does'. This should be refined to mean no other *available* hypothesis. In constructing an abductive problem, we must ensure that the set of possible hypotheses must be both comprehensive enough for the problem not to be trivial, and broad enough to ensure that the search for possible explanation is not pointless. We must make sure that we have not excluded *genuine* reasons for the event.

In RF I seek to produce explanations for why a user assessed a document as relevant; what caused a relevance assessment. In order to generate explanations we must first decide what are the consitituents of explanations. The elements of explanations could be based on information of different types, for example explanations could be based on background information on the user's experience, system knowledge, domain knowledge, etc. The latter

type of information can be contained within user modelling system, [BCT87], and could help provide explanations such as '*this document is relevant as it is a newly published paper on the user's doctoral topic*'. As shown in systems such as, [BCT87, CGR+92] this approach is potentially very complex and would require supporting with a dialogue system.

Alternatively we could try to incorporate situational or cognitive factors such as the user's task, searching behaviour or searching style, for example to help construct explanations such as '*this document is relevant as it contains a concise overview of the topic*'. However it is doubtful about whether we could infer this kind of information automatically. This is discussed in more detail in section 8.5.3.

Finally, the content of explanations could be based primarily on descriptions of the content of documents. As the observables in RF are relevance assessments the choice of the best explanation is guided by the relevance assessments themselves - how and what the user has marked relevant.

In Part II I used multiple representations of how terms are used within documents and collections – term and document characteristics. Thus explanations were of the type, 'this document is highly relevant as it mentions `donkeys` frequently', 'this document is relevant as it contains both `donkey` and `ass`' or 'this document is relevant as it is the `donkey` is one of the main topics of the document'.

As I did not use a complete set of possible characteristics, the possible components of an explanation were not the complete set of reasons for why a document may have been marked relevant. As discussed in Chapter Seven, section 7.4, it is unlikely whether such a complete set could be developed for such a purpose. This has the result that explanations may omit important reasons for relevance.

8.3.1.2 Explanations are *directed*

Explanations are not always chosen on the basis of what is most likely but often are chosen because they fulfil a purpose. Dromio's explanation, from section 8.3.1, for Antipholus's wife's anger is designed to persuade Antipholus to return home and deflect his wife's wrath from her servants; the explanation is constructed to be personal and convincing to Antipholus.

Although Dromio's mistress may have more than one cause for her anger, or even better causes than the ones given, these additional reasons may not be relevant to Dromio trying to lure Antipholus home. Dromio's explanation for his mistress's anger is, then, one that is designed to be relevant to Antipholus. Dromio's elegant explanation actually fails on this

point, as he has aimed it at his master's twin brother, Antipholus of Syracuse, and so none of his explanation makes sense to his audience.

Legal and political arguments are also often constructed in this fashion, with the intention of providing a particular explanation that not only fits a set of facts but which also supports a particular conclusion. The purpose of an explanation can affect the effort which we put into gathering evidence; a doctor may spend more time and resources in examining a patient believed to be suffering from a severe condition that one who is suffering from a minor complaint.

In certain cases, it may be sufficient simply to provide an explanation of why a set of documents are relevant. However, if we direct explanations to particular features of the relevance assessments we can tailor RF to particular retrieval situations. For example if the precision of a search is poor then we may concentrate on explanations that will increase precision. In other words we may require different types of explanations - each explaining different aspects of the relevance assessments - rather than a single method of creating explanations.

8.3.1.3 Causes may be multiple and connected

Dromio's explanation could have consisted of a single cause, *"She is so hot because you come not home,"* but he provides a stronger foundation for his explanation by asserting a chain of causal events. The initial cause *"She is so hot because the meat is cold,"* on it's own may not be relevant to Antipholus (section 8.3.1.2), so he personalises the argument with an additional explanation, *"The meat is cold because you come not home,".* This, in turn, he backs up with an explanation based on fact, *"You come not home ... having broke your fast;".* Explanations rely on the credibility of the evidence that supports them, as will be discussed in section 8.3.2. In this case, the chaining of events or causes provides a stronger explanation than the individual cause on its own.

Dromio's argument is an example of an explanation based on connected events. We may also have explanations which have multiple pieces of evidence that point to a conclusion, *"It walks like a chicken, it talks like a chicken, so it must be a chicken".*

Explanations may be capable of infinite regression: each element of an explanation itself may need explaining, e.g. what causes *A*, answer *B*, what causes *B*, answer *C*, what causes *C,* etc. This need not trouble us, [Lip97], as some facts are self-explanatory or can be understood without further explanation. Also explanations need not themselves be understood to be

useful, e.g. I do not need to understand the mechanics of my petrol gauge to accept it as an explanation for why my car breaks down even though the tank seems half full.

In a complex situation such as IR, it is unlikely that one single aspect of a document, such as the presence of an indexing term, is sufficient to determine its relevance. Rather it is more likely that a document will have to suffice several criteria before being assessed relevant. Explanations are likely to be composed of more than one component - more than one term or characteristic of a term.

8.3.1.4 Causes may have a temporal nature

In addition to causes of events being unavailable (section 8.3.1.1) the causes of events may not in the form that is required; some evidence will require processing before being suitable to be used in an explanation. For example, if I have a neural network I used to predict share prices and it is performing badly, I may generate a possible cause, such as 'It has learnt to recommend shares of companies that have an odd number of letters in their name'. It would very difficult to test this hypothesis using the internal weights of the network. I would need to convert it into some form that is suitable for analysis.

Some forms of evidence also take time to become apparent. A doctor investigating whether disease X caused her patient's head to swell so alarmingly may require the results of a series of tests before accepting X as an explanation, or *the* explanation, for her finding. Abductive explanation is then often time-limited and the process of providing an explanation may be tempered by the process of gathering evidence and discovering relationships between evidence.

RF, as a process of information-gathering, also has a temporal nature; the more evidence we have on what a user finds relevant and when they consider information relevant hopefully allows us to better estimate what will help retrieve more relevant information. I shall return to this in section 8.5.

8.3.2 Explanation and uncertainty

Evidence, in abduction, as in many forms of inference, is often uncertain. Abductive reasoning, as I use it in this chapter, produces a set of putative explanations, each associated with a plausibility measure which asserts how likely the explanation is to explain the data. It is possible to assert four sources of uncertainty in the abductive process,

8.3.2.1 Uncertainty of the events

The relationships between events such as relevance assessments can be complex and indistinct. An important process in modelling abduction is the reliability and measuring of the data to be explained. We can increase or refine our confidence in the data by gathering new evidence or testing existing evidence by more rigorous methods.

The uncertainty of events can also be affected by temporal factors in two ways. The first is that repetition of events over time can make some events more likely and others less likely. For example, if my doctor's hand slips whilst drawing blood three times in one visit, I may curse and assume he is having a bad day. If this happens on three successive visits, I may refuse to give him the benefit of the doubt and conclude he is incompetent (or drunk).

Secondly, the passage of time can also throw up new pieces of evidence or exclude existing evidence. This is related to the point made in section 8.3.1.4: time can change the evidence available and so change the likely explanations for the evidence.

In Part II, the components of explanations were characteristics of terms. These reflect static information derived from the document indexing process[71]. The uncertainty of the indexing process is reflected in weights attached to the characteristics, representing aspects such as the quality of the algorithm that implements the characteristic. A number of reasons for weighting characteristics was given Chapter Six. As will be shown in section 8.5.4 we also want to weight individual combinations of characteristics and terms to reflect their use in retrieving relevant documents. Uncertainty handling is thus important in abductive reasoning.

8.3.2.2 Uncertainty of the explanation generation process

The plausibility assigned to an explanation is dependent, in part, on the uncertainty of its composite elements. However many factors can lead us to be more of less confident in an individual explanation of a set of data. Factors that may affect this decision include the quality of evidential reasoning, uncertainty handling, or the evaluation carried out. Explanations themselves can be more certain that any of their components, that is explanations can display *emergent* certainty, [JJ94b]. I may, for example, be more confident in the overall theory of query expansion than I am convinced by any individual query expansion experiment.

Once we have selected a set of possible components of an explanation we need to construct a series of explanations. The quality of our explanation construction process will affect our

[71] With exception of the *context* characteristic which, being query dependent, was calculated during a search.

belief in the quality of the explanations as good queries as well as quality of individual explanations themselves. The quality of individual sub-tasks may be important in choosing between explanations, also some types of explanation are easier to build so we can be more confident of how accurate they are as explanations.

This is important for RF as we may need to decide how important it is, in individual retrieval situations, to generate a specific type of explanation. We may, for example, choose to use a simpler type of explanation if we are unsure of what type of explanation is required.

8.3.2.3 Uncertainty of the search for alternative explanations

The first explanation uncovered may not be the best one and, in most cases, we shall need to evaluate a number of explanations. Finding alternative explanations can be a computationally complex activity and the cost of finding alternative explanations must be weighed against practical constraints such as time and processing effort[72]. Our confidence in the degree to which we should accept an explanation will also be affected by how much attention was paid to finding alternative explanations for the same data, [JJ94b].

Although we want to select the best explanation from a series of known possible explanations, it may not be possible to create this set or create the set fast enough for our application. That is it may not be possible to generate *all* possible explanations for a set of data, instead we may have to heuristically select a set of good explanations and concentrate on evaluating or developing better explanations from within this set.

The creation of explanations for RF is limited in that RF is an interactive device. This means that the types of explanation that can be used are limited by the time it takes to create the explanations. This constraint may mean that our explanations are not as effective as they could be if we had more time to generate explanations.

8.3.2.4 Uncertainty regarding the use of an explanation

The purpose that the explanation is supposed to fulfil can also affect the likelihood of an explanation being accepted as correct or likely. Cecily in The Importance of Being Earnest, correctly separates the function of an explanation from the degree of likelihood of the explanation being correct: an explanation may be correct for one purpose but not for another.

[72]See section 8.6 for a discussion on the complexity of abduction.

> Cecily. [To Gwendolen.] *That certainly seems a satisfactory explanation, does it not?*
> Gwendolen. *Yes, dear, if you can believe him.*
> Cecily. *I don't. But that does not affect the wonderful beauty of his answer. [Wil86, Act 111, p301]*

For example, defence and prosecution lawyers will generally present very different explanations of the same set of evidence. What the lawyers themselves *believe* is the correct explanation may not correspond to the explanations they actually provide in court: the explanations serve to test the rigour of the opposing lawyer's explanations.

The uncertainty regarding the use of an explanation arises from three sources:

i. If a purpose for which an explanation is required is poorly specified then we will have poorer guidelines on how to create an explanation. In RF, for example, the less relevance information we have the more difficult it may be to decide what kind of material a user requires.

ii. If the task contains some element of prediction then we also may have more difficulty in giving good measures of plausibility to an explanation. In RF we want to use explanations to decide what kind of documents a user wants to retrieve. This in turn is based on the type of documents the user has already viewed. The assumed relationship between these two types of documents – the ones the user has assessed relevant and the ones the system thinks the user wants – may not hold well. For example the user may change their criteria for relevance during a search. In this case the predictive aspects of RF make explanation creation difficult.

iii. Our evidence for detecting what type of explanation is required may be poor as may be the method we use to detect the appropriate type of explanation required (point **i.**). In RF, we may have very few relevant documents upon which to decide how to modify a query and we may only have very general indications of how to choose a query modification technique.

The last three points are potential sources of uncertainty. The actual values for the uncertainty, and how we measure it, are dependent on the particular modelling approach used.

8.3.3 Explanation and error

As indicated before, section 8.2, abductive inference differs from deductive inference in that deductive inferences convey conclusive evidence: given a set of true premises, deductive

systems will generate true conclusions. Abductive inferences, on the other hand, are fallible because they rely on notions of likelihood and possibility.

If we exhaustively examine all possible explanations and reject all except one explanation (the best possible explanation) then we could represent abduction as a deductive problem, [JJ94b]. However, as described in sections 8.3.1.1 and 8.3.1.2, it is usually the case that abduction cannot consider all possible causes, and the causes themselves are not known with any certainty. Abductive inferences, then, provide *likely* rather than *true* conclusions, and as such are prone to error.

As an example, Banquo, on meeting the witches in Macbeth, attempts to use his previous experience to provide an explanation for his discovery. The hags' physical appearance suggests one possible explanation,

> *"You should be women,"* [Shak90, Act 1, Scene 3, 45]

but he rejects this explanation on an additional, physical, attribute possessed by the witches,

> *"And yet your beards forbid me to interpret*
> *That you are so"* [Shak90, Act 1, Scene 3, 46-8]

Rather than ignoring the potentially contradictory evidence, or reconstituting his beliefs, Banquo rejects the correct explanation - that the witches are real and bearded - and attempts to provide a new explanation for his perceptions.

> "*or have we eaten on the insane root*
> *that takes the reason prisoner*" [Shak90, Act 1, Scene 3, 84-5]

This new explanation justifies the perceptual data - he does see the witches - but allows for physical contradictions - imagined beings do not have to fit with his preconceptions.

This alternative explanation may fit better with his previous experience - certain foods have hallucinogenic properties. It could also, possibly, be justified by factual information such as knowledge of what he has eaten, the possibilities of him being given mind-altering drugs unaware. This explanation may also be preferable; if the witches are the product of a carelessly chosen mushroom or a badly digested piece of cheese, then he can safely ignore the

vision and wait for the effects to wear off; if not he must deal with the potentially unknown consequences of being in the presence of witches.

Although Banquo's second explanation is erroneous it has the advantage that it forces a *minimal* change in his beliefs - it is a conservative explanation. His first explanation may force him to reconsider and alter previously held beliefs as it adds information regarding the supernatural. This can be seen as an example of Lipton's, [Lip97], 'likely' explanations - ones which are most probable - and 'lovely' explanations - ones which, if true, would contribute most to our knowledge or understanding.

This aspect of abduction - the addition of knowledge - is one characteristic feature of abductive inference. Abduction inferences are *ampliative* inferences, abduction can generate information that was not part of the original knowledge, [JJ94b]. It may be the case that we, unlike Banquo, do actively want explanations that inform us more about the problem rather than ones that cost us least effort in accepting them. I shall return to the question of types of explanation in section 8.5.

As abductions are fallible, when constructing an explanation we should consider the pragmatic aspects of generating an explanation such as the cost of generating an incorrect explanation versus the benefits of generating a correct one, [JJ94b]. It may also be worth considering how important it is to generate an explanation or generate a new explanation weighed against the importance of seeking new information before creating an explanation. This process argument becomes important if we have to make implicit information explicit - we must consider whether the benefits of this will outweigh the extra processing involved in generating the new evidence.

In section 8.5.3 I will discuss the fact that there may be many different reasons for why a user performs a specific action. Our task in producing an explanation is to infer the most likely cause. As we are dealing with relatively blunt information our task is error-prone. In particular we may assign wrong reason to action or come up with wrong conclusion or wrong method of handling information.

8.3.4 Explanation and acceptance

If we have a set of explanations for an event, each associated with a score denoting how plausible the explanation, it would be straightforward to assume that the more plausible is an explanation, the greater our confidence should be in accepting it as the *best* explanation. This argument feels intuitive - the more plausible an explanation is the more certain we should be of accepting it. We could further increase our confidence in the relation between the

plausibility of an explanation and our confidence in acceptance of the explanation by asserting that we need only consider explanations whose plausibility is greater than a certain level. For example, we only consider those explanations whose plausibility of being correct is greater than their plausibility of being wrong.

However, Ku's, [Ku91], empirical investigations, reported in [JJ94b], suggest that this relationship between plausibility and acceptance is not as important as the *relative* plausibility of an explanation to the alternative explanations. An explanation whose plausibility is far greater than any of the alternative explanations should be accepted as the best explanation with a greater degree of confidence than one whose plausibility is only marginally better than the alternatives.

The absolute plausibility of an explanation is, of course, important - we should be careful about accepting unlikely explanations but in the general case it is the *relative* plausibility of one explanation over other explanations that should dictate which explanation should be accepted. In addition the relative number of explanations which competed for second best was important in confidence in accepting explanations. Ku's overall findings suggest that the choice of best explanation should be a factor, not primarily of the score of the explanation, but of how well the explanation stands out from the alternatives. We should have more confidence in an explanation that stands out from a small set of alternatives with low scores, than one that is the highest among a high set of highly-scoring alternatives.

In RF, if the plausibility of the best new query is not sufficiently high, or the new query fails some criteria for acceptance, then perhaps it may be better to use the previous query rather than use a new one. This is because we may not be confident enough of the value of any individual explanation as a new explanation and should prefer to use the existing query instead of creating a new query.

8.3.5 Summary

In the previous sections, I outlined some of the features of abductive inference, which I summarise here. Explanations are constructed from sets of causes, which, especially for complex systems, may not be the complete set of possible causes of an event. Even if we assume that the causes of an event are independent, the explanation may consist of many causes and these causes may be connected to provide a coherent explanation. Causes may also be linked to provide a chain of reasoning. The choice of which causes are used in an explanation partly results from the purpose to which the explanation is being put and partly from information on the uncertainty of the causes.

The uncertainty surrounding the causes is one source of uncertainty, other sources are the quality of the explanation generation mechanism and uncertainty about what the explanation is for. These sources of uncertainty mean that abductive inference is uncertain and is prone to error: we can only infer likely, as opposed to true, explanations and we do not have to accept an explanation.

In the next section I shall outline some standard methods of creating explanations. These shall serve to introduce some of the main features of what is important in defining explanations and their relation to RF.

8.4 Process of abduction

In this section I provide some definitions of explanations. These are based on descriptions from a variety of sources reflecting the diversity of abductive approaches to explanation-based systems. I start with a brief working example, section 8.4.1, which is used in the discussion to highlight the main points. In section 8.4.2 I present some criteria for explanations and I conclude with a short discussion of the process of creating explanations in section 8.4.3.

8.4.1 Working example

Consider a small collection, D, containing 10 documents $\{d_1,..., d_{10}\}$, with a set of 20 indexing terms, T, {`baboon`, `bear`, `canary`, `cat`, `chicken`, `cow`, `dog`, `eagle`, `elephant`, `frog`, `giraffe`, `horse`, `lizard`, `monkey`, `parrot`, `pig`, `snake`, `sparrow`, `toad`, `zebra`}. The index terms are indicators of the document's information content and are assigned as shown in Table 8.1. For the purposes of this example I assume that index terms are assigned automatically based on their presence in each document. Therefore the terms {`canary`, `chicken`, `eagle`, `parrot`, `sparrow`} appear in document d_1, terms {`canary`, `chicken`} appear in document d_2 and so on.

This example is based on a representation of documents as a set of weighted terms. The explanations themselves will be sets of terms taken from the set T. This is only for clarity of exposition. The model of abduction presented in this chapter does not depend on a specific document indexing or representation technique.

Document	Indexing terms
d_1	canary, chicken, eagle, parrot, sparrow
d_2	canary, chicken, parrot
d_3	eagle, sparrow
d_4	baboon, monkey
d_5	bear, cat, dog
d_6	cow, elephant, frog, giraffe, horse
d_7	lizard, pig, snake, toad
d_8	zebra
d_9	frog, toad
d_{10}	baboon

Table 8.1: Working example of a document indexing

8.4.2 Notation and definitions

In this section I present a standard definition of what constitutes an explanation and a best explanation relative to the RF problem. The definitions are based on those presented in [JJ94b]. This method of creating explanations is not the only method present in the literature but does form a good basis for presenting important aspects of how to create an explanation.

Definition 8.1: An abduction problem is a tuple $\langle D_{all}, H_{all}, e, pl \rangle$ where

- D_{all} is a finite set of all the data to be explained, in the RF case the documents marked relevant.
- H_{all} is a finite set of the individual hypotheses - the set of all indexing terms.
- e is a map from subsets of H_{all} to subsets of D_{all}. Hypothesis H explains $e(H)$ - for a given set of terms, $e(H)$ defines the relevant documents explained by H. Here, for simplicity, I assume that any term that appears in a document explains that document. For example, if the document d_3 is the only relevant document then $e(\{\texttt{eagle}\}) = \{d_3\}$, $e(\{\texttt{sparrow}\}) = \{d_3\}$, and $e(\{\texttt{eagle, sparrow}\}) = \{d_3\}$. e($\{H\}$), for all other subsets of H_{all}, = $\varnothing$, the empty set.
- pl is a map from subsets of H_{all} to a partially ordered set (H has plausibility $pl(H)$). pl calculates the plausibility of H being an explanation of D. pl may be measured by a probability function, fuzzy value or likelihood function, [JJ94a]. The actual method of creating the plausibility measure is not important, only that pl is

partially ordered. That is we need to be able to compare the *pl* values. If we assume, for example, that *pl* is given by the proportion of relevant documents explained, then *pl*({eagle}) = 1, *pl*({sparrow}) =1, and *pl*({eagle, sparrow}) = 1. *pl*({*H*}), for all other subsets of H_{all} = 0.

An important criterion for explanations is that an explanation should explain all the known data. This is reflected in the *completeness* criterion, Definition 8.2.

Definition 8.2: H is *complete* if $e(H) = D_{all}$. H is complete if it explains all the data in D_{all}

Example: If the relevant set is the set $\{d_{10}\}$ then the set {baboon} is the sole explanation as it is the only indexing term for d_{10}. This means that it is the only term that can explain d_{10} being relevant. If the relevant document set is $\{d_3, d_4\}$ then no indexing term on its own can serve to explain both documents. Possible explanations are {eagle, baboon}, {eagle, monkey}, {sparrow, baboon}, {sparrow, monkey}, {eagle, sparrow, baboon}, {eagle, sparrow, monkey}, {eagle, baboon, monkey}, {sparrow, baboon, monkey} and {eagle, baboon, sparrow, monkey}. All these possible explanations are complete.

A second important criterion is that explanations should contain no unnecessary elements, i.e. an explanation should contain no element that is not necessary to explain the data. This is reflected in the *parsimony* criterion, Definition 8.3.

Definition 8.3: H is *parsimonious* if $\forall_{H' \subset H}(e(H) \subset e(H'))$. H is parsimonious if it contains no superfluous elements, i.e. no proper subset of *H* explains all the data explained by H.

Example: If the relevant document set is $\{d_3, d_4\}$ then the sets {eagle, baboon}, {eagle, monkey}, {sparrow, baboon}, {sparrow, monkey}, are all parsimonious whereas the sets {eagle, sparrow, baboon}, {eagle, sparrow, monkey}, {eagle, baboon, monkey} and {sparrow, baboon, monkey} all contain superfluous elements.

The completeness and parsimony criteria can be combined to give a definition of an explanation, Definition 8.4.

Definition 8.4: H is an explanation if H is complete and parsimonious.

Example: If the relevant document set is $\{d_3, d_4\}$ then the sets {eagle, baboon}, {eagle, monkey}, {sparrow, baboon}, {sparrow, monkey}, are all explanations of the relevant document set as all four sets explain both documents and none contain superfluous elements. In this example, any set containing more than one indexing term from each document contains superfluous elements.

This definition of an abduction system only considers a relatively simple type of problem. For example we do not consider the interrelations between the elements of composite hypotheses, i.e. that fact that components of an explanation may be dependent on each other or may have some type of semantic relationship.

Definition 8.5: H is a *best* explanation if and only if it is an explanation and no other explanation, H', exists such that $pl(H') > pl(H)$. That is, H is only a best explanation if no other explanation can explain the data better than H.

Example: So far I have not assigned plausibility values to either elements or to explanations. If I calculate the plausibility of the elements by inverse document frequency measure (*idf*), [SJ72], for example, as shown in Table 8.2, it is possible to differentiate between components of explanations based on their discriminatory power.

Term	Occurrences	*idf*
baboon	2	1.61
eagle	2	1.61
sparrow	1	2.30
monkey	1	2.30

Table 8.2: *idf* values for elements of explanations of $\{d_3, d_4\}$

If we take the plausibility of an explanation to be the sum of the components of an explanation then the best explanation for the set $\{d_3, d_4\}$ is the set {sparrow, monkey} as this set has the highest overall plausibility as determined by *idf*. This is shown in Table 8.3.

H	Plausibility
{eagle, baboon}	1.61 + 1.61 = 3.22

{eagle, monkey}	1.61 + 2.30 = 3.91
{sparrow, baboon}	1.61 + 1.61 = 3.22
{sparrow, monkey}	2.30 + 2.30 = 4.60

Table 8.3: Calculation of plausibility of explanations

A best explanation, H, is defined as one which is complete, parsimonious and explains the data with the highest degree of plausibility. This definition ensures that no alternative explanation has a higher plausibility than H but does not ensure that there is a *unique* best explanation. H is therefore *a* best explanation but not necessarily *the* best explanation.

The parsimony criterion outlined in Definition 8.3 only considers one form of parsimony. Alternative definitions for parsimony were investigated by Tuhrim et al., [TRG91], who examined four[73] criteria for determining the most plausible explanation based on the notion of parsimony. Each of these definitions will create different explanations on what kind of queries can be created by an abductive RF algorithm. In this section I shall describe these types of explanations.

i. **minimal cardinality.** Under this definition, H is an explanation if and only H explains all the data and has the smallest number of elements amongst the possible explanations. This parsimony criterion is a form of Occam's Razor[74] and serves as a general guideline that the more simple an explanation, the more likely it is to be correct. Several applications have used this criterion to select between explanations of equal plausibility but different size.

For example, if we have two explanations, say {zebra, toad} and {frog}, with equal plausibility, then we should select the explanation {frog} as the shortest explanation. However in many situations the combination of two hypotheses may be more plausible than the simple sum (see section 8.3.2.2 - emergent uncertainty). For example, if the set $\{d_1, d_3\}$ is the set of relevant documents, then the sets {eagle} or {sparrow} are both potential explanations but the set {eagle, sparrow} is not an explanation. This is because the set {eagle, sparrow} contains elements that are not necessary to explain $\{d_1, d_3\}$. This may

[73] I ignore the two further definitions suggested, namely *single order* explanations - which can only consist of a single element - and *collapsed covers* - which are designed for problems with a spatial element.

[74] '*one should not increase, beyond what is necessary, the number of entities required to explain anything*', also known as the *principle of parsimony*, [Occ01].

be counterintuitive for IR since adding more good terms to a query may give better performance than only adding a minimal subset of terms.

ii. **irredundancy.** H is an explanation if and only if H is no longer complete if any element is removed. This criterion is less strict than minimal cardinality as it only considers the *coverage* of the data, not the comparative length of the explanation against other explanations. This definition of parsimony also does not allow the comparison of explanations with equal plausibility.

iii. **relevancy.** H is an explanation if and only every h in H explains a d in D_{all}. In other words, every element of an explanation must explain at least one element of data and the explanation as a whole must explain all the data. This is a loose version of parsimony as it does not consider the length or plausibility of an explanation. It also allows more than one component to explain the same d. It is still, however, a definition of parsimony as an explanation would not be parsimonious if it contained elements that did not explain an item of data.

iv. **most probable cover.** If we can attach a causal strength to each h – each component of our hypothesis H - and each d to represent how likely h is to explain d and a prior probability to each h to indicate how likely it is to occur then we can calculate $P(D|H)$. For RF this means that we can assess the probability that a set of indexing terms, H, will explain a set of relevant documents D. The probability function should be constructed in such a way that $P(D|H)$ is greater than 0 if and only if the set of indexing terms H explains all the relevant documents.

An explanation H is a best explanation if and only if $P(H \mid D) \geq P(H' \mid D)$ for any other possible explanation, H', of D. In RF this type of explanation would allow us to analyse the query as a whole, i.e. compare how each possible modified query performs as an explanation, rather than as the set of component parts.

Tuhrim et al., [TRG91], evaluated each of these types of parsimony criteria within a real-world problem. They took the problem of diagnosing possible explanations for a series of brain disorders and generated sets of explanations using the definitions given above. Human experts were then asked to assess the quality of the explanations produced by each method. The explanations were classified as being either an exact match to the expert's diagnosis, a close match to the expert's diagnosis, a partial match or an explanation that disagreed with the expert's explanation of the cause of the patient's disorder.

Overall the irredundancy method gave the most number of exact/close matches however it also produced a large number of possible explanations. That is it produced lots of possible explanations, some of which were very good. The minimal cardinality and most probable cover methods gave fewer good matches but produced a relatively small number of explanations. This investigation showed that not only do different definitions of what constitutes an explanation give different explanations but that the different definitions can also produce different numbers of explanations. This has computational implications if we try to generate all explanations before selecting the best explanation, see section 8.6.

An alternative approach, one which will be followed in this thesis, is to split the problem of creating an explanation into a number of sub-tasks, [JJ94b]. The important reason for this is that we can avoid generating all possible explanations and concentrate first on eliminating components that may be poor. This means that we can provide different methods to solve particular parts of a problem, as will be demonstrated in the following section on the model of RF. In the next section I introduce the model for RF based on abductive principles.

8.5 Abductive model of RF

In this section I outline a model for RF based on a process of abductive explanation. Section 8.5.1 describes the types of inference that are incorporated into the model, sections 8.5.2 outlines the various sub-tasks involved in creating explanations, and sections 8.5.3 – 8.5.6 describe the inference stages to obtain a list of possible components of explanations. Section 8.5.7 introduces the construction of explanations and the selection of the best explanation. I conclude in section 8.5.8.

To discuss the model I assume that explanations are composed of sets of characteristics of terms and documents. This assumption is solely for the purpose of outlining the model; the components of the model can be any representation of documents or retrievable objects. However, before discussing the model, I would like to make an important distinction in terminology.

The distinction is between is between the explanatory power of a component and how a component explains the data. The notion of explanatory power defines which are good terms to explain the documents. This corresponds to the notion of retrospective RF, Chapter One; providing a description of the known relevant documents. The notion of how we should use the terms to retrieve documents (how a component explains the data) corresponds to the predictive aspect of RF: using the explanation to retrieve more relevant documents. This

distinction is necessary because explanations are usually generated for a purpose. In RF, for example, we generate explanations to retrieve new documents. This means that we want to separate the process of selecting the components of an explanation, the terms themselves, from how we use the terms, selecting the characteristics of each term.

In the following discussion I shall refer back to this distinction where appropriate.

8.5.1 Types of inference

The goal is to obtain a set of characteristics of terms - an explanation - that can be used as a query[75]. The basic process of choosing an explanation is one of inference and this will correspond to a series of inference stages. The inferences are of two types:

i. inference within an iteration of feedback. This inference is primarily one of *content* in which we try to decide which term and document characteristics best distinguish the relevant documents from the irrelevant documents at the current search stage, independent of any other evidence.

ii. inference across iterations. This class of inference is one of change and brings in situation aspects of the search. In this inference we are looking at the current search stage in the context of the search as whole, in particular how the search is changing. This type of inference should incorporate some element of prediction of the search.

These two stages are often not handled consistently within RF models. Term reweighting approaches, e.g. the probabilistic model described in Appendix A, calculate relevance weights based on all the relevance information. All relevant documents are aggregated into a single set and term weights are recalculated at each iteration of feedback. New relevant documents and old relevant documents are, then, treated in the same way. Query expansion techniques, such as Rocchio, Appendix A, often have a cumulative effect: once a term has been added to a query it will not be removed unless its new weight – the one calculated by the term reweighting algorithm - becomes zero. New relevant documents, in this case, only serve to *modify* previous decisions. This can mean the terms that are currently poor query terms remain in the query, albeit with lower weights.

I explicitly separate these two stages of inference as this separation allows a distinction between new relevance information and previous relevance information. This distinction can

---

[75] I use characteristics of terms and documents as the basic components of explanations. However, terms themselves or any indexing unit can be used as components of explanations.

be used to investigate the relative utility of these two groups of relevance information in predicting what *should* be retrieved. I shall explain this in more detail in section 8.5.2.

Within each inference there are two sets of factors that may affect the choice of the best explanation:

i. system factors. These are the factors that derive from algorithmic properties of both term and document characteristics and the retrieval function used. These factors will include many of the factors outlined in Part II such as the quality of the characteristics.

ii. user factors. These are the factors that derive from how users search and how they assess documents. For example this set of factors will include aspects such as the use of non-binary relevance assessments (Chapter Five), the number of documents a user has assessed relevant and the order in which the user has assessed documents.

In the next section I outline the basic inference steps that compose the model of RF.

8.5.2 Abductive process

This model of explanation falls into six tasks. Each task contributes to the overall process of choosing a best explanation either by organising the data (selecting possible components of explanations, or ordering these components) or guiding the reasoning process (selecting the type of explanation required, constructing the explanation, selecting the best explanation).

A distinction can be made between *creating* explanatory hypotheses or explanations and *evaluating* the quality of each explanation, [JJ94b]. For the purposes of this work I shall not divide this process. One reason for this is that explanation can be complex entities, composed of many elements. A strategy that generates all possible explanations before evaluation of explanations may be too computationally expensive to be tractable (see section 8.6). In addition, a strategy that incorporates the evaluation of components of explanations *within* the hypothesis creation stage can reduce the number of possible explanations to be considered, [JJ94b].

I shall briefly introduce the tasks in this section to give an outline for a fuller discussion of each tasks in sections 8.5.3 – 8.5.8.

i. inference of explanation type. In this task I exploit the user's behaviour and information on the content of the relevant documents to infer what *kind* of explanation or

query is required at the current search stage. This task decides what is to be explained. This stage aims at to determining what features of the relevance assessments require explanation.

At each iteration of feedback some aspects of the relevance assessments may require explanation, other will not. This inference step examines both the overall search and the current iteration to estimate what explanations are required. This will be discussed in section 8.5.3.

ii. inference of the relevant document set. This stage takes the documents that have been marked relevant at the current iteration, summarised information on previous iterations, information on the process of making relevance assessments (such as the range of assessments, number of assessments, order of assessments) and selects which documents we should try to explain. The point of this inference, which is unusual for RF techniques, is that if we have new evidence on what constitutes currently relevant material then we may want to revise previous decisions. This will be discussed in 8.5.4.

iii. inference of possible components of explanation. This takes the set of terms and returns the set of terms that could form part of an explanation. This will be discussed in section 8.5.5.

iv. inference of *good* components of explanation. This stage takes the output from stage **iii.** (set of terms) and returns the terms with weights on the potential quality of each term providing a given type of explanation. This will be discussed in section 8.5.6.

v. building explanations. This stage constructs explanations according to the definitions outlined in section 8.4.2. I shall discuss this stage in section 8.5.7.

vi. selecting good explanations. This final stage selects and compares good explanations based on the plausibility of their component elements and the type of explanation required (point **i.** above) and returns the optimal explanation. This stage will be discussed in section 8.5.8.

The process is to infer what we want the query to achieve (what type of explanation), infer the relevant document set, from this set infer possible and then good components of explanations and then compose a number of explanations. From this set of explanations we choose one explanation to use as the best explanation.

Once we have created the best explanation we can then decide how each element of the explanation explains the relevant assessments. That is, once we have (retrospectively) created

a good explanation of the known relevant documents we have to decide how to use the explanation to retrieve a new set of documents. In this work this translates into selecting good term characteristics of each term in the explanation.

8.5.3 Inference of query type

Most statistical approaches to query modification, [RSJ76, Roc71], define a retrieval function that is used for all queries and all iterations of relevance feedback. Although these functions are applied to different sets of documents (different sets of relevant and irrelevant documents at each feedback iteration), most parameters (number of expansion terms, relative weighting of new and existing query terms, etc.) are identical for all iterations of feedback. Following these approaches, a query will be modified by the same mechanism at all iterations of feedback. This mechanism will typically be one that has been shown to give good average performance on a set of test collections.

However, if we view the process of modifying a query as one of *supporting* a user search, we should recognise that different searches, or different stages of a search, may require different query modification techniques. For example, if a user is moving from a browsing stage of a search - a stage where they are investigating general information on a topic - to a stage where they are looking for more specific information then it may be appropriate to change the query in different ways than if the user is moving from a specific to a general search. Depending on the type of search, we may want to vary the number or type of query terms added, the method of ranking the terms, and the degree to which we alter the existing query. One potentially powerful source of evidence for how to modify the query is the relevance assessments themselves.

The relevance assessments given by users are not only indications of what they find relevant but also of their decision-making process. For example Spink et al, [SGB98], note that the use of partial, or non-binary, relevance assessments correlate with stages of uncertainty as to search focus: the more partial relevance assessments, the more unfocused the search. Similarly, Florance and Marchionini, [FM95], demonstrate that the order in which users make assessments within an iteration can serve as good indicators of which documents are more central to the current search.

Information such as this has been used by several authors, e.g. [Kuh91, Kuh93, Ell89, ECH93] to show that discrete stages in searching can be detected and categorised. These stages often correspond either to a *task* (e.g. gathering information, checking for new information or to a *process* (e.g. orienting oneself in a database, focusing an information

need). The user's interaction with the IR system can, then, serve to distinguish one task or process from other alternatives.

In my query modification approach I could, therefore, try to infer what type of search state the user is involved in and modify the query to best support this search stage. For example a user who is trying to obtain an overview of a topic may be better served by a query that retrieves documents that contain different aspects of the topic. A user who has a very focused information need would require only documents relevant to particular aspects of the topic.

There are, however, a number of objections or difficulties with this approach. The first difficulty is that of inferring which evidence points to what conclusion. Although the classifications of information-seeking behaviour are based on user's interactions with an IR system, they require a certain amount of human interpretation and human-human interaction. In other words the classifications of stages and tasks within a search are not based *solely* on the interaction.

The point here is that for many aspects of making relevance assessments we are unable to automatically detect the cause of why assessments were made in a particular way. Partial assessments, for example, may be the result of a vague information need but they may also arise due to a poor retrieval session or a lack of highly relevant documents in the collection being searched. This means that we cannot assert, with any certainty, that a particular behaviour has a definite cause.

A related difficulty is that it is not clear what our actions should be - what kind of query modification we should attempt - even if we could identify search stages. Assume that we are able to identify a search stage in which a user has a vague information need. Our goal at this stage may be to help the user focus their information need, i.e. to move further on in the search process. Equally the goal may be to develop a query that will continue to retrieve documents similar to the ones the user has already got, i.e. support what type of search stage the user is involved in and allow the user to decide when it is appropriate to focus their search.[76]

[76]We could, of course, use an abductive system of reasoning to guess what is the cause of a particular set of actions but this would not help us decide how the system should react. An alternative action is to ask the user why they perform certain actions but it is doubtful whether the user is able, or willing, to make such reflective decisions.

However, as I have discussed above, although it is difficult to guess what the user intends and how to support the user, their searching behaviour can give useful indications of what is *important* about their search.

An alternative option is not to exploit the user's actions to decide upon what search stage we should base our query modification but to identify what is important about the relevance assessments. Here, I use the process of making relevance assessments to decide what features the system should attempt to explain, looking for important features in the relevance assessments and guiding query modification to these features. This detection of important features comes, in turn, from the *change* in relevance assessments over successive iterations of feedback. For example a drop in the number of relevance assessments, an increase in the number of partial relevance assessments, or a change in the similarity of relevant documents could provide valuable insights into how the search is changing. These indications of search change can be used to indicate how the query should be modified.

Different behavioural changes in the relevance assessments will lead to different query modifications. I use the change in behavioural evidence from the user to guide what evidence we use and how much of each evidence we use. For example if the consistency of the relevant document set increases (inter-document similarity) then we could infer that this increase in consistency is a reflection of a more focused need or a better retrieval session and target retrieval of documents that form a consistent set. Similarly if the use of partial relevance assessments over binary assessments decreases then we should concentrate on the retrieval of highly relevant documents.

Each of these possible methods on changing a query corresponds to different methods of explanation and the task in this inference is to decide which explanation is required. In Chapter Nine I show, experimentally, that different types of explanation give different retrieval results and, in Chapter Ten, I show that these different types of explanation can be used to detect which type of query modification is more appropriate for individual retrieval situations.

8.5.4 Inference of relevant document set

For any given iteration of RF, the first thing we have to consider is which documents we want to use as the relevant set of documents. These documents will be used to modify the query. This is not a question usually asked in RF - normally all the documents that the user has marked as relevant are used for query modification. However *how* the user assesses relevance may mean that we only want to consider some of the documents they marked as being relevant.

For example, we may choose only to use the documents that have a high relevance score or relevant documents that are very similar to each other. We may also take into account the order in which assessments were made. Users often deploy strategies when marking documents relevant, for example some users will simply go down a list starting from the first document and assessing or at least considering in some way each document until they have found enough information or until they stop searching the list. Other users act in a less ordered fashion, [FM95] - finding a good document and then relating the information in the other documents to this one. So order may be important in finding what the user thinks is a good document.

What is important here is that we are selecting what evidence is to be used to form explanations. To do this we may want to infer *how* to use the evidence. To do this we need to infer a change in the style of searching over time (*user factor*, *across iteration*).

Factors that affect choice of relevant documents can be used on an iteration-to-iteration basis to direct the choice of which documents are the best to use. Thus we can choose at each iteration how many of the relevant documents we want to consider: - all the relevant ones, only the highly relevant ones, the most consistent ones, or the ones that we feel may have been more central to the user's relevance assessments.

For example, if the assessments become more partial or the consistency changes then we may want to try a broader query than the one previously used. A broad search probably means we need to consider as many relevant documents as possible. If the number of partial relevance assessments lowers during the search or the number of high relevant assessments increases then this could correspond to a search that is becoming narrower may be better suited to only considering very relevant documents or only the documents most recently marked relevant. In other words, we could use the information on the current search state to automatically refine our previous decision on what we should have considered relevant at previous iterations: refining our previous decisions in the light of new information. This notion of selecting which documents we should concentrate on is back by experimental evidence, e.g. [SW99, Vak00a, Vak00b] which shows that searchers use different criteria for assessing relevance at different stages in their search. In other words, selecting which documents to explain is an attempt to select those documents that reflect the user's current criteria for relevance.

This type of inference gives us the basis for what we are explaining - which documents we are trying to explain. I shall present experimental evidence for this in Chapter Ten, section

10.2.3, where I show that better performance can be achieved by selecting which relevant documents are used for feedback.

8.5.5 Inference of components of explanations

Once we have decided which documents are to be used for feedback we should decide what are the possible components of explanations: which terms can explain the relevant documents. Potentially any set of terms can provide an explanation for the relevant set of documents. However we can cut down this search space in a number of ways.

The first way to cut down this search space is to assert that only terms that appear in a document can explain the document. This is a broad cut-off - a term that is not in a relevant document is not a good indicator of relevance at the current search stage. This inference is an example of an inference across iterations and is motivated by system factors - we cut down the number of possible components to help the computational properties of explanation generation.

The result of the previous two steps is a set of possible components of an explanation, each of which is an indexing term. We should now consider how important each of these are in an explanation. At present I have only identified which hypotheses, or terms, should be considered. I have not specified which are *good* hypotheses. This I do in section 8.5.6.

8.5.6 Inference of good components of explanations

The result of the previous inference stage is a set of terms that have some explanatory power in describing why the relevant documents are relevant. We can cut down this set further by considering the coverage and discrimination of the hypotheses. We are then performing an inference of which are *good* components of an explanation: those that explain more of the relevant documents and those that separate the relevant from the non-relevant documents. This reduction in the possible components is achieved in several stages.

The first stage is an inference across iteration and is based on system factors. In attempting to explain relevance assessments we must take into account how many of the relevant documents a term explains. A good term should explain as many of the relevant documents as possible, but it should also discriminate well between relevant and irrelevant documents so we next remove all the terms that are more likely to be present in irrelevant documents than relevant ones. We are then inferring which are good components of an explanation based on their discriminatory power.

This can be extended to take into account the temporal nature of a search, based on Campbell's, [CVR96], notion of ostensive relevance: the relevance weight of a term is a product of its discriminatory power over time. It can also be extended to incorporate partial relevance scores.

This allows us to eliminate all terms that are poor discriminators of relevance over time and allows us to order the remaining ones. As will be shown in Chapter Nine there are other ways of ordering terms for query expansion, each of which can be used to estimate the explanatory power of a term. For example we can order terms by their discriminatory power, by how much data they explain, or how likely they are to appear in documents. Each method of estimating the explanatory power of a term corresponds to a particular definition of what type of explanation is required. As mentioned in the introduction to section 8.5 this notion of explanatory power only considers how good a term is at explaining the relevance assessments, it is not used to decide how a term explains a document.

8.5.7 Composing explanations

The result of the previous step is a method of weighting terms according to their explanatory power in explaining the relevant documents. These components of explanations can be then combined to build potential explanations, each of which is a possible new query.

This set of terms may serve as an explanation on its own but it may be possible to derive a better explanation by only considering a subset of the terms. That is we may only require some of the components from the set of good components to explain the data. How we select the best explanation from this set depends on what kind of explanation we require.

In section 8.5.3 I argued that the choice of which type of explanation we want should be dictated by what we want the explanation to achieve: the effect we want the explanation to have on the search. This will allow us to select between good explanations (those that achieve what we want) and bad explanations (those that change the search in an inappropriate manner). By ordering the components of explanations, section 8.5.6, we can assume that we are dealing with the right kind of terms. For example, if we want an explanation that will broaden a search then we should order the terms according to how likely they are to broaden rather than narrow a search. This step allows us to concentrate on the terms that are likely to achieve what we want from an explanation.

However even though we are concentrating on the terms that are good for a particular type of explanation, some combinations of terms will form better explanations than others combinations. Hence we have to consider which combination is the best one; which is the

best explanation depends on how we define best. There are various ways we could define best and some of the criteria for selecting best explanation are, [JJ94b]:

i. *simplicity*. A better explanation will probably be a simpler one. Usually a small set of terms with good explanatory power is better than a larger set of terms with the same explanatory power.

ii. *plausibility.* A better explanation will be one that most plausibly explains the data. So far we have not discussed how we obtain plausibility of explanations but in part this will depend of the plausibility of the individual components of the explanation - their explanatory power.

iii. s*elf-consistency.* A good explanation will be one that is self-consistent. A poor explanation may be one that explains all the data, but which comprises a set of mutually exclusive sub-explanations, i.e. parts of the explanation explain some of the data, and other parts explain other parts, but there is no overlap.

iv. *consistent with background knowledge*. We should prefer an explanation that fits with what we already know about the retrieval situation. Although we may require radical changes to the query, if the choice is between two explanations, one that insists on a radical change and one a conservative change, it is probable that the conservative one is preferable.

v. *quality.* In this case, best is a question of explaining better; the quality of the explanation is more important than the number of documents explained. For example we may be able to explain all the relevance assessments but only by creating a large explanation or an explanation that contains unlikely components. In this case it may be better to eliminate some of the relevance assessments and concentrate on creating an explanation with better overall plausibility but which only explains part of the data.

vi. *quantitative*. In this case a better explanation explains more of the relevant documents, regardless of the plausibility of the explanation.

In practice all these issues are important but which is more important very much depends on what kind of explanation is required. There is also a trade-off between the method of creating explanations and the ability to guarantee the selection of the best explanation. For example we could create all possible explanations and iteratively test each explanation to see which

has the best overall explanatory power. However this method is impractical for real-time solutions. An alternative method is to heuristically select a good initial explanation and test the robustness of this explanation by adding or removing components. One method of doing this is to rank all components of the explanation and create the first explanation possible. Then, by adding new components or removing existing ones we can see to what degree the explanatory power of the explanation changes – testing how likely this explanation is to be the best one. As will be shown in section 8.6, the use of heuristics is often necessary to guide the system towards a good explanation. In Chapter Eleven I outline various formal techniques that can be used to select the best explanation.

8.5.8 Summary

The overall strategy for creating explanations is one of multiple inferences regarding what constitutes a good explanation for a current retrieval situation. A primary feature of this approach is the incorporation of more behavioural aspects of relevance feedback.

The inferences fall into several stages, each of which is guided by factors reflecting how retrieval systems work and how users assess relevance. There are four main inference stages:

i. *inference of query modification required.* This inference examines the search as a whole comparing the relevance assessments made at the current iteration against those made in previous iterations. The intention of this inference is to estimate what kind of query modification, what kind of explanation, is required. This inference is primarily governed by the user behaviour. That is, what the user marks relevant and how the user is assessing relevance.

ii. *inference of relevant document set*. The decision on which type of explanation is required also allows the inference of what documents are to be explained. In this inference the choice of what *kind* of explanation is to be generated allows a better estimate of *which* relevance assessments are to be explained. For example, if the user gives relevant documents higher scores in the current iteration than in previous iterations we could use this information to eliminate documents from previous iterations. This stage is primarily directed by the user behaviour and operates across iterations of feedback (selecting documents from earlier stages in the search) and the current iteration (selecting relevant documents from the current search iteration).

iii. *inference of components of explanations*. Once the system has decided what assessments are to be explained it can start to assemble the components of the

explanation. This set of components will be those that are capable of explaining the relevant assessments.

iv. *inference of good components of explanations.* This inference stage selects those components that are good at explaining the relevant documents.

Once these inferences have been made the choice of good explanations and best explanations can be implemented. The formation of good explanations concentrates on those components that have the best explanatory power, and the choice of best explanation typically will be guided by the principles outlined above. This means that we *tend* to want small, highly plausible explanations that explain all the data with the minimal change to the existing query. However this is only a tendency and sometimes a bigger explanation may be more plausible than a short explanation and sometimes it may be better to only explain the most important, rather than all, the data.

In the next section I discuss the complexity of producing explanations. This section is necessary as it shows that in most problems, we have to rely on some kind of heuristic reasoning to guide the process of creating explanations.

8.6 Complexity of abduction

Abductive inference is a theory of justification, based on inferring explanations for observed events. The judgement of how good a hypothesis, *H*, is as an explanation depends on a number of aspects, [JJ94b]: how accurately we have collected our data, section 8.3.1, how much effort we have expended on evaluating alternative explanations, section 8.3.3, how plausible *H* is an explanation and how much better *H* is as an explanation than the alternatives, section 8.3.4. As abductive inferences, unlike deductive inferences, can be wrong we need to balance the need to find an explanation against the effort of creating alternative explanations or finding more evidence.

In domains that have few elements, it may be possible to perform an exhaustive search for explanations. In domains such as IR the search space may become quite large. In IR we can reduce the search space by, for example, only considering a subset of possible components of explanations, but we cannot guarantee that the space will be small enough to ensure that an exhaustive search will be tractable, hence it is necessary to consider the *complexity* of abductive processes.

In [BAT+94], Bylander et al. consider the computational complexity of generating abductive explanations that are composed of individual elements such as indexing terms. In general, finding the most probable composite hypothesis is intractable, [Coop90]. However, Bylander et al. demonstrate that different *classes* of abductive problems are either polynomial[77] (*tractable*) or NP-hard[78] (*intractable*) depending on the complexity of calculating the uncertainty of the hypotheses and how much data is explained by the hypotheses. Bylander et al. argue that the computational complexity of the abduction task is not dependent on the representation or the method of reasoning but on the constraints on the explanatory process, and the ordering amongst hypotheses dictated by the plausibility measure. That is, certain types of abduction problems are hard irrespective of the reasoning methods that are applied to the problem. This is important because RF techniques are *interactive* techniques: solutions that are too computationally complex are unlikely to be appropriate for RF.

In the rest of this section I shall analyse my use of abduction based on the discussion in [BAT+94]. Their investigation is based on *'finding the most plausible composite hypothesis that explains all the data*' and is analogous to my use of abduction which is based on finding the most best set of indexing terms to use in a new query.

I shall discuss first the complexity of finding explanations in section 8.6.1, then in section 8.6.2 I shall discuss the complexity of finding the best explanation. In section 8.6.3 I shall discuss the complexity of my approach.

8.6.1 Complexity of finding explanations

In this section I analyse the complexity of different types of abduction problem based on the complexity of finding explanations, I do not consider the *plausibility* of explanations, i.e. I am

---

[77] A polynomial solution is one whose time complexity function is $0(n^k)$ for some k >=0. If problem is solvable in polynomial time then an algorithm can usually be found where k is relatively small, e.g. less than 5, [RS86]. This means that a solution is possible for this problem that can usually operate in an efficient time-scale. It does not, however, guarantee that such a solution is easy to find. Nor does it guarantee that a solution will be fast enough for the requirements of the user.

[78] An intractable solution is defined as one for which no polynomial solution exists but is solvable. That is a solution is possible but the time taken to give an answer may be exponential to the size of the number of components used to form the solution. NP (non-deterministic polynomial) solutions are solvable in polynomial time only by the use of heuristics or a non-deterministic algorithm, [RS86]. A problem is said to be NP-hard if an algorithm for solving it can be used to solve all other NP problems. A problem which is both NP and NP-hard is called an NP-complete Problem.

mainly concerned with finding any explanations not the best explanations. The plausibility of explanations will be dealt with in section 8.6.2.

8.6.1.1 Independent abduction problems

In the most simple abduction problems an explanation explains a datum if at least one of its component hypotheses explains the datum, regardless of what other hypotheses the explanation contains. In this situation we assume that the elements of an explanation do not interact and explanatory power is equal to set coverage, [BAT+94]. Formally an abduction problem is *independent* if it is the case that if at least one element of an explanation explains the datum then the complete explanation explains the datum, Equation 8.1. For RF this means that if term *t* explains a document *d* then any explanation containing *t* explains *d.*

$$\forall H \subseteq H_{all} (e(H) = \cup_{h \in H} e(h))$$

Equation 8.1: Independent abduction problem

If we are seeking the best explanation for an independent abduction problem, one method is to generate all possible explanations and to test the plausibility of each. However there may be an exponential number of explanations to be considered and so determining the *number* of explanations for an independent abduction problem is as hard as determining the number of solutions to an NP-complete problem, [BAT+94].

> **Theorem 8.1:** For the class of independent abduction problems, it is NP-complete to determine the number of explanations.

Therefore finding the number of possible explanations to this type of problem, regardless of how the plausibility function is measured, is intractable. However to find *an* explanation we do not need to consider all the possible explanations. For example if *h* is the most plausible component, and *h* explains all the data then *h* can be held to be the best explanation. This is only applicable if one individual hypothesis explains all the data, and the smaller a set is, the higher its overall plausibility. In most cases this ideal situation will not arise. For example, in RF although one term may explain all the relevant documents (very common terms may appear in all the relevant documents), when we take into account the plausibility of the individual terms we may find that composite explanations have a higher overall plausibility.

If we consider explanations with more than one component, it is easy to check whether *an* explanation exists (if the set of all possible hypotheses is not an explanation then no explanation exists). We can then test each individual hypothesis and generate a composite

working hypothesis; if adding a hypothesis to the working hypothesis increases the explanatory power then we retain the hypothesis else the hypothesis is removed. This creates a minimal explanation with maximum overall plausibility. If all the data is not explained by the explanation then no *full* explanation exists, and we can only achieve a partial explanation for problems of this type.

For independent abduction problems, it is tractable to find *an* explanation.

> **Theorem 8.2:** For the class of independent abduction problems, there is an $O(nC_e + n^2)$ algorithm for finding an explanation, if one exists. $n = |D_{all}| + |H_{all}|$. C_e is the complexity of calculating e

8.6.1.2 Monotonic abduction problems

The data explained by an explanation of an independent abductive problem is equivalent to the union of the data explained by each individual hypothesis. In the class of *monotonic* abduction problems, the data explained by a composite explanation may be greater than that explained by the individual hypotheses. This can arise from the fact that, together, hypotheses may interact to explain data that neither could explain separately. For example if I included phrases as the components of explanations, then the presence of the term *information* in a document or the presence of the term *retrieval* may not explain the document but the presence of the phrase *information retrieval* could explain the relevance of the document.

An abduction problem is *monotonic* if and only if for all explanations, H, any proper subset of H explains less data than H, Equation 8.2.

$$\forall H, H' \subseteq H_{all}(H \subseteq H' \rightarrow e(H) \subseteq e(H'))$$

Equation 8.2: Monotonic abduction problem

A composite explanation does not explain any less data than its individual hypotheses and may explain more data. All independent abduction problems are monotonic but not all monotonic problems are independent, [BAT+94]. This is because independent explanation insists that at least one hypothesis explains each datum.

As the independent abduction problems are included in the set of monotonic problems it is also intractable to determine the number of explanations for this class of problem. Bylander et al. also demonstrate that is it hard to enumerate a polynomial number of explanations.

Theorem 8.3: For the class of monotonic abduction problems, given a set of explanations, it is NP-complete to determine whether an additional explanation exists.

However the complexity of finding *an* explanation is as for the independent problems.

Theorem 8.4: For the class of monotonic abduction problems, there is an $0(nC_e + n^2)$ algorithm for finding an explanation, if one exists. $n = |D_{all}| + |H_{all}|$, C_e is the complexity of calculating *e*

Therefore it is tractable to find *a* solution, but intractable to find all explanations to a monotonic abduction problem.

8.6.1.3 Incompatibility abduction problems

So far we have assumed that any set of components of an explanation is possible. The class of *incompatibility* abduction problems refers to problems where elements can be mutually exclusive. That is some components of an explanation cannot jointly explain an event as the two events cannot occur together. For example if we try to explain why John and Mary's car crashed we may form explanations of the form '*John was driving and fell asleep*' or '*Mary was driving and was drunk*' but we could not form explanations that assert that both John and Mary were driving the car. In RF this type of situation is only problematic if we asserted that the fact that a term does not appear in a document means that the term could not represent the content of the document. For example, if the term `monkey` does not appear in a document then the document is not about monkeys. This assumption is not one we would wish to make in IR.

8.6.1.4 Cancellation abduction problems

The set of *cancellation* abduction problems refers to the class of problems in which one element may cancel out data explained by another. For example in a diagnostic situation, one disease may explain an increased body temperature and another disease explain an increased body temperature but the two diseases in combination would result in a normal body temperature.

In our model of RF this situation does not apply. A term or characteristic of a term simply explains a set of documents. If we combine terms with other terms in an explanation then the combination of terms does not explain less documents than either term individually: the addition of new information does not affect the explanatory coverage (in terms of set coverage) of a term. In my model explanatory power is cumulative not subtractive.

8.6.1.5 Summary

The framework for RF I have presented, based on terms, is an example of an independent abduction problem. However by widening the representations used to form explanations to include composite indexing elements such as phrases, the framework more properly is an example of a monotonic abduction problem. As shown in Table 8.4 this means that it is tractable to find an explanation and intractable to find all explanations. The complexity of finding a best explanation will be discussed in the next section.

	Condition to achieve		
Class of problems	Finding an explanation	Finding all explanation	Finding a best explanation
independent	P	NP	?
monotonic	P	NP	?

Table 8.4: Time complexity of generating explanations
P = known polynomial algorithm, NP = NP-hard. Adapted from [BAT+94]

8.6.2 Complexity of plausibility of finding a best explanation

In order to discuss the complexity of finding a best explanation I need to define how to compare the plausibilities of explanations. For the purposes of this discussion the plausibility criterion is based on comparing the plausibility of individual hypotheses in explanations. The overall plausibility of an explanation is therefore a function of the plausibility of its components.

8.6.2.1 Best-small plausibility criterion

It would be natural to assume that smaller explanations are preferable to larger ones and that more plausible individual explanations are preferable to less plausible ones, i.e. to assume that small and highly plausible explanations are better than large, less plausible ones. In RF this means that a small set of highly plausible terms is preferable as a query than a large set of less plausible terms.

However it is intractable to find best explanations using this best-small approach. A individual hypotheses may have different plausibilities we may reach the situation where a larger explanation has more plausible elements and a higher overall plausibility than a smaller explanation. Therefore we cannot distinguish between large, plausible explanations and small, implausible explanations, In addition, depending on the definition of parsimony we are using,

it may often be possible to increase the plausibility of an explanation simply by adding an extra element. Therefore it is not possible to order explanations based solely on the best-small plausibility criterion. We need additional information on how to order the explanations relative to each other, [BAT+94]

> **Theorem 8.5:** For the class of independent abduction problems using the best small plausibility criterion, it is NP-hard to find a best explanation.

8.6.2.2 Ordered abduction problem

If the plausibility of *all* the individual hypotheses are different and if their plausibilities can be totally ordered, i.e. all plausibility values are unique, then finding a best explanation using best-small is tractable.

An abduction problem is *ordered* if, given any two hypothesis, we can say which hypothesis has the greater value.

$$\forall h, h' \in H_{all} (h \neq h' \rightarrow \left(pl(h) < pl\left(h'\right) \vee pl(h) > pl\left(h'\right)\right)$$

Equation 8.3: Ordered abduction problem

It is tractable to find the best explanation for this kind of problem.

> **Theorem 8.6:** For the class of ordered monotonic abduction problems using the best-small plausibility criterion, there is an $0(nC_e + nC_{pl} + n^2)$ algorithm for finding a best explanation. $n = |D_{all}| + |H_{all}|$, C_e is the complexity of calculating *e*. C_{pl} is the complexity of calculating *pl*

Although it is tractable to find *a* best explanation for this kind of problem, it is difficult to determine whether it is *the* best explanation, without enumerating and testing all possible explanations.

> **Theorem 8.7:** For the class of ordered independent abduction problems using the best-small plausibility criterion, given a best explanation, it is NP-complete to determine whether there is another best explanation.

	Condition to achieve

Class of problems	Finding a best explanation	Finding more than one explanation
Ordered independent/ monotonic	P	NP
Unordered independent/ monotonic	NP	NP

Table 8.5: Complexity using best-small criterion based on plausibility of components
P = known polynomial algorithm, NP = NP-hard. adapted from [BAT+94]

8.6.3 Summary

The abductive problem, so far, is an independent problem, so it is possible to derive a tractable solution to find an explanation or to tell if *any* explanation exists. However, as I have based the model of explanation on the values of terms, I cannot assert that this problem is ordered: the values of terms do not allow the total ordering of all terms. However, this discussion has centred around the theoretical complexity of the problem, the practical nature may make this finding unimportant. For example, although in theory we have a large number of potential explanations, as discussed in section 8.5, most terms are usually ruled out before composite explanations are considered. It is also likely the use of heuristics can reduce the need to consider all our options. A practical approach to the problem may only require us to explain some of the data, so it may be the case that we only require partial explanations. That is, we may only require an explanation that is good enough. The point here is that we *can* theoretically determine how to select an explanation and, with appropriate definitions of plausibility and a definition of what constitutes the best explanation, we *can* select the best explanation(s). However in real systems we will often need to use heuristics to actually calculate explanations. I will demonstrate methods of doing this in the next chapter.

In the next section I shall complete this chapter with a short discussion.

8.7 Summary

In this chapter I proposed a framework for relevance feedback based on abductive inference. This model incorporates information on how user's make relevance assessments and uses a notion of explanation to generate modified queries.

The use of abductive reasoning here is a variant of Van Rijsbergen's, [VR86], proposal that relevance can be modelled as a process of *uncertain inference*. Van Rijsbergen's model asserts that the relevance of a document to a query can be measured by the probability that the information in a document infers the information in a query, Figure 8.5.

$$P(d \rightarrow q)$$

Figure 8.5: Relevance measured as uncertain inference

Inference is a particularly suitable process for IR as the information we have in a retrieval situation is usually underdetermined, [SJ99]. For example, queries do not usually specify exactly which documents will be relevant and the representations of documents do not adequately capture the user's reasons for relevance.

Also as Lipton [Lip97] points out, *'If inference is inductive, by definition it is underdetermined by the evidence and the rules of deduction'*. Often in a retrieval situation we cannot make clear deductions from evidence, we have to make educated guesses. If we expand our evidence from simply the content of the relevant documents to include how the users present their relevance assessments, I argue that better guesses can be made about what kind of RF is required for individual searches.

Van Rijsbergen's approach was encapsulated in the logical uncertainty principle, [VR86]:

> "Given any two sentences x and y; a measure of the uncertainty of $y \rightarrow x$ related to a given data set is determined by the minimal extent to which we have to add information to the data set, to establish the truth of $y \rightarrow x$."

In our case we are interested in the plausibility[79] that the information we have on relevance, the relevance assessments, *R*, infers a modified query, *q'*, where *q'* is an abductive explanation of *R,* equation 2.

$$Pl(R \rightarrow q')$$

Figure 8.6: Relevance measured as uncertain inference

[79]I do not, yet, specify which theory of uncertainty plausibility refers to. Plausibility should be treated as a general likelihood measure.

Our abductive situation starts with q and we want to reach q', we then have to abduce enough information to be able to reach q'. Van Rijsbergen's definition promotes a conservative approach to transformation (*minimal extent*), but as Banquo demonstrated, section 8.3.3, we may not always infer a minimal change to the previous query, q. Sometimes we may want a more radical change to the previous query to provide a better estimate of R[80]. As will be shown in the following chapters this is because more radical changes can be more appropriate to individual retrieval situations. This is because, although we may be able to explain the relevance assessments using a short explanation it may not be the preferred type of explanation for the retrieval situation. We may, instead, require a type of explanation that gives a bigger query modification.

In the following two chapters, Chapter Nine and Chapter Ten, I present an experimental investigation into different methods of creating explanations and their applicability to RF.

[80]Van Rijsbergen's approach was designed to provide a match between document and query rather than retrieval situation and query but the question of whether we want a minimal transformation holds.

Chapter Nine

Experiments on explanations

9.1 Introduction

In the previous chapter I outlined a general framework of RF based on abductive principles. In this and the following chapter I present an experimental investigation of some aspects of the framework. These experiments are carried out on test collections as test collections allow a large number of experiments to be run. However, the use of test collections means that certain aspects of the framework presented in Chapter Eight, e.g. the use of partial relevance assessments, could not be investigated as the test collections do not contain this information. The test collections also do not provide any notion of the development of an information need. I shall present a separate investigation on these aspects in Chapter Twelve where I discuss a separate, user-oriented, evaluation of the framework.

The fundamental argument outlined in Chapter Eight was that different retrieval situations should be supported by different RF techniques. This is to say that some RF techniques are more appropriate for particular types of query modification. For example, some RF techniques are better at improving precision than others. Furthermore, it was argued that it is possible to *select,* from the user's interaction, which RF technique(s) should be used at individual RF iterations. In this chapter, and in Chapter Ten, I experimentally investigate this proposal. I do this in a number of ways. In this chapter, I examine different criteria on what constitutes an explanation, i.e. how components of explanations should be ordered and what parsimony criterion should be used to select components of an explanation. Each definition of what constitutes an explanation should be created corresponds to a different method of reformulating a query based on relevance information. In Chapter Ten, I investigate factors that can be used to determine why individual query reformulation techniques work well on some queries and less well on others. Finally, I examine whether it is possible to automatically select an appropriate RF technique based on the user's interaction. This will also be discussed in Chapter Ten.

In the remainder of this introduction I shall discuss the relation between abductive and standard methods of query reformulation.

RF techniques, e.g. [Roc71, RSJ76, Har92c], aim to provide more effective queries based on a user's assessment of a set of retrieved documents. As discussed in Chapter One, RF methods typically concentrate on identifying good *indicators* of relevance: usually those terms that are good at discriminating documents that the user has assessed as containing relevant material. These terms can be given higher weights (*term reweighting*), e.g. [RSJ76], or be used as the basis for a new query (*query reformulation*), e.g. [Roc71].

The assumption behind RF approaches is that the more similar a document is to the relevant documents, then the more likely this document is to be relevant. RF techniques decide what features should be used in making this similarity comparison (query reformulation) and how important are each of these features (term reweighting). RF is then a process of detecting important features in the set of relevant documents. This detection of features is the basis behind the abductive interpretation of RF: select important features (components of explanations) and decide in what way the components explain the data.

Many techniques have been suggested for the selection and weighting of important terms in documents, [Har92c]. The performance of these technique in batch test collection evaluations, e.g. [SB90], and interactive evaluations, e.g. [FB00, KB96], have generally proved their utility in improving retrieval effectiveness. However, experimental evidence, e.g. [MVR97], has shown that the increase in retrieval effectiveness using these techniques is *variable*: some queries have increased effectiveness, whereas other queries have reduced effectiveness.

One of the possible reasons for this is that the same techniques are applied to all queries and many of the variables used in RF are held constant for different collections and queries. For example the same term reweighting function will be used to assess the importance of each term, and the same number of terms will often be used to reformulate each query. This is essentially a pragmatic decision, as the values of these variables will have been shown to give good performance over a range of conditions.

The abductive methods I suggest in this chapter, however, do not rely on fixed parameters such as these. An explanation is based on how many terms are required to explain the relevant documents, and the reweighting schemes (term and document characteristics) are used selectively for individual query terms. The experiments reported in this chapter demonstrate that an abductive interpretation of RF can give better and more consistent increases in retrieval effectiveness.

The overall research goal in this chapter is to investigate the applicability of abductive methods for RF in an experimental setting.

In section 9.2 I outline the abductive query reformulation techniques, each of which are based on a definition of what constitutes an explanation of a set of data. In section 9.3 I summarise the abductive term reweighting techniques. These techniques have already been described in detail in Part II. In sections 9.4 and 9.5 I outline the experimental methodology and the main findings from the experiments.

9.2 Explanations

In this section I describe the abductive query reformulation techniques used in my experiments: these techniques are responsible for the *content* of the modified query. I define an explanation as a set of terms that distinguish one set of documents (the relevant ones) from another set (the non-relevant ones). The explanation is a set of features that identify why the documents may be relevant. In these experiments the set of documents to be explained consists of the set of known relevant documents – the relevant documents used for feedback. I shall discuss the inference of the relevant document set in Chapter Ten.

Several definitions of what constitutes an explanation can be found in the literature, e.g. [JJ94b, TRG91]. Here I investigate four methods: Josephson, Minimal Cardinality, Relevancy and Coverage. These are based on definitions that have proved successful in other domains that rely on characterising a set of data. In sections 9.2.1 – 9.2.4 I describe these explanation types and how I implemented them in the experiments.

9.2.1 Josephson explanation

In [JJ94b], Josephson et al. proposed a method of creating an explanation that is based on a ranking of the possible components of explanations by their *explanatory* power. This type of explanation asserts that good explanations will contain elements that are good discriminators of the data.

To create an explanation, possible components of an explanation are ranked in decreasing order of their explanatory power. Starting at the top of the ranking of elements, each element is analysed in turn to see if explains any of the data. If the component does explain a datum it is added to a working explanation. If the component does not explain a datum, or only explains a datum that has already been explained, it is ignored. In this manner, an explanation is built up by adding the most likely components of an explanation to a working explanation.

This is a simple method of creating explanations that can be transferred to IR: isolate all those terms that have a positive explanatory power – these are the set of possible feedback terms. Then, rank all possible feedback terms and keep adding feedback terms to a working query until at least one term which appears in each relevant document has been added to the query. An example of this is shown in Figure 9.1.

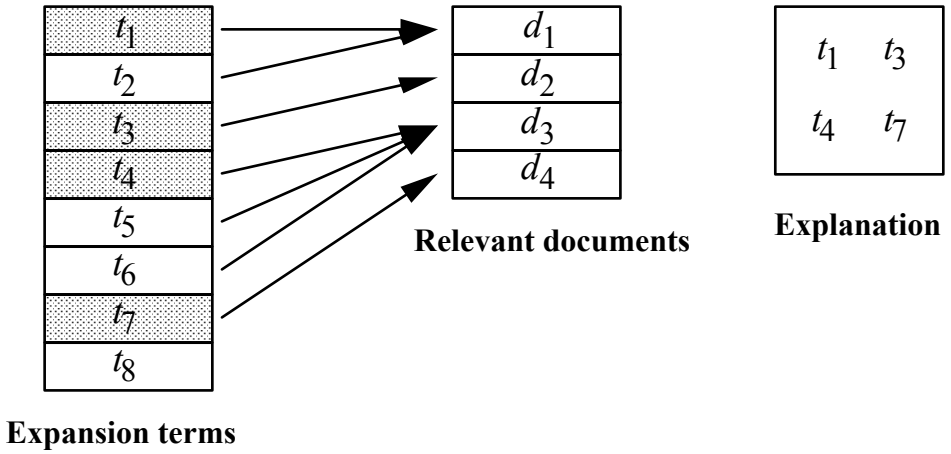

Figure 9.1: Josephson explanation

In Figure 9.1, term t_1 explains the first document, d_1 - it is contained within document d_1 and will retrieve the document. This term is added to the working explanation. Term t_2 only explains document d_1 which has already been explained, so term t_2 is not included in the explanation. t_3 explains d_2 and term t_4 explains d_3 and, as neither d_2 nor d_3 have been explained yet, both t_3 and t_4 are added to the explanation. Terms t_5 and t_6 do not explain documents that have not been already explained and are ignored. Finally, term t_7 explains the last relevant document, d_4, and is added to the explanation. The final explanation, in Figure 9.1, is an explanation according to the definition given in Chapter Eight, section 9.5. It is complete – it explains all the relevant documents – and it is parsimonious – it contains no superfluous elements.

This method of creating an explanation depends on a ranking of terms by explanatory power. For this type of explanation I use the F_4 reweighting scheme as a method of assessing the explanatory power of a term. The F_4 measure, [RSJ76], is a well-established scheme for assessing the discriminatory power of a term, section 1.2.2.3.

The F_4 weights produce a partial ordering of terms, i.e. they do not give unique values to terms. This means that although we can produce an explanation, we cannot assert that it is the single best explanation. Other explanations are possible, e.g. in this example the set of terms

$\{t_2, t_3, t_5, t_7\}$ also corresponds to an explanation. However if we assert that the explanatory power of the explanation is equal to the sum of the explanatory power of its components, we can assert that there is no shorter explanation with a higher explanatory power[81]. An explanation provided by the Josephson method is *a* best explanation but it may not be *the* best explanation.

The Josephson method of creating an explanation is similar to standard RF query reformulation techniques: adding a number of good discriminatory terms to the query. The major difference is that a *variable* number of terms are added to the explanation: only sufficient terms are used to explain the relevant documents. A further difference between this method and standard RF methods is that a non-consecutive set of terms is added to the query. In standard RF methods the top *n* consecutive terms would be added to the query.

9.2.2 Minimal cardinality explanation

An alternative method of creating an explanation is one that accords with the *minimal cardinality* criterion: a set of terms is an explanation if it explains all the data and has the shortest length amongst possible explanations, [TRG91]. The minimal cardinality type of explanation asserts that shorter explanations are better than longer ones. This is based on the hypothesis that short explanations are more believable than longer, more complex, explanations.

One method of creating short explanations is to base the explanation on those terms that are most likely to occur – terms that are more likely to appear in the unseen relevant documents.

We can create short explanations by selecting terms at the bottom of the F_4 ranking of feedback terms. These are terms that have low, but positive, discriminatory power but which appear in a large number of documents compared with those at the top of the ranking.

In Table 9.1 I show the average *idf* values for the query terms in the collections I used in my experiments (the collections are described in section 9.5), along with the average *idf* values of the top and bottom 10 feedback terms given by the F_4 ranking. As can be seen the terms at the top of the ranking appear in fewer documents – have a higher *idf* – than those at the bottom of the ranking or those chosen by the user (the original query terms).

[81] This means that higher F_4 weights correspond to terms with higher explanatory (discriminatory) power.

Collection	Original query terms	Top 10 feedback terms	Bottom 10 feedback terms
AP	34.2	49.9	11.6
SJM	34.1	49.2	13.0
WSJ	33.8	49.9	11.0

Table 9.1: average *idf* values for query and feedback terms

The terms chosen for this type of explanation are relatively poor at discriminating the known relevant documents from the rest of the collection. However, they do avoid the problem observed in some query reformulation methods, namely adding terms that are too specific to the relevant documents, e.g. terms that only appear in the known relevant documents. The terms chosen by this method are more general than those chosen by the Josephson method.

The same basic approach for creating explanations is followed for this type of explanation as for the Josephson type. Each feedback term is tested to see if explains an unexplained relevant document; if it does it is added to the working query, if it does not then the term is ignored and the next term is considered. The difference is that terms are added from the bottom, rather than the top, of ranking of expansion terms.

9.2.3 Relevancy explanation

A third type of explanation is the *relevancy* type, [TRG91]: a set of elements is an explanation of a set of data, if and only if each element explains at least one item of the data. This definition is therefore relatively loose and places no criteria on the characteristics of the explanation, such as length or explanatory power.

In an IR situation, any combination of terms that explains the set of known relevant documents will serve as a Relevancy explanation. Our method of creating an explanation of this kind is to regard the set of all feedback terms as an explanation, that is, all terms with a positive F_4 weight. The explanation created by the Josephson and Minimal Cardinality approach will also be explanations according to this definition of an explanation, however Relevancy explanations will be much longer.

9.2.4 Coverage explanation

One of the core criterion for explanations found in the literature is *coverage*, [TRG91]: a good explanation should explain as much of the data as possible. Therefore the components of an

explanation should explain, individually, as many of the relevant documents as possible. To test this type of explanation I implemented a form of coverage explanation which differed from the other explanations in that the expansion terms were ordered by how many relevant documents they appeared in, rather than F_4 weight.

Terms that appeared in most relevant documents were placed at the top of the expansion term ranking and those that appeared in least relevant documents were placed at the bottom of the term ranking. Terms that appeared in an equal number of relevant documents were sorted in decreasing order of F_4 weight. The creation of an explanation followed the same pattern as before: test each term to see if explains any unexplained data; if it does add the term to the current explanation; if it does not explain any additional data then ignore it.

9.2.5 Summary

The four methods of query reformulation differ in what they prioritise – Josephson explanations prioritise explanatory power, Minimal Cardinality explanation prioritise length, Coverage explanations emphasise the amount of data each component explains and the Relevancy explanation simply requires that all data is explained.

The four explanation types are somewhat related. For example, the Relevancy explanations are supersets of the other types of explanations: for an individual query all Coverage, Josephson and Minimal Cardinality explanations are subsets of the Relevancy explanation. The Minimal Cardinality and Coverage explanations will both tend to produce short explanations but will use different terms to compose explanations. How the performance of these explanations differ will indicate how important explanatory power is in creating good explanations.

9.3 Scoring Explanations

Once we have a modified query, we have to decide how terms should be used to score documents. In this section I describe the two methods of scoring the documents I investigated: weights derived from feedback (*relevance feedback weights*), section 9.3.1, and weights assigned at *indexing* time *(term and document characteristics*), section 9.3.2.

The research question I explore here is whether the abductive approach to selecting evidence (section 9.3.2) is better than relevance feedback weights based on a standard term reweighting scheme (section 9.3.1).

9.3.1 Relevance feedback weights

Relevance feedback weights are a standard method of assigning a weight to a term based on relevance information. The *same* function is typically used to score each term and a document score is given by the sum of the feedback weights of the query terms contained within the document. In these experiments I use the F_4 weighting function to calculate relevance feedback weights.

9.3.2 Term characteristics

In Part II proposed a technique of selecting which aspects of a term's use – term and document characteristics - indicated relevance. This is an attempt to abductively select why a term may indicate relevant material.

This approach *adapts* the method of scoring documents according to the relevance assessments: a query term's contribution to a document score is based on a variable set of characteristics. This method of reweighting terms and scoring documents is an example of abductive principles in that I select which aspects of a term's use indicate good explanatory aspects of a term's relevance.

The experiments reported in Part II concentrated only on reweighting the original query terms; no query reformulation methods were used. In this chapter I aim to complete this overall study by assessing how well the techniques perform under query reformulation, and the interaction between the reweighting and reformulation approaches.

Specifically I test the three main methods of weighting terms: indexing weights, scaling factors and discriminatory power of a characteristic of a term. To summarise:

i. *characteristics with no additional evidence*. In this method I use the index weights given by the term characteristics to score documents. The retrieval score of a document is given by the sum of the characteristic scores of each query term, i.e. sum of *idf* scores of each query term plus sum of *tf* scores of each query term, etc. Documents are given a score by the document characteristics, *specificity* and *information-noise*.

ii. *characteristics with evidence as to quality of characteristics*. In Part II I showed that incorporating information about the quality of the term characteristics could improve retrieval effectiveness. This is achieved by scaling the term and document characteristics weights using a set of scaling factors that are derived experimentally, Chapter Four. The retrieval score of a document is the same as for **i.** except that each index score is multiplied by the

corresponding scaling factor. The scaling factors used are: *idf* 1, *tf* 0.75, *theme* 0.15, *context* 0.5, *noise* 0.1, *specificity* and *information_noise* 0.1 This condition will be known as the weighting (**W**) condition, whereas case **i.** will be know as the non-weighting (**NW**) condition.

iii. *selection of characteristics and feedback evidence*. One of the most important conclusions from Part II was that, in RF, it is possible to select for each query term a set of characteristics that best indicate relevance. That is we can choose from analysing the relevant documents, which characteristics should be used for each query term to score the remaining documents. This technique is tested on both the weighting (**W**) and non-weighting (**NW**) conditions. The analysis of relevant documents can also be used to assign discriminatory scores to each query term characteristic selected for the new query. The discriminatory power is the average score of the combination of characteristic and query term, e.g. *tf* value of query term 1, in the relevant documents divided by the average in the non-relevant documents. The retrieval score for a document is the same as for **ii.** except that each index score is also multiplied by the discriminatory power of the characteristic and only selected characteristics for each term are used to calculate the retrieval score.

The three methods of weighting terms and documents incorporate principles of abductive reasoning, each of which uses different information. Scoring method **i.** uses indexing weights only to indicate how good a term is (its explanatory power). Scoring method **ii.** uses indexing weights combined with information on the quality of the source of the weights. Scoring method **iii.** uses the same information as **ii** combined with information about the discriminatory power of the characteristics. Method **iii.** also selects only those characteristics that have good explanatory power.

9.4 Experimental methodology

In this section I present the general experimental methodology. In sections 9.4.1 I outline two variations on the query expansion experiment and in section 9.4.2 I present the baseline comparison measures. The experimental procedure is as follows:

For each query,

i. all documents were ranked by the sum of the *idf*, *tf*, *theme*, *noise* characteristics of all query terms, and the *specificity* and *information_noise* characteristics of all documents.

ii. the relevant documents in the top 100 ranked documents were used to create a list of possible query expansion terms. These are the terms in the relevant documents that have a F_4 score greater than zero. The F_4 score gives a measure of how well a term discriminates the known relevant set from the remainder of the document collection. Terms are ranked in decreasing order of the F_4 score with higher scores indicating higher discriminatory power of a term[82].

iii. the query is reformulated. The method by which the query is modified differentiates the query reformulation experiments. Four explanation types, described in section 9.2, and two baseline methods, described in sections 9.4.3.1 and 9.4.3.2, are investigated.

iv. the modified query is used to score the remaining documents in the collection. The method of scoring the documents differentiates the term reweighting investigation and was discussed in section 9.3.2.

v. the new document ranking is evaluated using a freezing evaluation, [CCR71].

Steps **ii.** – **iv.** are repeated for four iterations of feedback, giving five document rankings for each query. The change in average precision between the initial document ranking and the ranking given after four iterations of feedback is used to assess the effectiveness of the query modification technique.

Each test was run on three collections: Associated Press (**AP** 1998), San Jose Mercury News (**SJM** 1991), and Wall Street Journal (**WSJ** 1990-1992), details of which are given in Table 9.2.

[82] For the coverage method of explanation, the terms were ranked according to the method described in section 9.2.4.

	AP	SJM	WSJ
Number of documents	79 919	90 257	74 520
Number of queries used[83]	48	46	45
Average document length[84]	284	163	326
Average words per query[85]	3.04	3.64	3.04
Average relevant documents per query	34.83	55.63	23.64
Number of unique terms in the collection	129 240	147 719	123 852

Table 9.2: Details of AP, SJM and WSJ collections

9.4.1 Query reformulation – query expansion and query replacement

All the RF techniques I am investigating select a number of terms – the *feedback terms* – to use in a new query. After selecting the feedback terms, they can either be added to the current query (*query expansion*) or used in place of the current query (*query replacement*).

Query replacement is motivated by the argument that if the set of feedback terms does not contain the original query terms, then the original query terms must be poorer at explaining the relevant documents than the terms chosen for the new query. Therefore we should exclude the original query terms from the new query as they are poorer at describing relevance than the feedback terms.

Query expansion is motivated by the argument that, even if query terms are not contained within the set of feedback terms, query terms still provide a valuable source of evidence as to what constitutes relevance because they have been chosen by the user. Salton and Buckley, [SB90], and Haines and Croft, [HC93], both showed experimentally that keeping the original query terms as part of the new query was useful in RF.

An important aspect of abduction is deciding what evidence is used to form explanations: query replacement explains only the relevance assessments, whereas query expansion explains all the relevance information – the relevance assessments and the original query. I shall present the results on this in section 9.5.1.

---

[83] These are queries with at least one relevant document in the collection.

[84] After the application of stemming and stopword removal.

[85] This row shows the average length of the queries that were used in the experiments.

9.4.2 Baseline measures

I compare the performance of the query reformulation methods against two baselines: expansion by the top *n* feedback terms (section 9.4.2.1), and expansion by a variable number of terms (section 9.4.2.2). I introduce a third baseline measure aimed specifically at testing the reweighting method (section 9.4.2.3).

9.4.2.1 Baseline 1

The first baseline comparison technique is a standard RF approach [MVR97]. This adds, to the query, the top *n* feedback terms from the top of the list of possible expansion terms. The F_4 weights of the query terms are used to score documents.

For each collection (and condition **NW** and **W**) I chose the value of *n* (where *n* varied between 1 and 20 expansion terms) that gave the best average precision. This optimum value gave a stricter baseline comparison for our experiments as I am using an optimum value for *n*. I only investigated the range 1..20 as this has previously been shown to be a useful range for setting *n,* [Har92b, MVR97]. This range is also important for another reason. These experiments are intended to simulate real user searches. In real searches it would be preferable to allow the user to modify the result of any query modification. Adding too many terms to the query (too high a value for *n*) then the query would be difficult for the use to modify. A low value of *n* is more suitable for comparison with the explanation methods.

The values of *n* for each collection and condition are shown in Table 9.3.

	AP (NW)	**AP W)**	**SJM (NW)**	**SJM (W)**	**WSJ (NW)**	**WSJ (W)**
n	18	20	20	18	20	20

Table 9.3: Optimum values for *n* in the range 1..20 expansion terms

The decision to use query expansion rather than query replacement for this baseline was made retrospectively as query expansion gave better results than query replacement.

9.4.3.2 Baseline 2

The Coverage, Josephson and Minimal Cardinality query reformulation methods (section 9.2) differ from the standard model of query expansion in two ways. First, they add a *variable* number of feedback terms to each query and iteration. Second, they do not add a consecutive set of terms from the top of the list of possible expansion terms: terms are drawn from

throughout the list of expansion terms. The second baseline is designed to test which of these two factors cause any change in retrieval effectiveness between the Baseline 1 measure and the explanation methods.

The Baseline 2 method adds a variable number of terms to the query. For this baseline I add one feedback term per relevant document to the query.

The difference between Baseline 2 and Baseline 1 is that Baseline 2 adds a variable number of terms to the query whereas Baseline 1 adds a fixed number. The difference between Baseline 2 and the Josephson method is that Josephson adds enough terms to explain the relevant documents whereas Baseline2 adds a number of terms relative to the number of relevant documents.

9.4.3.3 Baseline 3

The third baseline is aimed specifically at testing the selection method described in section 9.3.2, **iii.** In Part II I showed that this method performs well but did not test how well it performs when we use query terms that have been selected by the system rather than the user.

The third baseline, then, performs the same selection as described in section 9.3.2 but only performs this on the characteristics of the original query terms: no query terms are added in this baseline measure. The difference between this baseline and the query reformulation methods that use selection gives an indication of the relative performance of selection of characteristics against reformulation of queries.

This baseline measure differs from the default case (no feedback), only in the fact that I select good characteristics of the original query terms. The difference between this technique and no feedback gives a measure of how successful the selection process is in the absence of any other information.

9.4.3 Summary

The cross combination of scoring technique (F_4, term characteristics (**NW** and **W**), term characteristics with selection (**NW** and **W**)) and query modification (query expansion or replacement) gives 12 experimental tests for each method of creating a new query. In the following section I shall discuss the results of these experiments.

9.5 Results

Table 9.5 gives the percentage increase or decrease over no feedback for each modification technique (four explanations and three baselines) after four iterations of feedback. In section 9.5.1 I discuss the query reformulation experiments and in section 9.5.2 I discuss the reweighting experiments.

Query modification type	AP (NW)	AP (W)	SJM (NW)	SJM (W)	WSJ (NW)	WSJ (W)
Coverage Replacement	**2.89%**	**2.84%**	**1.55%**	-0.04%	**1.78%**	-0.89%
Coverage Expansion	**3.43%**	**5.57%**	**3.77%**	**3.28%**	**1.87%**	**1.60%**
Coverage Replacement F4	-0.47%	-1.24%	-5.20%	-9.15%	-0.06**%**	-2.73%
Coverage Expansion F4	**6.47%**	**6.53%**	**9.27%**	**0.39%**	**4.71%**	**0.52%**
Coverage Replacement Selection	**2.95%**	**2.41%**	**5.20%**	-1.69%	**2.63%**	-1.18%
Coverage Expansion Selection	**14.96%**	**10.79%**	**14.44%**	**7.78%**	**10.96%**	**2.20%**
Expansion	**6.53%**	**3.43%**	**5.67%**	**0.70%**	-1.06%	**0.67%**
Expansion F4 ***(Baseline 1)***	**8.83%**	**4.07%**	**11.47%**	**3.67%**	**9.22%**	**1.74%**
Expansion Selection	**9.47%**	**5.13%**	**8.92%**	**5.29%**	**3.68%**	**2.10%**
Josephson Replacement	**1.31%**	**2.55%**	**1.64%**	-4.31%	-0.81%	-1.83%
Josephson Expansion	**5.84%**	**5.36%**	**7.91%**	**4.75%**	**1.50%**	**1.25%**
Josephson Replacement F4	-1.01%	-0.67%	-3.43%	-11.86%	-2.08%	-3.26%
Josephson Expansion F4	**7.52%**	**5.18%**	**12.66%**	**1.63%**	**4.17%**	**0.86%**
Josephson Replacement Selection	**1.31%**	**2.05%**	**3.21%**	-5.69%	-0.35%	-2.08%
Josephson Expansion Selection	**9.33%**	**7.81%**	**18.04%**	**9.05%**	**9.33%**	**2.30%**
Just selection ***(Baseline 3)***	**6.44%**	**2.43%**	-4.99%	**4.76%**	**5.26%**	**0.69%**
Min Card Replacement	-11.21%	-10.04%	-25.22%	-24.67%	-7.89%	-7.92%
Min Card Expansion	-9.57%	-8.46%	-23.78%	-23.05%	-6.92%	-6.95%
Min Card Replacement F4	-11.21%	-9.97%	-25.13%	-24.49%	-7.96%	-7.94%
Min Card Expansion F4	**2.85%**	-0.65%	**6.23%**	-1.84%	**2.69%**	-0.86%
Min Card Replacement Selection	-11.24%	-10.07%	-25.20%	-24.04%	-7.80%	-7.87%
Min Card Expansion Selection	-1.86%	-4.12%	-8.81%	-14.49%	-0.31%	-4.45%
Relevancy Replacement	-3.38%	-4.39%	-21.08%	-21.22%	-7.58%	**0.16%**
Relevancy Expansion	-3.38%	-4.39%	-21.08%	-21.22%	-7.58%	**0.16%**
Relevancy Replacement F4	**28.40%**	**21.37%**	**18.68%**	**11.70%**	-7.69%	-6.73%
Relevancy Expansion F4	**28.40%**	**21.37%**	**18.68%**	**11.70%**	-7.80%	-6.73%
Replacement	**2.52%**	-0.40%	-8.25%	-5.42%	-8.44%	-2.18%
Replacement F4	-0.05%	-2.09%	-11.28%	-11.0%	-8.43%	-2.37%
Replacement Selection	**1.52%**	-0.96%	-22.37%	-6.98%	-7.84%	-2.74%
Variable Replacement	-6.81%	-6.71%	-3.24%	-4.53%	-7.96%	-4.53%
Variable Expansion	**1.42%**	-0.16%	**9.21%**	**2.39%**	-0.32%	-0.79%
Variable Replacement F4	-7.09%	-6.71%	-4.95%	-7.01%	-7.96%	-5.08%
Variable Expansion F4 ***(Baseline 2)***	**4.73%**	**1.01%**	**15.44%**	**5.20%**	-2.84%	**0.55%**
Variable Replacement Selection	-7.00%	-7.06%	-22.73%	-6.28%	-4.37%	-4.74%
Variable Expansion Selection	**7.00%**	**2.90%**	**10.91%**	**8.21%**	**5.42%**	**1.10%**

Table 9.4: Percentage change in average precision after four iterations of feedback.
bold indicate increased values

9.5.1 Query reformulation

9.5.1.1 Query expansion and query replacement

The first major conclusion from the query reformulation experiments is that query expansion almost always performs better than or at least as well as query replacement. There are at least three possible reasons for this. First, as noted in section 9.4.1, the queries terms are usually a good source of evidence for targeting relevant documents.

Second, query expansion will usually produce longer queries than query replacement. Therefore query expansion may retrieve more documents or provide more evidence upon which to rank the documents than query replacement.

Third, I can also suggest a third cause for the success of the query expansion methods: the relevance assessments themselves. In Table 9.5 I present the percentage of relevant documents, averaged across the queries, which have at least one query term. At least 75% of the relevant documents in each collection have at least one original query term. Therefore if the original query terms are retained, we can guarantee that at least 75% of the relevant documents will be retrieved. Any feedback terms added to the query serve to modify the order in which these documents are ranked, and to retrieve documents that do not contain a query term. If we do not use the original query terms then we have to rely on the feedback terms retrieving at least 75% of the relevant documents to equal the performance of the original document ranking. From Table 9.4, we can see that this does not happen: the majority of query replacement techniques perform worse than no feedback.

Collection	Percentage of relevant documents containing a query term
AP	74.88%
SJM	87.18%
WSJ	88.16%

Table 9.5: Percentage of relevant documents that contain at least one query term

9.5.1.2 Baseline measures

In this section I compare the performance of the three baseline measures against each other. The Baseline 1 measure adds an identical number of terms to each query, Baseline 2 adds a variable number of terms and Baseline 3 adds no new terms but selects good characteristics for the original query terms.

In Table 9.6 I list, in decreasing order of average precision after four iterations, which explanations performed best for each collection and condition[86]. From Tables 9.4 and 9.6, the most noticeable difference is that different baselines work better on different collections: different RF techniques give better performance on each of the three test collections I used. Baseline 1 was best on the AP and WSJ collections, whereas Baseline 2 was best on the on the SJM.

AP (NW)	**AP (W)**	**SJM (NW)**	**SJM (W)**	**WSJ (NW)**	**WSJ (W)**
Rel *13.77*	Rel *16.98*	Rel *14.28*	Rel *16.18*	Cov *14.13*	Jos *16.28*
Cov *12.33*	Cov *15.50*	Jos *14.2*	Jos *15.8*	Jos *13.92*	Cov *16.26*
Jos *11.73*	Jos *15.08*	B2 *13.89*	Cov *15.62*	B1 *13.91*	B1 *16.19*
B1 *11.67*	B1 *14.56*	Cov *13.76*	B2 *15.24*	B3 *13.40*	B3 *16.02*
B3 *11.41*	B3 *14.33*	B1 *13.41*	B3 *15.18*	B2 *12.88*	B2 *16.00*
B2 *11.23*	B2 *14.13*	B3 *13.40*	B1 *15.02*	NoFd *12.73*	NoFd *15.91*
MinC *11.03*	NoFd *13.99*	NoFd *12.03*	NoFd *14.49*	MinC *12.69*	Rel *15.94*
NoFd *10.72*	MinC *13.41*	MinC *10.97*	MinC *14.22*	Rel *11.77*	MinC *15.20*

Table 9.6: Highest average precision after four iterations of feedback (average precision figures in italic)
B1 = Baseline1, B2 = Baseline2, B3 = Baseline3, Cov = Coverage explanation, Jos = Josephson explanation, MinC = Minimal cardinality explanation, NoFd = No feedback

Overall, the Baseline 2 technique tended to perform less well than the other two baseline measures which suggests that simply varying the number of expansion terms in proportion to the number of relevant documents used for feedback does not yield any improvement over adding a constant number of terms. However, as I shall discuss in section 9.5.1.3, varying the number of expansion terms by the use of explanations does improve performance.

[86] This is the best performing case of each explanation, e.g. the best results achieved by a Coverage explanation, Josephson explanation, etc.

The Baseline 3 measure does not add query terms but selects good term characteristics of the original query terms. The Baseline 3 measure performs noticeably better than performing no feedback at all, performs better than the Minimal Cardinality expansion explanation and usually performs better than the query expansion Baseline 2 method. This demonstrates that appropriate selection of good indicators of term use is important for RF.

9.5.1.3 Explanations

In this section I analyse the relative performance of the explanation methods of query reformulation. From Table 9.7, the first observation is that the relative performance of explanations is fairly stable across the conditions: explanations that do well on the non-weighting condition for a collection also tend to perform well on the weighting condition. This occurs because, although different explanations select different terms for each query, an explanation method tends to select similar terms when using weighting or no weighting. The different retrieval results between the weighting and non-weighting conditions arise due to the ranking of documents rather than the content of the query.

On all collections the explanation methods based on the Minimal Cardinality method of creating an explanation – selecting terms with low F_4 weights but high collection frequency- performed poorly. The only conditions in which this method gave an increase in retrieval effectiveness was when we used query expansion, scored documents using the F_4 weighting scheme and did not weight the characteristics used to provide the initial ranking. However this query reformulation method performed more poorly than other methods that also used expansion and F_4 scores, suggesting that the choice of terms from this method was poor.

The Relevancy method – adding all possible expansion terms – was the most successful method on the AP and SJM collections. However it performed poorly on the WSJ collection. This method, although successful on two collections, is very expensive – we have to run a new retrieval using a large number of expansion terms. Consequently, this is not an appropriate method for interactive information retrieval, although it may be appropriate for filtering applications, [BSA94].

The Josephson method – selecting terms according to explanatory power - and Coverage method – selecting terms according to their occurrence in the relevant documents - increase retrieval effectiveness over the collections if we use query expansion. If we also use selection then we can gain even better performance. These explanations are examples of Relevancy explanations but each place a restriction on the creation of the explanation (explanatory

power and coverage of relevant items respectively). This extra restriction reduces the number of feedback terms added to the query, reducing retrieval processing time, but still give good overall increases in average precision.

9.5.1.4 Performance of explanations against baselines

The only baseline measure to give an increase in performance over all collections (**NW** and **W**) was Baseline 1: expansion by the top *n* terms using the F_4 weights of terms to score documents. The Baseline 2 measure will give an increase in all cases only if we expand the query and use selection of term and document characteristics.

The Coverage and Josephson expansion methods will give an increase across all collections (**NW** and **W**) if we use them to expand the query. This holds if we use a combination of all term characteristics[87], selection of term characteristics[88] or F_4 weights[89] to score documents. This means that these two explanation methods of expanding a query are stable across methods of scoring documents.

All the explanation methods add a variable number of terms to the query, as does the Baseline 2 measure. The Coverage explanation outperforms the Baseline 2 measure in five of the six cases in Table 4 and the Josephson explanation always outperforms the Baseline 2 measure. These two explanation methods always outperform the Baseline 1 measure that adds a fixed number of terms. This demonstrates that adding a variable number of terms does increase retrieval effectiveness (explanations compared against Baseline 1) but the variation in number of terms added is not dependent on the *number* of relevant documents but the *content* of relevant documents (explanations compared against Baseline 2).

On all collections, with the exception of SJM (**NW**) either a Josephson explanation or a Coverage explanation method gives better performance than all Baseline methods.

In Table 9.6 I present the percentage of queries, for each collection, that improved when using the different query reformulation techniques. The Minimal Cardinality method improved around 30% of queries on average but the majority of queries were either made worse or showed no improvement. The Baseline 2 method improved queries in the non-weighting case but the percentage of queries improved dropped for the weighting case. This method then works well for poor (**NW**) initial rankings.

---

[87] Josephson Expansion, Coverage Expansion in Table 9.3.

[88] Josephson Expansion Selection, Coverage Expansion Selection in Table 9.3.

[89] Josephson Expansion F4, Coverage Expansion F4 in Table 9.3.

No method improved more queries than it harmed on the WSJ weighting condition, indicating that this is a difficult collection for RF to gain improvements in retrieval effectiveness.

For all other conditions, the Coverage and Josephson explanations and Baselines 1 and 3 increased the performance of more queries than they harmed through feedback. The Coverage explanation always performed better than Baselines 1 and 3 whereas the Josephson explanation only performed more poorly than the Baselines 1 and 3 in the WSJ (**NW**) case.

The Baseline 3 method (selecting good characteristics of the original query terms) performs better overall the Baseline 1 (reweighting and query expansion) which reiterates the fact that how the original query terms are treated is important.

Overall the Coverage and Josephson methods not only increase the performance of more queries than they harm, they also increase the performance of more queries than the standard Baseline 1 method of RF. This demonstrates that the query reformulation techniques not only perform better on average but also perform better for more queries, i.e. they are more *consistent* in improving retrieval effectiveness.

	AP NW	**AP W**	**SJM NW**	**SJM W**	**WSJ NW**	**WSJ W**	**average**
B1	56%	50%	76%	65%	62%	62%	62%
B2	50%	42%	70%	50%	58%	56%	54%
B3	73%	54%	76%	67%	62%	56%	65%
Cov	**75%**	**69%**	**83%**	74%	**71%**	**67%**	**73%**
MinC	38%	27%	35%	22%	44%	49%	36%
Jos	**75%**	67%	80%	**78%**	60%	62%	70%

Table 9.7: Number of queries improved by each query reformulation method. Highest number shown in bold.

9.5.2 Method of scoring the documents

I now discuss the methods of scoring the documents I proposed in section 9.3.2. I first report on the performance of the three abductive approaches, sections 9.5.2.1 – 9.5.2.3, then the abductive approaches with the standard relevance weighting approach to term weighting, section 9.5.2.4 and I draw conclusions in section 9.5.2.5

9.5.2.1 Term and document characteristics

This method scored query terms by the combination of all term and document characteristics. If we use query expansion, rather than query replacement, then this method can give positive results but these are generally lower than those given by F_4 or the selection of characteristics. As demonstrated in Part II combination is a variable technique in that individual combinations can work very well but these improvements often do not hold over a set of queries. However, if we use query replacement then scoring by characteristics can give better results but this is variable. That is that the combination of all term and document characteristics can give better results for the expansion terms than the discriminatory F_4 weights. Therefore we may want to use existing query terms and expansion terms differently when scoring documents for a new retrieval.

9.5.2.2 Weighting characteristics

In Part II I demonstrated that the weighting condition (**W**), in which we treat characteristics as being of varying importance, usually gave better results than the non-weighting condition (**NW**) in which all characteristics were regarded as being equally important. In the experiments reported in this chapter, this finding held: weighting characteristics gives better overall retrieval effectiveness than non-weighting (Table 9.6). However, as in Part II, although the retrieval effectiveness is higher with weighting, the percentage increase in average precision in this case is not as high as in the non-weighting case.

9.5.2.3 Selection of characteristics

The basis behind selection of term characteristics is that different characteristics are better indicators of relevance for different query terms, and, if we select good term characteristics we can better rank documents. This is generally true if we use query expansion rather than query replacement. Applying the selection process to the original query terms also gives good performance (Baseline 3, Table 9.4).

9.5.2.4 F4

The relevance feedback weighting scheme (F_4), performs better than term and document characteristics (section 9.5.2.1) when using query expansion. However, if we use selection of characteristics, in nearly all cases the selection method outperforms the relevance feedback method. This is true in the weighting (**W**) or non-weighting (**NW**) conditions and whether we use query expansion or replacement.

The main exception to this rule is the Minimal Cardinality method in which selection tends to decrease performance when measured against F_4. As described in section 9.5.1.2, this method

chooses poor indicators of relevance. Consequently attempting to select good aspects of term use for poor indicators of relevance does not give good performance.

9.5.2.5 Summary

The research aim in this set of experiments was to demonstrate that we could use abductive methods to decide how query terms should be used to score documents for relevance feedback. The results indicate that the more information we have on which to base this decision the better (selection of characteristics works better than no selection, weighting works better than no weighting). That is the more information we have to describe *why* a term may be a good indicator of relevance, the better we can use the term to improve retrieval effectiveness. The selection method, in particular, gives good and consistent results over the collections tested.

9.6 Summary

The experiments reported in this chapter examine the process of RF from an abductive viewpoint. I have demonstrated that the two techniques I investigated – query reformulation and term reweighting – provide the basis for new RF algorithms that provide more consistent increases in retrieval effectiveness. Two differences between the explanations and the standard Baseline 1 RF technique is that the explanations add a different number of terms to the query and add different terms to the query.

In the next chapter I present a deeper analysis of these results to investigate two aspects. Firstly I investigate the stability of the results. That is, how do the results change when we change parameters such as the number of documents used for feedback and how explanatory power is measured. Secondly I investigate *why* individual query reformulation techniques perform better than others.

Chapter Ten

Further experiments on explanations

10.1 Introduction

The experiments I described in the previous chapter produced new queries based on relevance information. In this chapter I am concerned with investigating what factors affect the performance of the new query expansion and term weighting approaches. This investigation is composed of two sets of experiments.

First I investigate varying the experimental conditions used in the previous experiments (in particular the method of ranking terms, the number of documents used for feedback and which documents are used for feedback). These experiments investigate the affect of changing the evidence used for creating explanations and changing how we can measure explanatory power. This will be discussed in section 10.2. Second I investigate *which* queries have increased retrieval effectiveness using different methods of query modification. This second investigation is an attempt to uncover why individual feedback techniques work well for individual queries. This will be discussed in section 10.3. In section 10.4 I discuss how this analysis can be used to *select* when individual feedback techniques should be used. I conclude with a discussion on the overall approach to abductive-based relevance feedback in section 10.5.

10.2 Experiments on evidence and explanatory power

In the previous experiments three experimental parameters were held constant: the number of documents used for feedback was 100 per iteration, the method of ranking possible expansion terms used the F_4 term reweighting scheme and all known relevant documents were used for query modification. In this section I examine the effect of varying these three parameters. This is an attempt to investigate how sensitive the query modification techniques are to changes in the experimental conditions. The experimental conditions define what evidence is used to create explanations; this investigation, in effect, assesses the effect of changing the evidence used to create explanations.

In section 10.2.1 I change the number of documents used for feedback, in section 10.2.2 I change the term reweighting scheme, and in section 10.2.3 I change which documents are used for feedback.

For these experiments I only concentrate on the effective methods from the last set of experiments. These are the Coverage and Josephson explanations (sections 9.2.1 and 9.4.2)[90] and the three baselines (sections 9.4.2.1-9.4.2.3). Throughout I only use query expansion (rather than query replacement), and use weighting of characteristics (rather than no weighting). I only use query expansion as it was shown throughout the first set of experiments to be more effective. I only use weighting as the overall trends of the results were the same with or without weighting. That is weighting alters the average precision given by a feedback technique but does not alter the relative performance of the technique relative to other techniques. The weighting of characteristics as shown in Part II nearly always gave the higher average precision figures across the feedback techniques investigated.

10.2.1 Number of documents used for feedback

The same experiments were run as described in section 9.4, using 25 documents per iteration and 50 documents per feedback iteration. This is in contrast with the previous set of experiments that used 100 documents per feedback iteration. The number of documents used in feedback may affect the results of RF for two reasons:

i. the number of *relevant* documents used for feedback will change. If we use fewer documents per feedback iteration then we are likely to reduce the number of relevant documents used for feedback. Table 10.1 shows the number of relevant documents found in the top 25, 50 or 100 ranked documents after an initial query for each of the three test collections[91].

For each collection, as the number of documents used for feedback, *n*, increases the number of relevant documents used for feedback increases. This means, at higher values of *n*, that the feedback algorithms have more evidence of what constitutes relevance upon which to base feedback decisions.

---

[90] The Relevancy explanation was omitted as it is not practical for interactive IR which is the goal of these feedback techniques.

[91] This is only shown for the initial iteration. For subsequent iterations, i.e. after RF, the number of relevant documents found will be dependent on the success of the RF technique.

Also shown in Table 10.1 is the *concentration* of relevant documents (the percentages in Table 10.1), This shows that at higher values of *n*, although there have been more relevant documents found, the percentage of relevant to non-relevant documents in *n* is lower. That is, the set of documents that are used for feedback contains a lower percentage of relevant documents.

Number of documents used per iteration	AP	SJM	WSJ
25	199 (16.58%)	251 (21.83%)	130 (*11.56%)*
50	289 (12.04%)	375 (16.30%)	193 (*8.58%)*
100	379 (7.90%)	592 (12.87%)	279 (*6.20%)*

Table 10.1: Percentages of relevant documents found in top *n* documents after an initial query run

To summarise, at lower values of *n*, there are fewer relevant documents but a higher *proportion* of relevant documents. At larger values of *n*, there are more relevant documents but a lower proportion of relevant documents in the set of documents used for feedback.

This may affect the individual query modification techniques differently. The F_4 term reweighting approach, for example, is based on the difference between the relevant documents and the rest of the collection and so may benefit from more relevance information (higher *n* values). A technique, such as selection of term characteristics, that is only based on the retrieved set of documents may benefit from having a higher proportion of relevant documents (lower *n* values), as this technique tends to work better on documents with a similar content.

ii. the number of documents used may change the precision of the search. The evaluation method I use is the full-freezing method. This method fixes the rank positions of documents that were used for feedback. This means that documents used for feedback remain in the same ranks positions before and after feedback – the rank positions are *frozen*. Using a larger number of documents for feedback will increase the number of documents that are frozen.

The use of freezing is necessary to evaluate the effect of feedback on only the unseen relevant documents – the ones not used for feedback. However, the use of freezing places an upper limit on the *potential* improvement that can be gained by a feedback algorithm. A feedback algorithm can only move an unseen relevant document up the ranking to the top unfrozen rank position. The more documents that are used for feedback, the lower down the ranking

this top unfrozen rank position will appear. This means that the potential improvement to be gained from feedback may be reduced when using larger values of *n*, resulting in lower average precision figures.

A second potential reason for lower average precision when using larger values of *n* is that there are fewer unretrieved relevant documents to be retrieved. Table 10.2 lists the percentage of relevant documents that are unretrieved at the three levels of *n*. As the value of *n* increases, the percentage of unretrieved relevant documents – the ones that will be responsible for any change in average precision after feedback - decreases. Fewer unretrieved relevant documents means that there are less documents to change the average precision values.

Number of documents used for feedback (*n*)	AP	SJM	WSJ
25	88%	90%	92%
50	83%	85%	88%
100	77%	77%	83%

Table 10.2: Percentage of unseen relevant documents at different values of *n*

10.2.1.1 Results of varying *n*

Tables F.2, Appendix F, presents the average precision figures obtained using 25, 50 or 100 documents per feedback iteration. In Table 10.3 I summarise these results by presenting the percentage of queries that were improved (top half of Table 10.3) at each value of *n*. The bottom half of Table 10.3 shows the percentage of queries for which each technique gave the *greatest* increase in average precision. From the values in the bottom-half of Table 10.3 we can see that for many queries there was more than one technique which gave the best performance, i.e. the columns do not sum to 100%.

	AP			SJM			WSJ		
	25	**50**	**100**	**25**	**50**	**100**	**25**	**50**	**100**
B1	60%	56%	70%	56%	60%	70%	50%	54%	61%
B2	44%	50%	61%	38%	46%	54%	33%	44%	54%
B3	46%	52%	67%	60%	67%	70%	40%	50%	54%
Cov	**67%**	58%	**83%**	67%	71%	76%	50%	**60%**	**65%**
Jos	52%	**67%**	80%	**71%**	**73%**	**80%**	**52%**	58%	61%
B1	**42%**	31%	37%	**27%**	19%	20%	**35%**	**35%**	**46%**
B2	19%	17%	15%	10%	13%	13%	21%	25%	30%
B3	27%	27%	28%	21%	25%	30%	19%	19%	30%
Cov	35%	**35%**	**52%**	**27%**	**33%**	**39%**	27%	31%	**46%**
Jos	29%	**35%**	37%	23%	29%	33%	23%	29%	33%

Table 10.3: Affect of varying *n* when using F4 term weighting scheme

There are two main conclusions that can be drawn from this experiment.

i. more relevance information is generally better. All explanation techniques and all baselines improved a higher percentage of queries when using more relevance information (higher *n* values).

ii. increased relevance information evens out performance differences between the query modification techniques. At lower values of *n*, the technique that gives the *highest* increase in average precision is more likely to be the technique that is the *unique* best technique for a query, i.e. no other technique is as good as the best one. At higher values of *n* it is more likely that several techniques give equally good increases in average precision. This is also a factor of the freezing evaluation technique as, at higher *n* values, more relevant documents are retrieved in the initial iterations.

These conclusions are independent of how the terms are ranked for expansion: similar conclusions are drawn when I use alternative methods than F_4 to rank terms, Tables F.4 and F.5. These alternative methods will be discussed in section 10.2.2.

However even though more relevance information is generally better, the value of *n* does not affect all query modification techniques in the same way.

This is especially true if we compare the *average precision* rather than the number of queries improved, Tables F.1 – F.3. For example, if we do not use selection of characteristics, then better average precision is achieved at low *n* values. Similarly, for the Baseline 1 measure, when using the F_4 values to score documents, the best average precision is achieved at lower values of *n.* The SJM collection, when using selection of characteristics, also benefits more from lower values of *n*.

One reason for the difference in results is that some queries perform better than others when using feedback. Queries that do well with RF will be improved most with low values of *n*. For example, queries which retrieve a lot of relevant documents will perform better as there is more relevant information to modify the query. Lower values of *n* also means that these queries can be improved in fewer iterations. This is because less documents are frozen in the evaluation so relevant documents can move further up the ranking, improving the average precision. In addition, as relevant documents move further up the ranking they are more likely to be used in subsequent query modification. This is because they are more likely to fall within the set of documents used for feedback. At high values of *n* these queries are likely to give *lower* average precision than they would achieve at lower *n* values.

Conversely, queries for which RF performs poorly are more likely to benefit from high *n* values – more documents being used for feedback. This is because these queries will require more information on relevance to perform successful feedback. At low values of *n* there may not be enough relevant documents to modify the query successfully. The result is that at low *n* values some queries do very well, others are not improved or not improved by a great amount. At high *n* values queries for which RF performs well show small increases than at low *n* values but the other queries are more likely to be improved.

A second reason for the difference in performance at different *n* values is due to the individual query modification techniques. For example, experiments on the SJM which use selection of characteristics perform better at low *n* values. This collection has a higher number of relevant documents per query compared to the other collections. Consequently there may be insufficient evidence upon which to base the selection of characteristics. On the other collections, AP and WSJ, there are fewer relevant documents and so higher *n* values may be required to provide enough evidence to select good characteristics.

10.2.2 Explanatory power

The previous experiments, in Chapter Nine, used the F_4 term reweighting scheme in three ways:

i. to rank possible expansion terms. The F_4 scheme was used to rank terms for query expansion in the Josephson, Relevancy, Minimal Cardinality, Baseline 1 and Baseline 2 query reformulation techniques. The F_4 weight in this case was taken to be a measure of the explanatory power of a term.

ii. to order ties in the Coverage method. The Coverage explanation ranks terms by how many relevant documents the term explains. Terms that explain an equal number of relevant documents are then ordered by their F_4 value.

iii. as an alternative document scoring technique. The previous experiments, section 9.5, contrasted two methods of scoring documents given a query; using the term and document characteristics and using the F_4 values of the query terms in the documents.

Different term reweighting schemes, however, may give different results. This is partly because they perform at different levels. That is, some term reweighting schemes are better than others at discriminating relevance than others and hence better at providing new weights for terms. The different effectiveness of individual techniques may also be due to the fact that they select better terms for query expansion than other techniques. As noted by Robertson, [Rob90], an algorithm that is good at assigning discriminatory values to existing query terms may not be the best algorithm to use when selecting *new* query terms. I shall examine this in the experiments.

I ran the same experiments as in Chapter Nine, using two alternative term reweighting schemes, Porter's term weighting scheme, [PG88], and Robertson's *wpq* formula, [Rob90]. Porter's term reweighting formula[92] places emphasis on terms that occur more frequently in the set of relevant documents. Robertson's *wpq* formula was specifically suggested as a method of selecting new terms for query expansion[93] and incorporates the F_4 scheme.

The three term reweighting schemes – Porter, F_4, and *wpq* – are at varying levels of complexity. Porter's weighting scheme uses only the difference between the frequency of a

---

[92] Chapter One

[93] Appendix A

term in the relevant documents and the frequency of the term in the non-relevant documents. F_4 takes into account the *absence* of a term in the relevant and non-relevant documents as well as the presence of the terms. The *wpq* formula incorporates the F_4 scheme but multiplies this by a formula similar to Porter's scheme to calculate term weights.

I ran the experiments again, using 25, 50 or 100 documents per feedback iteration, using Porter's scheme and *wpq* in place of the F_4 measure.

10.2.2.1 Results on varying explanatory power

The results from varying explanatory power are shown in Appendix F, Tables F.1 – F.5. The major conclusions from this experiment are:

i. For Baselines 1 and 2 (these used Porter/F_4/*wpq* to score documents) better term ranking techniques gave better results. As expected, the more sophisticated the method of ranking terms for expansion, the better the average precision results. The results given when using *wpq* were generally better than those given by F_4 which, in turn, were better than those given by Porter's scheme.

ii. For the explanations, better term ranking algorithms generally give better average precision and improve a higher percentage of the queries. However this performance increase is usually less than that achieved by the Baselines. The performance of Baselines 1 and 2 were better than explanations when using *wpq* but explanations were better when using F_4 or Porter.

Table 10.4 shows the best performing query modification technique for each value of *n* and for each measure of explanatory power and the average precision of the best technique. As can be seen from Table 10.4 the best method when using Porter's scheme is always an explanation method, generally the best method when using F_4 is also an explanation. However at small values of *n* the selection procedures do not work so effectively. This was discussed in section 10.2.1.1.

When using the *wpq* scheme, expansion by a fixed number of terms gives better performance. In the AP collection a fixed number of terms, and combination of characteristics gave best average precision, for the SJM and WSJ collections the Baseline 1 (expansion by a fixed number of terms and weighting by *wpq*) gave the best performance.

Porter			
	25	**50**	**100**
AP	Josephson *6.22*	Coverage *6.26*	Coverage sel *10.79*
SJM	Josephson sel *16.07*	Josephson sel *11.69*	Coverage sel *8.85*
WSJ	Coverage sel *3.98*	Coverage sel *2.32*	Coverage sel *2.20*
F4			
AP	Expansion *11.93*	Coverage sel *11.17*	Coverage sel *10.79*
SJM	Josephson sel *16.37*	Josephson sel *12.93*	Josephson sel *9.04*
WSJ	Baseline 1 *4.70*	Josephson sel *2.97*	Josephson sel *2.33*
wpq			
AP	Expansion *38.51*	Expansion *35.52*	Expansion *24.47*
SJM	Baseline 1 *42.21*	Baseline 1 *32.51*	Baseline 2 *28.96*
WSJ	Baseline 1 *12.98*	Baseline 1 *7.81*	Baseline 1 *5.39*

Table 10.4: Best performing query modification technique for different values of *n* and for different term reweighting techniques
sel = selection of characteristics, Expansion = Baseline 1 using all term and document characteristics to score documents.

There are two possible reasons for the relative differences between expansion and explanations. First, the *wpq* measure selects better terms for expansion. Therefore expanding the query by larger numbers of terms (the Baseline1 measure) will give better results.

Secondly using the *wpq* measure for explanations will tend to prioritise the original query terms as these appear in a large number of relevant documents. This means that the explanations are biased towards adding very few terms other than the original query terms. A possible solution to this problem is to create an explanation from non-query terms only and then add this explanation to the query. This is in contrast to the current technique that forms an explanation from all terms, including the query terms. However the explanations still improve the majority of queries, Tables F.4 and F.5, and on some collections can still improve more queries than the Baseline 1 measure using the *wpq* weights.

10.2.3 Which documents are used for feedback

In almost all RF techniques, query reformulation is based on all known relevant documents: all the documents found so far in the search. The work on ostension, [CVR96], treats documents retrieved most recently as being more *important* to query reformulation but still uses all retrieved relevant documents for query reformulation. In this section I examine the affects on retrieval performance by using only a subset of the known relevant documents.

There are many methods that could be applied to select which documents are to be used to create explanations. In Chapter Eight I discussed how the user's search behaviour could be used to select the important documents, the ones that require explanation. In this section, as I am using test collections, I cannot infer the important documents from a user's search; instead I simulate the selection of important documents by only using relevant documents found in the most recent search iteration. That is, for each RF iteration, I only use the most recently found relevant documents to create explanations and ignore any documents that have been found in previous iterations.

Only the process of selecting the components of explanations used the current set of documents, i.e. only those terms that appeared in a relevant document at the current iteration were considered for query expansion. The process of assigning weights to terms or selecting characteristics for terms, however, used all relevance information. This is because if a term explains a document we want as much information as possible on how the term explains the document.

To avoid running all versions of the previous experiments I only ran the experiments using the first 25 documents from each ranking, $n = 25$. I chose this value as small values of n are more comparable to real user searching than larger values.

10.2.3.1 Results from varying documents used for feedback

In Appendix F, Tables F.4 – F.26 I present the results of this experiment. In each table I highlight when using some relevance information, those relevant documents found at the current iteration, is better than using *all* relevance information.

In Table 10.5 I summarise the results according to how many times using some relevance information gave an increase in retrieval effectiveness, how often it gave no difference and how often it gave a decrease in retrieval effectiveness. In Table 10.5 the general trend of the

results are shown in bold, i.e. for an individual collection and term weighting technique did the average precision tend to increase, decrease or remain the same?

Collection	Term weighting technique	Increase	No change	Decrease
AP	**porter**	**12**	0	0
	F_4	**8**	0	4
	wpq	1	**7**	4
SJM	**porter**	**8**	0	4
	F_4	**7**	0	5
	wpq	**6**	0	**6**
WSJ	**porter**	4	0	**8**
	F_4	1	1	**10**
	wpq	3	0	**9**
Total		**50**	8	**50**

Table 10.5: Affect of altering relevant documents used for query modification

As can be seen from Table 10.5 (row **Total**) overall there was no general trend, i.e. the results were increased as often as they decreased. However, examining the effect of changing the relevant documents did affect the individual collections and weighting schemes differently. For example the results of using only the current set of documents were generally better if we used less effective term weighting techniques, e.g. Porter's scheme. Also the collections for which there were more relevant documents, i.e. AP and SJM tended to perform better when only using the current relevant documents. On the WSJ collection using all the relevant documents was almost always a better approach. However the results on this collection were generally lower using either all relevant documents or only the current relevant documents than the other collections. Therefore the absolute number of relevant documents found in the current iteration is perhaps important.

The techniques for which using the current set of documents tended to work well was when documents were ranked using the term and document characteristics rather than the Porter/F_4/*wpq* term weighting schemes.

This experiment represents a crude estimate of how we should select the relevant set of documents. In a real searching environment, where a user is assessing relevance, we would have much more realistic relevance assessments and more evidence upon which to base our

decision of which documents are the most important or most representative of what a user wants. Nevertheless the experiment does show, in a limited way, that selecting relevant documents is an area that is worth pursuing.

10.2.4 Summary

In sections 10.2.1 – 10.2.3 I have explored different methods of varying the experimental conditions used in Chapter Nine. This examined varying the evidence used to create explanations and the method by which explanatory power was measured. A final analysis of the effects of varying *n* and the measure of explanatory power is to examine the *overlap* between the queries improved by each technique. That is, which set of techniques improves different queries and which set of techniques tend to improve the same queries.

The tables for this analysis are presented in Appendix F, Tables F.6 – F.14. An example, taken from Table F.6, is shown in Table 10.6[94]. From Table 10.6 we can see that 100% of the queries improved by Baseline 2 are also improved by Baseline 1 (the bold entry, row 4, column 2), whereas only 64% of the queries improved by Baseline 1 are also improved by Baseline 2 (the underlined entry, row 3, column 3). Baseline 1, then, improves the same queries as Baseline 2 but also improves additional queries.

	AP				
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	64%	56%	72%	68%
B2	**100%**	100%	69%	69%	75%
B3	64%	50%	100%	95%	91%
Cov	56%	34%	66%	100%	66%
Jos	71%	50%	83%	88%	100%

Table 10.6: Percentage overlap between query modification techniques

This analysis is also performed for the techniques that give the *highest* increase in average precision for a query, i.e. for what percentage of queries are techniques *equally* good at improving retrieval effectiveness, Tables F.15 – F.23. An example of this is shown in Table 10.7[95]

[94] This table is for the AP collection, using 25 documents per feedback iteration and using Porter's term weighting scheme.

[95] This table is also for the AP collection, using 25 documents per feedback iteration and using Porter's term weighting scheme.

AP					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	38%	25%	25%	25%
B2	**86%**	100%	57%	57%	57%
B3	29%	29%	100%	86%	64%
Cov	19%	19%	57%	100%	43%
Jos	29%	29%	64%	64%	100%

Table 10.7: Percentage overlap between query modification techniques

This table shows that, for the queries where Baseline 2 is the most effective modification techniques, Baseline 1 is often equally good (for 86% of the queries where Baseline 2 gives the highest average precision, Baseline 1 gives the same average precision). Baseline 2, on the other hand, is only as good as Baseline 1 in a small percentage (38%) of those queries where Baseline 1 is the most effective technique.

These tables can be used to analyse the relative similarity of the query modification techniques in terms of those queries they improve and those queries for which they are the best technique to use. The main question here is how different are the techniques?

Comparing first the case where any improvement in retrieval effectiveness is considered. There are two general conclusions. Firstly Baseline 1 and Baseline 2 tend to improve similar sets of queries, i.e. they show a strong overlap. Both these techniques add a consecutive set of terms from the top of the expansion term ranking and use the Porter, F_4 or *wpq* discriminatory weights to score documents. The main difference between these two techniques comes from the number of terms added – Baseline 1 tends to add more terms to a query than Baseline 2.

There is also a strong correlation between Baseline 3 and the Coverage and Josephson explanations. These techniques all use selection of characteristics to score documents. Where there is less overlap between the techniques, this difference comes from the terms they add to the query. If there is a difference between the Coverage and Josephson methods it means they have added different terms to some queries and if there is a difference between these two techniques and Baseline 3 it means the terms added by the explanations have caused the different retrieval results.

The overlap between these two sets of techniques (Baselines 1 and 2, and Baseline 3, Josephson and Coverage) can be relatively low even if the techniques improve a large number

of the queries. For example, for the Porter term weighting scheme on the AP collection the Coverage method improves 32 queries and Baseline 1 improves 25 queries, however the overlap between them is only 56%, i.e. 13 queries.

The second conclusion is that the measure of explanatory power and the number of documents used for feedback, *n*, have a strong effect on the similarity of queries improved. At high values of *n* the techniques and with better measures of explanatory power, more queries are improved by each technique and the overlap between techniques increases.

The previous analysis was for all queries for which a query modification technique gave an increase. If we look at the queries for which an individual technique gave the highest increase then there is a similar pattern: Baseline 1 and Baseline 2 tend to give the highest increase in average precision for the same queries (high overlap), and Baseline 3, Josephson and Coverage also tend to improve the same queries. The overlap between these two groups of techniques, however, tends to be lower: there is a clearer distinction between Baselines 1 and 2, and the other techniques when we only consider the queries for which they performed best.

The effects of explanatory power and *n* are also present but are less pronounced. This is most noticeable on SJM where there is a very low overlap between the Baseline1/2 techniques and the other techniques at low values of *n*.

As there is evidence that different query modification can improve different queries, we should ask why some queries perform well with individual techniques. I examine this in section 10.3.

10.3 Performance of explanations

In this section I attempt to elicit reasons for why some query modification techniques perform more effectively in RF than others. I do this by examining the features of the queries themselves. For example I examine the number of relevant documents for each query, the precision of the initial search, the order in which relevant documents were retrieved, and the number of relevant documents found in the initial iteration. I shall explain the reasons for examining these aspects in more detail in sections 10.3.1 – 10.3.5. This analysis is intended to see if it is possible to decide *when* an individual query modification technique should be used, i.e. is it possible to decide for individual queries which type of query modification is most appropriate?

In these analyses I only concentrate on the techniques that gave the greatest improvement for a query. As the overlap in this case is small, i.e. the queries that are most improved by individual techniques are often different, this case can be used to elicit those techniques that work best under different conditions. In the remainder of this section I will outline how this analysis is performed.

In Appendix F, Tables F.28.-F.38, I present the figures upon which these analyses are based. Table 10.8 shows an example of the tables used to analyse the query modification techniques regarding the numbers of relevant documents per query. For each value of *n* I calculate the number of queries that had highest improvement using each query modification (columns labelled *Queries*) the total number of relevant documents for these queries (columns labelled *Rels*) and the average number of relevant documents per query (columns labelled *Avg*).

In Table 10.8, for example, it can be seen that when the Josephson techniques give the greatest increase in average precision they do so on queries that have a relatively high number of relevance assessments (high *Avg*). Conversely Baseline 2, when it gives an increase in average precision, does so for queries that have a low number of relevant documents (low *Avg*).

These tables are used to analyse under what conditions the query modification techniques perform well. In Table 10.8, for example, the Josephson and Coverage methods perform well on queries with high numbers of relevance assessments, and the Baseline 2 method performs well on queries with low numbers of relevance assessments.

AP									
	25			50			100		
	Rels	**Queries**	**Avg**	**Rels**	**Queries**	**Avg**	**Rels**	**Queries**	**Avg**
B1	362	20	18.10	199	15	13.27	247	17	14.53
B2	46	9	5.11	31	8	3.88	25	7	3.57
B3	172	13	13.23	135	13	10.38	172	13	13.23
Cov	390	17	22.94	535	17	31.47	722	24	30.08
Jos	670	14	47.86	596	17	35.06	617	17	36.29

Table 10.8: Calculation of average relevant documents per query

As there is a high overlap between some of the query modification techniques I shall group the techniques into three broad groups:

i. techniques that gave small changes to the query (*Min*). This the Josephson and Coverage methods

ii. techniques that gave larger changes to the query (*Max*). This is the Baseline 1 and Baseline 2 methods.

iii. techniques that did not change the content of the query. This is the Baseline 3 method.

This is a relatively superficial analysis of the results but I am only looking for general trends amongst the data. I examine the data for all values of n but concentrate mostly on the trends where $n = 25$.

10.3.1 Number of relevant documents

The first aspect to be measured is the total number of relevant documents for each query. This is the number of relevant documents contained within the test collection. The more relevant documents there are for a query, the more relevant documents are likely to be retrieved and therefore the RF algorithm has more evidence on what constitutes relevance upon which to modify the query. As shown in section 10.2.1 more information on relevance usually leads to better effectiveness after feedback. Queries for which there are many relevant documents therefore can be regarded as *easier* queries for RF to improve retrieval effectiveness.

This is a relatively simplistic categorisation of which queries are easy for feedback to improve. I have not, for example, considered how similar the relevant documents are to the initial query. A low similarity between the query and relevant documents could lead to poor

retrieval effectiveness for some RF techniques, e.g. the Baseline 3 measure which only reweights the original query terms. However the basic correlation between the number of relevant documents and the ease with which a query can be improved allows a preliminary analysis of the techniques.

In Table 10.9 I show the results of this analysis.

Low				High			
	AP	**SJM**	**WSJ**		**AP**	**SJM**	**WSJ**
Porter	Max	Max	B3	**Porter**	Min	Min	Min
F_4	Max	Max	B3	**F_4**	Min	Min	Min
wpq	B3	B3	B3	***wpq***	Max	Max	Max

Table 10.9: Techniques that gave highest improvement on queries with the lowest numbers of relevant documents (**left**) and highest number of relevant documents (**right**)
where Min = Josephson/Coverage, Max = B1/B2

From Table 10.9 there is a difference between the collections and which measure of explanatory power was used to form the explanations. For queries which have a lot of relevance documents (right-hand side of Table 10.9), and for which we have a good method of ranking terms for expansion (*wpq*), then we should use as much information as possible from the relevant documents. This means using a technique that gives maximal query expansion and use discriminatory power to score documents. If, however, we have a poorer method of ranking terms (Porter and F_4) then we should use a more restrictive technique such as the minimal methods. These are more selective in which terms are added and also are consider more aspects of discrimination between how terms are used.

However, for the queries where there are fewer relevant documents, then a different pattern arises: we often want to use more evidence from poorer term ranking approaches to compensate for the poor term selection by these methods. When we have a good method of ranking terms it is often better to concentrate on weighting original terms.

10.3.2 Percentage of relevant documents found

A second analysis is to compare the percentage of relevant documents found in the initial iteration. If a high percentage of the relevant documents are found in the initial ranking then the system has more *representative* information upon which to base to query modification decisions, i.e. the known relevant documents are more representative of the entire relevant document set. Lower percentages of relevant documents found in the initial ranking means

that the query modification techniques are basing feedback decisions on *less* representative sets of relevant documents.

From Table 10.10, in most cases, the techniques that performed well when the retrieved set contained a high percentage of the relevant documents were those that also performed well when there were large numbers of relevant documents, and those techniques that work well on a small set of relevant documents are those that also work on a few relevant documents. Therefore minimal explanations are often better when we have retrieved a large percentage of the relevant documents and a poor method of ranking terms but maximal explanations are better when we have a good method of ranking terms.

When we have a low percentage of relevant documents found, then we often want to use a larger expansion to broaden the queries (Porter and F_4) or not change the content of the query at all (*wpq*).

Low				**High**			
	AP	**SJM**	**WSJ**		**AP**	**SJM**	**WSJ**
Porter	Max	Max	Max	**Porter**	Min	Min	Min
F_4	Max	Max	B3	**F_4**	Min	Min	Min
wpq	B3	B3	B3	***wpq***	Max	Max	Max

Table 10.10: Techniques that gave highest improvement on queries with the lowest percentage of found relevant documents (**left**) and highest percentage of found relevant documents (**right**)
where Min = Josephson/Coverage, Max = B1/B2

The previous two analyses used values that could be derived from the test collection: the total number of relevant documents and the percentage of relevant documents found. However these cannot be used to decide which query modification technique should be used in a real interactive environment: we cannot know how many relevant documents exist for a user's query and we cannot know how many of these have been found in real searches.

In sections 10.3.3 – 10.3.5 I examine three aspects of retrieval that may be used to indicate when individual query techniques are performing well. These can then be used to *select* which query modification techniques are appropriate for individual searches. In section 10.3.3 I examine the precision of the initial search, in section 10.3.4 I examine where in the document ranking the relevant documents are found and in section 10.3.5 I examine the similarity of the relevant documents. In each of these sections I shall present why the analysis

is important, how the analysis was carried out and the main results. I shall discuss the overall findings in section 10.3.6.

10.3.3 Initial precision

In this section I perform the same analysis as previously on the number of relevant documents found in the initial iteration: the *precision* of the initial search. The initial precision is calculated as the number of relevant documents found in the top *n* documents divided by *n*.

When the initial precision is good the search is performing well and either the query is a good match with the relevant documents or the retrieval function is good at emphasising those aspects of query terms that are good indicators of relevance. Higher initial precision will also tend to provide more evidence for RF algorithms on what constitutes relevant material. Low precision, on the other hand, will mean there are fewer relevant documents to provide a set of expansion terms and less information upon which to base term weights.

The results of the analysis of which techniques work best on low precision searches and high precision searches is shown in Table 10.11. For low precision searches the general trend is to add query terms, thereby broadening the topic of the search. If we are using a poorer term expansion technique (such as Porter or F_4) this is usually a minimal explanation; one that is more selective about terms are added. When we use a better term ranking technique (such as *wpq*) then a maximal explanation is usually better. This technique adds better terms but also uses good discriminatory weights to score documents.

Low				**High**			
	AP	**SJM**	**WSJ**		**AP**	**SJM**	**WSJ**
Porter	Min	Min	Min	**Porter**	Max	Max	Max
F_4	Min	Min	Min	**F_4**	Max	Max	Max
wpq	Max	Max	Max	***wpq***	B3	B3	B3

Table 10.11: Techniques that gave an improvement with low initial precision (**left**) and highest initial precision (**right**)
where Min = Josephson/Coverage, Max = B1/B2

If we have high precision, and therefore good evidence as to what relevant documents will look like, good strategies are either bigger expansions (maximal explanations) or simply reweighting the original query terms, (B3), as these terms are already good at retrieving relevant information.

10.3.4 Order of relevant documents in ranking

This analysis is based on where in the document ranking the relevant documents are found. The rankings were analysed using the same method as in Chapter Five, section 5.5.2, which gives each ranking a score based on the rank positions of relevant documents. A high score means that relevant documents tend to be found higher up the document ranking; a low score means relevant documents are found further down in ranking. Low scores, generally, will mean that the relevant documents have a poorer match with either the query or with the retrieval function. This means that either the retrieval function does not emphasise the features that make the documents relevant or that the wrong query terms are being emphasised.

	Low				**High**		
	AP	**SJM**	**WSJ**		**AP**	**SJM**	**WSJ**
Porter	Max	Max	B3	**Porter**	B3	Min	Min
F_4	B3	Max	B3	**F_4**	Max	B3	Max
wpq	B3	Max	B3	***wpq***	Max	B3	Max

Table 10.12: Techniques that gave an improvement using the poorer ranking of relevant documents (**left**) and better rankings of relevant documents (**right**)
where Min = Josephson/Coverage, Max = B1/B2

For this analysis, two types of query modification dominate: the maximal explanations and the B3 technique. These types of query modification differ in how they perform on individual collections. The B3 technique works best on poorer rankings of documents on the AP and WSJ collections and the maximal explanations work best on good rankings for these collections. However this is reversed for the SJM collections: the maximal explanations work best on poor rankings and the B3 technique works best on good rankings.

There are several differences between the SJM and AP/WSJ collections: the SJM collection, on average, has fewer terms per document, more relevant documents per query and longer queries. The difference in number of relevant documents per query means that the SJM collection generally has better evidence upon which to base query modification: more relevant documents.

Table 10.13 presents the average number of relevant documents found in the initial iteration for those queries for which a query reformulation technique gave the best performance. For example, for all those queries for which Baseline 1 was the best query reformulation technique, Table 10.13 presents the average number of relevant documents found in the initial

iteration. Table 10.13 only shows the results for $n = 25$ as this was the value of n which we concentrated upon for these analyses.

From Table 10.13 it can be seen that the SJM queries tend to retrieve more documents in the initial iteration for *any* technique. That is the difference between the collections in Table 10.12 probably comes from the fact that the SJM retrieves more documents whether the order is low or high. Therefore the absolute number of relevant documents may be important as well as the order in which they are retrieved.

The general trend from this analysis is therefore: if there are few relevant documents and these are retrieved further down the ranking or if there are lots of relevant documents and these are retrieved high up the ranking, the original query terms are the best source of evidence. In this case we should use a technique such as B3 which concentrates on how these terms are used to score documents for retrieval.

		Baseline 1	**Baseline 2**	**Baseline 3**	**Coverage**	**Josephson**
AP	**Porter**	1.63	2.00	7.14	**4.19**	2.79
	F_4	2.65	2.44	2.85	**3.82**	3.14
	wpq	4.05	**5.37**	1.64	3.08	3.40
SJM	**Porter**	4.55	1.67	4.63	6.40	**6.78**
	F_4	3.85	4.40	4.50	5.62	**7.27**
	wpq	5.92	**8.60**	1.71	5.17	4.43
WSJ	**Porter**	1.08	1.44	0.56	2.00	**2.85**
	F_4	1.76	1.80	0.56	2.08	**2.09**
	wpq	2.48	1.44	0.56	2.55	**3.70**

Table 10.13: Average of relevant documents found in initial iteration for cases where a query reformulation technique performed best
bold entries are highest number of relevant documents found

On the other hand if there are lots of relevant documents and these are retrieved further down the ranking, or if there are few relevant documents and these are retrieved high up the ranking we should use a query expansion technique. In the case where there are many relevant documents but these are retrieved lower down the ranking, a query expansion technique will add more terms from these documents to the query and improve the ranking of documents similar to the relevant ones. In the case where there are few relevant documents a query expansion technique will add more terms from the documents that are a good match with the query.

10.3.5 Similarity of relevant documents

This analysis is based on the similarity of the relevant documents to each other. If the relevant documents are very similar to each other, i.e. they share a lot of terms in common, then we may want to change the query in a different way than if the relevant documents are dissimilar to each others. For example, if the documents are similar then we may want to concentrate on techniques that add more of the shared terms whereas, if the documents are less similar, then we may want be careful about adding terms that only explain some of the relevant documents.

However we should also consider how similar the relevant documents are to other retrieved documents – how the relevant documents are separated from the non-relevant documents. This means that we should ask how similar the relevant documents are with respect to those terms that discriminate the relevant from the non-relevant documents.

There are many methods for analysing the similarity of the relevant documents. For example we may want to *cluster* the relevant documents, [TVR01]. However, as I am seeking a measure of similarity that can be used within an interactive situation, I developed a simple alternative that can be applied while the system is calculating the list of expansion terms.

This technique bases the measure of similarity on the relative number of discriminatory terms contained within the set of relevant documents, i.e. what proportion of the unique terms in the relevant documents have a positive discriminatory weight. Low proportions of discriminatory terms, then, indicate sets of relevant documents that differ in their content – the terms they contain are less likely to be contained within the other relevant documents.

Table 10.14 summarises the findings from this analysis. The general trends are similar to the ones found in the previous section in that the number of relevant documents found is an important factor. If the system has found few relevant documents and these are not similar to each other, then concentrating on the original query terms (B3) is often a good technique to use. Similarly when the similarity is high and there are many relevant documents, concentrating on the original query terms is effective as the original queries terms are giving good retrieval results.

However if the system has found a higher number of relevant documents and the similarity is low then some form of query expansion is useful, often a minimal explanation. Larger numbers of relevant documents will give better sets of expansion terms as there is more

evidence in the form of relevant documents. Minimal explanations are often better here as they can eliminate poor expansion terms.

High similarity of documents and low numbers of relevant documents are often best served by a larger query expansion. This will add more terms, increasing retrieval of a wider variety of documents.

Low				High			
	AP	**SJM**	**WSJ**		**AP**	**SJM**	**WSJ**
Porter	B3	Min	B3	**Porter**	Max	B3	Max
F_4	B3	Min	B3	**F_4**	Max	B3	Max
wpq	Min	Max	B3	***wpq***	Max	B3	Min

Table 10.14: Techniques that gave an improvement where the documents were least similar (**left**) and most similar (**right**)
where Min = Josephson/Coverage, Max = B1/B2

10.3.6 Summary

In this section I shall draw together the results from the previous three sections. I shall describe each type of query modification technique in turn and outline under what circumstances each technique performs well.

i. Baseline3 (B3). The B3 technique does not add any terms to the query but instead selects good term and document characteristics for the original query. This technique works well on cases where the expansion terms may be poorer than the existing query terms. This corresponds to two main cases: where the original query terms are very good (e.g. high precision, or high number of relevant documents which show a high similarity), or where the expansion terms may be poor (e.g. low precision). This type of query modification technique works where explanations may be poor relative to the original query. This relates to the discussion in Chapter Eight, section 8.3.4, where I discussed why we may prefer not to generate a new explanation due to the poor evidence available for explanation construction.

ii. Minimal explanations. These techniques add few terms to the query but are more selective about which terms they add to the query. That is, they only add terms that do not explain previously explained relevant documents. These techniques also select good characteristics of query terms. These techniques, then, add little information but pay extra attention on how these terms are weighted. These techniques also work

better where the original query terms may be a good source of evidence but where we want some type of query expansion: they are suitable for relatively poor retrieval situations. In particular they are successful where we have low precision and a poor method of ranking expansion terms, and where we have a large set of dissimilar relevant documents. These types of query modifications are suitable where we want to make a particular change to the query, e.g. to tackle low initial precision.

Although I only use two types of minimal explanations in this chapter the two types of explanations do tend to perform better for different types of retrieval situation. That is, even though there is a high overlap between the queries that are improved by the Coverage and Josephson explanations, there is often a preference for one type of explanation over another. For example, the Coverage explanation tends to work better in cases where there are more relevant documents retrieved, whereas the Josephson explanation works better where there are fewer relevant documents retrieved. Therefore we have the basis to choose explanations based on the type of retrieval situation presented to the system.

iii. Maximal query modification. These techniques add a larger number of terms to the query and, so far, have only used discrimination power to assess the value of a query term. These techniques are best suited to retrieval situations where we want a large change to the content of the query in order to broaden the query (e.g. if we have low precision and a good method of ranking expansion terms). As discussed before, Chapter Eight, section 8.7, it may be the case that we want to make a larger change to the content of the query – using an explanation that adds a large number of terms to the query.

The Relevancy explanation, examined in Chapter Nine, was a type of explanation that added a large number of terms to the query. This type of explanation was not considered for RF as it adds too many terms to the query to be suitable for interactive RF. However the maximal query modification techniques – Baselines 1 and 2 – which add larger numbers of terms to the query can be considered as examples of Relevancy explanations if they explanation formed does explain all the data, i.e. if the terms added to the query explain all the relevant documents. It is possible, therefore to use some form of the Relevancy definition of explanation in cases where we want to add a larger number of terms to the query.

Given that different methods of modifying queries and weighting terms perform better for different queries, it should be possible to *select* for individual queries which query modification technique is most appropriate for each query. The choice of which explanation to use is dependent on part on the features of the query (the precision of the search, the

position of the relevant documents in the document ranking and the similarity of the relevant documents). The choice of explanation also depends on the particular method used to measure the explanatory power of terms, as shown in section 10.3.3 – 10.3.5. In the next section I investigate this proposal.

10.4 Selection of explanations

In this section I report on an experiment that investigates the proposal that it is possible to select query modification techniques. The choice of which type of query modification to use is based on the features outlined in sections 10.3.3 – 10.3.5. For example if we use the F_4 measure to rank terms and the precision is low then the system should use a minimal explanation (such as Coverage or Josephson), whereas if the precision is high then we should use a maximal query modification (such as Baseline 1). Each feature can then be evidence for more than one type of query modification depending on the value of the feature (high or low).

This approach can be used to create a set of decision rules that decide which explanation is most suitable for the current search. The rules change according to which term ranking method is used. The rules for the Porter term weighting function are shown in Figure 10.1.

As can be seen from Figure 10.1 more than one piece of evidence can point to an individual technique. For example, both rules if (**precision** is *low*) and if (**order** is *high*) and (**number of relevant documents** is *low*) both indicate a maximal explanation. Also each feature, e.g. **precision**, can be evidence for more than one type of query modification technique. We therefore require a method of selecting the best modification technique from the decision rules.

This method was implemented as a form of voting procedure; each piece of evidence *votes* for which type of query modification is most suitable. The query modification technique with the most votes is the one chosen to modify the existing query. It may be the case that different pieces of evidence are better at deciding which query modification technique is best, e.g. **precision** may be a better source of evidence than **order**. I do not consider this so far in these experiments but some form of evidence weighting could be accomplished by weighting the rules.

if (term ranking method = *Porter*)
  if (**precision** is *high*) use minimal
    else if (**precision** is *low*) use maximal
  if (**order** is *low*) and (**number of relevant documents** is *high*) use maximal
    else if (**order** is *low*) and (**number of relevant documents** is *low*) use B3
    else if (**order** is *high*) and (**number of relevant documents** is *high*) use minimal
    else if (**order** is *high*) and (**number of relevant documents** is *low*) use maximal
  if (**similarity** is *low*) and (**number of relevant documents** is *high*) use minimal
    else if (**similarity** is *low*) and (**number of relevant documents** is *low*) use B3
    else if (**similarity** is *high*) and (**number of relevant documents** is *low*) use maximal
    else if (**similarity** is *high*) and (**number of relevant documents** is *high*) use B3

Figure 10.1: Rules for selecting query modification technique for the Porter term weighting scheme
where **bold** entries indicate features of the retrieval, *italic* entries indicate values of the features, and underlined entries indicate the query modification techniques suggested by the value of the feature

It may also be the case that no single query modification technique is the *absolute* best, i.e. all votes are split between different techniques. In this case the system will choose a default explanation to use. In the experiments reported in this section the default explanation was chosen to be the best performing explanation type for the collection[96]. I show, in Chapter Twelve, that a better way of handling this case is to get the user to provide more evidence, either in the form of more query terms or marking more documents relevant.

One point that has not yet been addressed is how to decide what constitutes *high* and *low*, e.g. if the similarity is *high* use maximal. One option is to use the values used in the analyses in section 10.3.3 – 10.3.5, e.g. the actual precision values that were used to decide if a query had high or low precision. However, these analyses were based on a fixed set of collections and queries and such values would not be available in real interactive searching on different collections. Hence I decided on a set of default values that can be applied to all collections. These values will be sub-optimal for most collections: they will not be the best values we could obtain for each individual collection. Better values could be determined by, for example, a study of important term statistics for individual collections such as the number of terms per document, the number of unique terms in the collection, etc. The values used are: *high* precision corresponds to precision of over 20%, *high* similarity corresponds to a

[96] For the AP and SJM collections the default explanation type was the Josephson explanation when using Porter and F_4 weighting schemes and Baseline 1 when using *wpq*. For the WSJ collection the default explanation was the Coverage explanation when using Porter's weighting scheme and Baseline 1 when using F_4 or *wpq*.

similarity of over 30%, and *high* order corresponds to most relevant documents appearing in the top half of the ranking of 25 documents, i.e. rank positions 1 – 12.

The results of this experiment are shown in Table 10.15 where I compare the results of against the other query modification techniques presented in this chapter. As before I only concentrate on $n = 25$. In Table 10.15 I present the percentage increase in average precision, after four iterations of feedback, for the three baseline techniques and the Coverage and Josephson explanations (columns 2-7). In column 8 I present the results of selecting query modification techniques (**Selection**) and in column 9 I show if the results of the selection procedure were significantly different to the best performing query modification technique (**sig**).

		B1	**B2**	**B3**	**Cov**	**Jos**	**Selection**	**sig**
AP	**Porter**	-0.89	-8.74	-1.38	2.96	3.07	**7.25**	yes t =-7.29
	F_4	9.91	-3.10	-1.38	2.96	5.06	**10.02**	no t = -0.08
	wpq	**35.10**	32.79	-1.38	2.96	12.36	32.12	no t = 2.23
SJM	**Porter**	-4.44	-8.27	6.49	12.38	**16.07**	9.83	yes t =3.01
	F_4	9.33	3.28	6.49	12.38	**16.37**	15.86	no t = 2.25
	wpq	42.41	36.36	6.49	12.38	16.07	**42.48**	no t = -0.19
WSJ	**Porter**	-1.90	-6.45	-1.14	**3.98**	0.73	2.22	yes t = 3.21
	F_4	**4.70**	-2.72	-1.14	3.98	3.21	3.94	yes t = 5.30
	wpq	**12.98**	-2.72	-1.14	3.98	6.27	9.77	yes t = 6.04

Table 10.15: Results of experiments on selecting query modification techniques each entry indicates the percentage increase over no feedback, **bold** entry indicates the highest performing non-selection technique

Comparing the results of the selection technique against the other techniques tested, it can be seen that the results are not conclusive. The results can be summarised as follows:

• On the AP collection the selection technique gives significantly better results than the best non-selection technique when using Porter's term ranking scheme, better performance than the best non-selection technique using the F_4 scheme, and poorer results when using *wpq*.

• On the SJM collection the selection technique gave better results when using *wpq*, and poorer results when using F_4 or Porter. The difference between the selection and the best non-selection technique was statistically significant when using Porter.

• On the WSJ collection the selection technique performed poorest overall: using any term ranking scheme, the selection technique performed poorer than the best non-selection technique.

Even though the results from this experiment are not conclusive there are several areas which could improve the results, in particular:

i. increasing the number of explanation types. In this experiment only four types of explanation, or query modification techniques, were used. In addition a relatively coarse-grained analysis was used to decide which types of explanation to use for individual retrieval situations. A more sophisticated method for detecting which explanations to use and a wider set of explanation types could give better results.

ii. tailoring rules to collections. A single set of rules was used for each term ranking algorithm regardless of which collection was used, e.g. the same set of rules were used for Porter's term ranking scheme on all collections. I deliberately avoided creating a new set of rules for each collection to test how effective a single set of results could be for all collection. In addition better default values for the rules would help improve performance.

However, from the analyses in section 10.3, the evidence used in the rules, e.g. precision, may indicate different types of explanation for different collections. For example, the low precision may suggest a minimal expansion on one collection and a maximal expansion on another collection. This is shown in Table 10.15 where the results from the SJM collection from are the opposite of the AP collection. That is, on the AP collection the selection technique works well using Porter on and poorly with *wpq* whereas selecting RF techniques works well using *wpq* and poorly with Porter on the SJM collection. Therefore, some kind of tailoring rules to individual collections may be beneficial. Such tailoring could be directed by the statistics of the collections themselves, e.g. number of documents, average length of document or number of unique terms in documents.

iii. weighting rules. A third method of potentially improving performance is to weight the rules according to the quality of the evidence they provide. For example, if we have empirical evidence that precision is the best indicator of what kind of query modification is required then we can weight the precision rules higher than the other rules in deciding how to modify the query.

A final observation is that, although the selection technique did not work better than the best non-selection technique, it did give relatively good performance compared with most of the other non-selection techniques. That is, it was not the best technique but it was still better than most alternatives.

In addition the best technique varied across collections and term ranking schemes. This means that we cannot know, in advance, which technique will perform best for a set of queries and which technique will perform best when users are making relevance assessments. Therefore the selection technique may be a *safer* technique to use because it does not depend on a single method of RF for all queries and relevance assessments.

10.5 Summary

In this chapter I examined three main aspects of the use of explanations: the evidence used for query reformulation, section 10.2, the features of individual queries that may account for the success of individual query reformulation techniques, section 10.3, and the selection of individual explanations for individual searches, section 10.4. In this section I will summarise the main conclusions from this chapter, relating the results obtained to the work presented in earlier chapters.

10.5.1 Evidence used for query reformulation

As discussed throughout Chapter Eight abductive reasoning is heavily dependent on the evidence used to form explanations. In section 10.2 I investigated how dependent the query reformulation techniques I proposed in Chapter Nine were to changes in the relevance evidence available. In particular I examined the *amount* of evidence available (the number of documents used for feedback), the method of assessing *explanatory power* (the term ranking schemes) and which *evidence* was used to form explanations (all relevant documents or only the most recently marked relevant documents).

There were three main conclusions from these experiments. First more relevance evidence was generally better. That is, the more evidence an abductive system has to form explanations the better explanations will be formed. However it is not the case that we required as much evidence as possible. As discussed in section 10.2.1.1, often techniques can be successful with less evidence. The conclusion therefore is good explanations should be based on *sufficient* evidence.

Second, minimal explanations (such as the Josephson and Coverage explanations) are more suited to situations where there is poor evidence upon which to base query modification

decisions. For example, if we have poorer methods of assessing the explanatory power of terms, or few relevant documents, then these minimal methods often perform better than query expansion techniques that give larger changes to the query. This is because the minimal techniques are more *selective* in which terms they add to the query. On the other hand if we have good evidence – good term ranking schemes or lots of relevant documents – then we can use larger explanations, e.g. Baseline1 or Baseline2 which are both examples of Relevancy explanations as long as they explain all the relevant documents.

Third, the choice of which relevant documents to use may be important. As introduced in Chapter Eight, we may want to select which documents to explain. This was based on the fact that, as searchers may refine or change the information they require, the documents that were previously assessed relevant may not be good examples of what the user currently wants. The experiment carried out in section 10.2.3 simulated this by only attempting to explain the relevant documents found in the previous search iteration. The results from this experiment were not conclusive, mostly because the test collections used in this experiment lacked any notion of *change* in a search. However they did indicate that some form of document selection can give increases in retrieval effectiveness over using *all* the relevant documents found. That is, selecting those documents that require explanation could well be a useful stage in creating explanations.

10.5.2 Features of individual queries

In section 10.3 I attempted to derive reasons for why some query reformulation techniques gave better performance than others. In particular this was achieved by analysing the queries for which individual query reformulation techniques gave the highest increase in retrieval effectiveness. This analysis showed that the types of queries for which individual techniques worked well varied. For example, the minimal explanations often worked well for queries that gave low precision and maximal explanations often worked well for queries that gave high precision.

These analyses were used to provide a means of detecting which query reformulation techniques should work well for individual queries. This selection procedure analyses various features of the search (order in which relevant documents have been retrieved, number of relevant documents, similarity of relevant documents) and selects the best query modification technique for the query. This selection procedure corresponds to the abductive principle of choosing the best type of explanation for a search, outlined in Chapter Eight.

10.5.3 Selection of query modification technique

The final investigation presented in this chapter is an experiment to test whether it is possible to select query modification techniques based on the features of retrieval. That is whether we can use behavioural information from *how* users search, as outlined in Chapter Eight, to *choose* how to modify the query. The results indicate that selection of RF techniques, in particular explanations, has the potential to improve retrieval effectiveness but this was not shown conclusively. More investigation and more types of query modification techniques will be needed to investigate this technique more fully. However I have indicated how improvement may be gained through such an investigation.

Chapter Eleven

Summary of the abductive framework for RF

11.1 Introduction

In this chapter I will summarise the overall approach to using abductive reasoning as a modelling tool for relevance feedback. In section 11.2 I will present the main experimental conclusions as they relate to the theoretical discussion from Chapter Eight. In section 11.3 I will discuss how the search for explanations and the best explanation can be represented within existing formal theories of knowledge discovery and reasoning. I will conclude in section 11.4 with an overall summary of Part III.

11.2 Abductive reasoning for RF

In Chapter Eight I presented a framework for RF based on abductive reasoning. The main focus of this framework was to use the set of relevance assessments (relevant documents with information on how the user assessed the documents) to decide how individual queries should be modified. The underlying motivation for using abductive reasoning was that we should form *explanations* of important features of the search, e.g. the precision of the search or the similarity of the relevant documents. An explanation is a description of the relevance assessments that can be used as the basis for a new query. What is important about the use of explanations is that the behavioural evidence given by the user when making relevance assessments directs what *type* of query modification is applied to the user's query.

This process of creating explanations for RF was converted into six stages, Chapter Eight, section 8.5.2. In the remainder of this section I revisit these stages and discuss how these stages relate to the experimental work presented in Chapters Nine and Ten.

i. inference of explanation type. The first stage in creating an explanation is to decide what type of explanation is required. That is, what features of the search are important. The choice of which features are important lead to the choice of which kind of explanation is required, e.g. if the search has low precision we should choose a explanation type that will increase the precision of the search. In Chapter Ten I showed that different types of explanations give

better results when applied to different types of queries and that it was possible to choose explanation types based on user search behaviour.

ii. inference of the relevant document set. This stage chooses which documents are the ones that should be used to form the explanation. As described in Chapter Eight, section 8.5.4, this is based on empirical evidence that users may assess relevance using different criteria at different stages in their search. Hence, some relevance assessments may be more representative of the type of documents a user requires. In Chapter Ten, section 10.2.3, I showed that choosing to explain only a subset of the relevant documents, i.e. only using a subset of relevant documents for RF, could give better results than using all relevant documents. This experiment was carried out using test collections so the hypothesis that selecting which documents require explanation has not been tested in a real search environment. However it did show that simply using *all* relevance information may not always be appropriate for RF.

iii. inference of possible components of explanation. This stage takes the set of hypotheses and returns the set of hypotheses, or components, that could form part of an explanation. In the experiments described in Chapters Nine and Ten, the possible components of explanations were indexing terms and the set of possible components of explanations were those terms that appeared in at least one relevant document. In these experiments I treated explanation as being analogous to retrieval: a term could explain a relevant document if it appeared in a document and could be used to retrieve that document.

iv. inference of *good* components of explanation. This stage takes the output from stage **iii.** (set of indexing terms) and returns the terms with weights on the potential quality of each term in providing a given type of explanation. The weights attached to terms reflect their discriminatory power in explaining the relevant documents. As shown in Chapter Nine, different types of explanations rank the terms differently: Josephson explanations rank terms by discriminatory power, Coverage explanations rank terms by the number of relevant documents in which a term appears, etc. The different methods of ranking terms mean that different terms will be added to each query by individual explanation types.

v. building explanations. This stage constructs explanations according to the definitions outlined in section 8.4.2. In Chapter Eight I outlined several definitions of what constitutes an explanation and in Chapters Nine and Ten I showed that different explanation types could give varying increases in retrieval effectiveness. Further, I showed that different types of explanation could be used for different retrieval situations.

vi. selecting good explanations. This final stage selects and compares good explanations based on the plausibility of their component elements and the type of explanation required (point **i.** above) and returns the optimal explanation. In Chapter Eight I showed that it is often necessary to rely on heuristics to create explanations due to the complexity of finding explanations. Moreover, it is often necessary for real-time applications to accept the first good explanation without searching for better explanations. In the next section I show how existing theories of reasoning can help find explanations.

11.3 Relationship with other theories

One of the main aims of this thesis was to incorporate aspects of *formal* reasoning into the RF process. In this section I examine the relationship between the work presented in this thesis and other formal theories. Specifically I show how the search for explanations and the best explanation can be formalised within mathematical models. In particular I examine Dempster-Shafer's Theory of Evidence, section 11.3.1, the theory of Rough Sets, section 11.3.2, and the use of Expert Systems, section 11.3.3.

11.3.1 Dempster-Shafer's Theory of Evidence

In Chapter Six I outlined how Dempster-Shafer's Theory of Evidence (**DS**) could be used to provide a model of relevance feedback for queries composed of term characteristics. These queries will come from an abductive process of query creation. In this section I show a stronger connection between the abductive process and DS.

As described in Chapter Six, DS theory assigns numerical scores to subsets of a set of elements – the *frame of discernment*. If we take the frame of discernment to be the set of terms within the set of relevant documents, T, then we can assign evidence directly to possible explanations. This means that we do not have to calculate individual plausibility scores for the components of explanations and do not have to compose individual explanations. In this case, Figure 11.1, each subset that receives a score – each *focal element* - is a possible explanation and the score assigned to each focal element reflects how good the focal element is as an explanation. Therefore, from Figure 11.1, we can assert that the set $\{t_1, t_4\}$, is a poorer explanation than the set $\{t_2, t_4\}$, and that the set $\{t_4\}$ is better as an explanation than any set than contains t_4, other than the frame of discernment itself. These scores can also be used to detect the *best* explanation(s): the focal elements(s) with the highest scores are the best explanation(s). Some of the focal elements may only be partial elements: only explain some of the relevant documents. So we may prefer to use a good partial explanation rather than a less good but complete explanation.

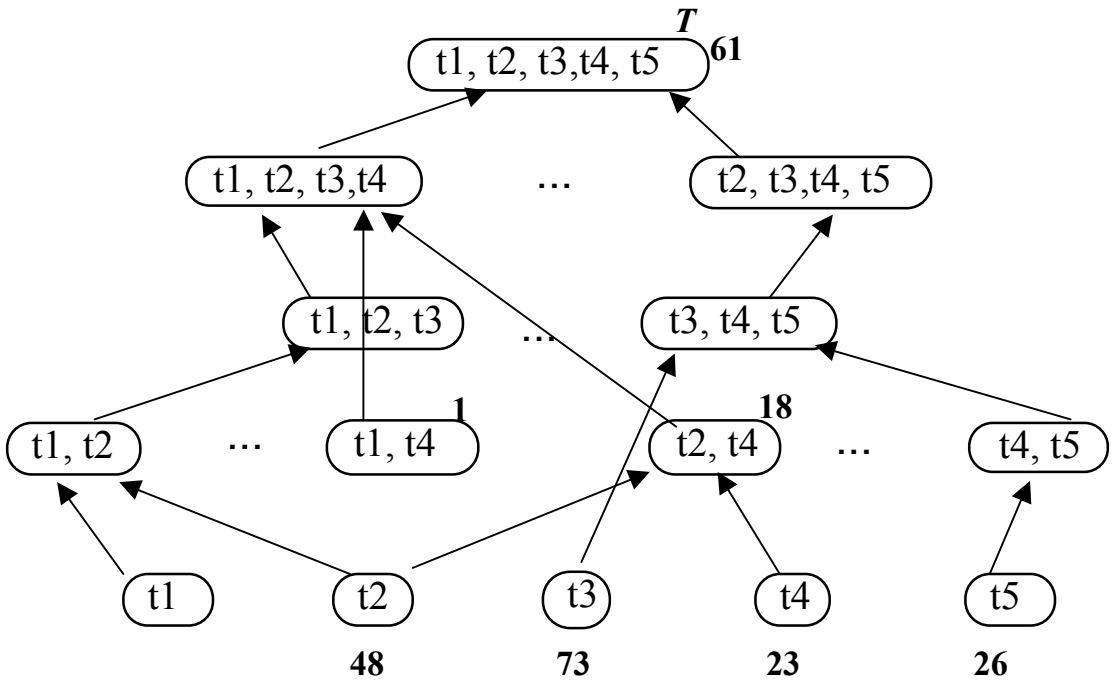

Figure 11.1: Mass distribution over the powerset of *T*

⟶ indicates subsumption

The distribution of evidence over the focal elements of *T*, the *mass* function, can also be used to decide how good an individual explanation is over alternative explanations. As described in Chapter Eight, one of the best indications of how good is a best explanation is the degree to which the best explanation is better than the alternative explanations. The difference in the mass scores of the focal elements can be used to make this assessment: if there are many focal elements and the mass values of the focal elements are similar, then the best explanation is only marginally the best explanation. In such a situation we may want to consider other factors, such as the length of the explanation or the similarity to the existing query, before using the best explanation as the basis of a new query. However, if one focal element has a much higher mass value than other focal elements then we can assert that this focal element is clearly the best explanation. Therefore, our assessment of the quality of an explanation corresponds to a confidence value in the focal element.

The different scoring functions described in Chapter Six, mass, belief and plausibility, can be also be used to help select explanations. For example, mass function will only calculate the exact support for an explanation whereas the belief function will include all support for an explanation from its sub-explanations. That is, if we want to know the total support for an explanation, *E*, then we can include the support for explanations that are subsets of *E*. This function may be useful if we are looking for long, e.g. Relevancy, explanations. The plausibility function will calculate all possible support for an explanation, i.e. will give a measure of how plausible an explanation *could* be. The difference between the belief and

plausibility values may be used to decide when we want to gather more information regarding a possible explanation. That is the difference between the belief in how good an explanation *is* and the *potential* quality (plausibility) of an explanation can serve as an indication of how information we have on the explanation. If this difference is large then we may have too little information upon which to base our decision on the quality of the explanation, and hence we may want to attempt to gather new information.

The mass distribution, here, is formed from the evidence given by the relevant documents. There are many methods by which such a mass distribution may be calculated. In this section I shall sketch one possible approach.

Let us assume that the frame of discernment, *T,* is the set of terms appearing in the relevant documents: explanations will be formed from the set of terms that appear in the documents to be explained[97]. Each relevant document will form a mass function over the subsets of *T* by assigning evidence to each subset. Any subset to which a document assigns evidence is a possible explanation of that document. Each relevant document will assign evidence differently as each relevant document will, in the majority of cases, be indexed by different sets of indexing terms. Therefore Dempster's combination rule will be used to combine the explanations given by the different relevant documents.

Given a relevant document, *d*, it is necessary to decide which focal elements receive a mass score and the value of the mass score itself. We can use the definitions of which kind of explanation is required to decide how to distribute the mass function over *T*. For example if we are interested in generating Josephson explanations we could assign the mass based on the discriminatory power of a set of terms; if we are creating Coverage explanation we could assign mass based on the terms' appearance in relevant documents.

11.3.2 Rough sets

The theory of rough sets (RS) was developed by Pawlak, [Paw82], to deal with vagueness and uncertainty. RS theory has been used in many areas of artificial intelligence and data analysis, e.g. for the discovery of patterns and dependency in data, data reduction, approximate classification of data.

RS theory is based on the assumption that knowledge is defined from our ability to classify objects. This classification is represented by information systems (also referred to as attribute-

[97] The frame of discernment could be composed of larger sets of terms, e.g. all indexing terms in the collection, or smaller sets of terms, e.g. just the discriminatory terms in the relevant documents.

values tables). For RF we can treat the attributes as terms and the value of the attributes as being either the presence of terms in the set of relevant documents, or some discriminatory function such as F_4 or *wpq*. One of the primary components of RS theory is the notion of a *reduct*; a reduct is a set of attributes (terms) from which we cannot remove an attribute without reducing the classification power of the set. This is similar to the notion of explanation introduced in Chapter Eight: like a reduct an explanation is parsimonious in that no element can be removed without reducing the explanatory power of the explanation, i.e. explanations should be complete – explain all the data.

If a set of attributes can classify the data into discrete sets, e.g. split the documents into relevant and non-relevant documents, then the set is said to be *precise,* otherwise the set is said to be *rough*. In RF we do not want precise sets of terms. If a set is precise it will only retrieve the known relevant documents and no others. Hence, the type of set we are interested in for RF are the rough sets. However, we are interested in rough sets that are good at splitting the documents into relevant and non-relevant. RS helps estimate how good a set is through the notion of lower and upper bound approximations which estimate those documents that *can* be classified as relevant using the set (lower bound) and those documents that could *possibly* be classified as relevant using the set (upper bound). The ratio of the lower bound to upper bound – the *accuracy* measure - gives a measure of how vague, or rough, is the set. This measure of roughness, then, can be used to help estimate which sets are good explanations; those that are non-precise, have a high accuracy measure, and form a reduct.

11.3.3 Expert systems

In Chapter Ten I used the analysis of which query reformulation techniques performed well on which queries to derive sets of decision rules. Examples of these rules are shown in Figure 11.2 (this is also Figure 10.1). This rule-based approach to selecting which query modification technique to use can be incorporated into an expert-system like system. In expert systems we can incorporate, formally, aspects of uncertainty in the explanation selection process. For example, we can weight the rules according to their utility in selecting explanations.

With expert systems we can also create more sophisticated methods of using the rules. In Figure 11.2 the rules are additive, i.e. all rules are tested and each provides a conclusion. However, we may not want to test all rules in every case. We may, for example, only want to test particular combinations of rules. Expert systems allow for *modelling* the combination of evidence through rules in a more flexible manner.

if (term ranking method = *Porter*)

if (**precision** is *high*) use minimal

else if (**precision** is *low*) use maximal

if (**order** is *low*) and (**number of relevant documents** is *high*) use maximal

else if (**order** is *low*) and (**number of relevant documents** is *low*) use B3

else if (**order** is *high*) and (**number of relevant documents** is *high*) use minimal

else if (**order** is *high*) and (**number of relevant documents** is *low*) use maximal

if (**similarity** is *low*) and (**number of relevant documents** is *high*) use minimal

else if (**similarity** is *low*) and (**number of relevant documents** is *low*) use B3

else if (**similarity** is *high*) and (**number of relevant documents** is *low*) use maximal

else if (**similarity** is *high*) and (**number of relevant documents** is *high*) use B3

Figure 11.2: Rules for selecting query modification technique for the Porter term weighting scheme

Expert systems have been suggested previously for modelling RF decisions, e.g. [KP94], but the techniques were not implemented or based on empirical evidence.

11.4 Summary

In Part III I examined techniques for creating explanations for RF through the abductive notion of explanation. These explanations choose which query terms to use in modifying a query. Once the new query has been created the best term characteristics of each query are selected and used to score documents. The work presented in Part II and Part III are complementary: Part III outlines *how* the query should be changed – the content of the explanation – and Part II describes how the query should be *used* for retrieval – in what way the explanation explains the relevant documents.

The successful aspects of the work described in Part III were shown to hold for test collections, and so require testing a more realistic searching environment – one where real end-users are making the relevance assessments. In Part IV I describe a series of experiments in which I investigate the performance of this selection technique in such a search environment. Part IV also demonstrates other methods of incorporating behavioural information into the process of explanation.

Part IV

User experiments

Chapter Twelve

User evaluation

12.1 Introduction

The user evaluation presented in this chapter examines three particular aspects of my model of RF. First, I evaluate the effectiveness of the RF algorithms outlined in Chapters Nine and Ten, when relevance assessments are made by individual users, rather than coming from a test collection. Second, I investigate the utility of presenting information to the user about the effect of RF on their search. Both of these aspects have been introduced in previous chapters. The third aspect investigates the incorporation of behavioural information into the term ranking component of explanations. All three investigations are carried out through laboratory experimentation and were particularly motivated by recent research, e.g. [HTP+00], which indicate that techniques that operate successfully using test collections can perform less well than expected when users make their own assessments of relevance.

In the following section I shall give more details about the third investigation which presents a new method of scoring terms based on relevance information. This method of ranking terms will be used throughout the experiments. In sections 12.3 I shall introduce the experiments themselves and give a general outline to the chapter.

12.2 Term ranking and user behaviour

The experiments carried out in this chapter involve real users. This means that it is possible to investigate some aspects of explanations that could not be investigated using test collections. One of the main claims of this thesis is that behavioural information – information on how users make relevance assessments – can help improve retrieval effectiveness. One of the goals of these experiments is then to incorporate behavioural information into the explanation process. In particular I investigate the role of ostension – the varying importance of documents according to when they were assessed relevant and the use of partial relevance assessments.

One method of including this kind of information into the explanation process is by incorporating partial relevance assessments and ostension into the term ranking process; the process by which the system decides which terms are good expansion terms. In the

experiments described in this chapter I investigate this by developing an extension to the standard F_4 term ranking[98] algorithm used throughout this thesis. The extension to F_4 will be called F_4*_po*[99] and the original F_4 algorithm will be referred to as F_4*_standard*.

The F_4*_po* algorithm incorporates information from two sources: partial relevance assessments and ostensive evidence. The weight of a term is composed of two components, one of which calculates the contribution coming from the partial evidence and one which reflects the contribution coming from the ostensive evidence, Equation 12.1.

$$F_4\_po_i = partial_i * ostensive_i$$

Equation 12.1: F_4*_po* term ranking scheme

In the remainder of this section I shall discuss how the two components are calculated,

i. *partial relevance component.* The F_4*_standard* term ranking scheme, Equation 12.2, treats relevance as a binary decision, i.e. all relevance assessments were taken to have a value of 1 (relevant) or 0 (non-relevant).

$$w_i = \log \frac{r_i / (R - r_i)}{(n_i - r_i) / (N - n_i - R + r_i)}$$

Equation 12.2: F_4*_standard* term ranking scheme

In all the experiments described in this chapter the subjects were asked to mark on a scale of 0-10[100] *how* useful a document was to their search. These non-binary assessments can be incorporated into the F_4 term ranking scheme by treating the value assigned to the document as *part* of a relevance assessment. A document that received a value of 10 was treated as a complete relevant document, a document that received a value of 5 was treated as half a relevant document, a document that received a value of 1 was treated as a tenth of a relevant

---

[98] The F_4 algorithm was designed to weight terms by the use of relevance information, i.e. it was used as a term *weighting* function. However it is often used to rank terms for query expansion, as was done in Chapter Nine, i.e. used as a term *ranking* function. As the main interest in this chapter is to investigate how terms should be ranked for query expansion I shall refer to this function, and the others described, as term ranking algorithms.

[99] F_4 *_p(artial)o(stensive)*

[100] 0 was the default value indicating not relevant, values of 1-10 were taken to indicate relevant or useful material. In the experiment *useful* was used instead of *relevant*, Chapter Twelve.

document, and so on. The aim is to test whether partial assessments can give better estimates of term utility than binary assessments.

Table 12.1 outlines the conversion from the binary, F_4 *_standard* weight to the partial, F_4*_po*, weight.

	F4 _*standard*	**F4 _*po***
$\mathbf{r}_i$	number of relevant documents containing term *i*	sum of relevance assessments of documents containing term *i*
R	number of relevant documents	sum of relevance assessments given in search
$\mathbf{n}_i$	number of documents containing term *i*	number of documents containing term *i* multiplied by maximum relevance assessment
N	number of documents in collection	number of documents in collection multiplied by maximum relevance assessment

Table 12.1: Conversion from binary F_4*_standard* to partial F_4*_po*

Table 12.2 gives examples of the difference between F_4*_standard* and F_4*_po*. This example is based on calculating the weight for term *i* which appears in 10 documents, 3 of which have been assessed relevant. The collection contains 100 documents, 7 of which have been assessed relevant. In row 2 the relevance assessments (column 2) are binary, all relevant documents have an equal relevance score. In rows 3 – 5 the relevance assessments for the three relevant documents containing *i* vary. The first case, where are relevant documents containing *i* are give the lowest relevance score (1), gives a negative weight for term *i.* This means, that although *i* may appear in relevant documents it is not useful in retrieving relevant documents. In the final case, where all assessments are equal and the documents all have a maximal relevance score (10), the new weight for *i* is identical to that of the binary case. The second case (row 4) shows the effect of varying relevance assessments.

	Rel ass	r_i	n_i	R	N	Weight
F4 _*standard*	1,1,1	3	10	7	100	2.22
F4_*po*	1,1,1	3	100	70	1000	-0.96
F4_*po*	3,5,7	17	100	70	1000	1.19
F4_*po*	10,10,10	30	100	70	1000	2.22

Table 12.2: Example comparison of binary F4_*standard* to partial F4_*po*

ii. *ostensive evidence component.* Evidence also comes from the *time* a document was marked relevant. In the experiments, although the subjects had a limited time to perform each search (15 minutes, section 12.8), they could run as many searches or feedback iterations as they felt necessary. This allowed me to investigate the potential effect of ostensive evidence: weighting terms according to *when* they indicated relevant material.

Ostensive evidence was incorporated into the term ranking algorithm by a similar means to the partial evidence. The equation used to calculate the ostensive value of the term is shown in Equation 12.3.

$$ostensive_i = \left(\sum_{j=1}^{s} j * r_{ji} \right) \Big/ \max_{ostensive}$$

Equation 12.3: Calculation of ostensive weight

where s = total number of feedback iterations, r_{ji} = number of relevant documents containing term i in iteration j , $\max_{ostensive}$ = maximum possible ostensive evidence

In Equation 12.3 the ostensive weight of term i, is based on a proportion of the ostensive evidence for i relative to the maximum ostensive weight that could be assigned to a term, $\max_{ostensive}$. This maximum ostensive weight will be equal to 1, if all relevant documents, at every iteration of feedback, contained the term i. The ostensive evidence for term i is the sum of the relevant documents containing i multiplied by the iteration in which the documents were marked relevant. Therefore the more relevant documents term i appears in, the higher weight it receives and the more recently-viewed relevant documents i appears in the higher weight it receives. An example of this is shown in Figure 12.1, based on the data given in Table 12.3.

In Table 12.3, we have 5 iterations of feedback. At each iteration a number of documents are marked relevant (row 5), some of which contain term *t*, (row 3), and some of which contain term *q* (row 4).

Iterations of feedback						
	1	**2**	**3**	**4**	**5**	**Total**
$\mathbf{r}_t$	1	0	0	1	5	7
$\boldsymbol{r}_{\mathbf{q}}$	5	1	0	0	1	7
R	5	2	3	1	10	21

Table 12.3: Example ostensive data

$$\text{max\_ostensive} = (5*1) + (2*2) + (3*3) + (1*4) + (10*5) = 72$$
$$r = (1*1) + (1*4) + (5*5) = 30$$
$$q = (5*1) + (1*2) + (5*1) = 12$$
$$\text{ostensive}_t = 30/72 = 0.417$$
$$\text{ostensive}_q = 12/72 = 0.167$$

Figure 12.1: Example ostensive calculation

The value of $\max_{ostensive}$ is identical for both terms: both terms could have appeared in all the relevant documents at all iterations. The incorporation of the ostensive evidence allows the F_4*_po* algorithm to incorporate when the documents containing term *t* or *q* were marked relevant. Even though both terms appear in the same number of relevant documents, term *t* receives a higher score as it appears in more of the documents that were marked relevant in the recent search iterations.

The ostensive evidence is used as a scaling factor. The partial component of the F_4*_po* weight is multiplied by the ostensive weight to give a final weight for each term. Terms are then ranked in decreasing order of this weight to reflect how useful they are at discriminating the user-selected relevant documents.

This new weight, then, incorporates information regarding the uncertainty of the utility of the term at detecting relevant material. This extension to the F_4*_standard* weighting scheme is similar in spirit to the *wpq* weighting scheme, section 1.2.2.2, Equation 1.12. The *wpq* scheme is also composed of two components – the F_4*_standard* weight and the difference between the probability of a term appearing based on relevance and non-relevance information, i.e. how likely a term is to appear in a relevant document. This latter component

– how likely a term is to appear in a relevant document – is analogous to the ostensive aspect of F_4*_po*. The major difference here is the use of weighted ostensive evidence rather than treating all appearances of a term in a relevant document as equally useful. The difference between F_4*_po,* F_4*_standard* and *wpq* will be analysed further in section 12.11.

The new weighting scheme will be investigated in several experiments, described in section 3.10. In the next section I discuss the main features of the experiments described in this chapter.

12.3 Introduction to experiments

These experiments described in this chapter are based on the TREC interactive track model of evaluation, [Ov98]. This model has been iteratively developed over a number of years, with input from many of the leading interactive and evaluation specialists in IR, [BRR96]. Although this approach is specifically designed to allow cross-site investigation of IR systems, it has produced a fairly rigorous experimental framework for evaluating interactive searching.

The type of searching investigated by the interactive track changes each year, resulting in a slightly different experimental methodology being used for each year's experiments. The specific experimental components I used are modified versions of the one used in TREC-6 [LO98, Ov98]. I chose this interactive track for several reasons; this particular track has been extensively evaluated, [LO98, Ov98], the modifications I made upon the search topics have been explored elsewhere, [BI99, Bo00b] (section 12.5), and this version of the track used relatively few experimental subjects, allowing me to carry out a variety of experiments.

In section 12.4 I present the data which was used in the experiments, in section 12.5 I discuss the search tasks that were given to the experimental subjects and in section 12.6 I discuss the experimental procedure that was followed in the experiments. In section 12.7 I describe a pilot test that was carried out to test the experimental methodology and the search topics. In section 12.8 I describe the common experimental methodology that was used in the experiments and in section 12.9 I describe how the results will be analysed. In section 12.10 I outline the five experiments I carried out, the specific research questions I addressed in each experiment and the results obtained. I discuss the overall findings in section 12.11 and provide a summary in section 12.12.

12.4 Data

The interactive track of TREC-6[101] used the Financial Times (**FT**) collection as the sole document collection. This collection consists of full-length newspaper articles from the Financial Times of London published from 1991 – 1994.

INTTREC6 used six topics for the interactive task. I retained five of these topics (topics numbered 303i, 307i, 326i, 322i, 347i[102], Table 12.4 columns 1 and 2). Topic 339i was not used, the reasons for this are discussed in section 12.5.1.

One of the conclusions from INTTREC6 was that the major variable in search success was the topics themselves: searchers across sites and systems found some topics easier to search on than other topics. One possible reason for this was the topics were poorly *covered* within the FT collection: there were few relevant documents to be found by the subjects.

Table 12.4 presents the coverage of the five INTTREC6 search topics used in these experiments. This is based on the number of documents from the FT collection that were assessed as relevant in the ad-hoc task[103], (column 2), compared to the total number of documents assessed relevant in the ad-hoc task, (column 3). Table 12.4, column 4 gives this ratio as a percentage.

Topic number	Relevant FT	Total ad-hoc relevant	%age of relevant documents in FT
303i	6	10	60.00%
307i	81	215	37.67%
322i	9	34	26.47%
326i	45	48	93.75%
347i	50	157	31.85%

Table 12.4: Statistics on topics selected for the user evaluation

From Table 12.4, three of the five topics had less than 40% of the ad-hoc relevant documents in the FT collection, the other two topics had a coverage of 60% of greater.

---

[101] Hereafter shortened to INTTREC6 for convenience

[102] The topic numbers relate to the TREC-6 non-interactive *ad-hoc* track, which uses fifty topics. The INTTREC6 track selected a number of these for interactive searching.

[103] The ad-hoc track used five document collections.

In the experiments I did not want to introduce a bias against some topics that were poorly covered within my document collection. By adding one of the other ad-hoc collections, the Los Angeles Times (**LA**) collection, I increased the coverage of the ad-hoc relevant documents contained within the test collection, Table 12.5. The addition of the LA collection means that four of our five topics have at least 79% of the ad-hoc relevant documents contained within the document collection. The remaining topic still has a coverage of only 38% but this topic has a large number of ad-hoc relevant documents.

As assessed by the ad-hoc task, three of the topics have relatively few ad-hoc relevant documents (303i, 322i and 326i), and two have a relatively large number of ad-hoc relevant documents (307i and 347i). Therefore, although the coverage of the topics was increased, I am not only considering documents with a large number of ad-hoc relevant documents.

Topic Number	Relevant FT/LA	Total ad-hoc relevant	%age of relevant documents in FT/LA
303i	10	10	100.00%
307i	83	215	38.60%
322i	33	34	97.06%
326i	45	48	93.75%
347i	125	157	79.62%

Table 12.5: Statistics on topics selected for user evaluation

The LA collection consists of a sample of approximately 40% of the articles published by this newspaper in the period from January 1989 to December 1990. The combination of the LA and FT collections gives a combined document set of over 340 000 documents (Table 12.6, column 4), which covers the period 1989 – 1994. This cannot be regarded as a set of currently topical documents and subjects would not be able to search using current new events. However, the collection is not out-of-date as regards the search topics given to the subjects (section 12.5).

Stopwords were removed using the stopword list found in [VR79] and the documents were indexed by the algorithms described in Chapter Three.

	FT	LA	Combined
Number of documents	210 158	131 896	342 054
Average document length	412	526	456
Number of unique terms in the collection	245 678	244 874	375 295

Table 12.6: Document collections used in evaluation

12.5 Topics

In this section I discuss the selection of search topics. In section 12.6 I outline the modifications I made to the topics for the experiments and in section 12.7 I discuss the results of a pilot experiment to test the appropriateness of the topics.

The search topics for this evaluation used five of the original INTTREC6 search topics (303i, 307i, 326i, 322i, 347i). I excluded topic number 339i which asked subjects to search for information on '*Alzheimer's drug treatment'*. This decision was made based on previous use of these topics by Borlund and Ingwersen, [BI99], whose experience suggested some searchers may feel uncomfortable searching on this topic.

This topic was replaced by ad-hoc topic number 321, '*Women in Parliaments'*. The choice of this topic was based on two reasons:

i. *Topic appropriateness*. Several of the remaining ad-hoc topics ask subjects to search for information on major diseases, including the topics '*Radio Waves and Brain Cancer'*, '*Poliomyelitis and Post-Polio*', '*Viral Hepatitis'*, '*Agoraphobia'*. These topics were not considered as suitable replacements for the excluded topic as they carried the same risk of upsetting searchers who may suffer from, or have relatives or friends who suffer from, these diseases.

Other possible replacement topics were deemed to be too similar to topics that were already contained within the INTTREC6 set, e.g. '*International Organized Crime'* and '*Industrial Espionage'* were felt to be too similar to '*International Art Crime'*, and '*Endangered Species* (*Mammals*)' was too close to '*Wildlife Extinction'*. Although the specific information that the subjects were asked to search for by these topics is not identical, the overall subject area of these topics was felt to be too similar to existing topics.

Some topics were not considered to be suitable as replacement topics because they were too specialised, e.g. '*Magnetic Levitation-Maglev*', or were considered to be less accessible for

the predominately UK searchers who were used in our experiments, e.g. the topic '*Best Retirement Country*', which was aimed specifically at American retirees.

ii. *Number of relevant documents per topic*. As mentioned in section 12.4, one aspect that I wanted to investigate in the experiments is the possible relationship between number of relevant documents assessed by the subject and the number of *potentially* relevant documents in the collection. By potentially relevant I mean those documents for which we have some external indication that they may contain relevant information.

This external indication takes the form of the relevance assessments obtained in the TREC-6 ad-hoc tasks as an approximate guide. The research question here is whether the number of relevant documents found in the ad-hoc task for a topic can serve as an indication of the ease with which an experimental subject can find relevant documents.

I took the five INTTREC6 topics and found that three had less than fifty relevant documents in the ad-hoc searching task. The other two tasks had over 150 relevant documents found in the ad-hoc task. I then examined the remaining topics for a topic that had a relatively large number of relevant documents. The basis behind this is that it is possible to examine number of relevant documents against search success.

Topic 321 '*Women in parliament'* had 234 relevant documents in the ad-hoc task, of which 133 were contained with the FT/LA collection, giving a coverage of 56%. As this topic appeared to be relatively neutral, was easily understandable and did not contain any emotive concepts I selected this topic for inclusion in my test set. Table 12.7 gives a summary of the topics I used in these experiments.

In the next section I discuss how these topic were converted into the search tasks given to the experimental subjects.

Topic Number	Topic Title	Relevant FT/LA	Total ad-hoc relevant	%age of relevant documents in FT/LA
303i	*Hubble telescope achievements*	10	10	100.00%
307i	*New hydroelectric projects*	83	215	38.60%
321	*Women in parliament*	133	234	56.84%
322i	*International art crime*	33	34	97.06%
326i	*Ferry sinkings*	45	48	93.75%
347i	*Wildlife extinctions*	125	157	79.62%

Table 12.7: Statistics on topics selected for user evaluation

12.6 Conversion of topics into search tasks

The INTTREC6 search topics contained detailed descriptions of what information searchers should attempt to retrieve and what constitutes a relevant document, Figure 12.2, [Ov98].

Number: 326i

Title: Ferry Sinkings

Description:

Any report of a ferry sinking where 100 or more people lost their lives.

Narrative:

To be relevant, a document must identify a ferry that has sunk causing the death of 100 or more humans. It must identify the ferry by name or place where the sinking occurred. Details of the cause of the sinking would be helpful but are not necessary to be relevant. A reference to a ferry sinking without the number of deaths would not be relevant.

Aspects:

Please save at least on RELEVANT document that identifies EACH DIFFERENT ferry sinking of the sort described above. If one document discusses several such sinkings, then you need not save other documents that repeat those aspects, since your goal is to identify different sinkings of the sort described above.

Figure 12.2: Interactive topic 326i

In INTTREC6 search topics were designed for a specific search task: *aspectual recall*. Aspectual recall is intended to measure how many different aspects of the topic could be found by the searchers. An aspect is defined as one of the possible answers to the question

posed by the topic, [Ov98]. Aspectual recall, then, is related to the *breadth* of the search: a search which provides documents on several ferry sinkings, regardless of how detailed the discussion is, would be assessed as better than a search which provides very detailed descriptions of fewer sinkings.

In this evaluation I did not want to place such a qualitative restriction on the searches. Instead I wanted to encourage more naturalistic search behaviour by our subjects. That is, I wanted our subjects to interact with the system as though they were performing their own search. Consequently, I modified the descriptions of the information needs, placing them within *simulated situations* as proposed in [BI99, Bo00a, Bo00b]. This technique, developed by Borlund, [Bo00a], asserts that searchers should be given search scenarios that reflect and promote a real information seeking situation. The simulated situations, such as the one shown in Figure 12.3, [Bo00b], are intended to achieve two main objectives. First, they are aimed at promoting a simulated information need in a subject. That is, the simulated situation should engage the subjects in the search by the identification of the searcher with the situation.

Second, the simulated situations position the search within a realistic context. This allows the experimental subject to provide his or her own interpretation of what information is required and allows the subject to develop the information need naturally. Unlike the topic description given in Figure 12.2, which asserts a particular definition of relevance, the simulated situations permit a dynamic interpretation of relevance on the part of the subject.

Simulated situation

Simulated work task situation: After your graduation you will be looking for a job in industry. You want information to help you focus your future job seeking. You know it pays to know the market. You would like to find some information about employment patterns in industry and what kind of qualifications employers will be looking for from future employees.

Indicative request: Find for instance something about future employment trends in industry, i.e. areas of growth and decline.

Figure 12.3: Simulated situation taken from [Bo00b]

Simulated situations can be composed of two parts: the simulated work task situation and an indicative request. The simulated work task situation is a short 'cover-story' designed to provide context for a search. The indicative request is an indication, rather than an instruction, of how a search may be initiated. The results from [Bo00b] show that the indicative request is not required for the simulated situation to engage the subject in the search and to promote natural searching behaviour on the part of the subject.

I decided to omit the indicative request from my simulated situations as several subjects in [Bo00b] reported using the indicative request, e.g. to select search terms or to assess relevance. The subjects, then, may be using information from the indicative request to which they would not normally have access to when searching.

Of the six INTTREC6 topics, Borlund used four in her experiments (topics number 303i, 326i, 322i, 347i), [Bo00b]. Topic 303i was used as a training topic and a new topic was developed. Borlund's simulated situations were modified heavily resulting in some situations which were very different from the original INTTREC6 topics. As I was interested in retaining the connection with the INTTREC-6 topics I did not use Borlund's situations and instead developed a new set. The simulated situations I developed, the original INTTREC6 topics and Borlund's versions are given in Appendix H.

One particular aspect of the design of simulated situations that is important is the degree of *semantic openness*, [Bo00b]. The simulated situation should allow the subjects to adapt, develop and use their own interpretations of what constitutes relevance. That is, the simulated situation should reflect the dynamic and personal nature of making relevance assessments. Semantic openness measures how well the simulated situation achieves this aim. A simulated situation that has *broad* semantic openness allows a greater degree of interpretation than a simulated situation with *narrow* semantic openness. Good simulated situations are those that have a broader semantic openness.

The semantic openness can be narrowed by the use of information that makes simulated situation specific to a person, place, or situation. For example, the simulated situation in Figure 12.3 could be narrowed by making the situation specific to the computing industry. This will narrow the semantic openness if the experimental subjects are not looking for a job in computing or do not have a computing background. Borlund investigated the creation of simulated information needs and proposed two techniques for increasing the semantic openness of the situations: *tailoring* the simulated situation, and by how *topical* the situation is to the subjects, [Bo00b].

Tailoring reflects the degree to which the simulated situations have been adapted to be a realistic scenario for the group of experimental subjects. The simulated situation shown in Figure 12.3 is an example of a situation that may be very relevant to the group of university students used as subjects in [Bo00b]. This situation remains semantically open because the subject has freedom to decide what industry is relevant and what is mean by employment patterns and qualifications.

The topical relevance criterion is related to the *topic* of the simulated situation – what is being searched for. Good simulated situations should be centred around a topic that is of interest to the subject group. The example in Figure 12.3 shows high topical relevance to university students. Topically relevant simulated situations are more likely to engage the subject in a search, and thus promoting naturalistic searching.

In Appendix H, I discuss, for each simulated situation, its semantic openness, tailoring and topical relevance. Table 12.8 summarises these aspects of the situations.

Topic number	Topic title	Semantic openness	Tailoring	Topical relevance
303i	*Hubble telescope achievements*	Narrow	None	None
307i	*New hydroelectric projects*	Fairly narrow	Some	Some
321i	*Women in parliaments*	Fairly broad	Some	High
322i	*International art crime*	Fairly broad	None	None
326i	*Ferry sinkings*	Fairly narrow	None	None
347i	*Wildlife extinctions*	Fairly narrow	None	None

Table 12.8: Semantic openness of simulated situations

The semantic openness for these topics was relatively low overall. This could cause the search topics to be less useful in stimulating realistic search behaviour on the part of the experimental subjects. Consequently I tested the search topics in a pilot test, described in section 12.7.

12.7 Pilot test

A pilot test was carried out prior to the main experiments in this chapter. The pilot test was designed to test questionnaires, elicit any system alterations that were necessary and to debug the experimental procedure. Minor changes were made to various aspects of the system and questionnaires as a result of the pilot test.

A more important aspect of the pilot test was to investigate the suitability of the search topics. Based on [Bo00b] and [Bo01] the most important factor in a good simulated situation was the

degree to which the topic engaged the subject's interest. I was keen to test the suitability of the INTTREC6-based topics in this respect.

As most of the potential subjects for the experiments were likely to be university students, I create a separate set of six simulated situations aimed specifically at students. These situations covered topics such as graduate employment, shared housing and exam marking, which were felt to be more pertinent to student subjects. These topics are presented in Appendix H. The pilot test was used to compare subjects reactions to the INTTREC6-based topics and the student-specific ones. The pilot test was carried out according to the experimental methodology described in section 12.8. One searcher was given only the INTTREC6 topics, one searcher was only given the new topics and the remaining four searchers were given three of each type of topic.

The results of the questionnaires and subject search logs and post-experiment discussion were used to gauge subject reaction to the two sets of topics. In particular I examined two sources of evidence: the subject's interest in the search topics and the subject's searching behaviour.

i. subject's interest in search topics. The subject was asked, after each search, to assign a score to each of the following questions: '*Was the search task realistic?*', '*How interested were you in the topic of the search task?*', and '*How enjoyable was this search?*'. The scores for each question ranged from 1 (*not at all*) to 5 (*extremely*).

This analysis was intended to elicit the degree to which the topics engaged the subjects' interest. The INTTREC6 topics scored lower on the question relating to the realism of the search tasks, (3.72 vs 3.94 for the student topics[104]). However the difference was not significant using a paired *t*-test (t = -0.7). The INTTREC6 topics scored lower on the question relating to the subjects interest in the search topic (3.3 vs 3.5 for the student topics). Again the difference between the two sets of results was not significant (t = -0.75). Finally, the INTTREC6 topics scored slightly higher on the question relating to the subjects' enjoyment of the search task (3.4 vs 3.3 for the student topics), although this difference was not significant (t = 0.25). None of the topics (INTTREC6 or student topics) scored noticeably lower than other topics across the questions, i.e. some topics scored lower for one question but higher on others. The results indicate that there was no major difference regarding the searchers' perceptions of the two sets of topics.

[104] These figures are averaged over all responses.

ii. subjects' searching behaviour. In this analysis I compared how the searchers interacted with the topics in terms of how many searches they ran per topic, how many documents they viewed per topic and how many relevant documents they found per topic. The intention is to discover whether the searchers searched differently when using the INTTREC6 topics or the student topics.

The subjects tended to run fewer searches on average for the INTTREC6 topics (4.2 for the INTTREC6 topics compared to 5 searches per topic for the student topics). They found slightly fewer relevant documents when using the student topics (9.22 relevant documents found per INTTREC6 topic against 8 relevant documents per student topic) and viewed fewer documents per topic with the student topics (INTTREC6 28.25 documents viewed per topic against 22 documents viewed per topic with the student topics). However the latter two differences are only of interest across the whole topic; the subjects examined the same number of documents and found the same number of relevant documents per *search* iteration. I conclude from this analysis that there was no real difference between the two sets of topics as regards the subjects' search behaviour.

The main goal is to find topics that are interesting to the potential subjects. The topicality and tailoring can increase our confidence that a topic will be interesting to a group of subjects, but, as also reported in [Bo00b], it is sometimes surprising which topics subjects will find of interest. For instance, in [Bo00b] Borlund found that topic number 303i, *'Positive achievements of the Hubble Telescope*' was unexpectedly popular. I also found that this topic was consistently rated highly by the subjects across the topics, section 12.11.2.

The major factor in the success of the simulated information needs seemed to be the subjects themselves: some subjects were willing to place themselves within a simulated search scenario, make subjective and dynamic relevance decisions and discuss coherently the kind of interpretations they made about the documents they discovered. Other subjects performed searches, made relevance assessments and, on examination of the search logs, seemed to have made as many search decisions. However, in the post-search interview these subjects claimed not to have found any tasks interesting or been willing to treat the simulated situation as a personal construct.

As there was no major preference for one set of topics, or a mixture of the two sets, I choose to use the INTTREC6 topics as this allowed more analysis regarding the topics, section 12.11.2.

12.8 Experimental methodology

In this section I describe the experimental procedure I followed for these experiments. The same methodology was used for each of the five experiments described in section 12.10. The only differences between the experiments are the control and experimental systems used in each experiment, and the subjects used in each experiment[105].

Each subject was asked to perform a search on each of the simulated information needs, three searches on the control system and three on the experimental system. The order in which topics were presented, and the choice of which system a subject used for each search, was determined by a randomised experimental matrix.

The INTTREC6 experiments used only four experimental subjects and the matrix, Figure 12.4, rotated the order in which systems were used to avoid possible system bias. The order of the systems were not interleaved to make the experiments smoother to run, and the order of search topic was not randomised across subjects.

	Topic					
Subject	**325i**	**322i**	**307i**	**347i**	**303i**	**339i**
1	E	E	E	C	C	C
2	C	C	C	E	E	E
3	E	E	E	C	C	C
4	C	C	C	E	E	E

Figure 12.4: INTTREC6 experimental matrix from [Ov98]
where C = Control system, E = Experimental system

In the experiments described in this chapter I used six experimental subjects per experiment. The matrix used in the experiments described here, Figure 12.5, randomises order of topics, distribution of topics across systems and order of systems. This is due to empirical evidence from [Bo00b] that the order in which the topics are given does affect the subject's search behaviour.

This number of subjects does not allow a complete randomisation of subject, system and topic so I have concentrated on randomisation of order in which subjects were presented topics and system. The same matrix was used for all experiments.

[105] No subject could take part in more than one experiment.

Subject	Topic	Topic	Topic	Topic	Topic	Topic
1	**303i**	*321*	**326i**	*307i*	**322i**	*347i*
2	*307i*	**322i**	*347i*	**321**	*326i*	**303i**
3	**307i**	*347i*	**326i**	*321*	**303i**	*322i*
4	*322i*	**307i**	*321*	**347i**	*303i*	**326i**
5	*326i*	**321**	*303i*	**322i**	*307i*	**347i**
6	**347i**	*322i*	**307i**	*326i*	**321**	*303i*

Figure 12.5: Experimental matrix

where **bold** figures = topics to be run on the experimental system,
italic figures = topics to be run on the control system

For each experiment the following steps were followed:

i. the subject was welcomed and was asked to read the short introduction to the experiments, (Appendix H). This set of instructions was written to ensure that each subject received precisely the same information.

ii. the subject was asked to complete the introductory search questionnaire (Appendix H). This contained general background information on the subjects' education, previous search experience and computer experience.

iii. the subject was given a tutorial on the search system, followed by a training topic. The training topic was the one given in the welcome sheet (Appendix H)

iv. the subject was given one of the simulated situations (Appendix H), and asked to answer a pre-search question to elicit information on how much the subject already knew about the topic (Appendix H).

v. after completion of the pre-search question, the subject was asked to perform the search and was given 15 minutes to search. Subjects could terminate a search early if they were unable to find any more relevant documents.

vi. after completion of the search, the subject was asked to complete the post-search questionnaire (Appendix H).

vii. The remaining topics were given to the subject, following steps iv. – vi. Subjects were offered a break after the third topic.

viii. at the end of the experiment, the subject was asked to complete the post-experiment questionnaire (Appendix H) and a post-experiment interview was conducted.

The post-search and post-experiment questionnaires varied according to the research questions that lay behind the experiment. All questionnaires are contained within Appendix H.

The experimental subjects themselves were students in the Computing Science Department at the University of Glasgow. Half of the subjects were undergraduate computing students, half were students on the Masters in Information Technology course. These latter students had previous degrees in a non-computing discipline. Thirty students took part in the experiments[106]; 9 of the subjects were female, 21 male, and their average age was 23.

The subjects had relatively high experience of on-line searching (average 4.28 years) which was mostly gained through library search facilities and web search engines. The subjects reported good experience on these two forms of IR system but little experience of any other search system. The subjects were also relatively frequent searchers searching daily or at least weekly. All had good previous experience of point-and-click interfaces such as the ones used in these experiments.

12.9 Analysis

For each experiment I shall analyse the results under three main headings. The first examines the subjects' overall search *behaviour*; this analysis looks for changes in how subjects searched on the control and experimental system. The second examines the search *effectiveness* of the two systems: did the subjects have a more effective search on the control or experimental system? Finally I shall examine the subjects' perceptions of the two systems: did the subjects prefer one system over the other? Where appropriate I shall also examine differences before and after feedback to isolate the effect of the feedback techniques on the search.

The results from the experiments will be assessed according to two types of criteria: criteria that are *generic* to all experiments, and criteria that are *specific* to the individual experiments. The specific criteria will examine aspects of searching that investigate the particular research question being address in each experiment. The generic criteria include qualitative data from the questionnaires and analyses of the search logs. Examples of the criteria used to compare the control and experimental systems include:

[106] Not including the pilot tests.

i. *number of relevant documents found*. Of the documents the subject viewed, how many did they consider to be relevant to their search.

ii. *degree of relevance*. Of the documents marked relevant by the subject, how highly did the subjects rate the documents' relevance.

iii. *degree of satisfaction with the search*. How satisfied were the subjects with the results of their search.

iv. *which topics were more/less successful*? Was there a difference between search success regarding the different simulated situations?

Where appropriate tests for statistical significance will be used. Specifically I will use a paired *t*-test for related samples, comparing subject aggregate performance on each topic using the control and experimental system.

12.10 Experiments

In the following sections I outline five experiments. For each experiment I describe the research question I addressed, the systems and interfaces I used[107] and the results obtained.

Each experiment involves two combinations of interface and system. For convenience of exposition, in each experiment I label one combination of algorithm and interface as the *control* system and the other combination as the *experimental* system. Table 12.9 summarises the five experiments according to the ranking algorithm used to rank expansion terms, the method by which the query was expanded and the interface used for the control and experimental systems.

Experiment	Term ranking algorithm	Query expansion technique	Interface	Term ranking algorithm	Query expansion technique	Interface
One	F_4_*standard*	Josephson	Two	F_4_*po*	Josephson	Two
Two	F_4_*standard*	Interactive	Three	F_4_*po*	Interactive	Three
Three	None	None	One	F_4_*po*	Josephson	Two
Four	F_4_*standard*	Top 6 terms	Two	F_4_*po*	Selection	Two
Five	F_4_*po*	Selection	Two	F_4_*po*	Selection	Four

Table 12.9: Summary of experiments

[107] The algorithms are described in Chapter Nine and Ten, and the interfaces are described in Chapter Twelve.

The experiments examine five basic research questions which I shall outline here; a more detailed introduction will be given in the description of each individual experiment.

Experiment One compares the performance of two term ranking algorithms: F_4*_standard* and F_4*_po*. Specifically I examine how good the two algorithms are at ranking terms for creating an explanation. The research issue here is whether the additional information used by the F_4*_po* algorithm, partial and ostensive evidence, leads to better retrieval.

Experiment Two compares the two term ranking algorithms as a means of suggesting terms for interactive query expansion. The research question here is which set of possible expansion terms the experimental subjects prefer for query modification.

Experiment Three compares the F_4*_po* algorithm and Josephson explanation against no feedback. The research question is whether abductive RF techniques can modify the query better than the experimental subject.

Experiment Four compares the technique of selecting explanation types against one single method of RF. This tests the selection technique, Chapter 10, section 10.4, when real searchers are making the relevance assessments.

Finally, in Experiment Five I examine the role of explanation at the interface: examining whether presenting the subject with information on how their query was changed will help the subject use RF more effectively.

12.10.1 Experiment One

All the explanations described in Part III *rank* possible expansion terms before creating an explanation. The intention behind the ranking of terms is to place terms that will be good as a component of an explanation at the top of the term ranking. In this experiment I compared two methods of ranking terms; the first method is the F_4*_standard* term ranking scheme that has been used throughout this these. The second term ranking scheme is the extension of the F_4*_standard* weighting scheme that incorporates ostensive and partial evidence, F_4*_po*.

Both term ranking schemes are used to provide a set of possible components for an explanation. I compare the performance of the two weighting schemes at providing components for a Josephson type of explanation. This type of explanation emphasises the *discriminatory* power of a term, so the main research question is whether the inclusion of

evidence on the subject's involvement in RF (F_4*_po*) causes any change in overall retrieval effectiveness. The system that uses the original F_4 weights, F_4*_standard*, to create explanations is the control system in this experiment and will be referred to as *Ab_standard*[108] for convenience. The control system is therefore the same RF technique as described in Chapter Nine. The system that uses the new version of F_4*_po* is the experimental system and will be referred to as *Ab_po*[109]. Once the system has chosen the new expansion terms the system then selects the best characteristics for each of the new query terms.

Both experimental and control systems use the same interface, Interface Two and only the underlying RF algorithm varied between the two systems.

Two additional features were added to the systems:

i. *timing control*. The performance of a RF iteration generally takes longer than an initial search. This is because the system has to calculate a list of expansion terms and select the best characteristics of these terms. Although these steps are performed in real-time, they can, depending on the features of the individual query, take longer than simply performing a new search. To avoid any noticeable time delay between RF and a new search, which could lead the subject to avoid RF, it was decided to artificially ensure that the RF and new search options took approximately the same time to complete. For each new search (after the initial search) the system would perform the same procedures as for an RF iteration: calculate a list of expansion terms based on the current set of relevant documents, choose a number of terms to add to the query and select characteristics of these terms. However, for a new search, the query itself was not actually modified: the RF procedures were executed simply to ensure that a new search would take as long as an RF iteration.

ii. *suppression of viewed documents*. RF aims to retrieve documents similar to the ones marked relevant by the subject. As such, the marked relevant documents will typically appear at the top of the new document ranking; the one obtained after running the modified query. This means that the subject is presented first with documents that they have already viewed and assessed rather than *new* documents. A common technique to avoid this problem is to only show those documents that the subject had not yet viewed. In both control and experimental systems I applied this

---

[108] Ab(ductive explanations)_standard(version of F_4).

[109] Ab(ductive explanations)_po(*p*artial and *o*stensive evidence version of F_4).

technique for document rankings obtained through RF. If the subject requested a new search, no documents were suppressed from the ranking.

12.10.1.1 Results from Experiment One

12.10.1.1.1 Overall search behaviour

In this section, I shall discuss the overall search behaviour of the experimental subjects. The subjects carried out a total of 49 new searches and 36 feedback iterations on the control system compared to 52 new searches and 25 feedback iterations on the experimental system. Neither the difference between new search iterations, feedback iterations nor combined feedback and new search iterations on both systems was found to be statistically significant, (t = -0.28, t = 0.86, t = 0.59 respectively). The difference between the number of feedback iterations and new search iterations on the same system was not found to be statistically significant (t = 1.83 control system, t = 1.93 experimental system).

Overall the subjects viewed more documents on the control system (549 total, 6.45 documents per search iteration) than on the experimental system (443 total, 5.75 documents per search iteration). Subjects also viewed the same documents slightly more often on the control system: of the documents viewed on the control system, 22% were viewed more than once, on the experimental system around 23% were viewed more than once. Neither the difference between documents viewed, the documents viewed once, documents viewed per search, nor documents viewed once per search was found to be statistically significant (t = 1.08, t = 1.14, t = 0.82, t = 0.28).

Overall, although there are more search iterations on the control system, the results indicate that the subjects did not interact differently with the two systems. That is, they did not submit a significantly different number of searches, neither did they perform a significantly different number of feedback iterations, they viewed roughly the same number of documents, and viewed approximately the same proportion of documents more than once. I shall now discuss differences in interaction before and after feedback, i.e. is there a difference in search behaviour after a new query and after feedback?

12.10.1.1.2 Search behaviour before and after feedback

Subjects viewed a similar proportion of documents before and after feedback on both systems (68%/32% before and after feedback on the control system, 67%/33% on the experimental system). The difference between documents viewed per new search (before feedback) was not significant (t = 1.1), neither was the number of documents viewed after feedback (t = 0.57). Comparing the number of documents found per search before and after feedback, neither case was found to be significant (t = 0.56 before feedback, t = 0.16 after).

Subjects also found a similar percentage of *relevant* documents before and after feedback (77%/23% before and after feedback on control, 80%/20% on experimental). As before, none of these differences are statistically significant (t = 1.03 before feedback all searches, t = 1.35 after feedback all searches, t = 0.71 before feedback measured as relevant documents found per search, t = –1.96 after feedback measured as relevant documents found per search). Although the last value – measuring the difference between relevant documents found after feedback per search iteration - is not significant it does lend some support to the experimental system helping to find relevant material.

The major conclusion is that subjects were not interacting in a noticeably different manner on the control and experimental systems before feedback or after feedback. In the next section I shall look at the *effectiveness* of the two search systems.

12.10.1.1.3 Search effectiveness

The overall precision (relevant documents found per documents retrieved) was lower on the experimental system (12.66% against 9.83%) (t = 1.20 – no statistical difference), as was the precision of *viewed* documents (52.15% versus 49.05%) (t = 0.46 – no statistical difference).

Precision before feedback (new searches only) gave similar values (61.55% control, 60.33% experimental). Precision after feedback (feedback iterations only) gave a difference (30.03% control, 18.06% experimental) with the control system seeming to perform better – retrieving more relevant documents per viewed document. However, again, there was no significant difference (t = 0.18, t = 0.97 before and after feedback respectively).

Table 12.10 gives the average precision for each of the topics (relevant documents per documents viewed). For more of the topics (topics 307i, 322i, 326i and 347i) the control system gave a higher precision value. Table 12.11 shows a higher precision after feedback although these are based on small numbers of values. On both systems there were two topics for which no relevant documents were found after feedback.

Condition	303i	307i	321	322i	326i	347i
Control	40.58%	**43.75%**	50.00%	**48.48%**	**61.62%**	**68.48%**
Experimental	**44.44%**	38.96%	**53.57%**	43.33%	59.72%	48.59%

Table 12.10: Results of documents relevant per viewed
bold figures indicate highest value.

Condition	303i	307i	321	322i	326i	347i
Control	**27.27%**	0.00%	0.00%	**41.18%**	**78.38%**	**33.33%**
Experimental	5.88%	**37.50%**	**11.76%**	0.00%	53.19%	0.00%

Table 12.11: Results of documents relevant per viewed after feedback
bold figures indicate highest value.

These values would appear to indicate a favour for the control, *Ab_standard*, (non partial, non ostensive) system in terms of search success. However the subjects' perceptions of the terms suggested by the system varied. I shall discuss this in the next section.

12.10.1.1.4 Subjects perceptions

In the post-search questionnaire (Appendix H) the subjects were asked how useful the terms added by the system were to their search. This was on a 5-point scale, rated from 1 (*Not at all* (useful)) to 5 (*Extremely* (useful)). The average response when the subjects rated the terms suggested by the control system was 1.67 compared with 2.44 when the subjects used the experimental system. This value was found to be statistically significant (t = -2.80).

The subjects also informally, whilst searching, remarked on the more obvious nature of the F_4*_po* term suggestions. An example of the type of terms added by F_4*_standard* and F_4*_po* systems is shown in Figure 12.6. This example is drawn from a real search, chosen at random. The subject submitted the query '*hubble space telescope*' and marked four documents relevant at the first iteration. Figure 12.6 shows the top ten terms ranked by F_4*_standard* and F_4*_po*.

The F_4*_standard* algorithm selected terms that are less usual in the collection (*accrete*, *chaisson*) whereas the F_4*_po* algorithm selected variants of existing terms (*telescopes*), and more obvious terms (*orbit*, *nasa*, *earth*). The F_4*_po* algorithm also returned the original query terms higher up than F_4*_standard*.

A further analysis was used to uncover how the expansion terms were actually treated by the subject: were the expansion terms often retained or removed by the subject. One justification for this kind of analysis is that subjects may be put off RF because the suggested terms do not appear useful, e.g. [RTJ01]. Consequently they may lose out on the potential benefits from RF. On the other hand, terms that appear useful to the search, even if they do not actually improve the precision of the search, may encourage subjects to interact more with the system,

for example by suggesting more query terms themselves. The results of this analysis are summarised in Table 12.12.

F_4_*standard*	**F_4_*po***
accrete	astronomer
chaisson	hubble
cullers	telescope
goldreich	universe
sandpile	astronomers
terrile	telescopes
borucki	scientists
machtley	orbit
nebula	nasa
astronomer	earth

Figure 12.6: Sample terms selected by F_4_*standard* and F_4_*po*

In Table 12.12 I present a count of how many terms per search iteration were used in the original query specification (row 2). In rows 3 and 4 I show the source of query terms that were added after the initial query: either added by the subject (row 3) or the system through RF (row 4). Finally I show how many of the terms the subject added were removed later by the subject (row 5) and how many of the terms added by the system were removed by the subject (row 6).

	Ab_standard	***Ab_po***	**Significant**
Original query terms	3.06	**3.22**	
Source of new terms			
subject	2.00	**2.33**	no, t = -0.36
system	**3.33**	1.11	yes, t = 3.78
Source of removed terms			
subject	0.72	**1.17**	no, t = -1.16
system	**2.28**	0.67	yes, t = 2.54

Table 12.12: Summary of query term addition and removal per topic
bold figures indicate highest value

Comparing the two version of the abductive system, Table 12.12 shows slightly longer initial queries for the experimental system (3.22 per search versus 3.06 per search on the control system, not significant t = -0.34). The subject added more of their own terms per search with the experimental system (2.33 experimental versus 2.00 control, not significant t = -0.36).

The system added more terms with the *Ab_standard* than the *Ab_po* algorithm per feedback iteration (1.11 experimental versus 3.33 control, significant t = 3.78). The main reason for this is that *Ab_po* emphasises the original query terms more than the *Ab_standard* algorithm, and is less likely to perform query expansion.

The subjects removed 36% of their own terms and 68% of the terms suggested by the system when using the *Ab_standard* system compared to 50% of their own terms and 60% of the system suggested terms with the *Ab_po* system. This suggests that subjects, on both systems, felt their own query terms were better, or more likely to retrieve relevant material.

The difference between the number of the subject's own terms removed was not significant (0.72 per search control system, 1.17 experimental, t = -1.16). However the difference between the number of *system* suggested terms removed was significant (2.28 search terms removed per search, 0.67 per search experimental, t = 2.54). This latter finding suggests that the terms suggested by the *Ab_po* system were felt to be better search terms by the subject.

Although the *Ab_po* system did not improve more queries or give better overall results, it was seen by the subjects as a better term suggestion technique. It led to increased satisfaction with the feedback process and subjects appeared to trust the systems suggestions more often. The next experiment tests the effectiveness of the two term ranking schemes when the subject is selecting new query terms – Interactive Query Expansion.

12.10.2 Experiment Two

The second experiment compared the effectiveness of the F_4*_standard* and F_4*_po* term ranking schemes in suggesting new expansion terms for selection by the subject. In this experiment the control system used the F_4*_standard* algorithm to suggest 20 possible expansion terms and the experimental system used the F_4*_po* algorithm to suggest expansion terms. Both control and experimental systems used the same interface (Interface One), the only difference between the two systems was the underlying term suggestion technique. As there was no automatic RF function in this experiment, the previously viewed documents were not suppressed: all searches were new searches.

12.10.2.1 Results from Experiment Two

12.10.2.1.1 Overall search behaviour

In Table 12.13 I summarise the overall search behaviour of the searchers on the topics. With the exception of the number of search iteration per topic, all values are for individual searches (rather for a topic as a whole).

	Control	Experimental
Search iterations per topic	**4.22**	4.17
Documents viewed	9.85	**10.65**
Unique documents viewed	6.75	**7.05**
Unique documents retrieved	15.90	**16.52**
Query terms	3.78	**5.19**
Unique query terms	2.10	**2.57**

Table 12.13: Summary of overall search behaviour for Experiment Two
bold figures indicate highest values

From Table 12.13 it can be seen that although subjects performed roughly the same number of searches per topic, they tended to view more documents with the experimental system, view these documents less often and retrieve more unique documents. That is, when using the experimental system, the subjects were less likely to retrieve documents that they had already retrieved in response to an earlier query.

The subjects also used more query terms, and more unique terms, per search with the experimental system. In Table 12.14 I present figures on the source of these query terms.

From Table 12.14, it can be seen that there was a (non-significant $t = 1.31$) difference in numbers of query terms used in the first search iteration (3.67 per search control system vs 3.00 on experimental system). There were also differences in how the subject added new terms. For example in the control system the subject was more likely to add their own terms to their query than ones suggested by the system, (on average per search subjects added 8.83[110] of their own terms compared against 1.61 of the expansion terms suggested by the system). On the experimental system, however, this was reversed: the subject was more likely to add terms suggested by the system (8.17 terms per search, compared against 6.67 of their own). The difference between the number of their own terms the subject added was not

[110]This does not include the original query terms.

significant (t = 0.69) however the difference in the number of the system-suggested terms added was significant (t = -3.16). That is, subjects were more likely to use the system-suggested terms when the system used the F_4_*po* term suggestion algorithm.

Control	**303i**	**307i**	**321**	**322i**	**326i**	**347i**	**Averages**
Initial	7	11	14	10	12	12	**3.67**
Subject added own	26	8	26	20	64	15	**8.83**
Subject added system	4	2	9	4	4	6	1.61
Subject removed own	16	6	29	18	63	0	**7.33**
Subject removed system	1	2	9	1	2	0	**0.83**
Experimental	**303i**	**307i**	**321**	**322i**	**326i**	**347i**	**Averages**
Initial	12	7	8	8	10	9	3.00
Subject added own	31	14	26	16	11	22	6.67
Subject added system	36	12	2	29	33	35	**8.17**
Subject removed own	20	4	23	2	10	10	3.83
Subject removed system	2	0	2	0	6	0	0.56

Table 12.14: Statistics on query terms in Experiment Two
bold figures indicate highest value

The subjects also tended to remove fewer expansion terms, either those suggested by the system or themselves, with the control system. Neither difference here was significant (difference in subject-suggested terms removed t = 1.14, difference in system-suggested terms t = 0.56).

12.10.2.1.2 Search effectiveness

The previous section showed that subjects tended to use more terms suggested by the F_4_*po* term ranking scheme. In this section I investigate whether the increase in term use lead to an increase in retrieval effectiveness: did using more expansion terms lead to the retrieval of more relevant documents?

In Table 12.15 I present the number of unique relevant documents found on average per topic and the average relevance score given by the subjects to the documents they assessed as relevant. From Table 12.15, it can be seen that on all topics, with the exception of topic 321, the subjects found at least as many relevant documents on average and the average relevance score given to the documents found was higher. The difference between numbers of

documents found was not significant (t = -0.69). However the difference between the average score given to a relevant document was significant, (t = -5.29). These results indicate that although the F_4*_po* suggested terms did not help find significantly more relevant documents, the F_4*_po* terms helped find *better* relevant documents.

Control	**303i**	**307i**	**321**	**322i**	**326i**	**347i**
Relevant documents found	10.00	**8.00**	**12.33**	7.33	9.67	8.00
Average relevance score	3.78	5.37	5.14	5.05	4.49	4.31
Experimental						
Relevant documents found	**11.00**	**8.00**	7.00	**9.33**	**21.67**	**9.33**
Average relevance score	**6.91**	**6.82**	**6.01**	**7.33**	**7.08**	**5.48**

Table 12.15: Comparison of relevant documents found and average relevance score
bold figures indicate highest value

12.10.2.1.3 Subject's perceptions

The subjects were asked to rate certain aspects of their search (Appendix H), relating to their perception of each search they performed. Table 12.16 summarises the subject's perceptions of the search as they relate to the expansion term suggestions. In particular I concentrate on the results to the questions '*Was it easy to search on this topic?'*, *'Are you satisfied with the results of your search?'*, *'Did you have enough time to do an effective search?'* and *'How useful do you think the query words, suggested by the system, were to your search?'*. All responses were on a scale of 1-5 with a score of '1' representing the category *'Not at all*' and a score of '5' representing the category *'Extremely*'.

	Easy to search	**Search satisfaction**	**Time to search**	**Utility of terms**
Control	2.72	2.61	3.33	1.53
Experimental	**3.72**	**3.83**	**3.89**	**3.53**
Significant	no, t = -0.172	yes, t = -2.99	no, t = -1.41	yes, t = -3.73

Table 12.16: Comparison of subject responses in Experiment Two
bold figures indicate highest value

For all questions the subjects rated the experimental system higher: they found it easier to perform searches upon, had higher search satisfaction and were generally happier with the time they were given to search. More importantly, the subjects rated the terms suggested by the experimental system as better than those suggested by the control system. As seen in

Table 12.17 where the average score per topic for this question is shown, this latter difference holds across topics[111]. The differences are also statistically significant (t = -3.73).

	Utility of terms (control)	Utility of terms (experimental)
303i	1.33	**3.33**
307i	2.33	**2.67**
321	1.33	**1.67**
322i	1.67	**3.67**
326i	2.00	**4.50**
347i	2.00	**5.00**

Table 12.17: Comparison of subject responses in Experiment Two regarding term utility
bold figures indicate highest value

This experiment showed that the terms suggested by the F_4*_po* weighting scheme could give better term suggestions: those that were preferred by the subject and which lead to the retrieval of better relevant documents. In the next experiment I test whether these results hold for automatic query expansion, where the system alone is choosing the expansion terms.

12.10.3 Experiment Three

The third experiment investigated the performance the abductive RF technique against no feedback. Both control and experimental systems used Interface One which only offered the *New Search* option. The control system performed a new search each time the subject modified the query, ranking documents by the combination of all term and document characteristics.

The experimental system performed the same search as the control system for the first query entered by the subject[112]. For the remainder of the topic, each time the subject entered a query and requested a new search a RF iteration was performed using the *Ab_po* function from Experiment One. The *Ab_po* algorithm used the current set of subject query terms and added new query terms before doing a new retrieval. The subject was not shown the new query terms that were added, nor were these highlighted in the full text of documents requested by the subject.

---

[111] These figures and the ones regarding term utility in Table 12.16 are only for searches in which the subject used the term suggestion option.

[112] That is the first query formulation for each topic.

For the subject there was no observable difference between the two systems at the interface level: both systems appeared to do a new search each time. The only difference between the control and experimental system was the method by which the query was modified and the documents were ranked – the RF method of the experimental system. Previously viewed documents were not suppressed and could be retrieved in response to a new query or feedback run. However, as in Experiment One, the timing of the new search option in the control system was altered to ensure that the control system searches took as long as the experimental system.

12.10.2.1 Results of Experiment Three

All searches on both systems started with a new search, subsequent search iterations on the experimental system were all feedback iterations, subsequent searches on the control system were all new searches. As I was interested only in the performance of feedback against no feedback the information regarding the initial search was excluded and the results from Experiment Two only refer to the searches carried out after the initial search. This allows a direct comparison of feedback only against no feedback.

12.10.2.1 Overall search behaviour

The subjects carried out twice as many post-initial searches on the control than experimental system (2.28 per topic control, 1.56 experimental). This was not found to be statistically significant (t = 1.81), although the t value lends some support to the argument that subjects performed more search iterations on the control system.

The subjects viewed slightly more documents on the experimental than control system (16.22 per search iteration, 292 total on control, 17.778 per search iteration, 320 total experimental). They also viewed slightly more unique documents on the experimental system (12.944 per search, 233 total on control system, 13.667 per search, 246 total on experimental system). Neither of these differences was found to be statistically significant (t = -0.33 documents viewed, t = -0.18 unique documents viewed).

To summarise, the subjects on the control system performed many more searches per topic and consequently viewed more documents over the entire *topic*. The subjects ran fewer searches and viewed fewer documents on the experimental system. The question to be answered is whether the subjects are running fewer searches because RF is more or less effective than the subject modifying their own query. This will be investigated in the next two sections.

12.10.2.2 Search effectiveness

The overall precision of the two systems, measured as the total number of unique relevant documents found divided by the total number of unique documents viewed was roughly similar (44.52% control vs 48.48% experimental). Again these figures only relate to search iterations performed after the initial search.

Table 8.12 breaks these overall figures down by topic. For topics 307i, 321, 322i and 347i there was an increase in precision of about 20% when using the experimental system. On topics 303i and 326i the control system gave much better performance (almost 50% increase over the experimental for topic 303i and around 24% for topic 326i). Topics 303i and 326i were the only topics for which the subjects viewed more documents on the experimental than control system.

The difference in precision between the two systems was not found to be statistically significant, (t = -0.31). However if we only consider the four topics where the experimental system is better (307i, 321, 322i and 347i) then the experimental system is significantly better than the control system (t = –9.33). On the topics where the control system is better (303i and 326i) the control system is not significantly better than the control system (t = 1.56).

Condition	303i	307i	321	322i	326i	347i
Control	**70.37%**	29.73%	34.78%	22.92%	**55.26%**	54.05%
Experimental	22.95%	**60.00%**	**56.52%**	**41.18%**	32.10%	**78.13%**

Table 12.18: Results of documents relevant per viewed
bold figures indicate highest value

Comparing the precision by measuring the number of relevant documents found by the number of documents *retrieved*, Table 12.19, it can be seen that the experimental system gives better precision for five of the six search topics. Again the results overall are not significant but if we consider only the topics where the experimental system is better than the control system, then the experimental system is significantly better (t = –4.99).

Condition	303i	307i	321	322i	326i	347i
Control	**31.67%**	4.07%	6.67%	4.07%	10.00%	13.33%
Experimental	7.78%	**6.00%**	**10.83%**	**11.67%**	**17.33%**	**20.83%**

Table 12.19: Results of documents relevant per retrieved
bold figures indicate highest value

Therefore the searchers are finding a higher percentage of relevant documents with the experimental system per documents retrieved and documents that the subject chooses to view. However this is not true for all topics – for some topics, e.g. topic 303i the subject performs better query modification than RF.

Finally, in Table 12.20 I compare the average relevance score given to the relevant documents by the subjects. For almost all topics the subject gives higher scores to documents retrieved by the control system – where the subject performs the query modification. So although RF is better at obtaining new relevant documents it may not be better at retrieving higher quality relevant documents. The difference in relevance score was not, however, significant (t = 1.46).

Condition	303i	307i	321	322i	326i	347i
Control	**3.87**	**4.41**	**4.78**	**5.74**	**5.77**	5.25
Experimental	3.65	2.77	5.00	3.41	5.74	**5.41**

Table 12.20: Average relevance score for control and experimental system
bold figures indicate highest value

In the next section I compare the subjects perceptions of searching on the two systems to see whether the searchers indicated a preference for one system over another.

12.10.3.1.3 Subject's perceptions

As in Experiment Two, the subjects were asked to rate certain aspects of their search (Appendix H), relating to their perception of each search they performed. For the question '*Was it easy to search on this topic?'*, *'Are you satisfied with the results of your search?'*, and *'Did you have enough time to do an effective search?'* the subjects rated the experimental system higher than the control system, however the results were not significant, Table 12.21 summarises the differences.

	Easy to search	Search satisfaction	Time to search
Control	3.50	3.06	3.56
Experimental	**3.83**	**3.44**	**3.94**

Table 12.21: Comparison of subject responses in Experiment Three
bold figures indicate highest value

The results from this experiment show some preference for feedback: the searchers found the same proportion of relevant documents in searching but found these documents using less searching with the experimental system. In this experiment I examined the performance of the F_4*_po* term ranking scheme when the experimental subjects were selecting the expansion terms. In the next experiment I compare two methods of automatically choosing query terms.

12.10.4 Experiment Four

The fourth experiment compared the technique of selecting which RF technique to use against a single method of implementing RF. The control system uses the F_4*_standard* algorithm and adds the six top terms to the query for each iteration of RF. Each iteration of RF, therefore, uses the same algorithm for query modification.

The experimental system *selects* which RF technique to use based on the behavioural evidence given by the searcher. This behavioural evidence is identical to the evidence described in Chapter Ten: order of relevant documents in retrieved set, similarity of relevant documents and precision of the search. As explained in Chapter 10, section 10.4, each expansion term ranking algorithm is associated with a set of rules which define how the behavioural evidence is to be used to decide on a method of query modification.

In Chapter Ten the rules were generated according to the empirical evidence drawn from experiments carried out on the test collections. For this set of experiments it was decided not to attempt to calculate new rules specifically for the data set used, i.e. not to define a good set of rules based on the queries and relevance assessments that are associated with the documents. This is because, in the majority of cases, the document collections associated with real-life IR systems cannot be used to calculate such rules, as there are no associated queries and relevance assessments for the collections. Hence, in this experiment I wanted to test a set of rules that could be applied to any document collection when the F_4*_po* term expansion technique was used. This means that the rules generated for this experiment are probably sub-optimal for this collection – it will be possible to create better rules for this data set – but that the experiment will give a better indication of how the RF techniques will work across collections rather than just for this collection of documents.

The specific rules used are shown in Figure 12.7, and are based on the ones derived for the *wpq* term ranking algorithm. In this experiment the Coverage and Josephson explanations were created as previously and the maximal explanation corresponds to the addition of the top six expansion terms drawn from the top of the F_4*_po* ranking of terms. Only six terms are added to avoid the query being flooded with expansion terms.

The maximal explanation, in this case, therefore corresponds to the query expansion technique in the control system. The only difference is the different term ranking scheme used to rank the terms. As shown in Experiment One the two term ranking schemes do not give noticeably different results when used to provide Josephson explanations. Therefore a main point in this experiments is to see if the two ranking schemes give the same results if we use *different* methods of choosing the expansion terms, i.e. selecting query expansion techniques compared against choosing the top six expansion terms.

if (term ranking method = F_4*_po*)
  if (**precision** is *high*) use josephson
    else if (**precision** is *low*) use maximal
  if (**order** is *low*) use coverage
    else if (**order** is *low*) use maximal
  if (**similarity** is *high*) and (**number of relevant documents** is *high)* use coverage
    else if (**similarity** is *high*) and (**number of relevant documents** is *low*) use josephson
    else if (**similarity** is *low*) use maximal
 if (**similarity** is *high*) use coverage
    else if (**similarity** is *low*) use maximal

Figure 12.7: Rules for selecting query modification technique for the F_4*_po* term ranking scheme

where **bold** entries indicate features of the retrieval, *italic* entries indicate values of the features, and underlined entries indicate the query modification techniques suggested by the value of the feature

12.10.4.1.1 Overall search behaviour

In Table 12.22 I summarise the main findings from the subjects interaction with the two systems. On the experimental system the subjects carried out more searching, more RF and viewed more documents than on the control system. They also used more query terms as a results of the increased searching. Although none of these results are significant, the t[113] levels lend some support to the hypothesis that the subjects were doing more searching on the experimental system and this was due to new search iterations rather than RF iterations. In the next section I compare the effectiveness of the two systems.

[113] t = -1.98 new search iterations, t = -0.79 RF iterations, t = -2.06, t = -1.46 viewed, t = -0.93 retrieved, t =-0.52 query terms, t = -0.62.

	Control	Experimental
New search iterations	2.34	**2.89**
RF iterations	1.06	**1.17**
Total search iterations	3.39	**4.06**
%unique RF	31.12%	**28.75%**
Unique viewed	16.95	**19.22**
Unique retrieved	57.87	**61.33**
Query terms	10.78	**11.78**
Unique query terms	5.06	**5.45**

Table 12.22: Comparison of searches on control and experimental system
bold figures indicate highest value

12.10.4.1.2 Search effectiveness

The overall precision of the control system was higher than the experimental system whether it is measured as the relevant documents found compared against the number of documents the subject viewed (54.80% control, 46.98% experimental) or against the number of documents retrieved (17.90% control, 14.82% experimental). Neither of these differences were significant (t = 0.85 viewed documents, t = 1.09 retrieved documents).

In the remainder of this section I shall compare the results only for RF iterations: the results of searches that were initiated by the subject selecting the *Improve search* option. This will give a clearer picture of the relative performance of the two RF techniques used in this experiment.

After feedback the subjects had relatively similar precision values, as measured by the number of documents found after feedback divided by the number of documents viewed after feedback (50.78% control, 52.08% experimental). The results are not significant (t = -0.07) and for two topics the control system gives better precision whereas the experimental system gives better precision for the other four topics, Table 12.23.

Condition	303i	307i	321	322i	326i	347i
Control	63.19%	**100.00%**	18.26%	**59.88%**	24.81%	38.57%
Experimental	**70.01%**	42.18%	**80.07%**	19.94%	**36.81%**	**63.49%**

Table 12.23: Precision of documents relevant per viewed after feedback
bold figures indicate highest value

In Table 12.24, I show the average relevance score for documents after a new search (**Con before**, **Exp before**), after RF (**Con after**, **Exp after**), and the ratio of the scores after and before feedback (**after**/**before**). This latter measure gives an indication of whether the documents found after RF are given higher relevance scores than after a new search. A value of greater than one indicates higher relevance scores after RF and a value of less than one indicates lower relevance scores after feedback.

From Table 12.24 it can be seen that, on average, the relevance scores for the experimental system are higher than the control system for new search and after RF (**Average**). However the ratio measures are virtually identical. This shows that, although, we achieve higher relevance scores with the experimental system, the experimental system does not retrieve better relevant documents after RF than it was retrieving after a new search.

	303i	**307i**	**321**	**322i**	**326i**	**347i**	**Average**
Con before	4.49	6.31	5.52	7.53	5.8	2.98	5.44
Con after	5.04	4.87	4.558	0	5	2.92	3.73
after/before	1.12	**0.77**	0.83	**0.00**	0.86	**0.98**	0.76
Exp before	5.52	5.3	4.94	4.56	6.47	7.01	5.63
Exp after	6.44	3	6.63	0	5.7	5.03	4.47
after/before	**1.17**	0.57	**1.34**	**0.00**	**0.88**	0.72	**0.78**

Table 12.24: Precision of documents relevant per viewed after feedback

where Con = control system, Exp = experimental system, before = average relevance score before feedback (after a new search), after = average relevance score after RF
bold figures indicate highest value

12.10.4.1.3 Subject's perceptions

In this section I compare the subjects' perceptions of the two systems. In particular I concentrate on the subjects' responses to three aspects: their satisfaction with the search, their assessment of whether they had sufficient time to search and their assessment of how useful RF was to their search.

In Table 12.25 I present the average response to these questions and whether the difference is significant. As can be seen the results are not conclusive in favour of one or other systems: the subjects had greater satisfaction with the control system but felt they had less time with this system and rated the RF component lower than the experimental system.

This set of results are important because they do not show a major difference: the systems were using different term ranking algorithms and different methods of choosing expansion

terms but there was no noticeable performance difference between the two systems. I shall discuss this in more detail in section 12.12.

Question	Average control	Average experimental	Significant
Search satisfaction	**3.72**	3.33	no, $t = 0.97$
Time for search	3.50	**3.67**	no, $t = -0.59$
Utility of RF	1.72	**3.01**	yes, $t = -3.50$

Table 12.25: Precision of documents relevant per viewed after feedback
bold figures indicate highest value

12.10.5 Experiment Five

The fifth experiment concentrates on the role of explanation at the interface. This is the only experiment in which the interfaces for the control and experimental system differ. The control system uses Interface Two and the selection RF algorithm. This was the experimental system from the previous experiment. The experimental system in this experiment uses the same RF algorithm and Interface Four. Interface Four is based on Interface Two but has the added component of an explanation summary. The explanation summary is a representation of the abductive RF process that highlights the main decisions made by the RF algorithm, e.g. which terms were regarded as being most important, which aspects of a term's use were more important than others. The mechanics of producing the summary and the different types of summary are explained in Chapter Twelve.

In this experiment I look at the effectiveness of these summaries in helping subjects to understand what effect the RF algorithm is having on the search. I am particularly interested in how successful the system is at increasing the subject's awareness of RF, any difference in searching behaviour due to the presence of explanations and the quality of the explanation. The data for these conclusions will be primarily gathered through extensions to the standard questionnaires (Appendix H) and post-search interviews. The behavioural question will also consider information from analysis of the search statistics.

Unlike the other experiments, the control and experimental systems differed at the interface rather than the underlying system. Therefore the main focus in the following sections is to highlight the main differences in the two systems regarding how the overall system was used rather than the effectiveness of the RF engine itself.

12.10.5.1.1 Overall search behaviour

In Table 12.26 I compare how often a subject performed a *New search* on the control and experimental systems compared with how often they performed an *Improve search* (RF). From Table 12.26 the subjects, on average, performed the same number of new searches on both systems. However they tended to perform more *Improve searches* on the experimental system.

	303i	**307i**	**321**	**322i**	**326i**	**347i**	**Average**
Control system							
New search	3.00	2.00	1.00	2.67	2.67	1.33	**2.11**
RF	1.67	1.33	1.67	1.33	1.33	1.67	1.50
Total search iterations	4.67	3.33	2.67	4.00	4.00	3.00	3.61
%age RF	0.36	0.40	**0.63**	0.33	0.33	0.56	43%
Experimental system							
New search	2.00	2.33	2.00	1.33	3.67	1.33	**2.11**
RF	2.00	2.00	1.67	2.33	1.67	2.00	**1.95**
Total search iterations	4.00	4.33	3.67	3.66	5.34	3.33	**4.06**
%age RF	**0.50**	**0.46**	0.46	**0.64**	0.31	**0.60**	**49%**

Table 12.26: Comparison of new searches against RF searches on Control and Experimental systems

bold figures indicate highest value

The number of total search iterations and new search iterations performed on the two systems was not statistically significant (t = -1.34 and t = 0.0 respectively). The difference in number of RF iterations was found to be statistically significant (t = 3.16). However, the *percentage* of all search iterations that were RF iterations (Table 12.26 rows 5 and 10) was not significant (t = -0.92). This means that although the subjects were doing more RF on the experimental system, there was not a significant preference for RF over a new search on the experimental system. The greater use of RF on the experimental system, therefore, does conclusively indicate that the explanations were leading the subjects to employ RF more often.

12.10.5.1.2 Search effectiveness

In Tables 12.27 and 12.28 I present the average ratio of documents assessed relevant to the number of documents viewed by the subject (Table 12.27) and the average ratio of documents assessed relevant to the number of documents retrieved (Table 12.28).

Condition	303i	307i	321	322i	326i	347i
Control	46.55%	**32.64%**	**42.65%**	**53.35%**	46.98%	**37.03%**
Experimental	**62.25%**	16.48%	37.50%	20.55%	**52.49%**	31.55%

Table 12.27: Ratio of documents assesed relevant per documents viewed
bold figures indicate highest value

Condition	303i	307i	321	322i	326i	347i
Control	15.76%	**12.69%**	**26.54%**	**21.78%**	20.21%	**16.16%**
Experimental	**19.30%**	6.52%	17.76%	11.22%	**20.90%**	12.62%

Table 12.28: Ratio of documents assesed relevant per documents retrieved
bold figures indicate highest value

In both Tables 12.27 and 12.28 the experimental system gave better performance for topics 303i and 347i, whereas the control system gave better performance on the other four topics. In neither case was the difference significant (t = 1.78 retrieved documents, t = 0.93 viewed documents).

There is a preference for the control system in terms of these performance measures. This is because in both cases, although the subjects found more relevant documents with the experimental system (average of 8.94 documents per topic on the experimental system compared to 8.62 per topic on the control system), they viewed more documents and retrieved more documents with the experimental system.

12.10.5.1.3 Subject's perceptions

An important aspect of this experiment is whether the use of explanations helped the subject understand feedback and to what degree they stimulated the subject's interest in RF. In particular I shall examine how useful the subjects rated the three features: *Improve Search*, the Explanation itself and the *Explain more* option.

In Table 12.29 I compare the average subject score for the three options. Each subject was asked how useful the option was to their search. As in previous questions the subject was asked to indicate the utility of the option using a 5 point scale with the value of '5' reflecting the highest utility. The values shown in Table 12.22 show the averaged results for the searches in which a subject employed RF. There was no detectable correlation with either the subjects' opinions on how easy the topic was or the success of RF.

Topic	Improve search	Explanation	Explain more
303i	3.00	**3.33**	3.00
307i	**2.50**	**2.50**	2.00
321	2.33	**3.00**	2.00
322i	1.67	**3.33**	3.00
326i	**3.33**	3.00	2.50
347i	1.50	**2.00**	0.00[114]

Table 12.29: Comparison of subject responses in Experiment Three
bold figures indicate highest value

The general tendency is for the Explanation to be rated higher than the *Improve search* (RF) option which, in turn, is rated higher than the *Explain more* option. The post-search interview was used to elicit the subject's perceptions on the relative worth of these options. The main reason given for the higher rating for the Explanation was that even if RF did not work, i.e. added unhelpful terms to the query, or if the wrong type of documents were retrieved the Explanation still gave useful information. This is because it still gives information on why the system modified the query. Therefore the success of the Explanation is not dependent on the success of RF.

The *Explain more* option was generally rated lower than the RF option. There are two reasons for this. Firstly, subjects had to explicitly request more information. This meant that subjects may not have requested information that may have been useful if they had viewed it. Secondly the information provided by the *Explain more* option was only useful relative to what was provided by the Explanation and RF: if the Explanation was not useful or RF led to a poor change in the subject's query then the *Explain more* option was not useful. This is because *Explain more* in this case gave more information about an aspect of the system that was not of interest. In addition, if the Explanation gave enough information to the subject about the effect of RF then the *Explain more* option was not necessary.

The situation where *Explain more* was most useful was where the subject was unsure why a query had retrieved a particular set of documents. In this case the subject could investigate the *Explain more* information to check what weighting schemes the system was using to retrieve

[114] No subject used the *Explain more* option for this topic.

documents. Although the subject could not change the retrieval scheme themselves they could remove terms from the query that were being prioritised by the system. A natural extension to the interface would be to allow the subject to alter the way terms were being used to retrieve documents. Overall the subjects found the *Explain more* option interesting but not always of use.

In general the subjects liked the use of explanations but most said that they would like more types of explanations and explanations that were more specific to their search. The first comment is valid and a wider range of explanations could be developed for such an interface. The second comment specifically relates to the selection of query terms. Most subjects who made this comment would have preferred a more semantic explanation of why a particular query term(s) was added to their query, e.g. an explanation of the form '*I am adding the word* space *to your query as you are searching for documents on the Hubble telescope and space is a word that is strongly related to this topic'*. This type of explanation is very difficult to create using the statistical techniques that underlie the experimental systems used in this thesis. Most subjects liked the presentation of explanations on the basis that *some* form of system explanation was useful and encouraging. As mentioned before this was because explanations can be helpful even when RF is not performing correctly.

12.11 Discussion

In this section I shall summarise the overall findings of the experimental analyses relating to the systems, section 12.11.1, the topics, section 12.11.2, and the term ranking schemes, section 12.11.3.

12.11.1 Search system

Most subjects found the experimental system easy to use and operate. Table 12.30 shows the average responses for three questions asked at the end of each experiment. The values are out of 5 with a score of 5 representing the category 'Extremely'.

Question	Average response
How easy was it to *learn* how to use this information system?	4.52
How easy was it to *use* this information system?	4.45
How well did you *understand* how to use this information system?	4.38

Table 12.30: Summary of subject exit responses

Subjects did have specific comments relating to the system that reflected their personal experiences, e.g. one subject did not like the use of grey backgrounds, several subjects would have liked a 'back button' and several subjects disliked the lack of control when using RF.

12.11.2 Topics

In this section I discuss the search topics used in the experiments. In section 12.4 I discussed the search topics used in these experiments. As mentioned in section 12.4 one of the motivations for using this set of topics was to allow a comparison of the interactive search results with the relevance assessments provided by the ad-hoc TREC track.

This analysis will be based on the figures given in Table 12.31. For each topic, Table 12.31 row 2 shows first the total number of relevance assessments for the topic in all experiments. That is the total number of documents marked relevant by any subject in any experiment using either the control or experimental system. In row 3 I calculate the total number of *unique* relevant documents, i.e. do not count a document twice if more than one subject marked it relevant. Row 4 presents the number of unique relevant documents as a percentage of the total number of relevant documents. Finally row 5 gives the number of unique relevant documents found in the ad-hoc, non-interactive, TREC task.

	303i	**307i**	**321**	**322i**	**326i**	**347i**
Total relevant documents	269	267	252	251	387	330
Unique relevant documents	72	134	101	97	112	133
% unique	26.77%	50.19%	40.08%	38.65%	28.94%	40.30%
TREC ad-hoc	10	83	133	33	45	125

Table 12.31: Details on topics used in the experiments

There are four main points to be made regarding the topics.

i. The first comparison is between the numbers of relevant assessments made by the subjects across the topics (row 2). For some topics, e.g. topics **326i** (*ferry sinkings*) it was easier to find relevant documents than others. For this topic subjects found 12 relevant documents on average whereas for topic **321** subjects only found around 8 documents on average. Although this is not a large absolute difference it does represent an increase of 50% in the number of relevant documents found per search.

ii. The second comparison is with the number of unique relevant documents found in the TREC ad-hoc track and by the experimental subjects. With the exception of topic **321** – women in parliament – the subjects found more unique relevant documents that were found in the ad-hoc, non-interactive, task. This is to be expected as the experimental subjects could modify their query according to the documents retrieved and could use additional terms not supplied by the TREC ad-hoc topic. The subjects

could also interpret what type of information was required – they were allowed to define what was meant by useful information. So the documents chosen by the user would not necessarily be assessed as relevant in the ad-hoc track. One potentially interesting feature is that if we omit topic **321** there is a correlation between the number of unique relevant documents found by the ad-hoc task and by the users (rows 3 and 5). This did not hold for the number of relevant documents found by the subjects (row 4). Therefore, although, it may be easy to find relevant documents for some topics, it is harder to find *different* relevant documents from the ones found by other subjects.

iii. subjects perceptions of the search tasks varied. At the end of the experiment the subjects were asked which topics they found most interesting, which they found most difficult to *start* a search on and which they found it most difficult to search. The subjects could mark more than one topic in each category. Table 12.32 gives the percentage of users who assessed a topic in each category. Also included in Table 12.32 is the average response to the question asked after each search '*How easy was it to judge how useful a document as to the search*?'. This question (with 1 being difficult to judge relevance and 5 being easy to judge relevance) was intended to elicit how easy/difficult a subject found it to make relevance decisions.

	Interesting	**Start**	**Finding**	**Assessment**
303i	**76%**	7%	**38%**	3.00
307i	24%	28%	14%	2.80
321	28%	28%	31%	3.00
322i	41%	24%	31%	**3.03**
326i	28%	**38%**	31%	2.90
347i	41%	7%	17%	**3.03**

Table 12.32: Subjects' views on search topics
bold figures indicate highest value

As can be seen from Table 12.32 there were three popular topics - **303i, 322i** and **347i** – and three less popular topics. The topic regarding the Hubble telescope – topic **303i** – was particularly marked out as being interesting with three-quarters of subjects rating it as one of the most interesting topics in the experiment. The three popular topics were slightly harder to perform a whole search on (Column 4), e.g. 38% of subjects rated topic **303i** as being a difficult topic for which to find useful documents but not necessarily difficult topics for which to start a search (Column 3).

12.11.3 Comparison of term ranking schemes

The final analysis is the comparison between the F4_*standard*, *F4_po*, and *wpq* term ranking algorithms. As discussed in section 12.2, the *wpq* function differs from F4_*standard* as it includes a component that measures the value of a term as an *expansion* term. This component is based on the difference between a term's appearance in the relevant documents and its appearance in the non-relevant documents.

The intention is to uncover how different the algorithms are in respect of which expansion terms they suggest given the same relevance information. For each topic I compare the expansion by the following method:

i. I take each log file – a complete search session on a topic - and extract the relevant documents found by the subject in the search. This is the set of relevant documents assessed by the subject who created the log, based on the relevance criteria for the subject performing the search.

ii. For each set of relevant documents I calculate the top 20 expansion terms for the F4_*standard, F4_po* and *wpq* algorithms. I only consider the top 20 terms as this was the number suggested to the subject in the interactive experiment (Experiment Two) and also because these are the terms that are most likely to be used for expansion in automatic query expansion.

iii. I then compare the *overlap* between the terms suggested by the three algorithms to see how similar are the lists of suggested terms.

iv. The results for individual topics are averaged, i.e. I calculate the average overlap for topic 303i, for topic 307i , etc., and for the complete set of logs.

The overlap results are presented in Table 12.33 as a percentage of terms suggested and as the number of terms, on average, that are in common. For example, in Table 12.33 for topic **303i**, the overlap between F4_*standard* and *F4_po* is 19.83% which corresponds to an average overlap of 3.97 terms in the top 20 terms suggested by the techniques.

From Table 12.33 it can be seen that the lowest overlap is between the F4_*standard* and *F4_po* algorithms: these algorithms differ most in the terms they suggest given the same set of relevant documents. On average the 20 terms suggested by these two term ranking schemes will only have 2.87 terms in common: the remaining terms will differ. The two differences between these two algorithms are the use of partial relevance assessments and the use of ostensive evidence. As will be discussed below it is the particular implementation of ostension that is likely to be having the main effect.

	303i	307i	321	322i	326i	347i	All topics
F4_*standard* vs F4_*po*	19.83%	14.33%	11.17%	17.17%	16.83%	6.67%	14.33%
	3.97	2.87	2.23	3.43	3.37	1.33	2.87
F4_*standard* vs *wpq*	21.67%	15.50%	17.17%	18.67%	17.50%	7.83%	16.39%
	4.33	3.10	3.43	3.73	3.50	1.57	3.28
F4_*po* vs *wpq*	94.50%	87.17%	87.33%	95.33%	94.00%	93.83%	92.03%
	18.90	17.43	17.47	19.07	18.80	18.77	18.41

Table 12.33: Comparison of term ranking algorithms

The *wpq* and F4_*standard* algorithms also differ, and differ almost to the degree that the F4_*standard* and *F4_po* algorithms differ. The only difference between these two algorithms is the additional component in the *wpq* algorithm that calculates the difference in a term's appearance between the relevant and non-relevant documents. In practice this component is influenced by the number of relevant documents in which a term appears and has the effect of eliminating terms that appear in very few relevant documents. The result is that terms which have a low collection frequency but appear in relevant documents, e.g. those terms that only appear in one or two documents, both of which are relevant are eliminated from the list of expansion terms. This component, then, prioritises more general terms that appear in many relevant documents.

The *wpq* and *F4_po* algorithms are most similar: on average the terms they suggest only differ by one or two terms. Both algorithms use two components to rank terms: a discriminatory component and a component that takes into account the number of relevant documents in which a term has appeared. In *wpq* the discriminatory component is F4_*standard* and in F4_*po* the discriminatory component is the version of F4_*standard* that uses partial relevance information. The component that is based on the number of relevant documents in F4_*po* is the ostensive evidence. The implementation of the ostensive evidence in F4_*po* implicitly takes into account the number of relevant documents in which a term appears. This is similar to the component in *wpq* that is based on a count of relevant documents. Given that these two factors are similar it is fair to assume that what makes these two algorithms similar is this component and what makes them different is the partial scores given to the relevant documents.

On reason for the low difference may also be due to the low use of multiple iterations of RF. The ostensive component includes information on when a document was marked relevant and

this biases the term ranking in favour of terms that were most recently marked relevant. However, few subject's performed multiple *consecutive* iterations of RF, consequently the ostensive evidence did not have a chance to accumulate.

12.12 Summary

In this section I shall give a short summary of the main findings from the experiments.

In Experiment One I compared two term ranking algorithms, examining how well they performed at providing terms for a Josephson method of query expansion. Specifically this compared the traditional F4 (F4*_standard*) term ranking algorithm against a version of F4 that incorporated partial relevance assessments and ostensive evidence. The results from this experiment were not conclusive in that, although the retrieval results pointed slightly in favour of the traditional version of F4*,* the subjects' perceptions were that the new version, F4*_po,* provided more useful terms. This experiment is interesting in the lack of correlation between what the subjects' reported (their view of the expansion terms) and their interaction with RF (the fact that they appeared to use the F4*_po* terms more and remove them less often) compared with how useful the documents retrieved by these terms were. That is, although the subjects liked the F4*_po* terms they did not necessarily lead to the retrieval of more relevant documents.

This result was replicated in Experiment Four in which I compared different methods of selecting terms; one using the F4*_standard* term ranking algorithm and one using the F4*_po* algorithm. In this experiment also, the results did not show a big difference in performance between the two different RF techniques.

However, as shown in Experiment Three, where the subjects selected the expansion terms themselves the F4*_po* algorithm was clearly shown to be better in terms of finding relevant documents. It therefore remains an important open question as to why different methods of ranking terms give similar results. One possible reason is that the original query terms in these experiments are not prioritised highly enough, i.e. I did not weight the original query terms relative to the expansion terms. A further experiment on this may reveal differences between the two ranking algorithms.

Finally, in Experiment Five, I investigated the presentation of RF at the interface. This experiment showed that engaging the user in the results of RF can lead to better more use of RF and a better understanding of the effect of RF on a subject's search.

The experiments, overall, have highlighted important issues regarding the overall goal of incorporating behavioural information into RF. They have also shown that selecting explanations can perform at least as well as using a single method of RF with the additional advantage that selecting RF techniques can be used to present explanations of RF to the user.

Part V

Conclusions

Chapter Thirteen

Conclusion and discussion

13.1 Introduction

In this thesis I have examined a number of aspects of using relevance information gained from a user to automatically modify the user's query. In Part II I examined selecting term weighting schemes based on relevance information; in Part III I examined selecting expansion terms using abductive inference techniques. In Part IV I examined the performance of the techniques from Parts II and III in a user study. In Part IV I also examined the presentation of RF at the interface. In this chapter I shall discuss the main findings and how these may be exploited in future work.

13.2 Selective relevance feedback

Part II of this thesis mainly concentrated on techniques for selecting which aspects of a term's use were good at indicating relevant material. The basic hypothesis was that RF should not be based solely on a term's appearance within relevant and non-relevant documents but on how the terms are *used* within relevant and non-relevant documents. That is, in RF we should concentrate on identifying what features of a term indicates relevant material. This is an attempt to move RF from simply a statistical model of term distribution, e.g. [RSJ76], to one that incorporates a stronger relation to the document text in which terms appear. By considering more information on how a term is used within documents IR systems can which documents decide containing a query term are likely to be relevant and which are not.

As introduced in Chapter One, IR is basically a process of *mediation*: the IR system mediates between the documents and the user's information need by means of *representations* of the document and information need (the indexed form of the document and user's query). RF algorithms form part of this mediation process by altering the query representation to one that is closer to the relevant document representations. The more flexible are the representations used, the more flexible is the mediation process. In widening the range of representations of individual terms – the term characteristics, Chapter Three – we can achieve a more flexible mediation process.

The most significant result, and one that was shown to hold over a range of conditions, Chapter Seven, was that it is possible to use relevance information to select which aspects of a term's use – which term characteristics –best represent each query term. That is we can use relevance information to select how query terms should be used to retrieve documents.

The use of multiple representations of terms has strong relations to Ingwersen's work on polyrepresentation, Chapter Four, [Ing94]. In this theory Ingwersen suggests that multiple representations of a single object can provide better insight into the object than a good single representation. In addition, Ingwersen suggests that, in individual cases, some representations are better than others [CHECK THIS]. In this thesis I demonstrate that multiple representations of terms – the characteristics – can provide better retrieval results than individual characteristics but that selecting characteristics is generally better. This accords with Ingwersen's theory on representations.

13.3 Abductive query modification

In Part III I proposed a framework for query modification that was based on abductive inference, or abduction. The main aim of this framework was to incorporate more aspects of how users assess documents into the RF process. In particular the framework depended on the abductive notion of *explanation*: query modification should be directed by an explanation of why the user made a set of relevance assessments. Therefore the process of query modification should not be a single procedure that is applied to all searches but should be an *adaptive* response to what information the user finds of interest (the relevant documents) and how the user is searching (the relevance assessments, Chapter Eight).

The framework presented in Part III is heavily dependent on evidential reasoning: choosing what documents to explain, what kind of query modification is required, how terms should be chosen and how many terms should be chosen. This connects to the work presented in Part II: we use abduction to construct a new query (the explanation) and then use the techniques from Part II to decide how the new query terms should be used to retrieve documents. The experimental study in Part III provided a basic experimental investigation of some of these issues. The experimental evidence shows that many of the techniques presented do lead to better retrieval results.

The abductive framework is an initial attempt to motivate the use of explanation as a means of constructing RF models. My approach shows how this may be accomplished but requires much more investigation and experimentation to develop the intuitions presented in Chapter Eight into a full model of RF. In particular the following aspects require addressing:

i. The definitions of explanations. In Part III I remained close to definitions of explanations that came from the abductive literature. These, for the most part, were definitions of explanations that have been shown to be successful in other domains. However, a closer study of the relationship between user searching behaviour and types of query modification would give a better understanding of what types of explanations are required.

ii. In Part III I concentrated mainly on the construction of explanations rather than the components of explanations themselves. Although I ranked possible expansion terms by how good they would be for a particular type of explanation, no attention was paid to the explanation as a whole. That is explanations, ultimately, were discrete sets of elements (terms) rather than a coherent explanation of the relevance assessments; no attention was paid to how the elements of the explanation interacted. Similarly in the experiment which selected which type of explanation was required, no attention was paid to how the different pieces of behavioural evidence interacted. These aspects of the framework require further development as, typically, the evidence used to supply the explanation must be coherent and the elements of the explanation should make sense as a whole, [TS97].

iii. The process by which the system chooses which explanation is required, Chapter Ten, was converted into a rule-based procedure. Although different rules may be used in an individual search, the evidence used to create the rules and decide which rules to use is fixed. What this approach lacks, so far, is a means of creating new knowledge. Explanations, as outlined in Chapter Eight, usually add to our knowledge of a problem by providing possible causes or reasons for an event. These are typically ones that are not known in advance. However, in this framework, we do not have a means of creating rules or new methods of finding information dynamically, [TS94].

iv. Although this model was extensively investigated, the main aspect that was not covered experimentally was the use of previous search history as an additional method of deciding what kind of query modification is required.

13.4 Users and RF

In Part IV I examined some of the successful techniques from Part III in a user-centred evaluation. In addition I showed how it was possible to incorporate more behavioural evidence into the explanation creation process.

The experiments were limited in terms of number of subjects employed and the number of experiments run. Nevertheless they do provide useful areas of study for more detailed experiments. One of the main findings from part IV was that how users interact with RF is important. In particular if users receive more information on how RF is changing their search and why, then this can lead to more use of RF by the user. This is important as users must *trust* RF before they will use it. In particular this is because RF has an unknown effect on the user's search: the user does not know what query terms will be added to their search, what way the query terms will be used to retrieve documents and what kind of documents will be returned after RF. This can lead users to stop using RF if it does not work, or not to try RF at all, preferring instead to modify their own query.

The use of explanations as a means of presenting the user with information on the process of RF was shown to be beneficial. This aspect of the user experiments should however be exploited in a much larger investigation.

13.4 Summary

One of the main motivations for starting this work was the diversity of research on how people serach, e.g. [Ell98, Kuh91, Vak00]. What these studies show are the range and complexity of user search behaviour. However, although users and searches are complex, the IR systems themselves are often relatively simple. If the machinery of IR is to keep up with the science of searching then we need more adaptive systems.

This thesis is concerned with increasing the adaptivity of RF techniques. Specifically I was interested in exploiting behavioural information to allow the system to better adapt to the user. Many of the techniques suggested in this thesis can impact on these studies of user searching. For example, the use of multiple term representations would allow a more detailed investigation of how a user's search changes over time. Similarly how the process of making relevance assessments maps to what kind of query modification can be effective in highlighting the relation between search and system. This thesis is, then, an attempt to bring the user and system closer.

References

[Aal92] I. J. Aalbersberg. *Incremental relevance feedback*. Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 11-22. Copenhagen. 1992.

[Alis96] A. Aliseda. *A unified framework for abductive and inductive reasoning in philosophy and ai.* ECAI '96 workshop on abductive and inductive reasoning. *http://www.cs.bris.ac.uk/~flach/ECAI96/Papers/aliseda.ps.gz*. 1996.

[Bar94] C. L. Barry. *User-defined criteria: an exploratory study*. Journal of the American Society for Information Science. **45**. 3. pp 149-159. 1994.

[BS98] C. L. Barry and L. Schamber. *Users' criteria for relevance evaluation: a cross-situational comparison*. Information, Processing and Management. **34**. 2/3. pp 219-237. 1998.

[Bat90] M. Bates. *Where should the person stop and the information search interface start?* Information, Processing and Management. **26**. 5. pp 575-592. 1990.

[Bay63] T. Bayes. *An Essay Toward Solving a Problem in the Doctrine of Chances*. Philosophical Transactions of the Royal Society of London. **53**. pp370-418. 1763.

[Beau97] M. Beaulieu. *Experiments with interfaces to support query expansion*. Journal of Documentation. **53**. 1. pp 8-19. 1997.

[BJ98] M. Beaulieu and S. Jones. *Interactive searching and interface issues in the Okapi best match probabilistic retrieval system*. Interacting with computers. **10**. 3. pp 237-248. 1998.

[BRR96] M. Beaulieu, S. Robertson and E. Rasmussen. *Evaluating interactive systems in TREC*. Journal of the American Society for Information Science. **47**. 1. pp 85-94. 1996.

[BCK+96] N. J. Belkin, C. Cool, J. Koenemann, K. Bor Ng, S. Park. *Using relevance feedback and ranking in interactive searching*. Proceedings of the Fourth Text Retrieval Conference (TREC-4). (D. Harman ed). NIST Special Publication 500-236. pp 181-210. 1996.

[BCS+95] N.J. Belkin, C. Cool, A. Stein, and U. Thiel. *Cases, scripts and information-seeking strategies: On the design of interactive information retrieval systems.* Expert Systems with Applications. **29**. 3. pp 325-344. 1993.

[BKF+95] N. J. Belkin, P. Kantor, E. A. Fox and J. A. Shaw. *Combining the evidence of multiple query representations for information retrieval*. Information Processing and Management. **31**. 3. pp 431-448. 1995.

[Borg97] C. L. Borgman. *Why are online catalogs still hard to use?* Journal of the American Society for Information Science. **47**. 7. pp 493-503. 1996.

[Bo00a]. P. Borlund. *Experimental Components for the Evaluation of Interactive Information Retrieval Systems*. Journal of Documentation. **56**. 1. pp 71 – 90. 2000.

[Bo00b]. P. Borlund. *Evaluation of interactive information retrieval systems*. PhD Thesis. Abo Akademi University. 2000.

[Bo01] P. Borlund. Personal communication.

[BI97] P. Borlund and P. Ingwersen. *The development of a method for the evaluation of interactive information retrieval systems.* Journal of Documentation. **53**. 5. pp 225-250. 1997.

[BI99] P. Borlund and P. Ingwersen. *The application of work tasks in connection with the evaluation of interactive information retrieval systems: empirical results*. Mira '99. S. Draper, M. Dunlop, I. Ruthven and C. J. van Rijsbergen (eds). Electronic Workshops in Computing. British Computer Society. 1999.

[BKL71] A. Borodin, L.Kerr and F. Lewis. *Query splitting in relevance feedback systems.* The SMART retrieval system - experiments in automatic document processing. G. Salton (ed). Chapter 19. pp 394-402. 1971.

[BCT87]. G. Brajnik, G. Guida and C. Tasso. *User modeling in intelligent information retrieval.* Information Processing and Management. 23. 4. pp 305 - 320. 1997.

[BG94] M. Buckland and F. Gey. *The relationship between recall and precision.* Journal of the American Society for Information Science. **45**. 1. pp 12-19. 1994.

[BSA94] C. Buckley, G. Salton and J. Allan. *The effect of adding relevance information in a relevance feedback environment*. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 292-300. Dublin. 1994.

[BAT+94] T. Bylander, C. Tanner, and J. R. Josephson. *Computational complexity of abduction*. Chapter 7. In Abductive Inference: Computation, Philosophy, Technology. J. R. Josephson and S. G. Josephson (eds).New York: Cambridge University Press. pp 157-179. 1994.

[Cam90] I. Campbell. *RdSystem technical notes*. University of Glasgow, Department of Computing Science, Glasgow. 1990. *unpublished*.

[Cam95] I. Campbell. *Supporting information needs by ostensive definition in an adaptive information space*. MIRO '95. electronic Workshops in Computing, Springer Verlag. Ian Ruthven (ed). 1995.

[Cam99] I. Campbell. *Interactive evaluation of the Ostensive Model, using a new test-collection of images with multiple relevance assessments*. Journal of Information Retrieval. **2**, 1, pp 89-114. 1999.

[CVR96] I. Campbell and C. J. van Rijsbergen. *Ostensive model of information needs.* Proceedings of the Second International Conference on Conceptions of Library and Information Science: Integration in Perspective (CoLIS 2). Copenhagen. pp 251-268. 1996.

[CGR+92] A. Cawsey, J. Galliers, S. Reece and K. Sparck Jones. *Automating the librarian: belief revision a base for system action and communication with the user.* The Computer Journal. **35**. 3. pp 221-232. 1992

[CCR71] Y. K. Chang, C. Cirillo and J. Razon. *Evaluation of feedback retrieval using modified freezing, residual collection & test and control groups*. The SMART retrieval system - experiments in automatic document processing. G. Salton (ed). Chapter 17. pp 355-370. 1971.

[CG91] E. Charniak and R. Goldman. *A probabilistic model of plan recognition.* Proceedings of the Ninth National Conference on Artificial Intelligence. pp 160 -1 65. Anaheim. 1991.

[CK66] C. W. Cleverdon and E. M. Keen. *Factors determining the performance of indexing systems.* Vol 1: Design. Vol 2: Results. Cranfield, UK: Aslib. Cranfield Ressearch Project. 1966.

[Coop90] G. F. Cooper. *The computational complexity of probabilistic inference using Bayesian belief networks.* Artificial Intelligence. **40**. 2 - 3. pp 393 - 405. 1990.

[Coop88] W. S. Cooper. *Getting beyond Boole*. Information Processing and Management. **24.** 3. pp 243-248. 1988.

[CLVR98] F. Crestani, M. Lalmas and C. J. van Rijsbergen. *Information retrieval: uncertainty and logics - Advanced models for the representation and retrieval of information*. Kluwer Academic Publishers. 1998.

[CRS+95] F. Crestani, I. Ruthven, M. Sanderson and C. J. van Rijsbergen. *The troubles with using a logical model of IR on a large collection of documents*. Proceedings of the Fourth Text Retrieval Conference (TREC-4). NIST special publication 500-236. D. K. Harman (ed). pp 509-525. 1995.

[CH79] W. Croft and D. Harper. *Using probabilistic models of information retrieval without relevance information*. Journal of Documentation. **35**. 4. pp 285-295. 1979.

[Dem68] A. P. Dempster. *A generalization of the Bayesian inference.* Journal of Royal Statistical Society. **30**. pp 205-447. 1968.

[DMB98] S. Dennis, R. McArthur and P. Bruza. *Searching the WWW made easy? The Cognitive Load imposed by Query Refinement Mechanisms*. Proceedings of the Third Australian Document Computing Symposium. 1998.

[DBM97] N. Denos and C. Berrut and M. Mechkour. *An Image Retrieval System based on the Visualization of System Relevance via Documents*. Database and Expert Systems Applications (DEXA). Toulouse. pp 214-224. 1997.

[Dra00] S. Draper. Personal communication.

[Efth93] E. N. Efthimiadis. *A user-centred evaluation of ranking algorithms for interactive query expansion.* Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 146-159. Pittsburgh. 1993.

[Efth95] E. N. Efthimiadis. *User-choices: a new yardstick for the evaluation of ranking algorithms for interactive query expansion.* Information processing and management. **31**. 4. pp 605-620. 1995.

[EB88] M. Eisenberg and C. Barry. *Order effects: a study of the possible influence of presentation order on user judgements of document relevance*. Journal of the American Society of Information Science. **39.** 5. pp 293-300. 1988.

[Ell89] D. Ellis. *A behavioural approach to information system design*. Journal of Documentation. **45**. 3. pp 171-212. 1989.

[ECH93] D. Ellis, D. Cox, and K. Hall. *A comparison of the information seeking patterns of researchers in the physical and social sciences.* Journal of Documentation. **49**. 4. pp 356-369, 1993.

[FST+99] V. I. Fants, J. Shapiro, I. Taksa and V. G. Voiskunskii. *Boolean search: current state and perspectives*. Journal of the American Society of Information Science. **50**. 1. pp 86-95. 1999.

[FlKa97] P. Flach and A. Kakas. *Abductive and Inductive Reasoning: report of the ECAI'96 workshop*. Logic Journal of the IGPL. **5**. 5. pp 773 - 778. 1997.

[FM95] V. Florance and G. Marchionini. *Information processing in the context of medical care.* Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle. pp 158-163. 1995

[FB00] H. Fowkes and M. Beaulieu. *Interactive searching behaviour: Okapi experiment for TREC-8*. IRSG 2000 Colloquium on IR Research. Cambridge. 2000.

[FBK+92] E. Fox, S. Betrabet, M. Koushik, and W. Lee. *Extended Boolean models*. Information Retrieval: Data Structures & Algorithms. Englewood Cliffs: Prentice Hall. W.B. Frakes and R. Baeza-Yates (eds). Chapter 15. pp 393-419. 1992.

[FMS91] H. P. Frei, S. Meienberg and P. Schauble. *The perils of interpreting recall and precision values.* Information Retrieval. N. Fuhr (ed). pp 1-10. Springer Verlag. 1991.

[FMW71] S. R. Friedman, J. A Maceyak and S. F. Weiss. *A relevance feedback system based on document transformations*. The SMART retrieval system - experiments in automatic document processing. (G. Salton ed). Chapter 23. pp 447-455. 1971.

[HC90] D. Haines and W. B. Croft. *Relevance feedback and inference networks.* Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 2-11. Pittsburgh. 1993.

[Har86] D. Harman. *An experimental study of factors important in document ranking.* Proceedings of the Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 186-193. Pisa. 1986.

[Har88] D. Harman. *Towards interactive query expansion.* Proceedings of the Eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 321-331. Grenoble. 1988.

[Har92a] D. Harman. *Ranking algorithms.* Information Retrieval: Data Structures & Algorithms. Englewood Cliffs: Prentice Hall. W.B. Frakes and R. Baeza-Yates (eds). Chapter 14. pp 363-392. 1992.

[Har92b] D Harman. *Relevance feedback revisited.* Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 1-10. Copenhagen. 1992.

[Har92c] D. Harman. *Relevance feedback and other query modification techniques.* Information Retrieval: Data Structures & Algorithms. Englewood Cliffs: Prentice Hall. (W.B. Frakes and R. Baeza-Yates ed). Chapter 11. pp 241-263. 1992.

[Har65] G. H. Harman. *The inference to the best explanation.* The Philosophical Review. **74**. 1. pp 88 - 95. 1965.

[Hea99] M. Hearst. *User interfaces and visualisation*. Modern information retrieval. R. Baeza-Yates and B. Riberio-Nelo (eds). Chapter 10. pp 257-323. Addison-Wesley/ACM Press. 1999.

[HP93] M. A. Hearst and C. Plaunt. *Subtopic structuring for full-length document access*. Proceedings of the Sixteenth ACM SIGIR Conference on Research and Development in Information Retrieval. pp 59-68. Pittsburgh. 1993.

[HTP+00] W. Hersh, A. Turpin, S. Price, D. Kraemer, B. Chan, L.Sacherek and D.Olson D. *Do batch and user evaluations give the same results? An analysis from the TREC-8 Interactive Track*, Proceedings of the Eighth Text Retrieval Conference (TREC-8). Gaithersburg. 2000.

[Hui96] T. Huibers. *An axiomatic theory for information retrieval*. PhD thesis. Utrecht University. 1996.

[Ide71] E Ide. *New experiments in relevance feedback.* The SMART retrieval system - experiments in automatic document processing. (G. Salton ed). Chapter 16. pp 337-354. 1971.

[IdS71] E. Ide and G. Salton. *Interactive search strategies and dynamic file organization in information retrieval*. The SMART retrieval system - experiments in automatic document processing. (G. Salton ed). Chapter 18. pp 373-393. 1971.

[Ing92] P. Ingwersen. *Information retrieval interaction*. Taylor-Graham. 1992.

[Ing94] P. Ingwersen. *Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction*. Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 101-110. Dublin. 1994.

[Iwa00] M. Iwayama. *Relevance feedback with a small number of relevance judgements: incremental relevance feedback vs document clustering*. Proceedings of the twenty-third annual international ACM SIGIR Conference on Research and development in information retrieval. pp 10-16. Athens. 2000.

[JJ91] J. W. Janes. *Relevance judgements and the incremental presentation of document representations*. Information Processing and Management. **27**. 6. pp 629-646. 1991.

[JAS+00] B. J. Jansen, A. Spink and T. Saracevic. *Real life, real users, and real needs: a study and analysis of users on the web.* Information Processing and Management. **36**. 2. pp 207-227. 2000.

[JJ94a] J. R. Josephson. *Plausibility*. Appendix B. In Abductive Inference: Computation, Philosophy, Technology. J. R. Josephson and S. G. Josephson (eds). New York: Cambridge University Press. pp 266 - 272. 1994.

[JJ94b] J. R. Josephson and S. G. Josephson (eds) *Abductive Inference: Computation, Philosophy, Technology*. New York: Cambridge University Press. 1994.

[Kel71] J. Kelly. Negative response relevance feedback. The SMART retrieval system - experiments in automatic document processing. (G. Salton ed). Chapter 19. pp 403-412. 1971.

[KP94] C. S. G. Khoo and D. C. C. Poo. *An expert system approach to online catalog subject searching*. Information Processing and Management. **30.** 2. pp 223-238. 1994.

[KB96] J. Koenemann and N. J. Belkin. *A case for interaction: a study of interactive information retrieval behavior and effectiveness*. Proceedings of the Human Factors in Computing Systems Conference (CHI'96). pp 205-212. Zurich. 1996.

[Ku91] I. O.-C. Ku. *Theoretical and empirical perspectives on the abductive confidence function*. Master's thesis. Department of Computer and Information Science. The Ohio State University. Columbus. 1991.

[Kuh91] C.C. Kuhlthau. *Inside the search process: information seeking from the user's perspective*. Journal of the American Society of Information Science. **42**. 5. pp 361 - 371. 1991.

[Kuh93] C.C. Kuhlthau. *Principle for uncertainty for information seeking*. Journal of Documentation. **49**. 4. pp 339-355. 1993.

[Lal96] M. Lalmas. *Modelling information retrieval with Dempster-Shafer's theory of evidence: a case study*. ECAI Workshop on Uncertainty in Information Systems: Questions of Viability. pp 29-36. Budapest. 1996.

[LaBr98] M. Lalmas and P.D. Bruza. *The use of logic in information retrieval modelling.* Knowledge Engineering Review. **13**. 2. pp 1-33. 1998.

[LR98] M. Lalmas and I. Ruthven. *Representing and retrieving structured documents using the Dempster-Shafer Theory of Evidence: modelling and evaluation.* Journal of Documentation. **54.** 5. pp 529-565. 1998

[Lea94] D. B. Leake. *Goal-based explanation evaluation*. Cognitive Science. **15**. 4. pp 509 - 545. 1991.

[Lea95] D. B. Leake. *Abduction, experience and goals: a model of everyday abductive explanation.* The Journal of Experimental and Theoretical Artificial Intelligence. 1995.

[Lip91] P. Lipton. *Inference to the best explanation.* Routledge. 1991.

[Lee98] J. H. Lee. *Combining the evidence of different relevance feedback methods for information retrieval.* Information Processing and Management. **34**. 6. pp 681-691. 1998.

[Lew92] D. D. Lewis *An Evaluation of phrasal and clustered representations on a text categorization task*. Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 37 – 50. Copenhagen. 1992.

[LO98] E. Lagergren and P. Over. *Comparing interactive information retrieval systems across sites: the TREC-6 interactive track matrix experiment.* Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Melbourne. pp 164-172. 1998.

[MVR97] M. Magennis and C. J. van Rijsbergen. *The potential and actual effectiveness of interactive query expansion*. Proceedings of the Twentieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 324-331. Philadelphia. 1997

[Mar64] M. E. Maron. *Mechanized documentation: the logic behinf a proababilistic interpretation.* Statistical Association Methods For Mechanized Documentation. National Bureau of Standards Miscellaneous Publications 269. (M. E. Stevens, V. E. Guiliano and L. B. Heilprin. eds). pp 9-13. 1964.

[MK60] M. E. Maron and J. L. Kuhns. *On relevance, probabilistic indexing and information retrieval.* Journal of the Association for Computing Machinery. **15**. pp 8-36. 1960. Reprinted in Readings in Information Retrieval. K. Sparck Jones and P Willet (eds). Morgan Kaufman. pp 39-46. 1997.

[Mar82] W. A. Martin. *Helping the less experienced user*. Proceedings of the 6th International Online Meeting. pp 67-76. 1982.

[MAGI71] D. Michelson, M. Amreich, G. Grissom and E. Ide. *An experiment in the use of bibliographic data as a source of relevance feedback in information retrieval* The SMART retrieval system - experiments in automatic document processing. (G. Salton ed). Chapter 22. pp 430-441. 1971.

[MFU99] Y. Miyata, T. Furuhashi and Y. Uchikawa. *Query Expansion for Information Retrieval Support System Using Fuzzy Abductive Inference*. The Transactions of The Institute of Electrical Engineers of Japan. **119-C**. 5. pp 632-637. 1999.

[Mull98] A. Müller. *A flexible framework for multimedia information retrieval.* Information Retrieval: Uncertainty and Logics - Advanced models for the representation and retrieval of information. (F. Crestani, M. Lalmas and C. J. van Rijsbergen (eds). Kluwer Academic Publishers. Chapter 5. pp 97-127. 1998.

[MT94] A. Müller and Ulrich Thiel. *Query Expansion in an Abductive Information Retrieval System*. Proceedings of RIAO Conference on Content-Based Multimedia Information Access. New York. pp 461-480. 1994.

[Nie89] J. Nie. *An information retrieval model based on modal logic.* Information Processing and Management. **25**. 5. pp 471-490. 1989.

[NM90] H. Ng and R. Mooney. *On the role of coherence in abductive explanation*. Proceedings of the eight national conference on artificial intelligence. pp 337 - 342. Boston. 1990.

[Occ01] F. Heylighen. *Occam's Razor*. Principia Cybernetica Web (Principia Cybernetica, Brussels), F. Heylighen, C. Joslyn and V. Turchin (eds). http://pespmc1.vub.ac.be/OCCAMRAZ.html. 2001. Date visited 18 September 2001.

[OR94] P. O'Rorke. *Abduction and explanation-based learning: case studies in diverse domains*. Computational Intelligence. **10**. 3. pp 295 - 330. 1994.

[OSG+96] J. H. Obradovich, P. J. Smith, S. Guerlain, J. W. Smith, S. Rudmann, L. Sachs, J. Svirbley, M. Kennedy and P. Stroh. *Design Concepts for an Instructional Tool: Teaching Abductive Reasoning in Antibody Identification*. CHI Conference Companion. pp 13 - 14. 1996

[Ov98] P. Over. *TREC-6 interactive track report*. Proceedings of the Sixth Text Retrieval Conference. Nist Special Publication 500-240. Gaitherburg. pp 73-82. 1998.

[PB96] F. Paradis and C. Berrut. *Experiments with theme extraction in explanatory texts*. Second International Conference on Conceptions of Library and Information Science (CoLIS 2). pp 433-437. Copenhagen. 1996.

[Pau93] G. Paul. *Approaches to abductive reasoning: an overview*. Artificial Intelligence Review. **7**. 2. pp 109 - 152. 1993.

[Paw82]. Z Pawlak. *Rough sets.* International Journal of Information and Computing Science. **5**. 11. pp 341 - 356.1982.

[Pei98] C. S. Peirce. *Reasoning and the logic of things: The Cambridge conferences lectures of 1898*. K. L. Ketner (ed.). Harvard University Press. 1992.

[Pei31] C. S. Peirce. Collected papers of Charles Sanders Peirce. C. Hartshorne and P. Weiss (eds.) Harvard University Press. 1931-1958.

[Pei58a] C. S. Peirce. *Charles S. Peirce: Selected writings*. P. P. Wiener (ed.). Dover Publications, Inc. 1966.

[Pet89] T. A. Peters. *When smart people fail: an analysis of the transaction log of an online public access catalog*. The Journal of Academic Librarianship. **15**. 5. pp 267-273. 1989.

[Por80] M. F. Porter. *An algorithm for suffix stripping*. Program. **14**. pp 130-137. 1980.

[PG88] M. Porter and V. Galpin. *Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute*. Program. **22**. 1. pp 1 - 20. 1988.

[RS86] V. J. Rayward-Smith. *A first course in computability*. Blackwell Scientific Publications. 1986.

[Rob77] S. E. Robertson. *The probability ranking principle in IR*. Journal of Documentation,. **33**. 4. pp 294-304. 1977.

[Rob86] S. E. Robertson. *On relevance weight estimation and query expansion*. Journal of Documentation. **42**. 3. pp 182-188. 1986.

[Rob90] S. E. Robertson. *On term selection for query expansion.* Journal of Documentation. **46**. 4. pp 359-364. 1990.

[RB78] S. E. Robertson and N. J. Belkin. *Ranking in principle*. Journal of Documentation. **34**. 2. pp 93-100. 1978.

[RSP76] S E Robertson and K Sparck Jones. *Relevance weighting of search terms.* Journal of the American Society for Information Science. **27**. 3. pp 129-146. 1976.

[RW94] S. E. Robertson and S. Walker. *Some simple effective approximations to the 2 Poission model for probabilistic weighted retrieval*. Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 232-241. Dublin. 1994.

[RWH+93] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull and M. Lau. *Okapi at TREC*. Proceedings of the First Text Retrieval Conference (TREC-1). NIST special publication 500-207. (D. K. Harman ed). pp 21-30. 1993.

[Roc71] J J Rocchio. *Relevance feedback in information retrieval* The SMART retrieval system - experiments in automatic document processing. (G. Salton ed). Chapter 14. pp 313-323. 1971.

[RL99] I. Ruthven and M. Lalmas. *Selective relevance feedback using term characteristics*. CoLIS 3, Proceedings of the Third International Conference on Conceptions of Library and Information Science. Dubrovnik. 1999.

[RTJ01] I. Ruthven, A. Tombros and J. Jose. *A study on the use of summaries and summary-based query expansion for a question-answering task.*. 23rd BCS European Annual Colloquium on Information Retrieval Research (ECIR '01). Darmstadt. 2001.

[Saf87] A. Saffioti. *An AI view of the treatment of uncertainty.* The Knowledge Engineering Review. **2**. 2. pp 75-97. 1987.

[Sal71] G Salton (ed). The SMART retrieval system - experiments in automatic document processing. 1971.

[SB90] G. Salton and C. Buckley. *Improving retrieval performance by relevance feedback.* Journal of the American Society for Information Science. **41**. 4. pp 288-297. 1990.

[Sal83] G. Salton and M. McGill. *Introduction to modern information retrieval*. McGraw-Hill Book Company. New York. 1983.

[SH93] S. Schocken and R. A. Hummel. *On the use of the Dempster Shafer model in information indexing and retrieval applications*. International Journal of Man-Machine Studies. **39**. pp 1-17. 1993.

[Seb83] T. A. Sebeok. *One, two, three spells UBERTY.* The sign of three: Dupin, Holmes and Pierce. U. Eco and T A Sebeok (eds). Chapter 1. pp 1 - 10. Bloomington: Indiana University Press. 1983.

[Seb94] F. Sebastiani. *A probabilistic terminological logic for modelling information retrieval.* Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 122-130. Dublin. 1994.

[Sch91] L. Schamber. *Users' criteria for evaluation in a multimedia environment*. Proceedings of the 59th Annual Meeting of the American Society for Information Science. **33**. pp 3-9. Medford. 1991.

[Sha76] G. Shafer. *A mathematical theory of evidence*. Princeton University Press. 1976.

[Shak88] W. Shakespeare. *The comedy of errors*. Cambridge University Press. 1988.

[Shak90] W. Shakespeare. *Macbeth*. Oxford's World Classics. Oxford University Press. 1990.

[Sim96] B. Simonnot. *Modélisation multi-agents d'un système de recherche d'information multimédia à forte composante vidéo*, (Multi-Agent Modelling of a multimedia information retrieval system for still images and videos collections). PhD thesis. Henri Poincaré University. 1996.

[Sme98] A. Smeaton. *Independence of contributing retrieval strategies in data fusion for effective information retrieval.* Proceedings of the 20th BCS-IRSG Colloquium. Springer-Verlag electronic Workshops in Computing. Grenoble. 1998.

[SJ72] K. Sparck Jones. *A statistical interpretation of term specificity and its application in retrieval.* Journal of Documentation. **28**. 1. pp 11-20. 1972

[SJ79] K. Sparck Jones. *Search term relevance weighting given little relevance information*. Journal of Documentation. **35**. 1. pp 30 -48. 1979.

[SJ99] K Sparck Jones. *IR lessons for AI*. Searching for Information: Artificial Intelligence and Information Retrieval Approaches. Glasgow. 1999.

[SSJ+00a] K. Sparck Jones, S. Walker and S. E. Robertson. *A probabilistic model of information retrieval: development and comparative experiments – Part 1*. Information Processing and Management. **36**. 6. pp 779-808. 2000.

[SSJ+00b] K. Sparck Jones, S. Walker and S. E. Robertson. *A probabilistic model of information retrieval: development and comparative experiments – Part 2*. Information Processing and Management. **36**. 6. pp 809-840. 2000.

[Spi96] A. Spink. *Study of interactive feedback during mediated information retrieval* Journal of the American Society for Information Sciece. **47**. 8. pp 603-609. 1996.

[SGB98] A. Spink, H. Greisdorf and J. Bateman. *From highly relevant to not relevant: examining different regions of relevance*. Information Processing and Management. **34**. 5. pp 599-621. 1998.

[SL96] A. Spink and R. M. Losee. *Feedback in information retrieval*. In: M. Williams (Ed.), Annual Review of Information. Science and Technology, **31**. pp 33-78. 1996.

[SW99] A. Spink, and T. D. Wilson. *Toward a theoretical framework for information retrieval (IR) evaluation in an information seeking context*. Mira '99: Evaluating Information Retrieval. S. Draper, M. Dunlop, I. Ruthven and C. J. van Rijsbergen (eds). electronic Workshops in Computing. 1999.

[Su94] L. T. Su. *The relevance of recall and precision in user evaluation*. Journal of the American Society for Information Science. **45**. 3. pp 207-217. 1994.

[TdSLM93] W. Teixeira da Silva and R. Luiz Milidiu. *Belief function model for information retrieval.* Journal of the American Society for Information Science. **44**. 1. pp 10-18. 1993.

[TS94] P. Thagard and C. Shelly. *Limitations of current formal models of abductive reasoning*. 1994.

[TS97] P. Thagard and C. Shelley. *Abductive reasoning: Logic, visual thinking, and coherence*. In: M.-L. Dalla Chiara et al (eds), Logic and Scientific methods. Dordrecht: Kluwer, p.413-427. 1997.

[TA88] M. A. Tiamiyu and I. Y. Ajiferuke. *A total relevance and document interaction effects model for the evaluation of information retrieval processes*. Information Processing and Management. **24**. 4. pp 391-404. 1988.

[TS98] A. Tombros and M. Sanderson. *The advantages of query-biased summaries in Information Retrieval* Proceedings of the Twenty-First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2-10. Melbourne. 1998.

[TVR01] A. Tombros and C. J. van Rijsbergen. *Query-sensitive similarity measures for the calculation of interdocument relationships*. Proceedings of the 10th ACM CIKM Conference. Atlanta. 2001.

[TRG91] S. Tuhrim, J. Reggia, and S. Goodall. *An experimental study of criteria for hypothesis selection*. The Journal of Experimental and Theoretical Artificial Intelligence. 3. 129 - 144. 1991.

[Vak00a] P. Vakkari. *Cognition and changes of search terms and tactics during task performance.* Proceedings of RIAO Conference on Content-Based Multimedia Information Access. Paris. pp 894-907. 2001.

[Vak00b] P. Vakkari. *Relevance and contributing information types of searched documents in task performance*. Proceedings of the twenty-third annual international ACM SIGIR Conference on Research and development in information retrieval. pp 2-9. Athens. 2000.

[VR79] C. J. van Rijsbergen. *Information retrieval*. Butterworths. 2nd edition. 1979.

[VR86] C. J. van Rijsbergen. *A non-classical logic for information retrieval*. The Computer Journal. **29**. 6. pp 48 -485. 1986.

[VR89] C. J. van Rijsbergen. *Towards an information logic*. Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 77-86. Cambridge. 1989.

[VRHP81] C J van Rijsbergen, D. Harper and M. Porter. *The selection of good search terms*. Information Processing and Management. **17**. 2. pp 77-91. 1981.

[VR92] C. J. van Rijsbergen. *Probabilistic retrieval revisited.* The Computer Journal. **35**. 3. pp 291-298. 1992.

[VH96] E. M. Voorhees and D. Harman. *Overview of the fifth Text REtrieval Conference (TREC-5)*. Proceedings of the 5th Text Retrieval Conference. pp 1-29. Nist Special Publication 500-238. Gaitherburg.1996.

[VH00] E. H. Voorhees and D. Harman. *Overview of the sixth text retrieval conference (TREC-6)*. Information Processing and Management. **36**. 1. pp 3 - 35. 2000.

[Wil86] O. Wilde. *The importance of being Earnest*. Penguin Plays. Penguin. 1986.

[WB95] S. Willie and P. Bruza. *Users' models of the information space: the case for two search models.* Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle. pp 205-210. 1995

[Wir98] U. Wirth, *What is Abductive Inference*? Encyclopaedia of Semiotics, ed. by Paul Bouissac, Oxford University Press. 1998.

[Zad94] W. Zadrozny. *Is there a prototypical rule of abduction? (Yes, e.g. in proximity based explanations)*. Journal of experimental and theoretical artficial intelligence. 6. pp 147 - 162. 1994.

[ZG00]. X. Zhu and S. Gauch. *Incorporating quality metrics in centralized/distributed information retrieval on the WWW*. Proceedings of the Twenty-Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 288-295. Athens. 2000.

Abduction, explanation and relevance feedback

Ian Ruthven

Department of Computing Science
University of Glasgow

UNIVERSITY
of
GLASGOW

Submitted for the Degree of Doctor of Philosophy
at the University of Glasgow
31st October 2001
Volume 2

Appendices

Appendix A

Retrieval models

A.1 Boolean model

The first operational IR retrieval model was the Boolean model, based on Boolean logic. In this model queries are keywords combined, by the user, with the conjunctive (AND), disjunctive (OR) or negation (NOT) operators. This is an *exact-match* model: the system only retrieves those documents that exactly match the user's query formula. For example, for the query *'information* AND *retrieval* AND *system'* the system will return all documents that contain the three words *'information*', '*retrieval'* and '*system'*, whereas the query *'information* OR (*retrieval* AND *system*)' will return those documents that contain the word *'information'* and those documents that contain both *'retrieval'* and *'system'*.

The Boolean model has been used in a large number of on-line public access catalogue (OPAC) systems but has been shown to demonstrate a number of difficulties. Firstly, traditional Boolean systems do not use term weights and consequently return the complete set of documents that match the query as an unordered set. This means the users may have to add or remove terms, or generate more complex query expressions to reduce the set of retrieved documents to a manageable size. Secondly, although expert users can perform effective searches with Boolean systems, inexperienced or novice users can find it difficult to issue good queries, [Pet89]. One cause of this is that Boolean operators do not always correspond to their English equivalents, e.g.

> "in English 'A and B' would typically refer to *more* entities than would 'A' alone, whereas in the information retrieval usage it refers to *fewer* documents.", [Coop88].

Some attempts have been made to make the Boolean operators less rigid, e.g. by weighting index terms, or using a looser interpretation of the Boolean operators. A summary of these approaches is given in [FBK+92].

Willie and Bruza, [WB95], argue that the problems with interacting with Boolean systems are not only a matter of the formal query language but a *conceptual* problem: the Boolean model does not lend itself to supporting how users think about searching and their individual search techniques. A further problem with Boolean systems is that the order in which operators are applied may not be consistent across systems, resulting in the fact that different systems may retrieve different documents for the same query, [Borg97]. Nevertheless Boolean systems do remain popular with users, perhaps because of the explicit control that is offered by these systems to the user. Web search engines often allow Boolean-style querying performed on an underlying best-match model (see sections A.2 - A.4).

Harman, [Har92a], suggests two possible methods for implementing RF on Boolean systems. The first is to present the user with a list of possible new query terms. These can be chosen, for example, by term distribution in the relevant documents. This means selecting those terms that appear more often in the relevant than non-relevant documents and which would be useful to include in a new query.

The second approach is for the system to automatically modify Boolean queries. An example of the latter type of query modification can be found in the system proposed by Khoo and Poo, [KP94], which is intended to automatically modify both the terms and the Boolean connectives of queries based on the documents marked relevant by a user.

An alternative to exact-match systems, such as the Boolean model, are *best-match* systems. These systems use term weights, such as *tf* and *idf*, to *rank* documents in decreasing order of matching score or estimation of relevance. The two most common best-match models are the *vector-space model*, which orders documents in decreasing *similarity* of query and document, [Sal71], and the *probabilistic model,* [RSJ76], which orders documents based on an estimate of the *probability of relevance* of a document to a query. In section A.2 I discuss the vector space model, and in section A.3 I discuss the probabilistic model.

A.2 Vector space model

In the vector-space model, a document is represented by a vector of *n* weights, where *n* is the number of unique terms in the document collection. Figure A.1 shows an example vector

where x_i is the weight[115] of the *i*th term in document *x* if *x* contains the term, and 0 if the term is not present in *x*.

$$x = (x_1, x_2, ..., x_n)$$

Figure A.1: Document vector

Queries are also represented as a vector of length *n*, and the similarity of the document vectors to a query vector gives a retrieval score to each document, allowing comparison and ranking of documents. A range of similarity measures exist to calculate this similarity, e.g. DICE, inner product, cosine correlation, [VR79, Chap 3]. Equation A.1 shows the cosine correlation, one of the more common vector-space matching functions.

$$\cos(doc_i, query_j) = \frac{\sum_{k=1}^{n}(term_{ik} \cdot qterm_{jk})}{\sqrt{\sum_{k=1}^{n}(term_{ik})^2 \cdot \sum_{k=1}^{n}(qterm_{jk})^2}}$$

Equation A.1: Cosine correlation between document doc_i and $query_j$

Unlike the Boolean model, which retrieves documents according to the query terms and query connectives, in the best-match models all documents that contain at least one query term will receive a non-zero score; the highest score going to documents that contain all the query terms. Documents that contain only some of the query terms will be ranked according to the sum of the weights of the query terms they contain. The documents that contain more query terms or contain query terms with a higher discriminatory power (term weight) will be retrieved above those that contain fewer query terms or query terms with lower weights. Similarity is then a function of term overlap between query and document, and the weights assigned to the terms.

Rocchio, [Roc71], is generally credited with the first formalisation of a RF technique, developed on the vector space model. In [Roc71] he defines the problem of retrieval as that of defining an optimal query; one that maximises the difference between the average vector of the relevant documents and the average vector of the non-relevant documents.

---

[115]Some implementations of the vector space model use 1 if a term occurs in a document, 0 if it does not occur. Most implementations will use some form of *tf*idf* weighting and some form of length normalisation will usually be performed to avoid retrieval bias towards long documents.

As discussed in Chapter One section 1.1, it may not always be possible for a user to submit such an optimal query, so RF is required to bring the query vector closer to the mean of the relevant documents, and further from the mean of the non-relevant documents. This is accomplished by the addition of query terms and by the reweighting of query terms to reflect their utility in discriminating relevant from non-relevant documents.

Rocchio's original formula for defining a new query vector in the vector space model, is as follows, Equation A.2

$$Q_1 = Q_0 + \frac{1}{n_1}\sum_{i=1}^{n_1} R_i - \frac{1}{n_2}\sum_{i=1}^{n_2} S_i$$

Equation A.2: Rocchio's original formula for modifying a query based on relevance information
where Q_o = initial query vector, Q_1 = new query vector, n_1 = number of relevant documents, n_2 = number of non-relevant documents, R_i = vector for the ith relevant document, S_i = vector for the ith non-relevant document

The new query vector is the original query vector plus the terms that best differentiate the relevant documents from the non-relevant documents. A modified query contains new terms (from the relevant documents) and has new weights attached to the query terms. If the weight of a query term drops to zero or below, it is removed from the query.

This formula is capable of being constrained further, e.g. by weighting the original query vector so that the original query terms contribute more to the modified query than the new query terms or by varying the amount of feedback considered. A variation of this formula was tested experimentally with positive results on the SMART retrieval system, [Roc71].

The small size of the document collection used in Rocchio's experiments meant that certain modifications had to be made to the formula. For example, although Rocchio tried to keep the size of the relevant and non-relevant feedback sets identical, this was not always possible. In addition a term was only considered if it was one of the original query terms or if it appeared in more relevant than non-relevant documents *and* in more than half the relevant documents. These modifications highlight the recurring difficulty of aligning theory with experimental practice.

Ide, [Ide71], extended the SMART relevance feedback experiments, examining different aspects of RF, such as only using relevant documents for feedback, varying the number of

documents used for RF, and using non-relevant documents. She found that using only relevant documents for feedback or varying the number of documents used at each iteration of feedback gave inconclusive or poor results.

Her third strategy was a variation of Rocchio's original formula, using only the first non-relevant document found, s_1. The formula used by Ide is shown in Equation A.3.

$$Q_1 = Q_0 + \sum_{i}^{n_r'} r_i - s_i$$

Equation A.3: Ide-dec-hi formula for modifying a query based on relevance information where Q_0 = initial query vector, Q_1 = new query vector, n_r = number of relevant documents, r_i = vector for the ith relevant document, s_i = vector for the first non-relevant document

This was compared against Rocchio's original formula. Although this technique, the *Ide-dec-hi* formula, did not improve results greatly it was more consistent in improval; improving the performance of more queries.

A further version of the Ide scheme, the Ide regular, [IdS71], scheme, uses all retrieved, non-relevant documents. The Ide-regular is based on the Rocchio formula but omits the normalisation of the relevant and non-relevant documents by the number of relevant/non-relevant documents. Equation A.4 shows the Ide-regular formula. This version of the Rocchio formula uses more non-relevant information but still generally performs less well than the Ide-dec-hi, [SB90].

$$Q_1 = Q_0 + \sum_{i=1}^{n_1} R_i - \sum_{i=1}^{n_2} S_i$$

Equation A.4: Ide-regular

A common modification to the vector space RF formulae, e.g. [IdS71], is to weight the relative contribution of the original query, relevant and non-relevant documents to the RF process. In Equation A.5, the α, β and γ values specify the degree of effect of each component.

$$Q_1 = \alpha.Q_0 + \beta/n_1 \sum_{i=1}^{n_1} R_i - \gamma/n_2 \sum_{i=1}^{n_2} S_i$$

Equation A.5: Rocchio modified relevance feedback formula

Various other suggestions as to how to use feedback information came out of the early SMART experiments. These include using negative feedback – feedback information based on what the user considers to be not relevant, [Kel97], query splitting - generating separate queries to detect different aspects of relevant documents [BKL71, IdS71], the use of bibliographic data - authors, citations, etc., [MAGI71], and modifying the document representation rather than the document representation, [FMW71]. Although these techniques did not show significant improvements in retrieval performance, some of the ideas are still being actively investigated, for example the use of negative feedback and forms of document modification, which will be discussed in sections A3 and A.4 respectively.

A.3 Probabilistic model

In the probabilistic model, suggested by Maron and Kuhns, [MK60], and developed by amongst others, Robertson and Sparck Jones, [RSJ76], and Van Rijsbergen, [VR79], documents and queries are also viewed as vectors but the vector space similarity measure is replaced by a probabilistic matching function. The probabilistic model is based on estimating the *probability* that a document will be relevant to a user, given a particular query. The higher this estimated probability, the more likely the document is to be relevant to the user[116]. This is instantiated in the probabilistic ranking principle, [Rob77].

> "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 relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."

The estimated probability of relevance can be expressed as $P_q(rel\,|\,x)$, the probability of relevance given a document x and a query q. This probability can be used to decide whether or not to retrieve a document: if $P_q(rel\,|\,x) = 0$ then the probability of relevance given x is 0, and x should not be retrieved[117].

---

[116]The probabilistic model measures the *probability* of relevance, i.e. the probability that a document will be relevant, not the *degree* of relevance as is sometimes suggested. A good discussion of the difference between these two notions is found in [RB78].

[117]In an operational system $P_q(rel|\,x)$ will generally only equal 0 if x does not contain any query terms. This rule then decides only to retrieve those documents that contain at least one query term.

This can be refined by also considering the probability of non-relevance given x and q, $P_q(\overline{rel}\,|\,x)$. If $P_q(rel\,|\,x) > P_q(\overline{rel}\,|\,x)$ then it can be asserted that the probability of relevance is greater than the probability of non-relevance and hence x should be retrieved[118]. Thresholds may also be used, i.e. the difference between the probability of relevance and the probability of non-relevance must be greater than some threshold value before x is retrieved, (($P_q(rel\,|\,x)$ - $P_q(\overline{rel}\,|\,x)$) > *threshold)*. In this case *threshold* is a value set by the user or system, in order to further restrict the retrieval function.

Having decided which documents to retrieve, the odds of relevance to non-relevance, Equation A.6, can be used as a document *ranking* function: the higher the ratio of the probability of relevance to non-relevance, given x, then the more likely document x is to be relevant to a user.

$$\frac{P_q(rel\,|\,x)}{P_q(\overline{rel}\,|\,x)}$$

Equation A.6: Odds of relevance to non-relevance for document x and query q

Bayes theorem, [Bay63], can be used to calculate $P_q(rel\,|\,x)$ and $P_q(\overline{rel}\,|\,x)$. Equation A.7 demonstrates this for the relevance case.

$$P_q(rel\,|\,x) = \frac{P_q(x\,|\,rel)P_q(rel)}{P(x)}$$

Equation A.7: Calculation of $P_q(rel\,|\,x)$ through Bayesian inversion

where $P_q(rel)$ is the prior probability that *any* document in the collection is relevant to q
$P_q(x\,|\,rel)$ is the probability of observing document x given relevance information
$P(x)$ is the probability of observing document x irrespective of relevance

After Bayesian inversion and deletion of $P(x)$ (which is identical for both the relevance and non-relevance case), the odds function from Equation A.7 turns into Equation A.8**a**.

The probability of relevance, $P_q(rel)$, and the probability of non-relevance, $P_q(\overline{rel})$, are identical for all x's, That is when we use the odds in Equation A.6 to rank documents, the

[118]In the case where the two probabilities are equal, it is arbitrarily decided that x is non-relevant [VR79].

ranking is dependent on the values of the probabilities $P_q(x \mid rel)$ and $P_q(x \mid \overline{rel})$, not on the values $P_q(rel)$ and $P_q(\overline{rel})$. We can therefore eliminate these elements and arrive at the odds in Equation A.8**b**. This is then the odds of observing *x* given relevance or non-relevance.

$$\frac{P_q(x \mid rel)P_q(rel)}{P_q(x \mid \overline{rel})P_q(\overline{rel})} \qquad \frac{P_q(x \mid rel)}{P_q(x \mid \overline{rel})}$$

a **b**

Equation A.8: Odds of relevance, or non-relevance, having observed document *x*

The odds in Equation A.8 refers to the probability of relevance, and non-relevance, after viewing the actual document text rather than the vector representation of the document. That is, it measures the odds of relevance to non-relevance based on the content of the document and is independent of the document representation. This means that the model can be used for many different types of document indexing but it also means that Equation A.8 must be ultimately be expressed as a retrieval function based on the specific document indexing technique used to represent the documents.

There are many probabilistic models based on the model outlined so far in this section. In the remainder of this section I shall describe the transformation from Equation A.8 to a function based on the term-based representation outlined in Chapter One, section 1.2.1. Specifically the discussion will be based on the probabilistic model known as the Binary Independence Model, as this is the most traditional variant of the overall probabilistic approach. This model was one of the first probabilistic models of IR, and will be used as an example of how the theoretical model is transformed into an actual retrieval model.

Before converting Equation A.8 into an equation that can be estimated based on the probability of relevance and non-relevance of the terms in document *x*, it is necessary to consider how the probabilities of relevance and non-relevance interact. In particular, two aspects of retrieval are important: the independence of terms and what information is used to order documents.

The probabilistic model assumes that terms are distributed independently of other terms, that is the probability of seeing term *t* in a document is not affected by seeing term *s* in the same

document. This is a simplifying assumption that reduces the computational complexity of the model.

However it is necessary to define over what sets the independence holds. Two versions of the *independence assumption* were proposed in [RSJ76]. Both term independence assumptions assume that terms, query terms in particular, are distributed independently in the set of relevant documents: the probability of a term appearing in the relevant documents is not dependent to the probabilities of other terms appearing in the relevant documents. The two assumptions differ in whether the relevant document set should be distinguished from the whole document collection or only from the set of non-relevant documents.

"*Independence assumption I1*: The distribution of terms in relevant documents is independent and their distribution in all documents is independent"

"*Independence assumption I2*: The distribution of terms in relevant documents is independent and their distribution in irrelevant[119] documents is independent"

These two versions of the independence assumption are important in distinguishing whether we should measure the difference in the probability of a term's occurrence against the non-relevant documents (*I2*) or against its probability of occurrence in the collection as a whole (*I1*).

The probabilistic model ranks documents according to their probability of being relevant to a query - the *ordering principle*. Two versions of this principle distinguish between the case where this probability is estimated based only on the *presence* of query terms within a document or presence *and* absence of the terms.

"*Ordering principle O1*: That probable relevance is based on the presence of search terms in documents"

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

Four weighting schemes, F_1-F_4, can be derived from the combination of the two variants of the independence assumption and the ordering principle, Table A.1.

---

[119] The labels *irrelevant* and *non-relevant* are treated as synonymous in this thesis.

In [RSJ76] each of these possible strategies was instantiated to give an actual method for weighting a query term, summarised in Figure A.2. The weighting methods themselves are based on a contingency table, Table A.2, which converts the probability values into values that can be calculated from term occurrence information.

	Independence assumption *I1*	Independence assumption *I2*
Ordering principle *O1*	**F_1**	**F_2**
Ordering principle *O2*	**F_3**	**F_4**

Table A.1: Term weighting functions derived from the combination of independence assumptions and ordering principles

	rel	$\overline{rel}$	
$x_i = 1$	*r*	*n-r*	*n*
$x_i = 0$	*R-r*	*N-n-R+r*	*N-n*
	R	*N-R*	

Table A.2: Contingency table to calculate term weights

where r = the number of relevant documents containing term x_i
n = the number of documents containing term x_i
R = the number of relevant documents for query q
N = the number of documents in the collection

$$w_{x_i} = \log \frac{P_q(x_i \mid rel)}{P_q(x_i)} = \log \frac{(r/R)}{(n/N)}$$

F_1

$$w_{x_i} = \log \frac{P_q(x_i \mid rel) P_q(rel)}{P_q(x_i \mid \overline{rel}) P_q(\overline{rel})} = \log \frac{(r/R)}{((n-r)/(N-R))}$$

F_2

$$w_{x_i} = \log \frac{P_q(x_i \mid rel) / P_q(\overline{x_i} \mid rel))}{P(x_i)/(P(\overline{x_i})} = \log \frac{r/(R-r)}{n/(N-n)}$$

F_3

$$w_{x_i} = \log \frac{P_q(x_i \mid rel) / P_q(\overline{x_i} \mid rel)}{P_q(x_i \mid \overline{rel}) / P_q(\overline{x_i} \mid \overline{rel})} = \log \frac{r/(R-r)}{(n-r)/(N-n-R+r)}$$

F_4

Figure A.2: Term weighting functions F_1 - F_4

Each of the four term weighting functions is a ratio of two proportions[120]:

- F_1 is the ratio of the proportion of relevant documents in which the query term *t* occurs (*ordering principle O1*) to the proportion of all documents in which *t* occurs (*independence assumption I1*).

- F_2 is the ratio of the proportion of relevant documents in which the query term *t* occurs (*ordering principle O1*)) to the proportion of all non-relevant documents in which *t* occurs (*independence assumption I2*).

F_3 and F_4 both use odds

- F_3, the ratio of 'relevance odds' (the ratio of relevant documents containing term *t* and relevant documents not containing *t* - *ordering principle O2*) and 'collection odds' (the ratio of documents containing *t* and documents not containing *t* - *independence assumption I1*).

- F_4 is the ratio of 'relevance odds' - *ordering principle O2* and 'non-relevance odds' (the ratio of non-relevant documents containing *t* and the non-relevant documents not containing *t* - *independence assumption I2*).

In [RSJ76], Robertson and Spark Jones used the four term weighting schemes to carry out two sets of experiments. The first set was based on *retrospective weighting.* This involves deriving optimal weights to retrieve the relevant documents already found – the *known relevant set*. The second group of experiments were based on *predictive weighting*. Predictive weighting uses the weights from the retrospective stage to retrieve new documents. If the known relevant set is a representative sample of all relevant documents, then predictive

[120]It may be the case, especially when using small samples, that some of the values in the weights could be zero, resulting in error when taking logs. The solution is to add 0.5 to each cell in the numerator and denominator of each function. An alternative is to use the ratio, [Rob86], n_i/N to replace the 0.5 correction factor.

weighting should be better at retrieving unseen relevant documents than the original term weights. Naturally, it is the latter case that is mainly of interest as RF is intended to retrieve relevant documents that the user has not yet seen.

All functions outperformed no relevance weighting, or the *idf* function. F_1 and F_2, and F_3 and F_4 perform within the same range with F_3 and F_4 outperforming F_1 and F_2, and F_4 slightly outperforming F_3. This confirms Robertson and Sparck Jones' intuition that ordering principles *O2* is correct and that it is necessary to consider both presence and absence of query terms. No conclusive evidence was provided to distinguish between the two versions of the independence assumption, however Robertson and Sparck Jones favour the second, *I2*, assumption as the more realistic assumption.

Given that the preferred weighting scheme is F_4, the odds function in Figure A.2 (Equation A.8**a**) can be converted to that of Equation A.8**b** by eliminating the division operators. By noting that $P_q(\overline{x_i} \mid rel) = 1 - P_q(x_i \mid rel)$, and $P_q(\overline{x_i} \mid \overline{rel}) = 1 - P_q(x_i \mid \overline{rel})$ it is possible to convert the representation of F_4 in Figure A.2 to that in Equation A.9**c**.

$$w_{x_i} = \log \frac{P_q(x_i \mid rel) / P_q(\overline{x_i} \mid rel)}{P_q(x_i \mid \overline{rel}) / P_q(\overline{x_i} \mid \overline{rel})} = \log \frac{P_q(x_i \mid rel) P_q(\overline{x_i} \mid \overline{rel})}{P_q(x_i \mid \overline{rel}) P_q(\overline{x_i} \mid rel)} = \log \frac{P_q(x_i \mid rel)(1 - P_q(x_i \mid \overline{rel}))}{P_q(x_i \mid \overline{rel})(1 - P_q(x_i \mid rel))}$$

a **b** **c**

Equation A.9: Term weighting function based on term's distribution in relevant and non-relevant documents
where w_{x_i} = the weight of term x_i

This equation (Equation A.9**c**), which expresses the F_4 function solely as a factor of the *presence* of a term in the relevant and non-relevant documents, can alternatively be represented as in Equation A.10.

$$w_{x_i} = \log \frac{p_i(1 - q_i)}{q_i(1 - p_i)}$$

Equation A.10: Term weighting function based on term's distribution in relevant and non-relevant documents
where w_{x_i} = the weight of term x_i, $p_i = P_q(x_i \mid rel)$ and $q_i = P_q(x_i \mid \overline{rel})$

The probability of relevance of a document, then, is measured as the sum of the term weights of the query terms in the document, i.e. the sum of the F_4 weights of each query term in the document.

The function in Equation A.10 was examined as a basis for ranking terms for query expansion. Robertson, [Rob90], argued that a weighting function that ranks terms for *matching* (as in Equation A.10) may not be appropriate for term *selection*[121]. That is, the degree to which a term indicates relevant material (matching) is not *necessarily* related to how well a term will improve retrieval effectiveness if added to a query (term selection).

For term selection, Robertson proposed the formula in Equation A.11, which provides a better estimate for how much a term will increase a search's effectiveness. Terms should be chosen for expansion based on the value shown in Equation A.11 rather than the *w* value from Equation A.10. Equation A.11 incorporates the *w* value of a term but also takes into account the different between the relevant and non-relevant distributions based on *i*.

$$a_i = w_i(p_i - q_i)$$

Equation A.11: Formula for ranking expansion terms based on term *t*'s distribution in relevant and non-relevant documents

where a_i = the value of term *i* for query expansion, w_i = weight of term *i* given by Equation A.9, $p_t = P_q(x_i \mid rel)$ and $q_i = P_q(x_i \mid \overline{rel})$

The formula in Equation A.11, with appropriate substitutions for p_i and q_i becomes the term ranking function in Equation A.12. This allows the calculation of Equation A.12 based on the distribution of terms within the relevant documents and the collection.

$$w_i = \log \frac{r_i / (R - r_i)}{(n_i - r_i) / (N - n_i - R + r_i)} \bullet \left(\frac{r_i}{R} - \frac{n_i - r_i}{N - R} \right)$$

Equation A.12: Term expansion ranking function

where r_i = the number of relevant documents containing term *i*
n_i = the number of documents containing term *i*
R = the number of relevant documents for query *q*
N = the number of documents in the collection

It should be made clear here that, although at each iteration of RF the same calculations are taking place (the weighting functions are identical even if that values are not), theoretically

[121] In [Rob86] Robertson also discussed the appropriateness of the 0.5 addition to the entries in the F_4 calculation, arguing that better estimations are more suitable for selecting new query terms.

different probabilities are being calculated at each iteration: the distribution that calculates $P_q(rel \mid x)$ and $P_q(\overline{rel} \mid x)$ are different at each iteration, [VR86].

The F_4 reweighting function calculates weights for terms based on their distribution in the relevant and non-relevant documents. The probabilistic model is then a retrieval model that is specifically designed for RF.

At the start of a search, of course, there is no relevance information to estimate the probabilities in Equation A.9. One standard solution to this problem is to use a weighting function that does not depend on relevance information, such as *idf*. After an initial ranking of documents and relevant information has been obtained, a function such as F_4 can be used to provide improved term weights. The use of *idf* comes from substitution of appropriate values for *r*, *R*, and *n* into the F_4 weight in Figure A.2.

It is possible to treat the query as an additional, and relevant, document and use the F_4 weight, however this will turn into something very like an *idf* weight, [RWH+93]. An alternative to this was proposed by Croft and Harper, [CH79], based on the formula in Equation A.7. This approach ranks documents by a function such as *idf*, assumes the top *n* documents are relevant, then uses these so-called *pseudo*-relevance assessments to estimate values for p_i and q_i in Equation A.10.

This fundamental approach to probabilistic modelling has been extended in many ways, in particular to incorporate within-document frequency information, [RW94]. Pertinent additions or modifications will be described, where appropriate, in later sections of this thesis. An historical overview of the probabilistic model can be found in [SSJ+a, SSJ+b].

A.4 Logical model

In [Mar64], Maron hinted at a potentially useful difference between the Boolean logic exact-match process and the process of logical implication. This difference distinguishes between the Boolean *matching* of text representations, in which the system is restricted to an exact formula, and the *inference* of information needs, by which process the system can infer more about what may be relevant than is stated in the query.

The advantages of implication or inference as the basis for a retrieval algorithm are demonstrated in the *logical modelling* approach to retrieval. This class of models originates from a proposal by Van Rijsbergen, [VR86], that relevance can be modelled as a process of *uncertain inference*. More precisely the relevance of a document representation can be

measured by the probability that the information in a document *infers* the information in a query[122], Equation A.13.

$$P(d \rightarrow q)$$

Equation A.13: Relevance measured as uncertain inference

This view was encapsulated in the logical uncertainty principle, [VR86]:

> "Given any two sentences x and y; a measure of the uncertainty of y $\rightarrow$ x related to a given data set is determined by the minimal extent to which we have to add information to the data set, to establish the truth of y $\rightarrow$ x."

That is if the information in a document, *d*, does not infer the information in a query *q* how much would *d* have to be changed to be relevant to *q*? The degree of necessary change to *d* allows the calculation of the probability of the inference.

As a simple example, if the query is about *animals* and a document mentions *dogs*, *ponies*, *cats*, but does not explicitly mention *animals,* then the document would not be retrieved by standard term-matching retrieval algorithms. By including information that *dogs*, *ponies*, and *cats* are kinds of *animals*, then it can be asserted that the document may be relevant and should be retrieved. Such an approach was taken by Lalmas, [Lal96], who used ontological relationships to express how many *transformations* or substitutions of this type would be necessary before a document's content inferred a query. In Lalmas's model, the number of substitutions gave a measure of the uncertainty associated with the inference.

The core logical models are based on non-classical logics as the classical notion of inference has several undesirable properties for retrieval, e.g. in classical logic the inference, $d \rightarrow q$, would hold even if *d* did not contain any information.

The majority of logical models of IR are based on a possible worlds semantics, in which each possible world represents a possible combination of events. One possible representation is one in which a possible world represents a possible combination of terms. For example, given a set of indexing terms $\{t_1, t_2, t_3, ..., t_{10}\}$, there would be 2^{10} worlds: a world in which all

[122]This is the most common version of the principle. Some authors have tried modelling the inverse; the degree to which the information in the query infers the information in the document $P(q \rightarrow d)$, or a combination of both measures, e.g. [Nie90]

terms are true, one in which all terms except $t_1$is true, one in which all terms except $t_1$and t_2 are true, and so on. In this representation each document and the query is associated with a world. The similarity of a document to the query is given by the *distance* between the document world and the query world[123].

Consider the example below, Figure A.3, containing two documents indexed by a number of terms drawn from the set of indexing terms $\{t_1, t_2, t_3, t_4, t_5\}$. d_1 is indexed by the conjunction of terms t_1 and t_2, d_2 is indexed by the conjunction of terms t_1, t_2 and t_3, and a query, q, indexed by $t_1$and t_5.[124]

$$d_1 = <1, 1, 0, 0, 0> \qquad d_2 = <1, 1, 1, 0, 0> \qquad q = <1, 0, 0, 0, 1>$$

Figure A.3: Possible worlds representation of d_1, d_2 and q

A simple retrieval model can be defined by asserting that all worlds (documents) have a distance of 1 from a query, q, if the intersection between the world and q is non-empty and the distance is 0 if the intersection is empty. This model would retrieve both d_1 and d_2 for q and corresponds to a Boolean disjunction of query terms. A Boolean conjunction of terms would be modelled by requiring the intersection of a world w and q to be identical to q.

Replacing the 1 and 0 in Figure A.3 by term weights, such as *idf* or *tf*, gives the representation used by the vector-space and probabilistic models described previously. The distance between the query and document worlds is given by the similarity or probability functions described before. Thus the logical model can be used to encapsulate the three retrieval models outlined previously, see [Hui96].

As in the example above, the principle of transforming documents and queries can be used to incorporate semantic information into the retrieval process. For example, consider a query t_2, and information that t_2 is a synonym of t_3 (from a thesaurus or dictionary). We can then assert that both d_1 and d_2 should both be retrieved, but that d_2 should be retrieved first as it undergoes fewer transformations than d_1 to be relevant

---

[123] This assumes the Closed World Assumption, i.e. any fact not known to be true is assumed false.

[124]Where 1 signifies that the proposition term t indexes the document is true, 0 signifies that the proposition is false.

We can also use representations based on different transformation principles, definitions of similarities, or definitions of possible worlds to give different retrieval models. [LaBr98] give a more detailed introduction to logical modelling of IR.

These models have the potential to be very powerful models in IR as they attempt to model the *semantics* of information and can incorporate, within a single framework, retrieval tools such as thesauri. In addition, they also allow for multiple relations to hold – they can be used to specify *which* relations cause relevance (see [VR86]). The formal nature of logical models mean that they also allow for formal *comparisons* between IR systems, e.g. [Hui96]. Crestani et al, [CLVR98], give an overview of current models and approaches in logic-based information retrieval.

RF has, so far, not been a major concern of existing logical models but it is possible to imagine several approaches to the problem. I shall describe these based on the following example of a concept based on an example given in [Seb94] which describes the class of documents which appeared in the proceedings of *SIGIR93*, whose author is a member of the institution *IEI-CNR* and which deal with *logic,* Figure A.4.

(**and** paper
(**func** appears-in (**sing** *SIGIR93*))
(**all** author (**func** affiliation (**sing** *IEI-CNR*)))
(**c-some** deals-with *logic*))

Figure A.4: Terminological representation of a concept
bold type indicates features of the representation language.

i. content modification. This approach is the most similar to that taken by the statistical RF models described previously. Here, the content of query is modified, e.g. by adding or deleting terms, or perhaps by altering connectives. For example, in the above example we could refine the query to retrieve only those papers that deal with *modal_logic*. This would retrieve only concepts that specifically mentioned *modal_logic,* Figure A.5, rather than the more general concept *logic*.

(**and** paper
(**func** appears-in (**sing** *SIGIR93*))
(**all** author (**func** affiliation (**sing** *IEI-CNR*)))
(**c-some** deals-with *modal_logic*))

Figure A.5: Terminological representation of a concept regarding *modal_logic*

or broaden the query by omitting one of the conditions, e.g. to retrieve all documents about logic written by a member of *IEI-CNR*, irrespective of where the paper was published. This would be a matching on only some of the components of our concept, as shown in Figure A.6.

(**and** paper
(**all** author (**func** affiliation (**sing** *IEI-CNR*)))
(**c-some** deals-with *logic*))

Figure A.6: Terminological representation of a concept

ii. personaliation of concepts. In addition to modifying the content of the query we could incorporate personalised thesaural knowledge. In the example, the term *logic* need not refer to a single term but could refer to a class of terms, e.g. *modal_logic*, *conceptual_graphs*, *cumulative_logic*, etc. This knowledge can be used as default values in retrieval but we could tailor this information to individual users based on feedback information. That is, the system automatically learns important synonymous concepts for individual users.

iii. uncertainty modelling. Logical concepts and rules reflecting thesaural knowledge are often associated with uncertainty values such as probabilities to reflect the importance of concepts or strength of relationship between concepts. These values can be changed in a similar fashion to the vector-space or probabilistic models to reflect important concepts in a search or the strength of association between concepts. Based on the example concept in Figure 1.8, for example, we could change the query to treat the author's affiliation as more important than the topic of the paper.

iv. rule modification or refinement. In this case, the information given by analysing the relevant documents is not only used to expand the query as in traditional feedback but is also used to modify the rules of the system. Examples of this approach include systems to *select* rules for retrieving documents, e.g. [DBM97] and the use of abductive logic to *create* new rules for retrieving documents, [Mull98].

Appendix B

Evaluation of IR systems and RF

B.1 Evaluation of retrieval systems and relevance feedback

In this Appendix I shall discuss the evaluation of IR systems and RF. The most common evaluation tool for IR systems is a *test collection*. This is a set of documents, a set of queries and a list of which documents are considered relevant for each query. The list of documents assessed as being relevant for each query – the *relevance assessments* – are usually not gathered from real-life search data. Rather test collections are usually constructed within a laboratory setting. Currently the foremost example of test collection construction is to be found within the TREC (Text REtrieval Conference) initiative, [VH96].

TREC follows a *pooling* method for creating test collections: a number of IR systems provide a ranking for a query, the top 100 documents from each ranking are pooled and the joint pool of documents are assessed by an assessor who decides which documents are relevant. The list of relevant documents is considered to be a representative set of relevant documents for the query. I discuss the difficulties and appropriateness of test collections for individual types of IR evaluation in Chapters Five and Twelve.

Test collections are primarily used for comparative evaluation: comparing the performance of two systems, or two versions of the same system on the same set of queries.

Two standard evaluation measures are commonly used with test collections: *precision* and *recall*. Recall is measured as the ratio of relevant documents retrieved to the number of relevant documents in the collection. Precision is the ratio of relevant documents retrieved to the number of documents retrieved. In a best-match, or ranking model, recall and precision figures can be calculated at various points in the document ranking to give an indication of performance at different levels of retrieval. Typically this would be done at 10% recall, i.e. 10% of relevant documents retrieved, 20% recall, 30% recall, etc. to give a set of 10 recall-precision figures), Figure B.1.

With a test collection, the recall-precision (RP) figures for each query are averaged to form a single set of recall-precision figures[125]. The averaged RP figures are often averaged across the recall points to give a single value – the *average precision* value, Figure B.1.

Recall	Precision
10	67.3
20	65.9
30	59.2
40	45.3
50	36.7
60	33.3
70	21.9
80	19.7
90	15.3
100	12.1
average precision	**37.67**

Figure B.1: Example recall and precision figures

RP figures are often represented graphically, Figure B.2 shows an example of a recall-precision graph drawn from the RP figures of two systems on the same test collection. As the line for System 1 is entirely above the line for System 2 we can infer that System 1 is better than System 2.

Figure B.3 shows the results of the two systems for a different test collection. In Figure B.3, the two lines cross at 70% recall, so we can say that, on the average of the queries tested, System 1 was better than System 2 at high recall levels (initially better at retrieving the relevant documents). On the other hand System 2 was better at lower recall levels (if the user is looking for *all* the relevant documents they will find them first with System 2).

[125]Interpolation measures are necessary for queries whose recall levels differ from the standard. For example in Figure B.1 RP is based on 10 recall levels, any query with a number of relevant documents different from a multiple of ten will require interpolation to give 10 recall levels. Interpolation is often used to calculate a 0% recall figure to give an 11pt recall-precision table.

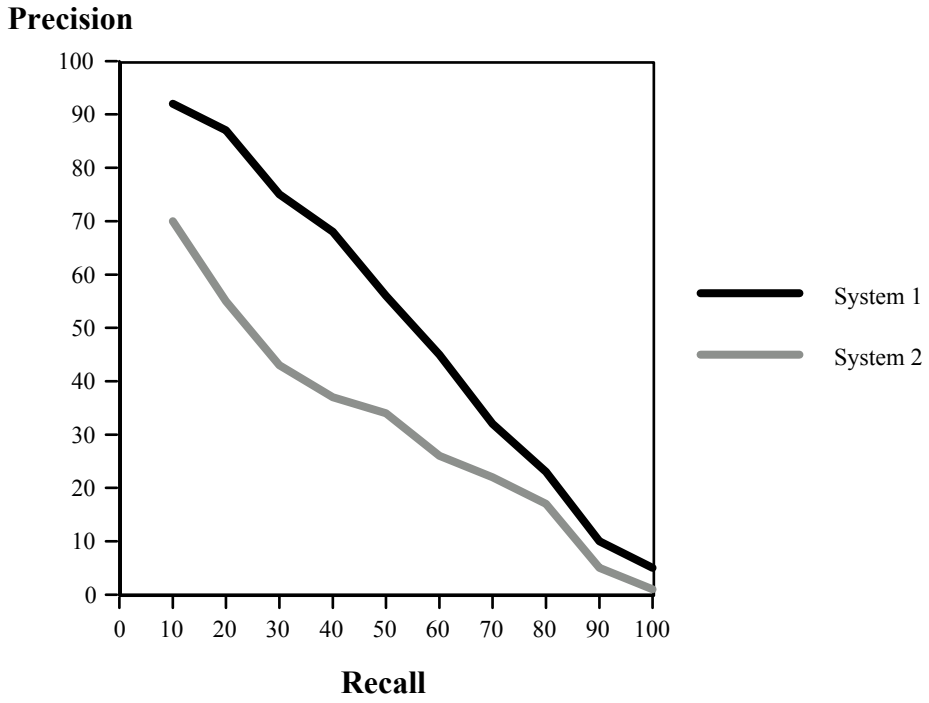

Figure B.2: Example RP graphs

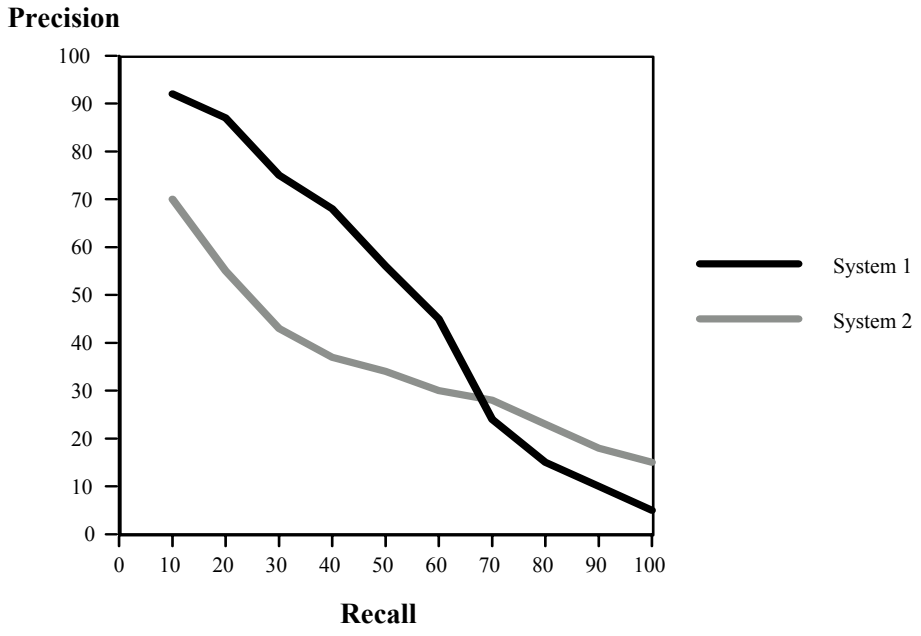

Figure B.3: Example RP graphs

Although these measures have been widely criticised for being capable of misrepresentation, [FMS91], not reflecting the dynamic, situational and subjective nature of information seeking, [BI97], and not reflecting *users'* evaluation criteria, e.g. [Su94], they have remained popular and standard measures of assessing an IR system performance.

However, as early as the early 1970's Chang et al., [CCR71], demonstrated that evaluation of RF algorithms poses certain problems for recall and precision. Given that RF, as described here, attempts to improve recall and precision by using information in marked relevant documents, it is usually the case that one of the main effects of RF is to push the known[126] relevant documents to the top of the document ranking. This *ranking effect*, will artificially improve RP figures for the new document ranking simply by re-ranking the known relevant documents. What is not directly tested is how good the RF technique is as improving retrieval of *unseen* relevant documents – the *feedback effect*. Chang et al., [CCR71], suggested three alternatives briefly outlined here to measure the effect of feedback on the unseen relevant documents:

- *residual ranking*: in this technique, the documents which are used in RF are removed from the collection before evaluation. This will include the relevant and some non-relevant documents. After RF, the RP figures are calculated on the remaining (*residual*) collection. The advantage of this method is that it only considers the effect of feedback on the unseen relevant documents but the main disadvantage is that the feedback results are not comparable with the original ranking. This is because the residual collection has fewer documents, and fewer relevant documents, than the original collection.

  A further difficulty is that, at each successive iteration of feedback, RP figures may be based on different numbers of queries. This arises because relevant documents are removed from the collection. If all the relevant documents are removed for a query, then this query cannot be used in subsequent iterations of feedback as there are no relevant documents upon which to calculate recall-precision figures. This method of evaluation is, then, biased somewhat towards queries that have more relevance assessments or those that perform poorly during initial iterations.

- *freezing*. The method known as freezing is based on the rank position of documents and comes in two forms: *full freezing* and *modified freezing*. In full freezing the rank positions of the top *n* documents, the ones used to modify the query, and are frozen. The remaining documents are re-ranked and RP figures are calculated over the whole ranking. As the only documents to change rank position are those below *n* (the ones used for RF) any change in RP happens as a result of the change of rank position of the unseen relevant documents. There is, then, no ranking effect. In modified freezing, the rank positions are frozen at the rank position of the last marked relevant document.

[126]These are the relevant documents that are used for RF.

The disadvantage of freezing approaches is that at each successive iteration of feedback a higher proportion of relevant documents are frozen. This means that the frozen section of the ranking contributes more to recall-precision at later iterations of RF, so although RF may work better at these later iterations, it can appear to be performing more poorly due to the higher contribution of the frozen documents.

In the discussion on the residual method of evaluating feedback runs, I mentioned that the residual collection method was forced to eliminate queries once all the relevant documents had been found. For the freezing methods, once all the relevant documents have been found for a query, recall-precision figures can still be calculated. However the recall-precision figures will not change once all the relevant documents have been frozen. Intuitively this seems correct: once we have found all the relevant documents for a query, feedback does not improve or worsen retrieval effectiveness.

- *test and control groups*. In this technique, the document collection is randomly split into two collections - the test group and the control group. Query modification is performed by RF on the test group and the new query is then run against the control group. RP is performed only on the control group, so there is no ranking effect. Successive queries can be run against the control group to assess modified queries on what can be regarded as a complete document collection unlike the residual ranking method. Unlike the freezing methods, all relevant documents in the control group are free to move within the document ranking. This means that recall-precision figures, before and after query modification, are directly comparable.

  The difficulty with this evaluation method is splitting the collection. It is easy to randomly split a document collection (e.g. by putting all evenly numbered documents in test group and all odd numbered documents in the control group). However, a random split will not ensure that the relevant documents are evenly split between the two collections. Neither will it ensure that the relevant documents in the test group are representative of those in the control group. Other factors such as document length or distribution of index terms may also be important to the RF method being tested, and may not be equally split between the two collections.

Each of these methods has advantages and disadvantages but all are standard methods of assessing RF algorithms. However, they only compare the performance of the algorithms in an idealised setting. For example, it is usual to use the same number of documents, per feedback iteration, to modify the query. A user, however, is unlikely to examine an identical

number of documents per search iteration. Also RF experiments based on recall-precision assume complete knowledge of the document collection: a fixed set of relevant documents is known beforehand. In interactive searching this is also unrealistic as what a user finds relevant may change over time, e.g. [Kuh93, Ell89, SW99, Vak00]. Additional methods are required to test the effectiveness of RF algorithms in more realistic settings. This requirement will be discussed more fully in Chapter Twelve.

A final point regarding these measures of RF evaluation is that they may not be directly comparable: each measure may appear to give different results depending on how the results are compared and on what factors affect the retrieval. An example of this is given next.

Table B.1 shows the results of RF on the same collection[127] but evaluated using the three RF evaluation schemes. An initial document ranking, for each query, was obtained using the *idf* weighting function, followed by four iterations of RF, in which the top 6 expansion terms were added, based on an F_4 ranking of expansion terms. 50 new documents were used in each iteration of feedback. After feedback all query terms were weighted using the F_4 term weighting scheme and these values were used to score documents. Table B.1 gives the percentage change, over no feedback, after four iterations of feedback using each of the three evaluation techniques.

AP	**Full freezing**	**Residual collection (removal)**	**Residual collection (no removal)**	**Test and control**
%age increase over no feedback	+1.75%	-72.65%	-15.04%	+3.82%

Table B.2: Example RF evaluation

As can be seen from Table B.1, the results vary according to how they describe the retrieval effectiveness of the system. Full freezing (column 2) gives a small increase in the effectiveness of the system. The test and control method gives a larger percentage increase in effectiveness (column 5).

These two approaches give different absolute performance figures (average precision) as they use different data to calculate *idf* values, F_4 values and do not have identical terms in the

[127] AP (Associated Press) collection 1988.

collection. The test and control method used one less query (as all the relevant documents for this query appeared in the test collection), and several of the queries were expanded by terms that appeared in the test collection but not the control collection[128]. These differences cause the different performance figures for the two evaluation methods.

The residual collection method (column 3) gives a large drop in retrieval effectiveness. This is because the residual collection method eliminates queries that have no relevant documents in the residual section of the collection. This means that queries, for which all relevant documents have been retrieved in early iterations of feedback, have been removed from the evaluation. The queries that are being used to calculate average precision are the ones for which the system finds it difficult to retrieve the remaining relevant documents[129].

If we do not remove queries when all relevant documents are found and, instead use the RP figures from the previous iteration, then we obtain the figure in column 4 for residual collection. This is an attempt to soften the effect of removing queries that perform well. This also shows a drop in retrieval effectiveness but not so severe a drop as in column 3. The drop in retrieval effectiveness is caused, again, by the effect of the queries for which the system finds it difficult to retrieve all relevant documents.

The difference in performance given by the three techniques is noticeable in this test as the RF technique is not proving to be very effective: no evaluation showed a significant increase in average precision. However, the problem of evaluation is applicable to all RF tests.

An alternative method of examining RF performance is to plot the average precision values at each iteration of feedback, as in Figure B.4. We can see that different methods give different shaped graphs. The freezing graph gives a slight, but steady, increases in retrieval effectiveness at each iteration of feedback. The test and control method gives an initial large increase followed by decreases at subsequent iterations of feedback.

The residual methods, however, give very different graphs: not removing queries gives a small drop at the first iteration followed by increases at subsequent iteration, whereas removing queries causes alternative increases and decreases in retrieval effectiveness.

---

[128] This was also true for one of the original query terms.

[129] The remaining queries may also include some queries that have a large number of relevant documents, but this is unlikely to be the case in this test as 200 documents have been used for feedback whereas the queries have an average of only 35 relevant documents per query.

The graphs can be used to highlight interesting areas – where RF is working well or where it is operating poorly. However as with recall and precision the graphs can be misleading: all four lines plotted in Figure B.4 are evaluating the same feedback technique on the same collection.

The point is that the evaluation measures are calculating different aspects of feedback: freezing is measuring *cumulative* effectiveness, residual collection is measuring the effectiveness of retrieving *only* the remaining relevant documents and test and control is measuring the relative performance of the modified queries produced at each iteration.

For the majority of the results presented in this thesis I shall use the full-freezing method of evaluation. This is because I believe that, of the three methods outlined here, it is the most realistic method for simulating interactive techniques as it gives a measure based on the whole search.

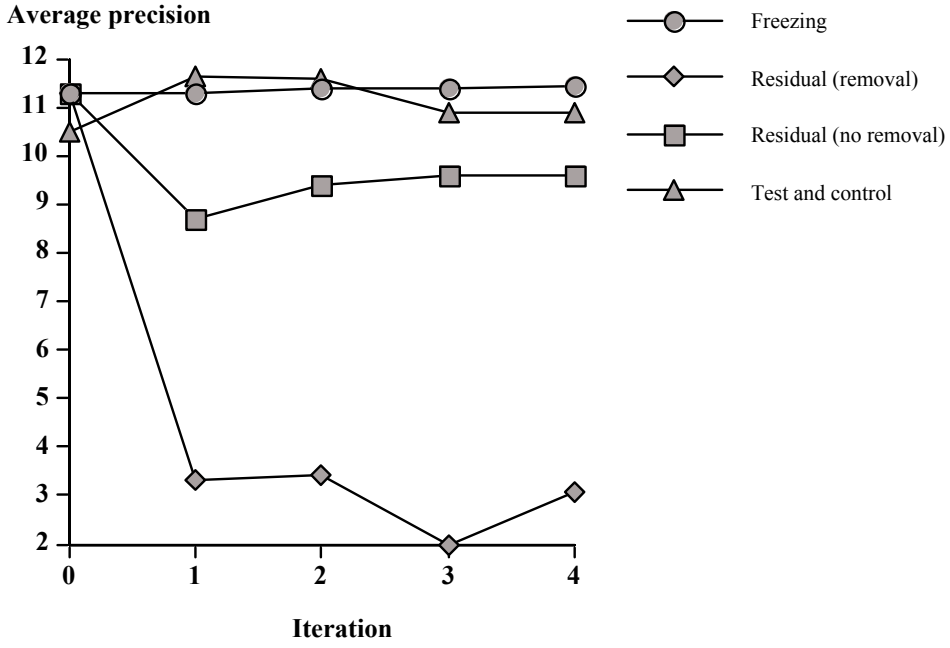

Figure B.4: Average precision over 4 iterations of feedback

Appendix C

Supplementary results from Chapter Four

CACM					
tf + nse	30.26	*idf + tf + con*	23.15	*idf*	22.00
idf + tf + nse	26.83	*idf + tf + th + con + inf*	23.10	*idf + th + con + spec*	21.97
idf + tf + nse + inf	25.74	*tf + th + con + spec + nse*	23.09	*idf + th + con*	21.92
tf + spec + nse	25.41	*tf + th + nse*	23.08	*idf + th + spec*	21.89
tf + con + nse	25.31	*idf + tf + th + con*	23.06	*tf + con + spec + inf*	21.87
idf + tf	25.21	*idf + tf + th + con + spec*	23.06	*idf + con + spec + inf*	21.85
idf + tf + th + nse	25.04	*idf + con + nse + inf*	23.05	*tf + th + spec + inf*	21.82
idf + tf + con + nse	24.79	*tf + th + con + nse + inf*	23.04	*tf + th + con + spec + inf*	21.79
idf + tf + spec + nse + inf	24.72	*idf + tf + th + inf*	23.03	*idf + th*	21.77
idf + tf + spec + nse	24.70	*idf + tf + con + inf*	23.02	*idf + con*	21.65
idf + tf + th + con + nse	24.61	*th + nse + inf*	22.98	*tf + th + con + spec*	21.63
tf + spec + nse + inf	24.42	*idf + tf + con + spec*	22.96	*tf + con + spec*	21.60
idf + tf + th + con + nse + inf	24.23	*tf + th + con + spec + nse + inf*	22.92	*con + nse*	21.49
tf + con + nse + inf	24.22	*idf + tf + th + spec*	22.79	*tf + th + con + inf*	21.44
idf + tf + th + nse + inf	24.20	*idf + con + spec + nse + inf*	22.79	*tf + spec*	21.30
idf + tf + con + nse + inf	24.19	*spec + nse*	22.77	*tf + spec + inf*	21.29
nse	24.15	*idf + tf + th + con + spec + inf*	22.72	*idf + con + spec*	21.28
idf + tf + th + spec + inf	24.00	*tf*	22.70	*tf + th + con*	21.14
idf + tf + th + spec + nse + inf	24.00	*idf + tf + th + spec + nse*	22.68	*tf + th + inf*	21.12
tf + nse + inf	23.89	*spec + nse + inf*	22.64	*tf + th + spec*	21.10
tf + th + nse + inf	23.89	*th + con + spec + nse*	22.58	*idf + spec + inf*	20.88
idf + th + nse	23.88	*idf + spec + nse + inf*	22.56	*th + con + spec + inf*	20.71
idf + tf + spec	23.82	*th + con + nse + inf*	22.54	*th + con + inf*	20.58
tf + con + spec + nse	23.82	*idf + tf + con + spec + inf*	22.54	*idf + spec*	20.44
nse + inf	23.81	*tf + con*	22.47	*con + inf*	20.41
idf + th + nse + inf	23.79	*tf + inf*	22.47	*th + con + spec*	20.29
idf + tf + con + spec + nse	23.74	*idf + th + con + inf*	22.40	*th + spec + inf*	20.29
all	23.69	*idf + con + nse*	22.38	*con + spec + inf*	19.89
idf + th + con + nse + inf	23.61	*con + nse + inf*	22.37	*th + nse*	19.88
tf + th + con + nse	23.59	*idf + con + inf*	22.36	*th + inf*	19.58
idf + nse	23.57	*th + spec + nse + inf*	22.31	*th + con*	19.39
idf + tf + th + con + spec + nse	23.57	*idf + spec + nse*	22.22	*tf + th*	19.35
idf + th + spec + nse + inf	23.49	*idf + con + spec + nse*	22.19	*con + spec*	19.11
idf + tf + con + spec + nse + inf	23.48	*tf + con + inf*	22.13	*th + spec*	19.01
tf + th + spec + nse	23.44	*con + spec + nse + inf*	22.13	*spec + inf*	18.51
idf + th + con + nse	23.39	*th + spec + nse*	22.12	*con*	14.80
idf + nse + inf	23.33	*con + spec + nse*	22.10	*th*	4.36
tf + con + spec + nse + inf	23.33	*idf + th + spec + inf*	22.10	*inf*	1.67
idf + tf + spec + inf	23.30	*idf + th + con + spec + inf*	22.10	*spec*	1.19
idf + th + con + spec + nse + inf	23.30	*th + con + nse*	22.08		
idf + th + spec + nse	23.24	*th + con + spec + nse + inf*	22.07		
idf + th + con + spec + nse	23.21	*idf + inf*	22.01		
tf + th + spec + nse + inf	23.21	*idf + tf + inf*	22.01		
idf + tf + th	23.18	*idf + th + inf*	22.01		

Table C.1: Summary of average precision figures for all combinations of characteristics on the CACM collection with no weighting of characteristics

CACM					
idf + tf + nse	25.68	*tf + th + spec + nse*	23.28	*idf + spec + inf*	22.00
idf + tf + spec + nse	25.68	*tf + th + nse + inf*	23.28	*spec + nse*	21.82
idf + tf + nse + inf	25.68	*tf + th + con + nse + inf*	23.28	*nse + inf*	21.82
idf + tf + con + nse + inf	25.68	*tf + con + nse*	22.91	*spec + nse + inf*	21.82
idf + tf + spec + inf	25.54	*tf + con + spec + nse*	22.91	*idf + con + nse*	21.04
idf + tf	25.45	*tf + con + nse + inf*	22.91	*idf + con + spec + nse*	21.04
idf + tf + spec	25.45	*tf + th + spec + nse + inf*	22.91	*idf + con + nse + inf*	21.04
idf + tf + th + nse	25.23	*tf*	22.70	*idf + th + spec + nse + inf*	21.04
idf + tf + th + spec + inf	25.23	*tf + spec*	22.70	*con + nse*	20.90
idf + tf + th + con + nse + inf	25.23	*tf + inf*	22.70	*th + con + nse*	20.90
idf + tf + th	24.97	*tf + spec + inf*	22.70	*con + spec + nse*	20.90
idf + tf + th + spec	24.97	*idf + th + con + nse*	22.65	*con + nse + inf*	20.90
idf + tf + th + inf	24.97	*idf + tf + spec + nse + inf*	22.65	*th + con + spec + nse*	20.90
idf + tf + th + spec + nse	24.97	*idf + th + con + spec + inf*	22.65	*th + con + nse + inf*	20.90
tf + nse	24.58	*idf + tf + con + spec + nse + inf*	22.65	*con + spec + nse + inf*	20.90
tf + spec + nse	24.58	*tf + con*	22.55	*tf + con + spec + nse + inf*	20.90
tf + spec + nse + inf	24.58	*tf + con + spec*	22.55	*idf + con*	20.67
nse	24.15	*tf + con + inf*	22.55	*idf + con + spec*	20.67
idf + th + nse	24.03	*tf + con + spec + inf*	22.55	*idf + con + inf*	20.67
idf + th + spec + nse	24.03	*tf + th + con + nse*	22.37	*idf + con + spec + inf*	20.67
idf + th + nse + inf	24.03	*idf + con + spec + nse + inf*	22.37	*th + con*	19.80
idf + th + con + nse + inf	24.03	*idf + th + con + spec + nse + inf*	22.37	*th + con + spec*	19.80
idf + tf + con + nse	23.84	*idf + th + con*	22.27	*th + con + inf*	19.80
idf + tf + th + nse + inf	23.84	*idf + th + con + spec*	22.27	*th + con + spec + inf*	19.80
idf + tf + con + spec + inf	23.84	*idf + th + con + inf*	22.27	*th + nse*	18.51
idf + tf + th + spec + nse + inf	23.84	*idf + th + con + spec + nse*	22.27	*th + spec + nse*	18.51
idf + th	23.79	*tf + th + con + spec + inf*	22.27	*th + nse + inf*	18.51
idf + th + spec	23.79	*tf + th*	22.25	*th + spec + nse + inf*	18.51
idf + th + spec + inf	23.79	*tf + th + spec*	22.25	*spec + inf*	18.51
idf + tf + th + con + nse	23.71	*tf + th + inf*	22.25	*con + spec*	14.96
th + con + spec + nse + inf	23.71	*tf + th + spec + inf*	22.25	*con + inf*	14.96
idf + tf + th + con + spec + inf	23.71	*idf + nse*	22.08	*con + spec + inf*	14.96
tf + th + con + spec + nse + inf	23.71	*idf + spec + nse*	22.08	*con*	14.80
all	23.71	*idf + nse + inf*	22.08	*th + spec*	14.68
idf + tf + con	23.64	*tf + th + con*	22.08	*th + inf*	14.68
idf + tf + con + spec	23.64	*idf + spec + nse + inf*	22.08	*th + spec + inf*	14.68
idf + tf + con + inf	23.64	*tf + th + con + spec*	22.08	*th*	4.36
idf + tf + con + spec + nse	23.64	*tf + th + con + inf*	22.08	*inf*	1.67
idf + tf + th + con	23.61	*tf + th + con + spec + nse*	22.08	*spec*	1.19
idf + tf + th + con + spec	23.61	*idf*	22.00		
idf + tf + th + con + inf	23.61	*idf + spec*	22.00		
idf + tf + th + con + spec + nse	23.61	*idf + inf*	22.00		
tf + th + nse	23.28	*idf + tf + inf*	22.00		
tf + nse + inf	23.28	*idf + th + inf*	22.00		

Table C.2: Summary of average precision figures for all combinations of characteristics on the CACM collection with weighting of characteristics

CISI					
idf + tf	12.87	*tf + th + con + spec + inf*	11.50	*idf + tf + con + nse + inf*	10.76
tf	12.51	*idf + spec + inf*	11.48	*tf + con*	10.74
idf + tf + th + inf	12.22	*tf + th + con*	11.48	*tf + con + spec + inf*	10.74
idf + tf + th	12.18	*idf + spec*	11.45	*spec + nse + inf*	10.72
idf + tf + spec + inf	12.09	*idf + th + con + spec*	11.45	*idf + tf + con + spec + nse + inf*	10.71
idf + tf + spec	12.00	*tf + th + con + spec*	11.44	*idf + con + inf*	10.69
idf + tf + th + spec	11.94	*idf + th + spec + nse + inf*	11.44	*tf + con + inf*	10.69
idf + tf + th + nse	11.89	*idf + th + spec + nse*	11.43	*idf + con*	10.66
idf + tf + th + spec + nse	11.84	*idf + th + con + nse + inf*	11.39	*idf + tf + con + spec + nse*	10.66
idf + tf + th + con + inf	11.80	*tf + th + con + nse + inf*	11.39	*nse + inf*	10.64
idf + tf + th + con	11.75	*idf + th + con + nse*	11.36	*tf + con + spec*	10.60
idf + tf + th + con + spec	11.75	*tf + th + con + nse*	11.34	*idf + con + spec + inf*	10.60
tf + spec	11.71	*tf + th + con + spec + nse*	11.33	*spec*	10.55
idf + tf + nse	11.71	*th + spec + inf*	11.32	*spec + nse*	10.53
tf + th + spec + inf	11.71	*idf + th + con + spec + nse*	11.32	*tf + con + nse*	10.46
tf + inf	11.70	*th + con + inf*	11.30	*idf + con + nse + inf*	10.46
idf + tf + th + nse + inf	11.69	*th + con + spec + inf*	11.29	*tf + con + nse + inf*	10.45
tf + th + inf	11.68	*th + nse + inf*	11.28	*idf + tf + th + con + spec + inf*	10.45
all	11.66	*tf + spec + nse + inf*	11.28	*idf + con + spec*	10.44
idf + th + con + inf	11.66	*tf + nse*	11.27	*tf + con + spec + nse + inf*	10.43
idf + tf + th + spec + inf	11.65	*tf + spec + nse*	11.26	*tf + con + spec + nse*	10.42
idf + inf	11.64	*th + spec + nse + inf*	11.26	*idf + con + spec + nse + inf*	10.41
idf + tf + inf	11.64	*th + inf*	11.23	*idf + con + nse*	10.38
idf + th + inf	11.64	*th + spec + nse*	11.21	*idf + tf + th + con + nse + inf*	10.38
tf + spec + inf	11.63	*th + con + spec*	11.20	*idf + nse*	10.35
idf + th + spec + inf	11.63	*th + nse*	11.15	*idf + tf + th + con + spec + nse*	10.33
tf + nse + inf	11.60	*th + con + nse + inf*	11.15	*idf + con + spec + nse*	10.29
tf + th + nse + inf	11.60	*th + con + spec + nse + inf*	11.14	*con + inf*	10.27
idf + tf + spec + nse	11.58	*th + spec*	11.13	*con + spec + inf*	10.23
idf + tf + nse + inf	11.58	*idf + tf + con + inf*	11.13	*idf + th + con + spec + nse + inf*	10.16
idf + tf + spec + nse + inf	11.58	*th + con*	11.12	*tf + th + con + spec + nse + inf*	10.14
tf + th + spec + nse + inf	11.58	*idf + tf + con*	11.12	*con + nse + inf*	10.10
idf + th + nse	11.57	*th + con + nse*	11.12	*con + spec + nse + inf*	10.09
idf + th + spec	11.56	*idf + nse + inf*	11.11	*con + spec*	10.02
idf + tf + th + con + nse	11.56	*tf + th*	11.10	*con + spec + nse*	9.97
idf + th + nse + inf	11.55	*idf + tf + con + spec + inf*	11.09	*con + nse*	9.96
tf + th + spec + nse	11.55	*idf + spec + nse + inf*	11.08	*con*	9.57
idf	11.54	*th + con + spec + nse*	11.08	*th*	5.11
idf + th	11.54	*idf + tf + con + spec*	11.01	*inf*	4.08
idf + th + con	11.53	*idf + tf + th + spec + nse + inf*	11.01		
tf + th + nse	11.53	*nse*	11.00		
tf + th + spec	11.52	*idf + spec + nse*	10.96		
idf + th + con + spec + inf	11.52	*spec + inf*	10.90		
tf + th + con + inf	11.50	*idf + tf + con + nse*	10.78		

Table C.3: Summary of average precision figures for all combinations of characteristics on the CISI collection with no weighting of characteristics

CISI					
idf + tf	12.84	*idf + spec + inf*	11.54	*tf + th + con + nse + inf*	10.85
idf + tf + spec	12.84	*tf + th + con + inf*	11.50	*idf + con*	10.78
idf + tf + th	12.79	*idf + th + con + spec*	11.45	*idf + con + spec*	10.78
idf + tf + th + spec + nse	12.79	*tf + th + con + spec*	11.44	*idf + con + inf*	10.78
idf + tf + spec + nse + inf	12.66	*idf + th + spec + nse*	11.43	*idf + tf + con + nse*	10.78
idf + tf + th + spec + inf	12.58	*idf + th + con + nse*	11.36	*idf + con + nse*	10.75
idf + tf + th + nse + inf	12.58	*tf + th + con + nse*	11.34	*tf + con + nse*	10.75
idf + th + spec	12.57	*idf + nse*	11.33	*idf + con + spec + nse + inf*	10.75
tf	12.51	*idf + spec + nse*	11.33	*tf + con + spec + nse + inf*	10.75
tf + spec	12.51	*idf + nse + inf*	11.33	*tf + con + spec + inf*	10.74
tf + inf	12.51	*idf + tf + con*	11.31	*idf + th + con + spec + nse + inf*	10.71
tf + spec + inf	12.51	*idf + tf + con + spec + inf*	11.31	*tf + th + con + spec + nse + inf*	10.69
idf + tf + th + spec + nse + inf	12.49	*th + con + spec + inf*	11.29	*idf + con + spec + inf*	10.60
tf + nse	12.35	*idf + tf + th + con + spec*	11.29	*spec*	10.55
tf + spec + nse	12.35	*idf + tf + th + con + inf*	11.29	*idf + con + nse + inf*	10.46
idf + th + nse	12.30	*tf + spec + nse + inf*	11.28	*tf + con + nse + inf*	10.45
idf + th + spec + nse + inf	12.30	*th + spec + nse + inf*	11.26	*th + con + nse*	10.44
idf + tf + th + inf	12.22	*idf + tf + th + con + nse*	11.26	*th + con + spec + nse + inf*	10.44
tf + th + nse	12.15	*idf + tf + th + con + spec + inf*	11.25	*tf + con + spec + nse*	10.42
tf + nse + inf	12.15	*idf + tf + con + spec + nse*	11.21	*th + con*	10.41
tf + th + spec + nse + inf	12.15	*idf + tf + con + nse + inf*	11.21	*th + con + spec*	10.41
tf + th	12.11	*idf + tf + con + spec + nse + inf*	11.21	*th + con + inf*	10.41
tf + th + spec	12.11	*idf + tf + th + con + spec + nse*	11.16	*idf + con + spec + nse*	10.29
tf + th + inf	12.11	*idf + tf + th + con + nse + inf*	11.16	*con + spec + nse + inf*	10.09
idf + tf + spec + inf	12.09	*th + con + nse + inf*	11.15	*con + nse*	10.08
all	12.02	*idf + tf + con + inf*	11.13	*con + spec + nse*	10.08
idf + tf + th + spec	11.94	*idf + spec + nse + inf*	11.08	*con + nse + inf*	10.08
idf + tf + th + nse	11.89	*th + con + spec + nse*	11.08	*con + spec*	9.78
idf + tf + th + con	11.75	*idf + th + con*	11.06	*con + inf*	9.78
tf + th + spec + inf	11.71	*idf + th + con + spec + inf*	11.06	*con + spec + inf*	9.78
idf + tf + nse	11.66	*th + nse*	11.02	*spec + nse*	9.70
idf + th + con + inf	11.66	*th + spec + nse*	11.02	*nse + inf*	9.70
idf + th + spec + inf	11.63	*th + nse + inf*	11.02	*spec + nse + inf*	9.70
tf + th + nse + inf	11.60	*idf + tf + con + spec*	11.01	*con*	9.57
idf + tf + spec + nse	11.58	*nse*	11.00	*th + spec*	9.56
idf + tf + nse + inf	11.58	*idf + th + con + spec + nse*	10.96	*th + inf*	9.56
idf + th + nse + inf	11.55	*idf + th + con + nse + inf*	10.96	*th + spec + inf*	9.56
tf + th + spec + nse	11.55	*tf + th + con*	10.94	*th*	5.11
idf	11.54	*tf + th + con + spec + inf*	10.94	*inf*	4.08
idf + th	11.54	*spec + inf*	10.90		
idf + spec	11.54	*tf + con*	10.89		
idf + inf	11.54	*tf + con + spec*	10.89		
idf + tf + inf	11.54	*tf + con + inf*	10.89		
idf + th + inf	11.54	*tf + th + con + spec + nse*	10.85		

Table C.4: Summary of average precision figures for all combinations of characteristics on the CISI collection with weighting of characteristics

MEDLARS					
th + nse	48.64	*idf + th + con + spec + nse*	44.89	*tf + th + con + spec + inf*	43.00
tf + th + nse	48.60	*idf + tf + con + spec*	44.82	*tf + con*	42.92
idf + th + nse	47.79	*idf + tf + con + inf*	44.82	*idf + con + spec + inf*	42.86
idf + tf + th	47.68	*idf + th + spec + nse + inf*	44.61	*tf + con + spec + nse + inf*	42.72
tf + nse	47.63	*idf + th + con + nse*	44.60	*idf + tf + con + spec + inf*	42.69
th + spec + nse + inf	47.29	*idf + tf + con*	44.58	*idf + spec + nse*	42.62
tf + nse + inf	46.71	*tf + spec + nse*	44.57	*th + con + spec + inf*	42.49
tf + th + spec + nse	46.62	*idf + tf + th + con + inf*	44.54	*idf + con*	42.38
tf + th + nse + inf	46.62	*idf + th + con + spec*	44.51	*idf + nse + inf*	42.31
idf + tf + nse	46.39	*idf + th + con + inf*	44.51	*spec + nse*	42.28
th + spec + nse	46.33	*tf + th + con + spec + nse + inf*	44.51	*tf + spec*	42.16
th + nse + inf	46.31	*tf + th + con + nse*	44.44	*idf + con + spec + nse + inf*	42.16
idf + tf + th + nse + inf	46.17	*idf + tf + con + spec + nse*	44.43	*con + spec + nse*	42.09
idf + tf + th + nse	46.14	*idf + tf + th + spec + nse*	44.40	*th + spec + inf*	42.03
tf + spec + nse + inf	46.05	*idf + th + con + spec + nse + inf*	44.31	*tf + con + spec*	41.81
idf + tf + th + spec	46.04	*th + con + spec + nse*	44.20	*con + spec + nse + inf*	41.78
idf + tf + th + inf	46.04	*th + con + nse + inf*	44.20	*con + nse + inf*	41.70
idf + tf + th + con + nse	45.98	*th + spec*	44.15	*idf + spec*	41.67
idf + tf + spec + nse	45.95	*idf + tf + spec*	44.07	*idf + con + spec*	41.60
idf + tf + nse + inf	45.95	*idf + tf + th + con + spec + inf*	44.04	*nse + inf*	41.58
all	45.92	*th + con + spec + nse + inf*	44.03	*tf + inf*	41.54
idf + tf	45.73	*idf + tf + con + nse + inf*	44.00	*idf + con + inf*	41.48
idf + th	45.70	*tf + th + con + spec*	43.96	*idf + inf*	41.46
th + con + nse	45.70	*tf + th + con + inf*	43.96	*idf + tf + inf*	41.46
idf + tf + spec + inf	45.69	*nse*	43.92	*idf + th + inf*	41.46
idf + tf + th + con + spec	45.59	*th + inf*	43.92	*tf + con + inf*	41.26
idf + th + spec + inf	45.42	*tf + con + spec + nse*	43.80	*spec + nse + inf*	40.72
idf + tf + th + con + nse + inf	45.40	*tf + con + nse + inf*	43.80	*tf + th + con*	40.50
idf + tf + th + con + spec + nse	45.39	*tf*	43.75	*idf + spec + inf*	40.38
idf + th + spec + nse	45.37	*idf + th + con + spec + inf*	43.72	*tf + spec + inf*	40.05
idf + th + nse + inf	45.37	*idf + tf + spec + nse + inf*	43.65	*tf + th + spec*	39.20
idf + tf + th + spec + inf	45.27	*th + con*	43.49	*con + spec*	38.04
idf + tf + th + spec + nse + inf	45.27	*tf + th*	43.41	*con + spec + inf*	37.70
idf + tf + th + con	45.23	*idf + nse*	43.40	*con + inf*	37.54
tf + th + con + spec + nse	45.10	*idf + tf + con + spec + nse + inf*	43.35	*con*	36.14
tf + th + spec + inf	45.06	*idf + con + nse*	43.18	*spec + inf*	35.79
idf + th + con + nse + inf	45.06	*con + nse*	43.13	*th*	11.12
tf + th + spec + nse + inf	45.06	*tf + con + spec + inf*	43.12	*inf*	8.67
idf + th + spec	45.02	*idf*	43.10	*spec*	4.62
tf + th + con + nse + inf	44.95	*th + con + spec*	43.08		
tf + th + inf	44.92	*th + con + inf*	43.07		
tf + con + nse	44.92	*idf + spec + nse + inf*	43.05		
idf + th + con	44.90	*idf + con + spec + nse*	43.00		
idf + tf + con + nse	44.9	*idf + con + nse + inf*	43.00		

Table C.5: Summary of average precision figures for all combinations of characteristics on the MEDLARS collection with no weighting of characteristics

MEDLARS					
th + nse	47.29	***idf + tf + con***	44.82	***idf + spec + nse + inf***	43.05
th + spec + nse	47.29	***idf + tf + con + spec***	44.82	***idf + spec***	43.03
th + nse + inf	47.29	***idf + tf + con + inf***	44.82	***idf + inf***	43.03
tf + th + nse	46.62	***idf + tf + con + spec + inf***	44.82	***idf + tf + inf***	43.03
tf + nse + inf	46.62	***idf + tf + th + con***	44.71	***idf + th + inf***	43.03
tf + th + spec + nse + inf	46.62	***idf + th + con + spec + nse***	44.60	***idf + spec + inf***	43.03
idf + tf + th + spec + inf	46.14	***idf + th + con + nse + inf***	44.60	***idf + th + spec + inf***	43.02
idf + tf + th + nse + inf	46.14	***idf + th + con + spec + nse + inf***	44.60	***idf + con + nse***	43.00
idf + tf + th + spec + nse + inf	46.14	***idf + th + con***	44.51	***idf + con + spec + nse***	43.00
tf + nse	46.05	***idf + th + con + spec + inf***	44.51	***idf + con + nse + inf***	43.00
tf + spec + nse	46.05	***tf + th + con + spec + nse***	44.44	***tf + th + con + spec***	43.00
tf + spec + nse + inf	46.05	***tf + th + con + nse + inf***	44.44	***tf + th + con + inf***	43.00
idf + tf + th	46.04	***tf + th + con + spec + nse + inf***	44.44	***idf + con + spec + nse + inf***	43.00
idf + tf + th + spec + nse	46.04	***tf + spec***	44.23	***idf + con***	42.86
idf + tf + nse	45.95	***tf + inf***	44.23	***idf + con + spec***	42.86
idf + tf + spec + nse	45.95	***tf + spec + inf***	44.23	***idf + con + inf***	42.86
idf + tf + nse + inf	45.95	***th + con + nse***	44.20	***idf + con + spec + inf***	42.86
idf + tf + spec + nse + inf	45.95	***th + con + spec + nse + inf***	44.20	***th + con***	42.49
idf + tf	45.69	***tf + th + con***	43.96	***th + con + spec***	42.49
idf + tf + spec	45.69	***tf + th + con + spec + inf***	43.96	***th + con + inf***	42.49
idf + tf + spec + inf	45.69	***nse***	43.92	***con + nse***	41.78
tf + th + spec + nse	45.43	***tf + th + spec + inf***	43.91	***con + spec + nse***	41.78
tf + th + nse + inf	45.43	***tf + con + nse***	43.80	***con + nse + inf***	41.78
idf + th	45.42	***tf + con + spec + nse***	43.80	***con + spec + nse + inf***	41.78
idf + th + spec	45.42	***tf + con + nse + inf***	43.80	***th + con + spec + nse***	40.04
idf + tf + th + con + nse	45.41	***tf + con + spec + nse + inf***	43.80	***th + con + nse + inf***	40.04
idf + tf + th + con + spec + nse	45.41	***tf***	43.75	***th + spec***	38.39
idf + tf + th + con + nse + inf	45.41	***tf + th + con + nse***	43.64	***th + inf***	38.39
idf + th + nse	45.37	***spec + nse***	43.55	***th + spec + inf***	38.39
idf + th + spec + nse + inf	45.37	***nse + inf***	43.55	***th + con + spec + inf***	37.50
idf + tf + th + nse	45.35	***spec + nse + inf***	43.55	***con***	36.14
all	45.29	***idf + th + con + nse***	43.34	***spec + inf***	35.79
idf + tf + th + spec	45.27	***idf + th + con + spec***	43.19	***th + spec + nse + inf***	35.12
idf + tf + th + inf	45.27	***idf + th + con + inf***	43.19	***con + spec***	34.94
idf + tf + th + con + spec	45.23	***idf + th + spec + nse***	43.17	***con + inf***	34.94
idf + tf + th + con + inf	45.23	***idf + th + nse + inf***	43.17	***con + spec + inf***	34.94
idf + tf + th + con + spec + inf	45.23	***tf + con***	43.12	***th***	11.12
tf + th	45.06	***tf + con + spec***	43.12	***inf***	8.67
tf + th + spec	45.06	***tf + con + inf***	43.12	***spec***	4.62
tf + th + inf	45.06	***tf + con + spec + inf***	43.12		
idf + tf + con + nse	44.90	***idf***	43.10		
idf + tf + con + spec + nse	44.90	***idf + nse***	43.05		
idf + tf + con + nse + inf	44.90	***idf + spec + nse***	43.05		
idf + tf + con + spec + nse + inf	44.90	***idf + nse + inf***	43.05		

Table C.6: Summary of average precision figures for all combinations of characteristics on the MEDLARS collection with weighting of characteristics

AP					
idf + tf + con + nse	15.31	***idf + spec + nse + inf***	11.32	***idf + th + spec***	9.88
idf + tf + nse	15.28	***tf + th + con + nse***	11.22	***idf + con + nse + inf***	9.88
idf + tf + con + spec + nse	15.04	***idf + th + nse + inf***	11.14	***tf***	9.86
tf + th + nse + inf	14.53	***tf + spec + inf***	11.13	***idf + spec + nse***	9.86
tf + con + nse	14.44	***idf + tf + th + spec***	11.13	***con + spec***	9.77
idf + tf + con	14.26	***idf + tf + th + spec + nse***	11.12	***idf + spec***	9.66
idf + tf + con + spec	14.13	***idf + con + nse***	11.09	***con + spec + inf***	9.60
tf + spec + nse	14.04	***idf + con + spec + nse + inf***	10.98	***con***	9.57
idf + tf + con + nse + inf	14.02	***idf + th + con + spec + nse + inf***	10.88	***tf + th + con + inf***	9.52
idf + tf + con + spec + nse + inf	13.78	***idf + tf + th + inf***	10.87	***idf + nse + inf***	9.51
idf + tf + spec + nse + inf	13.77	***idf + con + spec + inf***	10.86	***con + inf***	9.46
idf + tf + spec	13.65	***idf + th + con + spec + nse***	10.74	***th + con + nse + inf***	9.42
idf + tf	13.63	***th + spec + nse + inf***	10.69	***idf + spec + inf***	9.32
tf + con + spec	13.61	***con + nse***	10.67	***tf + nse + inf***	9.26
tf + con	13.60	***idf + tf + th + nse***	10.65	***tf + th + spec + inf***	9.26
idf + tf + spec + inf	13.57	***idf + th + con + spec***	10.65	***tf + th + spec***	9.25
idf + tf + con + inf	13.44	***idf + th + con + spec + inf***	10.64	***th + spec + inf***	9.24
tf + con + spec + inf	12.94	***con + spec + nse***	10.61	***tf + th + inf***	9.07
tf + con + spec + nse + inf	12.94	***idf + con***	10.59	***idf + th + nse***	9.05
idf + tf + con + spec + inf	12.84	***tf + th + spec + nse + inf***	10.54	***spec + nse + inf***	8.79
tf + con + nse + inf	12.69	***idf + con + spec***	10.52	***idf + th***	8.77
tf + spec	12.60	***idf + th + con + nse + inf***	10.49	***idf + inf***	8.77
tf + con + inf	12.46	***th + con + spec + inf***	10.47	***idf + tf + inf***	8.77
idf + tf + spec + nse	12.18	***th + con + spec + nse + inf***	10.45	***idf + th + inf***	8.77
tf + con + spec + nse	12.16	***idf + tf + nse + inf***	10.44	***spec + nse***	8.69
idf + tf + th + con + spec + nse	12.14	***idf + th + con***	10.40	***nse + inf***	8.48
idf + tf + th + con + nse	12.13	***con + nse + inf***	10.39	***th + spec + nse***	7.98
all	12.04	***idf + tf + th***	10.37	***th + nse + inf***	7.89
idf + tf + th + con	11.83	***tf + th + spec + nse***	10.36	***spec + inf***	7.86
idf + tf + th + con + spec	11.83	***tf + spec + nse + inf***	10.29	***th + spec***	7.70
idf + tf + th + con + nse + inf	11.70	***idf + th + con + nse***	10.28	***th + inf***	7.67
tf + th + con + spec	11.62	***idf + th + spec + nse + inf***	10.27	***tf + th + nse***	7.67
tf + nse	11.61	***th + con + inf***	10.26	***tf + th***	7.18
idf + tf + th + con + spec + inf	11.59	***th + con + spec + nse***	10.22	***th + nse***	5.25
tf + inf	11.55	***th + con + nse***	10.21	***th***	4.63
tf + th + con + spec + nse + inf	11.52	***idf + nse***	10.13	***nse***	1.00
tf + th + con + spec + nse	11.51	***idf***	10.10	***spec***	0.47
idf + tf + th + con + inf	11.45	***idf + th + con + inf***	10.09	***inf***	0.44
idf + tf + th + nse + inf	11.39	***idf + con + inf***	10.08	***con + spec + nse + inf***	0.00
tf + th + con + nse + inf	11.37	***idf + th + spec + nse***	10.08		
tf + th + con	11.36	***idf + con + spec + nse***	10.07		
tf + th + con + spec + inf	11.35	***idf + th + spec + inf***	10.02		
idf + tf + th + spec + inf	11.34	***th + con + spec***	9.98		
idf + tf + th + spec + nse + inf	11.34	***th + con***	9.89		

Table C.7: Summary of average precision figures for all combinations of characteristics on the AP collection with no weighting of characteristics

AP					
all	14.09	***tf + th + con + spec***	13.04	***con + spec + nse***	10.15
idf + tf + con + nse	14.07	***tf + th + con + inf***	13.04	***con + nse + inf***	10.15
idf + tf + con + spec + nse	14.07	***tf + th + con + spec + inf***	13.04	***con + spec + nse + inf***	10.15
idf + tf + con + nse + inf	14.07	***idf + th + con + nse***	12.27	***tf + nse***	10.11
idf + tf + con + spec + nse + inf	14.07	***idf + th + con + spec + nse***	12.27	***tf + spec + nse***	10.11
idf + tf + con	13.99	***idf + th + con + nse + inf***	12.27	***tf + spec + nse + inf***	10.11
idf + tf + con + spec	13.99	***idf + th + con + spec + nse + inf***	12.27	***idf***	10.10
idf + tf + con + inf	13.99	***idf + th + con***	12.02	***idf + spec***	10.10
idf + tf + con + spec + inf	13.99	***idf + th + con + spec***	12.02	***idf + inf***	10.10
idf + tf + th + con + nse	13.92	***idf + th + con + inf***	12.02	***idf + tf + inf***	10.10
idf + tf + th + con + spec + nse	13.92	***idf + th + con + spec + inf***	12.02	***idf + th + inf***	10.10
idf + tf + th + con + nse + inf	13.92	***th + con + nse***	11.18	***idf + spec + inf***	10.10
idf + tf + th + con	13.88	***th + con + spec + nse***	11.18	***idf + nse***	10.09
idf + tf + th + con + spec	13.88	***th + con + nse + inf***	11.18	***idf + spec + nse***	10.09
idf + tf + th + con + inf	13.88	***th + con + spec + nse + inf***	11.18	***idf + nse + inf***	10.09
idf + tf + th + con + spec + inf	13.88	***idf + th + nse***	11.12	***idf + spec + nse + inf***	10.09
idf + tf + nse	13.86	***idf + th + spec + nse***	11.12	***tf***	9.86
idf + tf + spec + nse	13.86	***idf + th + nse + inf***	11.12	***tf + spec***	9.86
idf + tf + nse + inf	13.86	***idf + th + spec + nse + inf***	11.12	***tf + inf***	9.86
idf + tf + spec + nse + inf	13.86	***th + con***	11.03	***tf + spec + inf***	9.86
idf + tf	13.67	***th + con + spec***	11.03	***con***	9.57
idf + tf + spec	13.67	***th + con + inf***	11.03	***con + spec***	9.57
idf + tf + spec + inf	13.67	***th + con + spec + inf***	11.03	***con + inf***	9.57
idf + tf + th + nse	13.65	***idf + th***	10.97	***con + spec + inf***	9.57
idf + tf + th + spec + inf	13.65	***idf + th + spec***	10.97	***spec + inf***	7.86
idf + tf + th + nse + inf	13.65	***idf + th + spec + inf***	10.97	***th + nse***	5.04
idf + tf + th + spec + nse + inf	13.65	***idf + con + nse***	10.96	***th + spec + nse***	5.04
idf + tf + th	13.64	***idf + con + spec + nse***	10.96	***th + nse + inf***	5.04
idf + tf + th + spec	13.64	***idf + con + nse + inf***	10.96	***th + spec + nse + inf***	5.04
idf + tf + th + inf	13.64	***idf + con + spec + nse + inf***	10.96	***th***	4.63
idf + tf + th + spec + nse	13.64	***idf + con***	10.67	***th + spec***	4.63
tf + con + nse	13.43	***idf + con + spec***	10.67	***th + inf***	4.63
tf + con + spec + nse	13.43	***idf + con + inf***	10.67	***th + spec + inf***	4.63
tf + con + nse + inf	13.43	***idf + con + spec + inf***	10.67	***nse***	1.00
tf + con + spec + nse + inf	13.43	***tf + th + nse***	10.65	***spec + nse***	1.00
tf + con	13.38	***tf + nse + inf***	10.65	***nse + inf***	1.00
tf + con + spec	13.38	***tf + th + spec + nse***	10.65	***spec + nse + inf***	1.00
tf + con + inf	13.38	***tf + th + nse + inf***	10.65	***spec***	0.47
tf + con + spec + inf	13.38	***tf + th + spec + nse + inf***	10.65	***inf***	0.44
tf + th + con + nse	13.09	***tf + th***	10.54		
tf + th + con + spec + nse	13.09	***tf + th + spec***	10.54		
tf + th + con + nse + inf	13.09	***tf + th + inf***	10.54		
tf + th + con + spec + nse + inf	13.09	***tf + th + spec + inf***	10.54		
tf + th + con	13.04	***con + nse***	10.15		

Table C.8: Summary of average precision figures for all combinations of characteristics on the AP collection with weighting of characteristics

WSJ					
idf + tf	15.65	***tf + th + con + nse***	12.79	***idf + inf***	11.69
idf + tf + nse	15.64	***tf + th + con***	12.76	***idf + th + spec + nse + inf***	11.60
idf + tf + con + nse	15.48	***con + inf***	12.73	***th + con + nse***	11.58
idf + tf + con	15.45	***tf + th + con + spec + nse + inf***	12.68	***idf + th + spec + inf***	11.57
idf + tf + con + nse + inf	14.92	***tf + th + con + spec + inf***	12.66	***idf + th***	11.56
idf + tf + con + inf	14.88	***idf + th + con + nse + inf***	12.64	***th + con***	11.55
tf + con + nse	14.86	***idf + th + con + inf***	12.64	***idf + th + spec + nse***	11.55
tf + con	14.79	***idf + tf + th + spec + nse + inf***	12.49	***idf + th + spec***	11.53
idf + tf + con + spec + nse	14.55	***idf + tf + th + spec + inf***	12.49	***tf + th + spec + nse***	11.44
idf + tf + con + spec	14.52	***idf + tf + th + spec + nse***	12.48	***con + spec + nse + inf***	11.38
tf + con + nse + inf	14.35	***tf + th + con + spec + nse***	12.40	***tf + th + spec***	11.37
idf + tf + nse + inf	14.35	***tf + th + con + spec***	12.37	***con + spec + inf***	11.32
idf + tf + con + spec + nse + inf	14.35	***idf + th + con + spec + nse + inf***	12.32	***con + spec + nse***	11.11
idf + tf + spec + nse	14.33	***idf + th + con + nse***	12.32	***th + spec + nse + inf***	11.09
idf + tf + con + spec + inf	14.33	***idf + th + con + spec + inf***	12.30	***con + spec***	11.07
idf + tf + spec + nse + inf	14.31	***idf + th + con + spec + nse***	12.27	***th + spec + inf***	11.06
idf + tf + spec	14.31	***idf + th + con + spec***	12.24	***con + nse***	10.83
tf + con + inf	14.30	***idf + th + con***	12.24	***idf + spec + nse + inf***	10.75
idf + tf + spec + inf	14.28	***idf + tf + th + nse + inf***	12.22	***idf + spec + inf***	10.73
tf + con + spec + nse	14.05	***idf + tf + th + spec***	12.21	***th + nse + inf***	10.69
tf + con + spec	14.00	***idf + nse***	12.19	***th + spec + nse***	10.67
tf + con + spec + nse + inf	13.80	***idf***	12.19	***tf + nse***	10.60
tf + con + spec + inf	13.75	***idf + con + spec + nse + inf***	12.15	***th + spec***	10.59
tf + spec + nse	13.65	***th + con + nse + inf***	12.08	***th + inf***	10.57
idf + con + nse + inf	13.63	***tf + th + nse + inf***	12.07	***idf + spec + nse***	10.42
tf + spec	13.61	***tf + nse + inf***	12.07	***idf + spec***	10.37
idf + tf + th + con + inf	13.60	***idf + con + spec + inf***	12.07	***nse + inf***	10.11
idf + tf + th + con + nse + inf	13.59	***th + con + spec + nse + inf***	12.00	***spec + nse + inf***	9.73
idf + con + inf	13.57	***tf + th + spec + nse + inf***	11.99	***spec + inf***	9.67
tf + spec + nse + inf	13.47	***tf + th + inf***	11.98	***tf + th + nse***	9.66
idf + tf + th + con + spec	13.42	***th + con + spec + inf***	11.97	***tf + th***	9.48
idf + tf + th + con	13.42	***tf + th + spec + inf***	11.97	***spec + nse***	9.42
idf + tf + th + con + nse	13.41	***th + con + inf***	11.94	***tf***	7.39
tf + spec + inf	13.38	***idf + th + nse + inf***	11.92	***th + nse***	6.98
tf + inf	13.36	***idf + con + spec + nse***	11.89	***nse***	1.05
all	13.33	***idf + con + spec***	11.88	***th***	1.00
tf + th + con + nse + inf	13.06	***th + con + spec + nse***	11.73	***inf***	0.48
idf + tf + th + con + spec + inf	13.04	***th + con + spec***	11.73	***spec***	0.42
idf + tf + th + nse	13.02	***idf + th + nse***	11.73	***con***	0.04
idf + tf + th + inf	12.99	***idf + con + nse***	11.72		
idf + tf + th	12.96	***idf + con***	11.72		
tf + th + con + inf	12.93	***idf + nse + inf***	11.70		
idf + tf + th + con + spec + nse	12.92	***idf + th + inf***	11.69		
con + nse + inf	12.87	***idf + tf + inf***	11.69		

Table C.9: Summary of average precision figures for all combinations of characteristics on the WSJ collection with no weighting of characteristics

WSJ					
all	15.73	***tf + con + spec***	14.75	***con + nse***	10.82
idf + tf + nse	15.67	***tf + con + inf***	14.75	***con + spec + nse***	10.82
idf + tf + spec + nse	15.67	***tf + con + spec + inf***	14.75	***con + nse + inf***	10.82
idf + tf + nse + inf	15.67	***th + con + nse***	13.60	***con + spec + nse + inf***	10.82
idf + tf + spec + nse + inf	15.67	***idf + th + con***	13.53	***tf + th + nse***	10.60
idf + tf	15.66	***idf + th + con + spec***	13.53	***tf + nse + inf***	10.60
idf + tf + spec	15.66	***idf + th + con + nse***	13.53	***tf + th + spec + nse***	10.60
idf + tf + spec + inf	15.66	***idf + th + con + inf***	13.53	***tf + th + nse + inf***	10.60
idf + tf + th + con + nse	15.59	***idf + th + con + spec + nse***	13.53	***tf + th + spec + nse + inf***	10.60
idf + tf + th + con + spec + nse	15.59	***idf + th + con + spec + inf***	13.53	***tf + th***	10.47
idf + tf + th + con + nse + inf	15.59	***idf + th + con + nse + inf***	13.53	***tf + th + spec***	10.47
idf + tf + th + con	15.58	***idf + th + con + spec + nse + inf***	13.53	***tf + th + inf***	10.47
idf + tf + th + con + spec	15.58	***idf + th***	13.32	***tf + th + spec + inf***	10.47
idf + tf + th + con + inf	15.58	***idf + th + spec***	13.32	***con + spec***	10.42
idf + tf + th + con + spec + inf	15.58	***idf + th + nse***	13.32	***con + inf***	10.42
idf + tf + con + nse	15.41	***idf + th + spec + nse***	13.32	***con + spec + inf***	10.42
idf + tf + con + spec + nse	15.41	***idf + th + spec + inf***	13.32	***tf + nse***	10.15
idf + tf + con + nse + inf	15.41	***idf + th + nse + inf***	13.32	***tf + spec + nse***	10.15
idf + tf + con + spec + nse + inf	15.41	***idf + th + spec + nse + inf***	13.32	***tf + spec + nse + inf***	10.15
idf + tf + con	15.40	***th + con + spec + nse***	12.60	***tf + spec***	10.03
idf + tf + con + spec	15.40	***th + con + nse + inf***	12.60	***tf + inf***	10.03
idf + tf + con + inf	15.40	***th + con + spec + nse + inf***	12.60	***tf + spec + inf***	10.03
idf + tf + con + spec + inf	15.40	***th + con***	12.55	***spec + inf***	9.67
idf + tf + th	15.37	***th + con + spec***	12.55	***tf***	7.39
idf + tf + th + spec	15.37	***th + con + inf***	12.55	***th + nse***	6.95
idf + tf + th + nse	15.37	***th + con + spec + inf***	12.55	***th + spec + nse***	6.95
idf + tf + th + inf	15.37	***idf***	12.19	***th + nse + inf***	6.95
idf + tf + th + spec + nse	15.37	***idf + spec***	12.19	***th + spec + nse + inf***	6.95
idf + tf + th + spec + inf	15.37	***idf + nse***	12.19	***th + spec***	6.70
idf + tf + th + nse + inf	15.37	***idf + inf***	12.19	***th + inf***	6.70
idf + tf + th + spec + nse + inf	15.37	***idf + tf + inf***	12.19	***th + spec + inf***	6.70
tf + th + con + nse	14.85	***idf + th + inf***	12.19	***nse***	1.05
tf + th + con + spec + nse	14.85	***idf + spec + nse***	12.19	***th***	1.00
tf + th + con + nse + inf	14.85	***idf + spec + inf***	12.19	***spec + nse***	0.93
tf + th + con + spec + nse + inf	14.85	***idf + nse + inf***	12.19	***nse + inf***	0.93
tf + th + con	14.84	***idf + spec + nse + inf***	12.19	***spec + nse + inf***	0.93
tf + th + con + spec	14.84	***idf + con + nse***	11.75	***inf***	0.48
tf + th + con + inf	14.84	***idf + con + spec + nse***	11.75	***spec***	0.42
tf + th + con + spec + inf	14.84	***idf + con + nse + inf***	11.75	***con***	0.04
tf + con + nse	14.76	***idf + con + spec + nse + inf***	11.75		
tf + con + spec + nse	14.76	***idf + con***	11.74		
tf + con + nse + inf	14.76	***idf + con + spec***	11.74		
tf + con + spec + nse + inf	14.76	***idf + con + inf***	11.74		
tf + con	14.75	***idf + con + spec + inf***	11.74		

Table C.10: Summary of average precision figures for all combinations of characteristics on the WSJ collection with weighting of characteristics

CACM							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	25	30	28	26	22	28
		78.13%	93.75%	87.50%	81.25%	68.75%	87.50%
tf	30	-	30	31	30	31	31
	93.75%		93.75%	96.88%	93.75%	96.88%	96.88%
theme	21	9	-	21	19	13	20
	65.63%	28.13%		65.63%	59.38%	40.63%	62.50%
context	12	12	16	-	13	7	14
	37.50%	37.50%	50.00%		40.63%	21.88%	43.75%
spec	5	8	14	7	-	5	9
	15.63%	25.00%	43.75%	21.88%		15.63%	28.13%
noise	30	30	29	31	29	-	31
	93.75%	93.75%	90.63%	96.88%	90.63%		96.88%
info_noise	19	11	23	20	21	16	-
	59.38%	34.38%	71.88%	62.50%	65.63%	50.00%	

Table C.11: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the CACM collection with no weighting of characteristics

CACM							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	29	30	30	29	28	28
		90.63%	93.75%	93.75%	90.63%	87.50%	87.50%
tf	30	-	30	30	30	29	28
	93.75%		93.75%	93.75%	93.75%	90.63%	87.50%
theme	16	5	-	15	14	11	15
	50.00%	15.63%		46.88%	43.75%	34.38%	46.88%
context	5	4	13	-	10	8	13
	15.63%	12.50%	40.63%		31.25%	25.00%	40.63%
spec	7	7	8	8	-	4	11
	21.88%	21.88%	25.00%	25.00%		12.50%	34.38%
noise	21	22	23	24	20	0	24
	65.63%	68.75%	71.88%	75.00%	62.50%		75.00%
info_noise	11	10	12	14	11	9	-
	34.38%	31.25%	37.50%	43.75%	34.38%	28.13%	

Table C.12: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the CACM collection with weighting of characteristics

CISI							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	*25*	*26*	*28*	*27*	*26*	*25*
		78.13%	*81.25%*	*87.50%*	*84.38%*	*81.25%*	*78.13%*
tf	*27*	-	*27*	*28*	*28*	*28*	*27*
	84.38%		*84.38%*	*87.50%*	*87.50%*	*87.50%*	*84.38%*
theme	*22*	*20*	-	*27*	*24*	*26*	*23*
	68.75%	*62.50%*		*84.38%*	*75.00%*	*81.25%*	*71.88%*
context	*3*	*2*	*6*	-	*2*	*2*	*4*
	9.38%	*6.25%*	*18.75%*		*6.25%*	*6.25%*	*12.50%*
spec	*4*	*6*	*9*	*5*	-	*6*	*7*
	12.50%	*18.75%*	*28.13%*	*15.63%*		*18.75%*	*21.88%*
noise	*2*	*3*	*7*	*2*	*3*	*0*	*3*
	6.25%	*9.38%*	*21.88%*	*6.25%*	*9.38%*		*9.38%*
info_noise	*21*	*18*	*24*	*25*	*25*	*21*	-
	65.63%	*56.25%*	*75.00%*	*78.13%*	*78.13%*	*65.63%*	

Table C.13: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the CISI collection with no weighting of characteristics

CISI							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	22	24	26	26	25	26
		68.75%	75.00%	81.25%	81.25%	78.13%	81.25%
tf	26	-	24	26	28	28	28
	81.25%		75.00%	81.25%	87.50%	87.50%	87.50%
theme	23	18	-	27	22	24	23
	71.88%	56.25%		84.38%	68.75%	75.00%	71.88%
context	1	0	7	-	6	5	8
	3.13%	0.00%	21.88%		18.75%	15.63%	25.00%
spec	12	10	13	11	-	11	13
	37.50%	31.25%	40.63%	34.38%		34.38%	40.63%
noise	12	11	17	14	14	0	17
	37.50%	34.38%	53.13%	43.75%	43.75%		53.13%
info_noise	9	9	11	11	11	18	-
	28.13%	28.13%	34.38%	34.38%	34.38%	56.25%	

Table C.14: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the CISI collection with weighting of characteristics

MEDLARS							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	25	26	31	26	18	25
		78.13%	81.25%	96.88%	81.25%	56.25%	78.13%
tf	29	-	26	31	30	27	30
	90.63%		81.25%	96.88%	93.75%	84.38%	93.75%
theme	27	23	-	26	25	25	25
	84.38%	71.88%		81.25%	78.13%	78.13%	78.13%
context	6	3	10	-	7	4	9
	18.75%	9.38%	31.25%		21.88%	12.50%	28.13%
spec	7	8	11	6	-	6	7
	21.88%	25.00%	34.38%	18.75%		18.75%	21.88%
noise	26	29	23	28	28	0	29
	81.25%	90.63%	71.88%	87.50%	87.50%		90.63%
info_noise	7	9	12	6	8	8	-
	21.88%	28.13%	37.50%	18.75%	25.00%	25.00%	

Table C.15: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the MEDLARS collection with no weighting of characteristics

MEDLARS							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	25	26	26	26	25	26
		78.13%	81.25%	81.25%	81.25%	78.13%	81.25%
tf	26	-	24	26	28	28	28
	81.25%		75.00%	81.25%	87.50%	87.50%	87.50%
theme	23	18	-	27	22	24	23
	71.88%	56.25%		84.38%	68.75%	75.00%	71.88%
context	1	0	7	-	6	5	8
	3.13%	0.00%	21.88%		18.75%	15.63%	25.00%
spec	12	10	13	11	-	11	13
	37.50%	31.25%	40.63%	34.38%		34.38%	40.63%
noise	12	11	17	14	14	0	17
	37.50%	34.38%	53.13%	43.75%	43.75%		53.13%
info_noise	9	9	11	11	11	18	-
	28.13%	28.13%	34.38%	34.38%	34.38%	56.25%	

Table C.16: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the MEDLARS collection with weighting of characteristics

AP							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	28	29	28	28	14	26
		87.50%	90.63%	87.50%	87.50%	43.75%	81.25%
tf	31	-	30	31	31	31	39
	96.88%		93.75%	96.88%	96.88%	96.88%	121.88%
theme	10	4	-	9	12	10	14
	31.25%	12.50%		28.13%	37.50%	31.25%	43.75%
context	29	29	29	-	27	26	26
	90.63%	90.63%	90.63%		84.38%	81.25%	81.25%
spec	20	22	28	22	-	31	25
	62.50%	68.75%	87.50%	68.75%		96.88%	78.13%
noise	26	23	25	22	22	22	24
	81.25%	71.88%	78.13%	68.75%	68.75%		75.00%
info_noise	13	13	22	10	19	19	-
	40.63%	40.63%	68.75%	31.25%	59.38%	59.38%	

Table C.17: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the AP collection with no weighting of characteristics

AP							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	32	32	32	32	32	32
		100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
tf	31	-	30	31	31	31	29
	96.88%		93.75%	96.88%	96.88%	96.88%	90.63%
theme	10	4	-	9	12	10	14
	31.25%	12.50%		28.13%	37.50%	31.25%	43.75%
context	29	29	29	-	27	26	26
	90.63%	90.63%	90.63%		84.38%	81.25%	81.25%
spec	20	22	28	22	-	31	25
	62.50%	68.75%	87.50%	68.75%		96.88%	78.13%
noise	26	23	25	22	22	22	24
	81.25%	71.88%	78.13%	68.75%	68.75%		75.00%
info_noise	13	13	22	10	19	19	-
	40.63%	40.63%	68.75%	31.25%	59.38%	59.38%	

Table C.18: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the AP collection with weighting of characteristics

WSJ							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	31	32	32	32	32	31
		96.88%	100.00%	100.00%	100.00%	100.00%	96.88%
tf	31	-	32	32	32	32	31
	96.88%		100.00%	100.00%	100.00%	100.00%	96.88%
theme	12	2	-	12	16	13	12
	37.50%	6.25%		37.50%	50.00%	40.63%	37.50%
context	28	30	32	-	32	30	32
	87.50%	93.75%	100.00%		100.00%	93.75%	100.00%
spec	4	8	10	7	-	10	8
	12.50%	25.00%	31.25%	21.88%		31.25%	25.00%
noise	23	25	25	26	29	22	27
	71.88%	78.13%	78.13%	81.25%	90.63%		84.38%
info_noise	20	16	30	23	23	23	-
	62.50%	50.00%	93.75%	71.88%	71.88%	71.88%	

Table C.19: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the WSJ collection with no weighting of characteristics

WSJ							
	idf	*tf*	*theme*	*context*	*spec*	*noise*	*info_noise*
idf	-	32	31	31	32	31	32
		100.00%	96.88%	96.88%	100.00%	96.88%	100.00%
tf	31	-	32	32	32	32	31
	96.88%		100.00%	100.00%	100.00%	100.00%	96.88%
theme	24	24	-	32	28	27	27
	75.00%	75.00%		100.00%	87.50%	84.38%	84.38%
context	17	25	32	-	24	24	25
	53.13%	78.13%	100.00%		75.00%	75.00%	78.13%
spec	2	2	2	1	-	0	2
	6.25%	6.25%	6.25%	3.13%		0.00%	6.25%
noise	17	28	21	28	24	-	25
	53.13%	87.50%	65.63%	87.50%	75.00%		78.13%
info_noise	0	2	1	1	0	1	-
	0.00%	6.25%	3.13%	3.13%	0.00%	3.13%	

Table C.20: Percentage of times a characteristic (row) improved a combination containing another characteristics (column) on the WSJ collection with weighting of characteristics

Appendix D

Supplementary results from Chapter Five

Level	*tf*	*idf* + *tf*	*idf* + *theme*	*idf* + *context*	*tf* + *theme*	*tf* + *context*	*theme* + *context*
1	**56.82**	56.73	54.92	50.43	55.96	56.76	51.73
2	**54.13**	54.00	51.13	47.27	53.09	54.17	48.12
3	51.62	51.50	48.68	45.13	50.40	**51.66**	46.11
4	**49.60**	49.48	46.24	41.93	47.79	49.49	43.59
5	**46.85**	46.79	44.00	39.88	45.02	46.82	41.91
6	**44.23**	44.22	40.53	36.24	41.36	44.09	38.26
7	42.80	42.83	39.06	34.70	40.43	**42.90**	37.17
8	**48.08**	47.95	40.00	36.38	46.02	47.99	38.28
9	**49.88**	49.67	39.88	36.26	48.56	49.68	37.43
10	41.77	41.54	39.99	32.56	41.13	41.60	**56.14**

Table D.1: Average precision figures for retrieval using combinations of two characteristics, varying the importance of characteristics.
Highest value shown in bold.

.

Level	tf	*tf* + *idf* + *context*	*tf* + *idf* + *theme*	*tf* + *theme* + *context*	*idf* + *theme* + *context*
1	**56.82**	54.63	56.68	56.75	51.73
2	54.13	51.15	53.81	**54.18**	48.14
3	51.62	48.66	51.50	**51.70**	46.14
4	**49.60**	45.97	49.35	49.53	43.57
5	46.85	43.92	46.56	**46.86**	41.80
6	44.23	42.07	43.60	**44.31**	38.23
7	42.80	40.71	42.22	**43.00**	37.16
8	48.08	45.58	47.57	**48.14**	38.29
9	49.88	47.33	49.25	**49.90**	37.50
10	41.77	40.91	41.53	**41.86**	34.98

Table D.2 Average precision figures for retrieval using combinations of three characteristics, varying the importance of characteristics.
Highest value shown in bold.

Level	*tf*	*all*
1	**56.82**	56.75
2	54.13	**54.18**
3	51.62	**51.70**
4	**49.60**	49.53
5	46.85	**46.86**
6	44.23	**44.31**
7	42.80	**43.00**
8	48.08	**48.14**
9	49.88	**49.90**
10	41.77	**44.48**

Table D.3: Average precision figures for retrieval using combinations of four characteristics, varying the importance of characteristics.
Highest value shown in bold.

		Relevance level									
Topic	**Char**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	***tf***	**55.28**	**51.65**	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	idf + tf	55.13	51.27	49.10	47.95	45.62	42.07	41.91	43.02	49.34	37.95
	tf + th	55.11	51.61	**50.09**	**48.33**	**46.37**	39.27	38.49	40.97	46.58	44.52
	tf + co	55.08	51.52	49.32	48.10	45.94	**42.36**	**42.12**	43.09	49.49	38.19
	idf+ th	46.40	42.69	41.47	41.37	39.95	36.01	39.39	41.67	38.46	36.33
	idf+ co	51.73	46.63	46.63	44.20	43.26	39.24	39.38	38.41	36.87	37.16
	th+ co	48.13	44.40	43.19	42.43	41.28	37.88	43.23	**44.92**	42.01	**45.43**
B	***tf***	**55.28**	**51.65**	**49.53**	**48.25**	**45.94**	**42.32**	**41.95**	**43.03**	**49.62**	**38.07**
	idf + tf	51.30	49.47	46.63	47.09	43.80	38.47	36.36	42.86	39.50	33.95
	tf + th	49.49	46.86	43.88	44.49	40.89	36.01	33.70	36.84	36.32	31.78
	tf + co	51.30	49.29	46.43	46.55	43.44	38.49	36.52	42.84	39.46	33.92
	idf+ th	48.66	47.26	43.93	44.90	42.05	38.87	33.40	33.71	32.51	29.79
	idf+ co	49.94	49.18	47.30	47.54	44.01	39.05	36.93	37.42	37.15	29.99
	th+ co	49.48	47.68	44.63	45.48	42.51	39.75	34.69	34.85	33.02	29.92
C	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	idf + tf	59.32	59.79	57.79	54.07	51.89	**44.96**	41.80	**40.89**	47.48	33.80
	tf + th	57.95	58.73	56.11	51.83	49.37	39.93	40.11	40.78	45.58	33.38
	tf + co	**59.85**	**60.45**	**58.36**	**54.66**	**52.45**	44.88	42.22	41.22	47.95	34.39
	idf+ th	50.08	48.02	48.96	43.83	41.50	32.61	34.15	36.11	41.45	34.19
	idf+ co	58.58	55.66	53.21	48.39	47.22	42.71	**43.22**	40.50	42.35	35.74
	th+ co	53.24	51.49	51.55	46.38	45.57	36.72	38.75	40.66	46.51	**39.48**
D	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	idf + tf	57.32	53.65	49.25	49.41	44.56	**43.21**	**45.26**	**44.26**	42.22	40.53
	tf + th	55.91	52.47	47.59	47.27	42.81	41.56	41.63	42.78	40.20	40.31
	tf + co	**57.44**	**53.79**	**49.64**	**49.54**	**44.70**	43.19	45.09	44.03	41.97	40.26
	idf+ th	50.11	47.17	43.08	39.72	37.85	34.78	32.35	36.04	35.99	32.40
	idf+ co	55.33	51.23	45.68	45.69	43.14	41.92	44.17	43.58	47.73	**54.87**
	th+ co	52.60	47.92	44.13	43.10	41.55	38.40	37.52	36.75	38.14	35.89
Own	***tf***	55.28	51.65	49.53	**48.25**	**45.94**	42.32	**41.95**	**43.03**	**49.62**	**38.07**
	idf + tf	**57.25**	52.00	**49.94**	45.95	44.09	**43.80**	41.34	40.35	31.66	30.64
	tf + th	56.10	50.63	49.03	44.66	42.22	42.11	39.29	37.25	33.77	29.62
	tf + co	57.10	**52.34**	49.97	46.02	44.04	43.75	41.24	40.40	31.43	30.68
	idf+ th	52.19	47.18	42.30	38.75	37.97	38.62	35.28	33.85	29.83	28.96
	idf+ co	56.66	51.61	48.08	46.03	42.87	39.57	36.76	36.49	33.55	31.46
	th+ co	53.93	48.14	44.28	41.42	40.60	39.12	35.63	35.79	30.99	30.10
TR	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	49.62	38.07

	idf + tf	58.67	56.48	**52.61**	**52.16**	**52.07**	**58.24**	55.07	**55.57**	**53.23**	57.22
	tf + th	**60.61**	**57.23**	52.17	50.58	50.47	55.20	52.97	53.78	52.02	55.89
	tf + co	58.45	56.34	52.52	52.03	51.90	58.06	**55.08**	55.52	53.20	**57.23**
	idf+ th	54.56	51.49	49.78	44.52	43.39	40.99	37.53	33.76	27.18	28.35
	idf+ co	57.52	52.86	48.30	45.23	45.01	45.70	42.33	40.42	35.68	44.78
	th+ co	51.70	48.31	46.38	42.07	42.16	40.54	36.45	33.08	28.14	30.61

Table D.4: *th* - theme, *co* - context. Combining combinations of two characteristics against *tf* for each relevance level and for each topic, varying the importance of the characteristics. Highest value shown in bold.

		Relevance level									
Topic	**Char**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	***tf***	**55.28**	**51.65**	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	Best	**55.28**	**51.65**	**50.09**	**48.33**	**46.37**	42.32	42.12	44.92	**49.62**	45.43
tf*+ *idf* + *co		50.96	46.59	44.31	42.63	41.32	37.76	40.92	41.57	43.27	35.90
tf*+ *idf* + *th		55.00	50.89	48.80	47.43	45.36	40.39	40.06	41.06	45.72	37.77
tf*+ *th* + *co		55.04	51.48	49.28	48.06	45.96	**42.36**	42.14	43.11	49.49	38.19
idf* + *th* + *co		48.15	44.35	43.17	42.28	41.20	37.82	**43.10**	**44.99**	42.16	**45.85**
B	***tf***	**55.28**	**51.65**	**49.53**	**48.25**	**45.94**	**42.32**	**41.95**	**43.03**	**49.62**	**38.07**
	Best	**55.28**	**51.65**	**49.53**	**48.25**	**45.94**	**42.32**	**41.95**	**43.03**	**49.62**	**38.07**
tf* + *idf* + *co		51.76	49.43	47.11	46.94	43.45	40.07	35.06	36.84	34.60	32.55
tf* + *idf* + *th		51.11	49.08	46.68	46.66	43.35	38.18	36.17	42.55	38.87	33.08
tf* + *th* + *co		51.05	49.07	46.36	46.47	43.34	38.40	36.43	42.86	39.46	33.92
idf* + *th* + *co		49.18	47.75	44.79	45.54	42.52	39.78	34.65	34.82	32.99	29.87
C	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	**43.03**	**49.62**	38.07
	Best	59.85	60.45	58.36	54.66	52.45	44.96	**43.22**	40.89	**49.62**	**39.48**
tf* + *idf* + *co		**58.93**	59.48	57.41	52.18	50.03	38.79	38.54	40.72	45.18	36.64
tf* + *idf* + *th		59.19	59.67	57.58	53.94	51.73	45.01	41.81	40.86	47.52	33.92
tf* + *th* + *co		59.88	**60.48**	**58.41**	**54.69**	**52.50**	**45.39**	42.24	41.22	47.95	34.39
idf* + *th* + *co		52.85	51.04	51.21	46.04	45.20	36.62	38.65	40.51	46.44	39.40
D	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	Best	57.44	53.79	49.64	49.54	**44.70**	43.21	45.26	44.26	**49.62**	**54.87**
tf* + *idf* + *co		53.87	48.73	44.44	42.75	40.63	40.02	36.34	36.68	35.82	37.98
tf* + *idf* + *th		57.38	53.53	49.44	49.60	44.52	**43.28**	**45.68**	**44.91**	43.20	42.05
tf* + *th* + *co		**57.51**	**53.92**	**49.78**	**49.69**	44.85	**43.28**	45.64	44.72	42.95	41.78
idf* + *th* + *co		52.54	47.87	44.01	42.92	41.38	38.25	37.50	36.73	38.12	35.90
Own	***tf***	55.28	51.65	49.53	**48.25**	**45.94**	42.32	**41.95**	**43.03**	**49.62**	**38.07**
	Best	**57.25**	52.34	49.94	**48.25**	**45.94**	43.80	**41.95**	**43.03**	**49.62**	**38.07**
tf* + *idf* + *co		54.74	48.72	45.89	43.20	43.59	**44.24**	40.04	40.37	33.60	31.49
tf* + *idf* + *th		57.21	51.96	**50.41**	46.41	44.44	44.01	41.55	41.07	33.02	30.82
tf* + *th* + *co		57.16	**52.39**	50.11	46.14	44.17	43.86	41.35	40.52	31.77	30.78
idf* + *th* + *co		54.35	48.56	44.59	41.81	40.50	39.20	35.74	35.94	31.36	30.61
TR	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	49.62	38.07
	Best	**60.61**	**57.23**	52.61	52.16	52.07	**58.24**	55.08	**55.57**	**53.23**	57.23
tf* + *idf* + *co		55.57	53.27	50.61	49.25	48.87	58.07	**55.70**	54.52	51.94	46.17
tf* + *idf* + *th		58.92	56.59	**52.64**	**52.25**	**52.10**	58.21	55.05	55.51	53.18	57.00
tf* + *th* + *co		58.42	56.32	52.52	52.06	51.91	57.94	54.82	55.35	53.22	**57.26**

idf* + *th* + *co	51.85	48.45	46.55	42.26	42.12	40.52	36.54	33.15	28.33	30.80

Table D.5: *th* - theme, *co* - context. Combining combinations of two characteristics against *tf* for each relevance level and for each topic, varying the importance of the characteristics. **Best** is the highest average precision achieved from comparing *tf* against combinations of two characteristics. Highest value shown in bold.

		Relevance level									
Topic	**Char**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	***tf***	**55.28**	**51.65**	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	Best	**55.28**	**51.65**	**50.09**	**48.33**	**46.37**	**42.36**	**43.10**	**44.99**	**49.62**	**45.85**
	all	55.04	51.48	49.28	48.06	45.96	**42.36**	42.14	43.11	49.49	38.19
B	***tf***	**55.28**	**51.65**	**49.53**	**48.25**	**45.94**	**42.32**	**41.95**	**43.03**	**49.62**	**38.07**
	Best	**55.28**	**51.65**	**49.53**	**48.25**	**45.94**	**42.32**	**41.95**	**43.03**	**49.62**	**38.07**
	all	51.05	49.07	46.36	46.47	43.34	38.40	36.43	42.86	39.46	33.92
C	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	**43.03**	**49.62**	38.07
	Best	58.93	**60.48**	**58.41**	**54.69**	**52.50**	**45.39**	**43.20**	**43.03**	**49.62**	**39.48**
	all	**59.88**	**60.48**	**58.41**	**54.69**	**52.50**	**45.39**	42.24	41.22	47.95	34.39
D	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	**49.62**	38.07
	Best	**57.51**	**53.92**	**49.78**	**49.69**	44.70	**43.28**	**45.68**	**44.91**	**49.62**	**54.87**
	all	**57.51**	**53.92**	**49.78**	**49.69**	**44.85**	**43.28**	45.64	44.72	42.95	41.78
Own	***tf***	55.28	51.65	49.53	**48.25**	**45.94**	42.32	**41.95**	**43.03**	**49.62**	**38.07**
	Best	**57.25**	**52.39**	**50.41**	**48.25**	**45.94**	**44.24**	**41.95**	**43.03**	**49.62**	**38.07**
	all	57.16	**52.39**	50.11	46.14	44.17	43.86	41.35	40.52	31.77	30.78
TR	***tf***	55.28	51.65	49.53	48.25	45.94	42.32	41.95	43.03	49.62	38.07
	Best	**60.61**	**57.23**	**52.64**	**52.25**	**52.10**	**58.24**	**55.70**	**55.57**	**53.23**	**57.26**
	all	58.42	56.32	52.52	52.06	51.91	57.94	54.82	55.35	53.22	**57.26**

Table D.6: Combining combinations of all characteristics (*all*) against *tf* and for each relevance level and for each topic, varying the importance of the characteristics.
Best is the highest average precision achieved from comparing *tf* against combinations of two or three characteristics. Highest value shown in bold.

Topic	Char	Relevance level									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	**Fback 1**	52.34	46.49	42.82	42.47	38.22	34.41	35.88	33.83	27.26	25.65
	Fback 2	52.52	46.84	43.21	42.85	38.68	35.22	36.34	34.20	27.95	26.02
	Fback 3	52.1	46.14	42.67	42.33	38.2	34.26	35.92	33.73	27.18	25.46
	Fback 5	51.62	45.97	42.50	42.15	37.97	34.12	35.93	33.74	27.27	25.54
	idf	51.16	45.74	42.08	41.65	37.49	33.62	35.47	33.34	27.24	25.83
	Comb	**55.04**	**51.48**	**49.28**	**48.06**	**45.96**	**42.36**	**42.14**	**43.11**	**49.49**	**38.19**
	F_4	51.24	45.81	42.14	41.78	37.62	33.79	35.56	33.43	27.29	25.98
B	**Fback 1**	47.27	45.85	43.33	43.62	40.51	36.92	32.17	31.66	30.95	31.67
	Fback 2	47.47	46.03	43.53	43.73	40.32	36.53	31.93	31.79	31.28	32.26
	Fback 3	47.13	45.78	43.28	43.51	40.37	36.71	32.09	31.55	30.94	31.57
	Fback 5	46.95	45.63	43.17	43.52	40.13	36.38	31.86	31.47	31.05	31.69
	idf	47.10	44.81	42.27	43.13	39.81	36.10	30.95	30.50	30.06	30.51
	Comb	**51.05**	**49.07**	**46.36**	**46.47**	**43.34**	**38.40**	**36.43**	**42.86**	**39.46**	**33.92**
	F_4	47.32	45.88	43.36	43.52	40.06	36.61	31.57	31.34	30.81	31.71
C	**Fback 1**	53.14	50.15	49.23	44.20	41.03	30.84	30.65	29.30	27.41	26.67
	Fback 2	52.84	50.01	48.98	44.12	40.96	31.05	30.79	29.36	27.57	27.11
	Fback 3	53.15	50.13	49.23	44.34	41.16	30.98	30.76	29.37	27.48	26.72
	Fback 5	53.05	50.05	49.17	44.28	41.06	31.03	30.80	29.36	27.59	26.88
	idf	50.47	47.86	46.26	42.49	40.54	30.53	30.41	28.98	27.26	26.58
	Comb	**59.88**	**60.48**	**58.41**	**54.69**	**52.50**	**45.39**	**42.24**	**41.22**	**47.95**	**34.39**
	F_4	52.53	49.65	48.79	43.88	40.64	30.49	30.17	28.80	27.03	26.21
D	**Fback 1**	50.81	44.42	39.67	37.46	35.77	34.22	34.26	37.76	38.51	35.34
	Fback 2	50.77	44.54	39.90	37.69	35.91	34.28	34.29	37.93	38.70	35.46
	Fback 3	50.56	44.38	39.65	37.51	35.82	34.27	34.27	37.83	38.6	35.2
	Fback 5	50.51	44.34	39.67	37.45	35.68	33.97	34.03	37.75	38.47	35.13
	idf	50.34	44.24	39.48	37.23	35.51	33.86	33.90	37.67	38.35	35.00
	Comb	**57.51**	**53.92**	**49.78**	**49.69**	**44.85**	**43.28**	**45.64**	**44.72**	**42.95**	**41.78**
	F_4	50.27	44.21	39.55	37.37	35.62	33.93	34.14	37.77	38.50	35.23
Own	**Fback 1**	53.15	46.82	42.45	37.56	34.34	33.26	30.89	30.87	27.07	26.81
	Fback 2	53.33	47.01	42.46	37.68	34.32	33.25	30.85	30.95	26.97	26.82
	Fback 3	52.95	46.91	42.59	37.73	34.49	33.47	31.07	31.19	27.44	26.99
	Fback 5	52.58	46.45	42.43	37.90	34.63	33.45	31.06	31.46	27.33	26.74
	idf	55.23	46.60	42.28	37.68	34.22	32.74	30.49	30.66	26.44	26.30
	Comb	**57.16**	**52.39**	**50.11**	**46.14**	**44.17**	**43.86**	**41.35**	**40.52**	**31.77**	**30.78**

	F_4	52.77	46.57	42.23	37.49	34.17	32.76	30.54	30.81	26.52	26.38
TR	**Fback 1**	54.73	48.93	45.00	43.35	42.46	40.50	36.89	32.89	24.42	30.93
	Fback 2	54.81	49.18	45.23	43.63	43.15	40.39	36.74	33.10	24.57	31.54
	Fback 3	54.66	48.98	45.12	43.35	42.48	40.39	36.94	32.88	24.48	31.05
	Fback 5	54.31	48.76	44.63	42.82	42.21	40.17	36.68	32.77	24.37	30.85
	idf	56.74	51.32	44.64	42.84	42.48	39.91	36.32	32.42	23.85	30.35
	Comb	**58.42**	**56.32**	**52.52**	**52.06**	**51.91**	**57.94**	**54.82**	**55.35**	**53.22**	**57.26**
	F_4	55.09	49.35	44.99	43.31	42.88	40.41	36.66	32.79	24.06	30.79

Table D.7: Comparison of average precision across topics for the four relevance feedback functions, F_4 and *idf*.

Fback1 - Feedback 1 strategy, **Fback2** - Feedback 2 strategy, **Fback3** - Feedback 3 strategy, **Fback5** – Feedback 5 strategy, **Comb** - best combination (no feedback). Highest value shown in bold.

	Feedback 1							
Level	***tf***	***theme***	***context***	***tf* + *theme***	***tf* + *context***	***theme* + *context***	***tf* + *theme* + *context***	***Total***
1	6.1	16.4	5.9	4.3	15.7	8.3	**16.8**	73.6
2	6.2	**15.9**	6.4	4.1	15.0	9.3	14.3	71.2
3	5.8	**16.4**	7.0	4.9	15.4	7.5	13.0	70.0
4	5.5	**16.6**	6.7	5.6	16.0	7.4	11.7	69.5
5	5.6	**17.5**	5.4	4.8	14.1	7.9	12.2	67.5
6	4.7	**18.1**	5.3	4.2	14.2	7.2	11.3	65.0
7	5.3	**19.7**	5.3	4.7	12.9	6.6	12.8	67.4
8	4.1	**19.1**	5.3	4.9	12.9	6.6	13.1	66.0
9	4.8	**17.4**	4.8	4.3	12.8	6.2	12.1	62.3
10	3.8	**15.6**	4.6	4.2	13.7	6.5	12.9	61.2

Table D.8: %age of times each characteristic was used in modified query for each relevance level for Feedback 1 strategy.
Total is the total % of of query terms a characteristic could have been applied to.
Highest value at each relevance level shown in bold.

	Feedback 2							
Level	***tf***	***theme***	***context***	***tf* + *theme***	***tf* + *context***	***theme* + *context***	***tf* + *theme* + *context***	***Total***
1	10.4	18.5	2.5	3.6	10.9	3.6	**27.7**	77.2
2	9.9	17.8	2.4	4.1	10.7	3.3	**30.4**	78.7
3	9.3	17.2	2.4	4.1	10.4	3.1	**32.6**	79.2
4	9.3	16.7	2.5	4.3	11.3	3.1	**32.4**	79.5
5	9.3	15.7	2.5	4.4	11.4	3.0	**34.6**	80.9
6	9.7	15.5	2.3	4.2	11.4	2.6	**37.1**	82.8
7	11.1	16.0	2.6	4.4	11.2	2.5	**36.3**	84.1
8	11.2	14.3	2.3	3.4	11.7	2.2	**42.2**	87.4
9	9.2	12.3	2.2	2.8	10.9	2.3	**48.9**	88.6
10	9.7	11.7	1.4	2.6	9.6	2.5	**50.3**	87.6

Table D.9: %age of times each characteristic was used in modified query for each relevance level for Feedback 2 strategy.
Total is the total % of of query terms a characteristic could have been applied to. Highest value at each relevance level shown in bold.

Topic	Char	Relevance level									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	**Fback 1**	65.48	63.80	61.47	61.45	62.73	60.40	64.44	70.10	71.76	72.67
	Fback 2	56.21	52.55	49.95	50.93	50.37	47.00	44.53	49.01	59.85	42.99
	Fback 3	**68.06**	**66.8**	**67.17**	**68.12**	**70.12**	**68.06**	**72.34**	**81.21**	**80.01**	**81.80**
	Fback 5	55.29	54.03	54.59	53.73	55.53	52.04	58.07	61.45	62.54	62.76
	idf	51.16	45.74	42.08	41.65	37.49	33.62	35.47	33.34	27.24	25.83
	Comb	55.04	51.48	49.28	48.06	45.96	42.36	42.14	43.11	49.49	38.19
	F_4	51.25	47.20	43.91	42.60	38.17	34.46	36.62	33.66	28.77	27.59
B	**Fback 1**	64.54	62.54	61.82	58.65	56.84	53.14	50.18	53.37	51.66	42.47
	Fback 2	55.19	52.08	48.51	49.83	46.42	43.51	37.83	43.24	44.10	38.88
	Fback 3	**68.77**	**68.3**	**67.28**	**67.94**	**66.00**	**59.46**	**60.01**	**68.31**	**66.13**	**49.09**
	Fback 5	57.69	56.27	55.49	54.94	52.96	48.15	45.79	51.35	51.98	40.95
	idf	47.10	44.81	42.27	43.13	39.81	36.10	30.95	30.50	30.06	30.51
	Comb	51.05	49.07	46.36	46.47	43.34	38.40	36.43	42.86	39.46	33.92
	F_4	50.38	48.85	45.97	45.73	42.55	37.73	32.14	31.83	30.87	29.76
C	**Fback 1**	67.38	66.42	66.37	65.14	63.20	58.94	59.33	58.95	72.32	68.55
	Fback 2	61.20	61.26	58.21	54.99	53.62	49.51	50.45	50.02	59.95	48.54
	Fback 3	**70.73**	**71.47**	**71.22**	**68.24**	**67.98**	**61.97**	**63.64**	**61.11**	**72.89**	**70.63**
	Fback 5	63.19	60.54	59.96	54.85	54.07	49.42	51.84	48.63	58.12	59.29
	idf	50.47	47.86	46.26	42.49	40.54	30.53	30.41	28.98	27.26	26.58
	Comb	59.88	60.48	58.41	54.69	52.50	45.39	42.24	41.22	47.95	34.39
	F_4	56.35	53.68	54.06	50.00	46.34	35.09	35.25	31.80	30.30	27.68
D	**Fback 1**	64.75	61.21	55.43	55.12	51.43	51.95	53.21	52.70	53.55	59.59
	Fback 2	58.66	54.11	49.38	49.04	44.06	44.07	46.75	45.85	45.93	47.64
	Fback 3	**67.63**	**64.85**	**62.33**	**63.37**	**60.84**	**60.54**	**59.63**	**60.48**	**62.3**	**65.93**
	Fback 5	59.32	55.97	54.21	55.73	53.59	52.52	53.57	49.81	55.72	54.54
	idf	50.34	44.24	39.48	37.23	35.51	33.86	33.90	37.67	38.35	35.00
	Comb	57.51	53.92	49.78	49.69	44.85	43.28	45.64	44.72	42.95	41.78
	F_4	51.01	44.81	40.00	38.96	37.34	35.34	36.42	40.10	39.13	33.48
Own	**Fback 1**	65.79	61.36	58.21	58.08	53.71	52.68	50.02	50.02	43.78	48.77
	Fback 2	56.92	52.16	50.00	45.20	43.42	43.82	41.25	40.25	34.93	30.26
	Fback 3	**71.03**	**66.76**	**61.87**	**61.39**	**59.07**	**58.79**	**58.51**	**55.92**	**52.98**	**51.16**
	Fback 5	61.90	56.96	52.19	52.13	48.80	46.28	44.20	41.18	43.49	39.84

	idf	55.23	46.60	42.28	37.68	34.22	32.74	30.49	30.66	26.44	26.30
	Comb	57.16	52.39	50.11	46.14	44.17	43.86	41.35	40.52	31.77	30.78
	F_4	54.36	48.89	43.91	39.35	36.59	35.65	33.46	33.14	27.31	27.21
TR	**Fback 1**	65.44	61.58	60.02	59.54	60.03	64.05	57.56	54.90	52.07	48.95
	Fback 2	58.95	55.36	51.92	51.88	50.78	53.62	51.71	54.59	54.97	54.34
	Fback 3	**68.94**	**65.99**	**68.02**	**65.2**	**67.41**	**68.35**	**65.29**	**64.89**	**60.74**	**59.27**
	Fback 5	54.61	51.67	51.90	48.41	47.75	47.34	44.16	43.29	40.51	41.79
	idf	56.74	51.32	44.64	42.84	42.48	39.91	36.32	32.42	23.85	30.35
	Comb	58.42	56.32	52.52	52.06	51.91	57.94	54.82	55.35	53.22	57.26
	F_4	54.14	48.05	44.25	41.70	41.38	40.07	36.57	32.90	24.91	31.55

Table D.10: Comparison of average precision across topics for retrospective feedback using four relevance feedback functions, F_4 and *idf*.
Fback1 - Feedback 1 strategy, **Fback2** - Feedback 2 strategy, **Fback3** - Feedback 3 strategy, **Comb** - best combination (no feedback). Comparison of average precision across topics for three relevance feedback functions, F_4 and *idf*.

	Feedback techniques				Baselines			
Level	**Feedback Strategy 1**	**Feedback Strategy 2**	**Feedback Strategy 3**	**Feedback Strategy 5**	***idf***	***tf***	**Best Combination**	**F_4**
1	5	1	**17**	0	0	0	0	0
2	3	0	**20**	0	0	0	0	0
3	2	0	**21**	0	0	0	0	0
4	2	0	**21**	0	0	0	0	0
5	1	1	**20**	1	0	0	0	0
6	3	1	**19**	0	0	0	0	0
7	1	1	**21**	0	0	0	0	0
8	1	2	**20**	0	0	0	0	0
9	1	3	**17**	2	0	0	0	0
10	5	1	**10**	0	0	1	0	0

Table D.11: Average precision figures for retrospective feedback techniques compared with *idf* ranking.
Highest value shown in bold.

	Feedback 1							
	Possible	***tf***	***theme***	***context***	***tf*** + ***theme***	***tf*** + ***context***	***theme*** + ***context***	***tf*** + ***theme*** + ***context***
1	912	64	115	43	48	173	101	**179**
2	903	63	120	61	48	**155**	111	151
3	872	61	132	54	42	**158**	91	143
4	823	57	141	52	39	**155**	87	115
5	772	50	**137**	47	43	136	73	108
6	663	32	118	38	38	**126**	61	84
7	619	39	101	31	39	**105**	54	88
8	512	29	80	31	34	**91**	45	69
9	438	26	**70**	27	32	67	41	63
10	263	17	42	18	18	42	20	**44**

Table D.12: Number of times each characteristic was used in modified query for each relevance level.
Possible is the number of times a characteristic could have been used.

	Feedback 2							
	Possible	***tf***	***theme***	***context***	***tf* + *theme***	***tf* + *context***	***theme* + *context***	***tf* + *theme* + *context***
1	20298	1572	**4073**	717	939	2350	1149	3847
2	19464	1504	**4427**	671	948	2272	1082	3954
3	18289	1411	**4047**	640	860	2225	1025	3937
4	16954	1354	**3722**	576	848	2129	940	3681
5	15883	1282	3299	568	784	2060	878	**3642**
6	13773	1160	2831	481	662	1834	751	**3315**
7	12160	1007	2481	449	602	1616	695	**3025**
8	9707	896	1960	330	485	1316	463	**2590**
9	7634	634	1520	266	364	1078	301	**2304**
10	5135	397	1044	186	197	652	221	**1546**

Table D.13: Number of times each characteristic was used in modified query for each relevance level.
Possible is the number of times a characteristic could have been used.

	Feedback 1							
Level	***tf***	***theme***	***context***	***tf* + *theme***	***tf* + *context***	***theme* + *context***	***tf* + *theme* + *context***	***Total***
1	7.0	12.6	4.7	5.3	19.0	11.1	**19.6**	79.3
2	7.0	13.3	6.8	5.3	**17.2**	12.3	16.7	78.5
3	7.0	15.1	6.2	4.8	**18.1**	10.4	16.4	78.1
4	6.9	17.1	6.3	4.7	**18.8**	10.6	14.0	78.5
5	6.5	**17.7**	6.1	5.6	17.6	9.5	14.0	76.9
6	4.8	17.8	5.7	5.7	**19.0**	9.2	12.7	75.0
7	6.3	16.3	5.0	6.3	**17.0**	8.7	14.2	73.8
8	5.7	15.6	6.1	6.6	**17.8**	8.8	13.5	74.0
9	5.9	**16.0**	6.2	7.3	15.3	9.4	14.4	74.4
10	6.5	16.0	6.8	6.8	16.0	7.6	**16.7**	76.4

Table D.14: %age of times each characteristic was used in modified query for each relevance level.
Total is the total % of query terms a characteristic could have been applied to. Highest value at each relevance level shown in bold.

	Feedback 2							
Level	***tf***	***theme***	***context***	***tf* + *theme***	***tf* + *context***	***theme* + *context***	***tf* + *theme* + *context***	***Total***
1	7.7	**20.1**	3.5	4.6	11.6	5.7	19.0	72.2
2	7.7	**22.7**	3.4	4.9	11.7	5.6	20.3	76.3
3	7.7	**22.1**	3.5	4.7	12.2	5.6	21.5	77.3
4	8.0	**22.0**	3.4	5.0	12.6	5.5	21.7	78.2
5	8.1	20.8	3.6	4.9	13.0	5.5	**22.9**	78.8
6	8.4	20.6	3.5	4.8	13.3	5.5	**24.1**	80.1
7	8.3	20.4	3.7	5.0	13.3	5.7	**24.9**	81.2
8	9.2	20.2	3.4	5.0	13.6	4.8	**26.7**	82.8
9	8.3	19.9	3.5	4.8	14.1	3.9	**30.2**	84.7
10	7.7	20.3	3.6	3.8	12.7	4.3	**30.1**	82.6

Table D.15: %age of times each characteristic was used in modified query for each relevance level.
Total is the total % of query terms a characteristic could have been applied to.
Highest value at each relevance level shown in bold.

Appendix E

Supplementary results from Chapter Six

	AP
Retrieval function	**Average precision**
idf + tf + context	13.8
idf + tf	12.9
tf + context	12.3
all	11.2
tf + theme + context	10.8
idf + context	10.4
idf	10.1
idf + theme + context	9.9
idf + tf + theme	9.9
tf	9.9
context	9.6
theme + context	9.4
tf + theme	8.8
idf + theme	5.1
theme	4.6

	WSJ
Retrieval function	**Average precision**
idf + tf	15.2
idf + tf + context	15.0
tf + context	14.3
all	12.7
idf + tf + theme	12.6
tf + theme + context	12.4
idf	12.2
idf + theme + context	11.6
idf + theme	11.2
idf + context	11.0
theme + context	11.0
tf + theme	9.3
tf	7.4
theme	1.0
context	0.0

	AP
Retrieval function	**Average precision**
idf + tf + context	13.4
all	13.3
idf + tf	13.1
idf + tf + theme	13.1
tf + theme + context	12.5
tf + context	12.4
idf + theme + context	11.5
theme + context	10.6
idf + theme	10.5
idf + context	10.2
tf + theme	10.2
idf	10.1
tf	9.9
context	9.6
theme	4.6

	WSJ
Retrieval function	**Average precision**
idf + tf	15.4
idf + tf + context	15.2
all	15.1
tf + theme + context	14.5
idf + tf + theme	14.4
tf + context	14.2
idf + theme + context	13.3
idf + theme	13.1
idf	12.2
theme + context	12.2
idf + context	11.5
tf + theme	10.3
tf	7.4
theme	1.0
context	0.0

Table E.1: Combination of characteristics using the **simple** method, ordered by decreasing average precision, with no weighting of characteristics (Top) and weighting of characteristics (Bottom)

	AP
Retrieval function	**Average precision**
idf + theme + context	16.6
idf + theme	14.2
idf + tf + context	13.0
idf + context	12.6
idf	10.1
tf	9.9
context	9.6
theme + context	8.9
all	8.5
tf + theme	7.4
idf + tf	6.6
tf + context	5.4
theme	4.6
tf + theme + context	3.5
idf + tf + theme	1.9

	WSJ
Retrieval function	**Average precision**
idf + tf + theme	19.9
tf + theme + context	15.8
idf + tf	15.6
tf + context	15.2
idf + tf + context	15.1
all	14.7
theme + context	14.6
idf + theme + context	13.5
idf	12.2
idf + theme	11.2
tf +theme	9.5
tf	7.4
idf + context	5.8
theme	1.0
context	0.0

	AP		WSJ
Retrieval function	**Average precision**	**Retrieval function**	**Average precision**
all	16.5	***idf + tf***	15.8
idf + tf + theme	14.8	***idf + tf + theme***	15.3
idf + tf	13.0	***tf + context***	15.2
idf + theme + context	12.9	***idf + theme + context***	14.8
idf + context	12.5	***all***	14.2
idf + theme	12.2	***theme + context***	14.0
idf	10.1	***idf + tf + context***	13.8
theme + context	9.9	***idf + theme***	12.6
tf	9.9	***idf***	12.2
context	9.6	***idf + context***	12
tf + theme	7.7	***tf***	7.4
theme	4.6	***theme***	1.0
tf + theme + context	3.1	***tf + theme + context***	1.0
tf + context	2.9	***tf + theme***	0.6
idf + tf + context	2.2	***context***	0.0

Table E.2: Combination of characteristics using Dempster's combination rule, ordered by decreasing average precision, with no weighting of characteristics (Top) and weighting of characteristics (Bottom)

CISI				
Combination	**simple, no weighting,**	**DS, no weighting**	**simple, weighting,**	**DS, weighting**
all	11.6	9.4	**12.7**	11.7
context	**9.6**	**9.6**	**9.6**	**9.6**
idf	**11.5**	**11.5**	**11.5**	**11.5**
idf* + *context	**12.7**	8.4	**12.7**	11.2
idf* + *tf	**12.9**	8.5	12.8	11.3
idf* + *tf* + *context	11.0	8.4	**11.2**	**11.2**
idf* + *tf* + *theme	12.1	10.1	**12.7**	11.3
idf* + *theme	11.4	**11.5**	11.4	**11.5**
idf* + *theme* + *context	11.4	**12.7**	10.9	11.8
tf	**12.5**	**12.5**	**12.5**	**12.5**
tf* + *context	10.6	5.0	**10.8**	4.9
tf* + *theme	11.0	7.3	**12.0**	5.1
tf* + *theme* + *context	**11.4**	5.0	10.8	5.0
theme	**9.6**	**9.6**	**9.6**	**9.6**
theme* + *context	**11.0**	3.2	10.3	2.8

Table E.3: Summarised results of combining characteristics, using Dempster's combination rule (**DS**), summing characteristic scores (**simple**), either weighting the characteristics scores (**weighting**) or treating characteristics as equally important (**no weighting**).

CISI			
	No weighting	**Weighting**	**Total**
simple	4	5	**9**
DS	2	2	**4**
Total	**6**	**7**	

Table E.4: Number of times each strategy gave highest average precision for a combination of characteristics

CISI				
Recall	***idf* 1** ***tf* 1** ***theme* 1** ***context* 1**	***idf* 1** ***tf* 0.75** ***theme* 0.15** ***context* 0.5**	***idf* 0.25** ***tf* 0.5** ***theme* 0.75** ***context* 1**	***idf* 0.5** ***tf* 0.25** ***theme* 0.25** ***context* 0.5**
10	20.9	26.2	23.4	24.6
20	14.9	18.6	16.4	17.3
30	12.3	14.9	13.6	14.3
40	10.1	12.6	11.2	11.7
50	8.2	10.3	9.1	9.5
60	7.3	9.1	8.2	8.6
70	6.3	7.8	7.2	7.5
80	5.5	6.9	6.2	6.5
90	4.7	5.8	5.4	5.6
100	3.7	4.7	4.1	4.3
average	**9.4**	**11.7**	**10.5**	**11.0**

Table E.5: Recall precision figures for combination of all characteristics, using Dempster's Combination Rule, and various characteristic weighting functions on the CISI collection. *idf* 0.5 signifies that all *idf* values have been mulitplied by a weighting value of 0.5

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	26.2	26.5	26.6	26.6	26.6
20	18.6	20.8	21.1	21.2	21.2
30	14.9	18.8	19.0	19.5	19.4
40	12.6	16.9	16.9	17.6	17.9
50	10.3	15.3	15.1	15.7	15.8
60	9.1	13.3	13.3	14.0	14.3
70	7.8	11.2	11.0	11.8	11.8
80	6.9	8.9	8.9	9.3	9.3
90	5.8	6.8	7.1	7.4	7.4
100	4.7	5.1	4.9	5.0	4.9
average	**11.7**	**14.4**	**14.4**	**14.8**	**14.9**

Table E.6: RP figures for the Feedback 5.1 method

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	21.0	21.1	21.2	21.3	21.3
20	14.9	17.3	17.7	17.7	17.8
30	12.3	15.0	15.7	16.0	16.1
40	10.1	13.7	14.3	14.8	14.9
50	8.2	12.3	12.7	13.0	13.1
60	7.3	10.7	11.0	11.2	11.4
70	6.3	8.9	9.2	9.5	9.5
80	5.5	6.8	7.3	7.4	7.6
90	4.7	5.5	5.8	5.8	5.9
100	3.7	3.9	3.8	3.8	3.8
average	**9.4**	**11.5**	**11.9**	**12.0**	**12.1**

Table E.7: RP figures for Feedback 5.2 method

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	26.8	28.6	30.3	28.4	28.2
20	19.5	21.9	23.6	22.4	22.8
30	14.7	17.6	18.6	18.5	18.5
40	12.1	15.2	15.9	16.2	16.4
50	10.2	14.1	14.4	14.5	14.9
60	9.0	12.7	12.3	12.8	13.3
70	7.4	10.8	10.1	11.0	11.2
80	6.1	8.2	7.9	8.6	8.8
90	5.2	6.2	6.4	6.8	6.9
100	4.0	4.3	4.1	4.2	4.3
average	**11.5**	**14.0**	**14.4**	**14.3**	**14.5**

Table E.8: RP figures for Feedback 5.3 method

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	26.2	26.6	26.6	26.6	26.6
20	18.6	21.1	21.2	21.3	21.3
30	14.9	19.0	19.1	19.5	19.5
40	12.6	17.4	17.3	17.9	18.1
50	10.3	15.5	15.6	15.9	16.1
60	9.1	13.7	13.7	14.1	14.5
70	7.8	11.6	11.4	11.9	12.0
80	6.9	9.0	9.3	9.5	9.6
90	5.8	6.9	7.3	7.5	7.5
100	4.7	5.1	4.9	5.0	5.0
average	**11.7**	**14.6**	**14.6**	**14.9**	**15.0**

Table E.9: RP figures for Feedback 5.4 method

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	25.5	25.7	25.7	25.7	25.7
20	18.4	20.1	20.1	20.1	20.1
30	15.3	17.6	17.8	17.8	17.8
40	12.6	16.4	16.1	16.3	16.3
50	10.7	15.1	15.1	15.2	15.0
60	9.4	13.4	13.5	13.4	12.9
70	7.9	11.5	11.5	11.4	11.2
80	6.8	9.0	8.6	8.9	8.8
90	5.9	6.5	6.4	6.2	6.1
100	4.7	4.7	4.4	4.4	4.4
average	**11.7**	**14.0**	**13.9**	**13.9**	**13.8**

Table E.10: RP figures for F4 using default combination of characteristics as an initial ranking

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	21.0	21.0	21.0	21.0	21.0
20	14.9	14.9	14.9	14.9	14.9
30	12.3	12.3	12.3	12.3	12.3
40	10.1	10.1	10.1	10.1	10.1
50	8.2	8.2	8.2	8.2	8.2
60	7.3	7.3	7.3	7.3	7.3
70	6.3	6.3	6.3	6.3	6.3
80	5.5	5.5	5.5	5.5	5.5
90	4.7	4.7	4.7	4.7	4.7
100	3.7	3.7	3.7	3.7	3.7
average	**9.4**	**9.4**	**9.4**	**9.4**	**9.4**

Table E.11: RP figures using no weighting of characteristics and no selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	26.2	26.2	26.2	26.2	26.2
20	18.6	18.6	18.6	18.6	18.6
30	14.9	14.9	14.9	14.9	14.9
40	12.6	12.6	12.6	12.6	12.6
50	10.3	10.3	10.3	10.3	10.3
60	9.1	9.1	9.1	9.1	9.1
70	7.8	7.8	7.8	7.8	7.8
80	6.9	6.9	6.9	6.9	6.9
90	5.8	5.8	5.8	5.8	5.8
100	4.7	4.7	4.7	4.7	4.7
average	**11.7**	**11.7**	**11.7**	**11.7**	**11.7**

Table E.12: RP figures using weighting of characteristics and no selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	21.0	21.4	21.4	21.4	21.4
20	14.9	17.0	17.1	17.1	17.2
30	12.3	14.8	15.2	15.1	15.1
40	10.1	12.7	13.5	13.3	13.4
50	8.2	11.3	11.8	11.9	11.6
60	7.3	9.7	9.9	9.9	10.0
70	6.3	7.5	8.0	8.3	8.2
80	5.5	6.1	6.7	6.6	6.7
90	4.7	5.0	5.2	5.1	5.1
100	3.7	3.7	3.7	3.7	3.8
average	**9.4**	**10.9**	**11.3**	**11.3**	**11.3**

Table E.13: RP figures using no weighting of characteristics and selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	26.2	26.8	26.8	26.8	26.8
20	18.6	20.7	20.7	20.7	20.7
30	14.9	17.1	17.5	17.5	17.6
40	12.6	15.2	15.7	15.7	15.7
50	10.3	12.9	13.1	13.3	13.5
60	9.1	11.1	11.3	11.4	11.6
70	7.8	8.9	9.0	9.2	9.6
80	6.9	7.4	7.7	7.9	7.9
90	5.8	6.2	6.3	6.2	6.3
100	4.7	4.7	4.7	4.8	4.8
average	**11.7**	**13.1**	**13.3**	**13.4**	**13.5**

Table E.14: RP figures using weighting of characteristics and selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	25.2	25.4	25.4	25.4	26.0
20	17.0	20.2	20.1	20.3	20.7
30	14.6	18.9	19.3	19.6	20.0
40	12.1	17.1	17.7	17.9	18.2
50	10.1	15.9	15.7	16.5	16.5
60	9.1	13.8	13.8	14.6	14.9
70	7.6	11.7	11.5	12.4	12.5
80	6.6	9.3	9.3	10.0	9.9
90	5.7	6.8	7.3	7.5	7.6
100	4.5	4.9	4.8	4.8	4.9
average	**11.7**	**14.4**	**14.5**	**14.9**	**15.1**

Table E.15: RP figures using weighting of characteristics, selection of characteristics and additional weights given by quality of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	25.6	25.7	25.7	25.7	25.9
20	19.2	21.2	21.4	21.4	21.5
30	15.2	19.7	20.2	20.3	20.5
40	12.7	17.9	18.6	18.8	18.8
50	10.9	16.1	16.8	17.2	17.3
60	9.6	14.3	14.7	15.3	15.3
70	7.9	12.2	12.2	12.8	12.9
80	6.8	9.5	9.7	10.1	10.3
90	5.8	6.9	7.5	7.6	7.8
100	4.5	4.9	5.0	4.9	5.0
average	**11.7**	**14.8**	**15.2**	**15.4**	**15.5**

Table E.16: RP figures using weighting of characteristics, selection of characteristics and additional weights given by quality and strength of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	21.0	21.3	21.3	21.3	21.3
20	14.2	16.3	16.6	16.6	16.6
30	12.2	14.2	14.9	15.1	15.1
40	10.1	12.5	12.8	13.3	13.4
50	8.4	11.0	11.0	11.4	11.7
60	7.6	9.5	9.6	10.0	10.4
70	6.3	7.8	7.8	8.0	8.5
80	5.5	6.5	6.4	6.6	6.7
90	4.8	5.1	5.1	5.3	5.4
100	3.7	3.8	3.8	3.8	3.9
average	**9.4**	**10.8**	**10.9**	**11.2**	**11.3**

Table E.17: RP figures for the full model of RF, scoring by index weights with selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	25.5	25.9	25.9	25.9	25.9
20	18.7	20.4	20.6	20.6	20.6
30	15.2	18.6	18.9	19.0	18.9
40	12.7	15.7	15.9	16.3	16.4
50	10.8	13.6	13.6	14.1	14.1
60	9.5	11.3	11.7	12.1	12.1
70	7.8	9.4	9.4	9.8	9.8
80	6.7	7.7	7.6	8.1	7.9
90	5.8	6.1	6.0	6.3	6.3
100	4.5	4.5	4.5	4.6	4.6
average	**11.7**	**13.3**	**13.4**	**13.7**	**13.7**

Table E.18: RP figures for the full model of RF, scoring by index weights and characteristic strength with selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	25.2	25.2	25.2	25.2	25.2
20	17.0	17.1	18.3	18.5	18.5
30	14.6	14.4	16.5	17.0	17.2
40	12.1	12.0	14.4	14.9	15.2
50	10.1	10.9	13.0	14.0	14.2
60	9.1	8.7	10.4	12.0	12.4
70	7.6	7.0	8.4	10.1	10.6
80	6.6	5.3	6.7	7.8	8.4
90	5.7	4.7	5.5	6.1	6.1
100	4.5	4.1	4.5	4.5	4.4
average	**11.3**	**10.9**	**12.3**	**13.0**	**13.2**

Table E.19: RP figures for the full model of RF, scoring by index weights and characteristic quality with selection of characteristics

CISI					
Recall	**Iteration 0**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
10	25.5	25.5	25.5	25.6	25.6
20	18.7	19.0	20.2	20.3	20.3
30	15.2	15.4	17.0	17.4	17.6
40	12.7	12.7	15.3	16.1	16.4
50	10.8	11.6	14.1	15.1	15.2
60	9.5	9.3	11.8	13.4	13.8
70	7.8	7.3	9.8	12.0	12.2
80	6.7	5.9	7.6	9.7	9.7
90	5.8	4.9	5.9	6.8	6.8
100	4.5	4.2	4.5	4.6	4.7
average	**11.7**	**11.6**	**13.2**	**14.1**	**14.2**

Table E.20: RP figures for the full model of RF, scoring by index weights and characteristic strength and quality with selection of characteristics

Appendix F

Supplementary results from Chapter Ten

Porter		AP			SJM			WSJ	
	25	**50**	**100**	**25**	**50**	**100**	**25**	**50**	**100**
Cov	**5.15**	**6.26**	**5.57**	**7.24**	**3.74**	**3.28**	**2.67**	**1.65**	**1.60**
Cov Porter	-4.58	**0.17**	**4.60**	-1.01	-6.05	-3.77	-1.07	-2.42	-0.28
Cov Selection	**2.96**	**5.39**	**10.79**	**12.38**	**9.31**	**8.85**	**3.98**	**2.32**	**2.20**
Exp	**6.06**	**3.82**	**1.64**	**2.84**	**1.23**	-1.22	**0.80**	-0.86	-0.83
Exp Porter	-0.89	**0.40**	**0.50**	-4.44	-4.50	-3.18	-1.90	-1.52	-0.76
Exp Selection	**0.23**	**4.43**	**4.28**	-1.82	**3.32**	**5.29**	-0.40	-0.14	**1.01**
Jos	**6.22**	**6.08**	**4.03**	**11.92**	**10.55**	**4.75**	**2.15**	**1.89**	**1.59**
Jos Porter	-1.11	**1.71**	**1.53**	**3.56**	**2.89**	**5.65**	-3.48	-2.44	**0.25**
Jos Selection	**3.07**	**5.08**	**4.19**	**16.07**	**11.69**	**7.78**	**0.73**	**0.45**	**1.42**
Just selection	-1.38	**1.40**	**2.42**	**6.49**	**6.37**	**4.81**	-1.14	-0.12	**0.68**
Relevancy	-15.21	-8.22	-4.51	-40.89	-33.38	-21.22	-19.20	-11.93	-6.77
Relevancy Porter	**25.77**	**26.61**	**20.18**	**25.77**	**7.51**	**11.70**	-11.38	-4.28	-0.12
Var	-1.61	-1.24	-0.67	-0.95	-0.80	-1.16	-3.64	-2.79	-1.39
Var Porter	-8.74	-4.45	**1.01**	-8.27	-6.39	**2.39**	-6.45	-3.8	-1.49
Var Selection	**0.98**	**2.22**	**2.60**	**5.62**	**6.05**	**8.21**	-0.18	**0.10**	**0.49**

Table F.1: Percentage increase over no feedback for query reformulation techniques using Porter weighting scheme and 25, 50 or 100 documents per feedback iteration

F_4		AP			SJM			WSJ	
	25	50	100	25	50	100	25	50	100
Cov	**7.68**	**7.00**	**5.58**	**7.24**	**3.74**	**3.24**	**2.67**	**1.65**	**1.57**
Cov F_4	-1.08	**6.29**	**6.50**	**2.45**	-1.17	**7.45**	**0.93**	-0.66	**0.50**
Cov Selection	**2.96**	**11.17**	**10.79**	**12.38**	**9.31**	**7.80**	**3.98**	**2.32**	**2.20**
Exp	**11.93**	**6.92**	**3.43**	**9.64**	**5.13**	**0.69**	**2.91**	**0.88**	**0.44**
Exp F_4	**9.91**	**5.65**	**4.07**	**9.33**	**5.78**	**5.18**	**4.70**	**2.39**	**1.76**
Exp Selection	**3.57**	**6.47**	**5.15**	**1.34**	**5.37**	**5.24**	**1.30**	**1.73**	**1.95**
Jos	**6.78**	**7.13**	**5.36**	**10.66**	**7.05**	**4.76**	**1.71**	**1.94**	**1.26**
Jos F_4	-0.92	**2.16**	**5.15**	**4.95**	**1.95**	**1.59**	**0.55**	**0.36**	**0.88**
Jos Selection	**5.06**	**6.52**	**7.79**	**16.37**	**12.93**	**9.04**	**3.21**	**2.97**	**2.33**
Just selection	-1.38	**1.40**	**2.42**	**6.49**	**6.37**	**4.81**	-1.14	-0.12	**0.68**
Relevancy	-15.14	-8.03	-4.43	-41.01	-33.32	-21.26	-19.19	-11.92	**0.19**
Relevancy F_4	**26.56**	**27.80**	**21.37**	-0.09	**8.52**	**11.66**	-12.98	-3.61	-6.73
Var	**0.04**	**0.08**	-0.14	**3.58**	**5.82**	**2.35**	-2.36	-2.07	-5.72
Var F_4	-3.10	-0.86	**1.00**	**3.28**	**7.25**	**5.18**	-2.72	-0.82	**1.13**
Var Selection	**1.65**	**3.02**	**2.93**	**11.27**	**10.41**	**8.21**	**1.50**	**1.25**	**1.45**

Table F.2: Percentage increase over no feedback for query reformulation techniques using F4 weighting scheme and 25, 50 or 100 documents per feedback iteration

wpq		AP			SJM			WSJ	
	25	**50**	**100**	**25**	**50**	**100**	**25**	**50**	**100**
Cov	**5.15**	**4.11**	**3.18**	**7.24**	**3.74**	**3.28**	**2.67**	**1.65**	**1.60**
Cov *wpq*	-4.73	-1.83	**0.24**	-2.27	-7.37	-3.08	-1.07	-2.42	-0.28
Cov Selection	**2.96**	**5.39**	**5.26**	**12.38**	**9.31**	**7.78**	**3.98**	**2.32**	**2.20**
Exp	**38.51**	**35.52**	**24.47**	**2.84**	**1.23**	**24.58**	**0.80**	-0.86	**0.41**
Exp *wpq*	**35.10**	**28.96**	**20.86**	**42.41**	**32.51**	**25.15**	**12.98**	**7.81**	**5.39**
Exp Selection	**15.37**	**20.20**	**15.07**	**23.37**	**20.79**	**21.43**	**1.30**	**1.73**	**1.95**
Jos	**17.09**	**15.65**	**12.39**	**12.33**	**14.39**	**11.41**	**2.15**	**1.89**	**0.25**
Jos *wpq*	**4.51**	**6.23**	**8.46**	-0.26	**0.95**	**1.73**	-3.48	-2.44	**1.73**
Jos Selection	**12.36**	**15.18**	**15.56**	**16.07**	**16.95**	**14.23**	**6.27**	**4.82**	**3.68**
Just selection	-1.38	**1.40**	**2.42**	**6.49**	**6.37**	**4.81**	-1.14	-0.12	**0.68**
Relevancy	-16.63	-9.77	-5.91	-41.24	-33.94	-22.05	-19.19	-11.92	-6.76
Relevancy *wpq*	**35.00**	**29.55**	**21.16**	**46.16**	**33.17**	**25.30**	-12.95	-3.61	**0.16**
Var	-0.67	-0.67	-0.67	**38.97**	**28.25**	**20.34**	-2.36	-2.07	-0.79
Var *wpq*	**32.79**	**22.78**	**18.25**	**36.36**	**28.96**	**28.96**	-2.72	-0.82	**0.55**
Var Selection	**14.83**	**16.02**	**13.76**	**26.44**	**22.61**	**19.12**	**1.50**	**1.25**	**1.11**

Table F.3: Percentage increase over no feedback for query reformulation techniques using *wpq* weighting scheme and 25, 50 or 100 documents per feedback iteration

	AP			SJM			WSJ		
	25	**50**	**100**	**25**	**50**	**100**	**25**	**50**	**100**
B1	52%	50%	61%	42%	40%	46%	42%	51%	22%
B2	33%	40%	61%	27%	29%	33%	31%	40%	44%
B3	46%	52%	67%	60%	67%	70%	42%	53%	53%
Cov	**67%**	**58%**	**83%**	**67%**	**71%**	76%	53%	**64%**	**67%**
Jos	50%	**58%**	67%	65%	**71%**	**78%**	**56%**	56%	60%
B1	33%	29%	30%	23%	13%	11%	29%	31%	22%
B2	15%	15%	17%	6%	4%	4%	20%	24%	31%
B3	15%	27%	28%	33%	35%	30%	20%	24%	31%
Cov	**44%**	**40%**	**57%**	**42%**	**44%**	**43%**	**31%**	**42%**	**44%**
Jos	29%	31%	35%	38%	19%	37%	29%	25%	35%

Table F.4: Affect of varying *n* when using Porter term weighting scheme

	AP			SJM			WSJ		
	25	**50**	**100**	**25**	**50**	**100**	**25**	**50**	**100**
B1	**73%**	**69%**	80%	**79%**	**81%**	**85%**	62%	69%	**78%**
B2	67%	63%	76%	73%	77%	80%	36%	47%	56%
B3	46%	52%	67%	60%	67%	70%	44%	53%	56%
Cov	67%	58%	80%	67%	71%	76%	56%	64%	67%
Jos	65%	67%	**83%**	65%	67%	76%	**64%**	**73%**	73%
B1	**44%**	**56%**	**61%**	**54%**	**54%**	**59%**	**51%**	**62%**	**64%**
B2	40%	31%	33%	31%	40%	46%	20%	22%	29%
B3	23%	21%	24%	15%	15%	11%	20%	22%	29%
Cov	27%	23%	33%	13%	19%	15%	24%	24%	33%
Jos	31%	27%	33%	15%	25%	22%	21%	21%	30%

Table F.5: Affect of varying *n* when using *wpq* term weighting scheme

			AP		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	64%	56%	72%	68%
B2	100%	100%	69%	69%	75%
B3	64%	50%	100%	95%	91%
Cov	56%	34%	66%	100%	66%
Jos	71%	50%	83%	88%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	79%	71%	83%	79%
B2	100%	100%	84%	84%	89%
B3	68%	64%	100%	92%	92%
Cov	71%	57%	82%	100%	82%
Jos	68%	61%	82%	82%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	86%	75%	82%	82%
B2	86%	100%	71%	79%	75%
B3	68%	65%	100%	100%	90%
Cov	61%	58%	82%	100%	76%
Jos	74%	68%	90%	94%	100%

Table F.6: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the Porter weighting scheme on the AP collection

			SJM		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	50%	65%	75%	75%
B2	77%	100%	77%	69%	85%
B3	45%	34%	100%	90%	90%
Cov	47%	28%	81%	100%	88%
Jos	48%	35%	84%	90%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	63%	79%	84%	84%
B2	86%	100%	93%	93%	93%
B3	47%	41%	100%	94%	91%
Cov	47%	38%	88%	100%	88%
Jos	47%	38%	85%	88%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	67%	81%	86%	86%
B2	93%	100%	87%	87%	93%
B3	53%	41%	100%	100%	94%
Cov	51%	37%	91%	100%	91%
Jos	50%	39%	83%	89%	100%

Table F.7: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the Porter weighting scheme on the SJM collection

			WSJ		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	58%	58%	74%	74%
B2	79%	100%	64%	79%	86%
B3	58%	47%	100%	89%	100%
Cov	58%	46%	71%	100%	92%
Jos	54%	46%	73%	85%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	70%	65%	83%	70%
B2	89%	100%	78%	89%	83%
B3	63%	58%	100%	92%	96%
Cov	66%	55%	76%	100%	79%
Jos	64%	60%	92%	92%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	100%	80%	80%	80%
B2	50%	100%	75%	80%	80%
B3	33%	63%	100%	96%	92%
Cov	27%	53%	77%	100%	83%
Jos	30%	59%	81%	93%	100%

Table F.8: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the Porter weighting scheme on the WSJ collection

			AP		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	66%	55%	79%	66%
B2	90%	100%	57%	71%	67%
B3	73%	55%	100%	95%	91%
Cov	72%	47%	66%	100%	72%
Jos	76%	56%	80%	92%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	85%	70%	85%	81%
B2	96%	100%	75%	83%	88%
B3	76%	72%	100%	92%	96%
Cov	82%	71%	82%	100%	93%
Jos	69%	66%	75%	81%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	84%	72%	81%	84%
B2	96%	100%	71%	79%	86%
B3	74%	65%	100%	100%	97%
Cov	68%	58%	82%	100%	87%
Jos	73%	65%	81%	89%	100%

Table F.9: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the F_4 weighting scheme on the AP collection

			SJM		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	56%	70%	78%	81%
B2	83%	100%	67%	72%	78%
B3	66%	41%	100%	90%	90%
Cov	66%	41%	81%	100%	91%
Jos	65%	41%	76%	85%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	69%	72%	76%	79%
B2	91%	100%	73%	77%	82%
B3	66%	50%	100%	94%	94%
Cov	65%	50%	88%	100%	88%
Jos	66%	51%	86%	86%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	71%	71%	81%	84%
B2	88%	100%	80%	84%	88%
B3	69%	63%	100%	100%	100%
Cov	71%	60%	91%	100%	97%
Jos	70%	59%	86%	92%	100%

Table F.10: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the F_4 weighting scheme on the SJM collection

			WSJ		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	63%	54%	75%	75%
B2	94%	100%	63%	75%	75%
B3	68%	53%	100%	89%	100%
Cov	75%	50%	71%	100%	92%
Jos	72%	48%	76%	88%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	77%	65%	85%	77%
B2	95%	100%	71%	81%	81%
B3	71%	63%	100%	92%	96%
Cov	76%	59%	76%	100%	86%
Jos	71%	61%	82%	89%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	86%	57%	68%	68%
B2	96%	100%	60%	68%	72%
B3	64%	60%	100%	96%	96%
Cov	63%	57%	80%	100%	90%
Jos	68%	64%	86%	96%	100%

Table F.11: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the F_4 weighting scheme on the WSJ collection

			AP		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	89%	49%	74%	71%
B2	97%	100%	50%	78%	75%
B3	77%	73%	100%	95%	95%
Cov	81%	78%	66%	100%	94%
Jos	81%	77%	68%	97%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	91%	61%	73%	76%
B2	100%	100%	67%	73%	77%
B3	80%	80%	100%	92%	92%
Cov	86%	79%	82%	100%	93%
Jos	83%	77%	77%	87%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	95%	70%	84%	89%
B2	100%	100%	71%	83%	89%
B3	84%	81%	100%	100%	94%
Cov	84%	78%	84%	100%	92%
Jos	87%	82%	76%	89%	100%

Table F.12: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the *wpq* weighting scheme on the AP collection

			SJM		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	84%	61%	76%	71%
B2	91%	100%	60%	69%	71%
B3	79%	72%	100%	90%	90%
Cov	91%	75%	81%	100%	88%
Jos	87%	81%	84%	90%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	87%	67%	72%	69%
B2	92%	100%	68%	70%	70%
B3	81%	78%	100%	94%	91%
Cov	82%	76%	88%	100%	91%
Jos	84%	81%	91%	97%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	87%	69%	77%	77%
B2	92%	100%	68%	73%	73%
B3	84%	78%	100%	100%	97%
Cov	86%	77%	91%	100%	97%
Jos	86%	77%	89%	97%	100%

Table F.13: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the *wpq* weighting scheme on the SJM collection

			WSJ		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	***Jos***
B1	100%	57%	57%	75%	82%
B2	100%	100%	69%	81%	81%
B3	80%	55%	100%	90%	100%
Cov	84%	52%	72%	100%	100%
Jos	79%	45%	69%	86%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	***Jos***
B1	100%	61%	58%	77%	84%
B2	90%	100%	71%	81%	86%
B3	75%	63%	100%	92%	100%
Cov	83%	59%	76%	100%	100%
Jos	79%	55%	73%	88%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	***Jos***
B1	100%	69%	57%	71%	77%
B2	96%	100%	60%	68%	76%
B3	80%	60%	100%	96%	100%
Cov	83%	57%	80%	100%	97%
Jos	82%	58%	76%	88%	100%

Table F.14: Overlap between query modification techniques that gave an increase in retrieval effectiveness using the *wpq* weighting scheme on the WSJ collection

			AP		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	38%	25%	25%	25%
B2	86%	100%	57%	57%	57%
B3	29%	29%	100%	86%	64%
Cov	19%	19%	57%	100%	43%
Jos	29%	29%	64%	64%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	43%	36%	36%	36%
B2	86%	100%	71%	71%	71%
B3	38%	38%	100%	85%	69%
Cov	26%	26%	58%	100%	47%
Jos	33%	33%	60%	60%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	43%	36%	36%	36%
B2	75%	100%	63%	63%	63%
B3	38%	38%	100%	92%	69%
Cov	19%	19%	46%	100%	35%
Jos	31%	31%	56%	56%	100%

Table F.15: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the Porter weighting scheme on the AP collection

			SJM		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	18%	0%	0%	0%
B2	67%	100%	0%	0%	0%
B3	0%	0%	100%	69%	50%
Cov	0%	0%	55%	100%	65%
Jos	0%	0%	44%	72%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	33%	0%	0%	0%
B2	100%	100%	0%	0%	0%
B3	0%	0%	100%	71%	6%
Cov	0%	0%	57%	100%	5%
Jos	0%	0%	11%	11%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	40%	20%	20%	20%
B2	100%	100%	50%	50%	50%
B3	7%	7%	100%	86%	14%
Cov	5%	5%	60%	100%	10%
Jos	6%	6%	12%	12%	100%

Table F.16: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the Porter weighting scheme on the SJM collection

			WSJ		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	54%	38%	38%	38%
B2	78%	100%	56%	56%	56%
B3	56%	56%	100%	78%	78%
Cov	36%	36%	50%	100%	50%
Jos	38%	38%	54%	54%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	64%	43%	43%	43%
B2	82%	100%	55%	55%	55%
B3	67%	67%	100%	78%	78%
Cov	32%	32%	37%	100%	42%
Jos	50%	50%	58%	67%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	100%	80%	80%	80%
B2	71%	100%	57%	57%	57%
B3	57%	57%	100%	64%	64%
Cov	40%	40%	45%	100%	50%
Jos	50%	50%	56%	63%	100%

Table F.17: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the Porter weighting scheme on the WSJ collection

			AP		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	30%	20%	20%	20%
B2	67%	100%	44%	44%	44%
B3	31%	31%	100%	92%	69%
Cov	24%	24%	71%	100%	53%
Jos	29%	29%	64%	64%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	40%	33%	33%	33%
B2	75%	100%	63%	63%	63%
B3	38%	38%	100%	85%	69%
Cov	29%	29%	65%	100%	53%
Jos	29%	29%	53%	53%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	35%	29%	29%	29%
B2	86%	100%	71%	71%	71%
B3	38%	38%	100%	92%	69%
Cov	21%	21%	50%	100%	38%
Jos	29%	29%	53%	53%	100%

Table F.18: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the F_4 weighting scheme on the AP collection

			SJM		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	22%	0%	0%	0%
B2	33%	100%	0%	0%	0%
B3	0%	0%	100%	75%	25%
Cov	0%	0%	56%	100%	13%
Jos	0%	0%	20%	13%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	22%	0%	0%	0%
B2	33%	100%	0%	0%	0%
B3	0%	0%	100%	75%	25%
Cov	0%	0%	56%	100%	13%
Jos	0%	0%	20%	13%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	22%	11%	11%	11%
B2	33%	100%	17%	17%	17%
B3	7%	7%	100%	93%	33%
Cov	5%	5%	74%	100%	26%
Jos	6%	6%	31%	31%	100%

Table F.19: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the F_4 weighting scheme on the SJM collection

			WSJ		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	41%	29%	29%	29%
B2	70%	100%	50%	50%	50%
B3	56%	56%	100%	78%	78%
Cov	38%	38%	54%	100%	62%
Jos	45%	45%	64%	73%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	53%	35%	35%	35%
B2	75%	100%	50%	50%	50%
B3	67%	67%	100%	78%	78%
Cov	40%	40%	47%	100%	53%
Jos	43%	43%	50%	57%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	57%	38%	38%	38%
B2	86%	100%	57%	57%	57%
B3	57%	57%	100%	93%	71%
Cov	38%	38%	62%	100%	52%
Jos	53%	53%	67%	73%	100%

Table F.20: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the F_4 weighting scheme on the WSJ collection

			AP		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	48%	19%	19%	19%
B2	53%	100%	21%	21%	21%
B3	36%	36%	100%	91%	91%
Cov	31%	31%	77%	100%	92%
Jos	27%	27%	67%	80%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	44%	19%	19%	19%
B2	80%	100%	33%	33%	33%
B3	50%	50%	100%	90%	90%
Cov	45%	45%	82%	100%	91%
Jos	38%	38%	69%	77%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	43%	18%	18%	18%
B2	80%	100%	33%	33%	33%
B3	45%	45%	100%	100%	100%
Cov	33%	33%	73%	100%	80%
Jos	33%	33%	73%	80%	100%

Table F.21: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the *wpq* weighting scheme on the AP collection

			SJM		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	35%	0%	0%	0%
B2	60%	100%	0%	0%	0%
B3	0%	0%	100%	43%	29%
Cov	0%	0%	50%	100%	67%
Jos	0%	0%	29%	57%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	54%	0%	0%	0%
B2	74%	100%	0%	0%	0%
B3	0%	0%	100%	71%	71%
Cov	0%	0%	56%	100%	89%
Jos	0%	0%	42%	67%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	52%	0%	0%	0%
B2	67%	100%	0%	0%	0%
B3	0%	0%	100%	100%	100%
Cov	0%	0%	71%	100%	86%
Jos	0%	0%	50%	60%	100%

Table F.22: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the *wpq* weighting scheme on the SJM collection

			WSJ		
***n* = 25**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	30%	22%	22%	22%
B2	78%	100%	56%	56%	56%
B3	56%	56%	100%	78%	78%
Cov	45%	45%	64%	100%	82%
Jos	50%	50%	70%	90%	100%
***n* = 50**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	29%	21%	21%	21%
B2	80%	100%	60%	60%	60%
B3	60%	60%	100%	70%	70%
Cov	55%	55%	64%	100%	91%
Jos	60%	60%	70%	100%	100%
***n* = 100**					
	B1	**B2**	**B3**	**Cov**	**Jos**
B1	100%	34%	28%	28%	28%
B2	77%	100%	62%	62%	62%
B3	62%	62%	100%	92%	85%
Cov	53%	53%	80%	100%	80%
Jos	57%	57%	79%	86%	100%

Table F.23: Overlap between query modification techniques that gave the highest increase in retrieval effectiveness using the *wpq* weighting scheme on the WSJ collection

Porter 25	**AP all *R***	**AP new *R***	**SJM all *R***	**SJM new *R***	**WSJ all *R***	**WSJ new *R***
Coverage Expansion	**5.15%**	**24.87%**	**7.25%**	**16.56%**	**2.70%**	**2.77%**
Coverage Expansion *wpq*	-4.57%	**4.22%**	-1.04%	-0.83%	-1.07%	-1.70%
Coverage Expansion Selection	**2.93%**	**16.58%**	**12.35%**	**18.15%**	**3.96%**	-13.14%
Expansion	**6.08%**	**16.44%**	**2.83%**	**23.53%**	**0.82%**	**2.39%**
Expansion *wpq*	-1.36%	**6.43%**	-1.79%	**1.79%**	-1.89%	-6.35%
Expansion Selection	**0.21%**	**2.36%**	**23.33%**	**19.25%**	-0.38%	-4.09%
Josephson Expansion	**6.22%**	**20.94%**	**11.87%**	**28.57%**	-6.47%	**2.70%**
Josephson Expansion *wpq*	-1.14%	**11.51%**	**3.52%**	**15.94%**	-3.46%	-0.63%
Josephson Expansion Selection	**3.07%**	**9.44%**	**16.08%**	**11.53%**	**0.75%**	-0.57%
Variable Expansion	-1.64%	**0.71%**	-0.97%	-0.90%	-3.65%	-3.90%
Variable Expansion *wpq*	-8.79%	-4.72%	-8.28%	-13.18%	-6.47%	-6.66%
Variable Expansion Selection	**0.93%**	**1.43%**	**5.59%**	**19.25%**	-0.19%	-1.01%

Table F.24: Change in retrieval effectiveness when using only the current set of relevant documents (new *R*) against all relevant documents (all *R*) using the Porter weighting scheme **bold** entries represent increase in retrieval effectiveness over no feedback, underlined entries represent increase of new *R* over all *R*

F4 25	AP all *R*	AP new *R*	SJM all *R*	SJM new *R*	WSJ all *R*	WSJ new *R*
Coverage Expansion	**7.65%**	**24.87%**	**7.25%**	**16.56%**	**2.70%**	**2.77%**
Coverage Expansion *wpq*	-1.07%	**4.22%**	**2.42%**	-0.83%	**0.57%**	-1.70%
Coverage Expansion Selection	**2.93%**	**16.58%**	**12.35%**	**18.15%**	**3.96%**	-13.14%
Expansion	**11.94%**	**16.44%**	**9.59%**	**23.53%**	**2.89%**	**2.39%**
Expansion *wpq*	**9.94%**	**6.43%**	**9.32%**	**1.79%**	**4.71%**	-6.35%
Expansion Selection	**3.57%**	**2.36%**	**1.31%**	**19.25%**	**1.32%**	-4.09%
Josephson Expansion	**6.79%**	**20.94%**	**10.63%**	**28.57%**	**2.70%**	**2.70%**
Josephson Expansion *wpq*	-0.93%	**11.51%**	**4.97%**	**15.94%**	**0.94%**	-0.63%
Josephson Expansion Selection	**5.08%**	**9.44%**	**16.36%**	**11.53%**	**3.21%**	-0.57%
Variable Expansion	**0.00%**	**0.71%**	**3.59%**	-0.90%	-2.39%	-3.90%
Variable Expansion *wpq*	-3.07%	-4.72%	**3.24%**	-13.18%	-2.70%	-6.66%
Variable Expansion Selection	**1.64%**	**1.43%**	**11.25%**	**19.25%**	**1.51%**	-1.01%

Table F.25: Change in retrieval effectiveness when using only the current set of relevant documents (new *R*) against all relevant documents (all *R*) using the F_4 weighting scheme bold entries represent increase in retrieval effectiveness over no feedback, underlined entries represent increase of new *R* over all *R*

wpq 25	AP all *R*	AP new *R*	SJM all *R*	SJM new *R*	WSJ all *R*	WSJ new *R*
Coverage Expansion	**14.22%**	**14.22%**	**7.25%**	**10.35%**	**2.70%**	**2.07%**
Coverage Expansion *wpq*	-16.44%	-16.44%	-2.28%	-29.95%	-1.07%	-99.56%
Coverage Expansion Selection	**16.15%**	**13.01%**	**12.35%**	**12.63%**	**3.96%**	-14.52%
Expansion	**36.53%**	**36.53%**	**2.83%**	**37.47%**	**0.82%**	**5.66%**
Expansion *wpq*	**2.43%**	**2.43%**	**42.37%**	-5.04%	**12.70%**	-1.95%
Expansion Selection	**8.58%**	**8.58%**	**23.33%**	**9.11%**	**1.32%**	-3.46%
Josephson Expansion	**16.38%**	**18.73%**	**12.28%**	**14.22%**	**2.14%**	**1.26%**
Josephson Expansion *wpq*	**14.62%**	-15.51%	-0.28%	-28.50%	-3.46%	-99.56%
Josephson Expansion Selection	**18.58%**	**18.58%**	**16.08%**	**20.98%**	**6.29%**	**2.51%**
Variable Expansion	**37.17%**	**26.30%**	**38.92%**	**37.06%**	-2.39%	**3.52%**
Variable Expansion *wpq*	**16.87%**	**2.43%**	**36.37%**	**21.67%**	-2.70%	-1.95%
Variable Expansion Selection	**17.66%**	**17.66%**	**22.57%**	**28.57%**	**1.51%**	**2.58%**

Table F.26: Change in retrieval effectiveness when using only the current set of relevant documents (new *R*) against all relevant documents (all *R*) using the *wpq* weighting scheme **bold** entries represent increase in retrieval effectiveness over no feedback, underlined entries represent increase of new *R* over all *R*

AP Porter_25	Total rels	no of queries	Average	SJM Porter_25	Total rels	no of queries	Average
Baseline1	249	16	15.56	**Baseline1**	491	11	44.64
Baseline2	34	7	4.86	**Baseline2**	113	3	37.67
Baseline3	451	14	32.21	**Baseline3**	838	16	52.38
Coverage	491	21	23.38	**Coverage**	1102	20	55.10
Josephson	408	14	29.14	**Josephson**	1048	18	58.22
AP Porter_50	**Total rels**	**no of queries**	**Average**	**SJM Porter_50**	**Total rels**	**no of queries**	**Average**
Baseline1	275	14	19.64	**Baseline1**	220	6	36.67
Baseline2	25	7	3.57	**Baseline2**	102	2	51.00
Baseline3	135	13	10.38	**Baseline3**	781	17	45.94
Coverage	608	19	32.00	**Coverage**	920	21	43.81
Josephson	430	15	28.67	**Josephson**	808	9	89.78
AP Porter_100	**Total rels**	**no of queries**	**Average**	**SJM Porter_100**	**Total rels**	**no of queries**	**Average**
Baseline1	132	14	9.43	**Baseline1**	122	5	24.40
Baseline2	82	8	10.25	**Baseline2**	102	2	51.00
Baseline3	172	13	13.23	**Baseline3**	489	14	34.93
Coverage	888	26	34.15	**Coverage**	833	20	41.65
Josephson	420	16	26.25	**Josephson**	1038	17	61.06

WSJ Porter_25	Total rels	no of queries	Average
Baseline1	120	13	9.23
Baseline2	46	9	5.11
Baseline3	24	9	2.67
Coverage	235	14	16.79
Josephson	161	13	12.38
WSJ Porter_50	**Total rels**	**no of queries**	**Average**
Baseline1	128	14	9.14
Baseline2	71	11	6.45
Baseline3	30	9	3.33
Coverage	399	19	21.00
Josephson	185	12	15.42
WSJ Porter_100	**Total rels**	**no of queries**	**Average**
Baseline1	28	10	2.80
Baseline2	89	14	6.36
Baseline3	76	14	5.43
Coverage	374	20	18.70
Josephson	317	16	19.81

Table F.27: Average number of relevant documents for queries whose average precision was improved by the greatest amount by query modification techniques when using the Porter weighting scheme

AP F_4_25	Total rels	no of queries	Average	SJM F_4_25	Total rels	no of queries	Average
Baseline1	362	20	18.10	**Baseline1**	605	13	46.54
Baseline2	46	9	5.11	**Baseline2**	433	5	86.60
Baseline3	172	13	13.23	**Baseline3**	430	10	43.00
Coverage	390	17	22.94	**Coverage**	625	13	48.08
Josephson	670	14	47.86	**Josephson**	924	15	61.60
AP F_4_50	**Total rels**	**no of queries**	**Average**	**SJM F_4_50**	**Total rels**	**no of queries**	**Average**
Baseline1	199	15	13.27	**Baseline1**	366	9	40.67
Baseline2	31	8	3.88	**Baseline2**	561	6	93.50
Baseline3	135	13	10.38	**Baseline3**	516	12	43.00
Coverage	535	17	31.47	**Coverage**	702	16	43.88
Josephson	596	17	35.06	**Josephson**	736	15	49.07
AP F_4_100	**Total rels**	**no of queries**	**Average**	**SJM F_4_100**	**Total rels**	**no of queries**	**Average**
Baseline1	247	17	14.53	**Baseline1**	289	9	32.11
Baseline2	25	7	3.57	**Baseline2**	584	6	97.33
Baseline3	172	13	13.23	**Baseline3**	508	14	36.29
Coverage	722	24	30.08	**Coverage**	951	18	52.83
Josephson	617	17	36.29	**Josephson**	791	16	49.44

WSJ F_4_25	**Total rels**	**no of queries**	**Average**
Baseline1	215	17	12.65
Baseline2	54	10	5.40
Baseline3	24	9	2.67
Coverage	215	13	16.54
Josephson	112	11	10.18
WSJ F_4_50	**Total rels**	**no of queries**	**Average**
Baseline1	250	17	14.71
Baseline2	79	12	6.58
Baseline3	30	9	3.33
Coverage	185	15	12.33
Josephson	303	14	21.64
WSJ F_4_100	**Total rels**	**no of queries**	**Average**
Baseline1	235	21	11.19
Baseline2	99	14	7.07
Baseline3	147	14	10.50
Coverage	374	21	17.81
Josephson	193	15	12.87

Table F.28: Average number of relevant documents for queries whose average precision was improved by the greatest amount by query modification techniques when using the F_4 weighting scheme

AP *wpq*_25	**Total rels**	**no of queries**	**Average**	**SJM *wpq*_25**	**Total rels**	**no of queries**	**Average**
Baseline1	466	21	22.19	**Baseline1**	110	2	55.00
Baseline2	839	19	44.16	**Baseline2**	1622	26	62.38
Baseline3	73	11	6.64	**Baseline3**	1338	15	89.20
Coverage	153	13	11.77	**Coverage**	147	7	21.00
Josephson	399	15	26.60	**Josephson**	194	6	32.33
AP *wpq*_50	**Total rels**	**no of queries**	**Average**	**SJM *wpq*_50**	**Total rels**	**no of queries**	**Average**
Baseline1	941	27	34.85	**Baseline1**	1853	26	71.27
Baseline2	574	15	38.27	**Baseline2**	1544	19	81.26
Baseline3	56	10	5.60	**Baseline3**	114	7	16.29
Coverage	129	11	11.73	**Coverage**	151	9	16.78
Josephson	375	13	28.85	**Josephson**	346	12	28.83
AP *wpq*_100	**Total rels**	**no of queries**	**Average**	**SJM *wpq*_100**	**Total rels**	**no of queries**	**Average**
Baseline1	703	28	25.11	**Baseline1**	1692	27	62.67
Baseline2	380	15	25.33	**Baseline2**	1492	21	71.05
Baseline3	76	11	6.91	**Baseline3**	72	5	14.40
Coverage	210	15	14.00	**Coverage**	141	7	20.14
Josephson	686	15	45.73	**Josephson**	519	10	51.90

WSJ *wpq*_25	Total rels	no of queries	Average
Baseline1	398	23	17.30
Baseline2	40	9	4.44
Baseline3	24	9	2.67
Coverage	130	11	11.82
Josephson	109	10	10.90
WSJ *wpq*_50	**Total rels**	**no of queries**	**Average**
Baseline1	554	28	19.79
Baseline2	48	10	4.80
Baseline3	112	10	11.20
Coverage	114	11	10.36
Josephson	80	10	8.00
WSJ *wpq*_100	**Total rels**	**no of queries**	**Average**
Baseline1	533	29	18.38
Baseline2	94	13	7.23
Baseline3	130	13	10.00
Coverage	167	15	11.13
Josephson	89	14	6.36

Table F.29: Average number of relevant documents for queries whose average precision was improved by the greatest amount by query modification techniques when using the *wpq* weighting scheme

AP Porter_25	Rels found	Initial precision	SJM Porter_25	Rels found	Initial precision
Baseline1	1.63	10.47%	**Baseline1**	4.55	10.19%
Baseline2	2	41.18%	**Baseline2**	1.67	4.43%
Baseline3	7.14	22.16%	**Baseline3**	4.63	8.84%
Coverage	4.19	17.92%	**Coverage**	6.4	11.62%
Josephson	2.79	9.57%	**Josephson**	6.78	11.65%
AP Porter_50	**Rels found**	**Initial precision**	**SJM Porter_50**	**Rels found**	**Initial precision**
Baseline1	2.86	14.56%	**Baseline1**	2.83	7.72%
Baseline2	3.29	92.12%	**Baseline2**	0	0.00%
Baseline3	3.69	35.53%	**Baseline3**	6.82	14.85%
Coverage	5.74	17.94%	**Coverage**	7.1	16.21%
Josephson	3.93	13.71%	**Josephson**	11.89	13.24%
AP Porter_100	**Rels found**	**Initial precision**	**SJM Porter_100**	**Rels found**	**Initial precision**
Baseline1	2.5	26.52%	**Baseline1**	3.2	13.11%
Baseline2	5.13	50.05%	**Baseline2**	0.5	0.98%
Baseline3	5.08	38.40%	**Baseline3**	7.86	22.50%
Coverage	7.08	20.73%	**Coverage**	8.25	19.81%
Josephson	6.56	24.99%	**Josephson**	18	29.48%

WSJ Porter_25	Rels found	Initial precision
Baseline1	1.08	11.70%
Baseline2	1.44	28.17%
Baseline3	0.56	21.00%
Coverage	2	11.91%
Josephson	2.85	23.01%
WSJ Porter_50	**Total rels**	**no of queries**
Baseline1	2.14	23.41%
Baseline2	2.73	42.30%
Baseline3	2.73	81.90%
Coverage	4.11	19.57%
Josephson	3.08	19.98%
WSJ Porter_100	**Total rels**	**no of queries**
Baseline1	1.5	53.57%
Baseline2	4.14	65.12%
Baseline3	2.64	48.63%
Coverage	3.35	17.91%
Josephson	6.13	30.94%

Table F.30: Average initial precision for queries whose average precision was improved by the greatest amount by query modification techniques when using the Porter weighting scheme

AP F_4_25	Rels found	Initial precision	SJM F_4_25	Rels found	Initial precision
Baseline1	2.65	14.64%	**Baseline1**	3.85	8.27%
Baseline2	2.44	47.74%	**Baseline2**	4.4	5.08%
Baseline3	2.85	21.54%	**Baseline3**	4.5	10.47%
Coverage	3.82	16.65%	**Coverage**	5.62	11.69%
Josephson	3.14	6.56%	**Josephson**	7.27	11.80%
AP F_4_50	**Rels found**	**Initial precision**	**SJM F_4_50**	**Rels found**	**Initial precision**
Baseline1	3	22.61%	**Baseline1**	5.33	13.11%
Baseline2	3.38	87.23%	**Baseline2**	12	12.83%
Baseline3	3.69	35.53%	**Baseline3**	6.5	15.12%
Coverage	4.76	15.13%	**Coverage**	8.67	19.76%
Josephson	5.65	16.12%	**Josephson**	6	12.23%
AP F_4_100	**Rels found**	**Initial precision**	**SJM F_4_100**	**Rels found**	**Initial precision**
Baseline1	7.25	49.90%	**Baseline1**	6.22	19.37%
Baseline2	3.29	92.12%	**Baseline2**	22.33	22.94%
Baseline3	5.08	38.40%	**Baseline3**	8.29	22.85%
Coverage	5.96	19.81%	**Coverage**	10.28	19.46%
Josephson	7	19.29%	**Josephson**	13.5	27.31%

WSJ F_4_25	Rels found	Initial precision
Baseline1	1.76	13.92%
Baseline2	1.8	33.33%
Baseline3	0.56	21.00%
Coverage	2.08	12.58%
Josephson	2.09	20.53%
WSJ F_4_50	**Rels found**	**Initial precision**
Baseline1	2.29	15.57%
Baseline2	2.92	44.35%
Baseline3	1.56	46.80%
Coverage	3.33	27.00%
Josephson	4.64	21.44%
WSJ F_4_100	**Rels found**	**Initial precision**
Baseline1	5	44.68%
Baseline2	5.07	71.70%
Baseline3	2.71	25.81%
Coverage	3.95	22.18%
Josephson	3.2	24.87%

Table F.31: Average initial precision for queries whose average precision was improved by the greatest amount by query modification techniques when using the F_4 weighting scheme

AP *wpq*_25	Rels found	Initial precision	SJM *wpq*_25	Rels found	Initial precision
Baseline1	4.05	18.25%	**Baseline1**	5.92	10.76%
Baseline2	5.37	12.16%	**Baseline2**	8.6	13.79%
Baseline3	1.64	24.71%	**Baseline3**	1.71	1.92%
Coverage	3.08	26.17%	**Coverage**	5.17	24.62%
Josephson	3.4	12.78%	**Josephson**	4.43	13.70%
AP *wpq*_50	**Rels found**	**Initial precision**	**SJM *wpq*_50**	**Rels found**	**Initial precision**
Baseline1	5.85	16.79%	**Baseline1**	9.65	13.54%
Baseline2	7.53	19.68%	**Baseline2**	12.68	15.60%
Baseline3	2.5	44.64%	**Baseline3**	5.14	31.56%
Coverage	4.18	35.64%	**Coverage**	6	35.76%
Josephson	4.77	16.54%	**Josephson**	5.67	19.66%
AP *wpq*_100	**Rels found**	**Initial precision**	**SJM *wpq*_100**	**Rels found**	**Initial precision**
Baseline1	6.14	24.46%	**Baseline1**	14.48	23.11%
Baseline2	9.13	36.04%	**Baseline2**	18.95	26.67%
Baseline3	3	43.42%	**Baseline3**	6	41.67%
Coverage	5.6	40.00%	**Coverage**	7.71	38.28%
Josephson	8.07	17.65%	**Josephson**	9.5	18.30%

WSJ *wpq*_25	Rels found	Initial precision
Baseline1	2.48	14.33%
Baseline2	1.44	32.40%
Baseline3	0.56	21.00%
Coverage	2.55	21.58%
Josephson	3.7	33.94%
WSJ *wpq*_50	**Rels found**	**Initial precision**
Baseline1	4.04	20.42%
Baseline2	2.1	43.75%
Baseline3	1.5	13.39%
Coverage	4	38.60%
Josephson	2.4	30.00%
WSJ *wpq*_100	**Rels found**	**Initial precision**
Baseline1	4.55	24.76%
Baseline2	4.62	63.89%
Baseline3	2.15	21.50%
Coverage	3.4	30.54%
Josephson	2.57	40.43%

Table F.32: Average initial precision for queries whose average precision was improved by the greatest amount by query modification techniques when using the *wpq* weighting scheme

AP Porter_25	Total order score	no of queries	Average	SJM Porter_25	Total order score	no of queries	Average
Baseline1	112	16	7.00	**Baseline1**	71	11	6.45
Baseline2	32	7	4.57	**Baseline2**	9	3	3.00
Baseline3	79	7	11.29	**Baseline3**	147	16	9.19
Coverage	145	21	6.90	**Coverage**	190	20	9.50
Josephson	69	14	4.93	**Josephson**	162	18	9.00
AP Porter_50	**Total order score**	**no of queries**	**Average**	**SJM Porter_50**	**Total order score**	**no of queries**	**Average**
Baseline1	219	14	15.64	**Baseline1**	42	6	7.00
Baseline2	68	7	9.71	**Baseline2**	0	2	0.00
Baseline3	146	13	11.23	**Baseline3**	280	17	16.47
Coverage	254	19	13.37	**Coverage**	344	21	16.38
Josephson	156	15	10.40	**Josephson**	180	9	20.00
AP Porter_100	**Total order score**	**no of queries**	**Average**	**SJM Porter_100**	**Total order score**	**no of queries**	**Average**
Baseline1	198	14	14.14	**Baseline1**	173	5	34.60
Baseline2	103	8	12.88	**Baseline2**	80	2	40.00
Baseline3	167	13	12.85	**Baseline3**	482	14	34.43
Coverage	547	26	21.04	**Coverage**	682	20	34.10
Josephson	287	16	17.94	**Josephson**	662	17	38.94

WSJ Porter_25	Total order score	no of queries	Average
Baseline1	50	13	3.85
Baseline2	25	9	2.78
Baseline3	23	9	2.56
Coverage	71	14	5.07
Josephson	42	13	3.23
WSJ Porter_50	**Total order score**	**no of queries**	**Average**
Baseline1	97	14	6.93
Baseline2	69	11	6.27
Baseline3	74	11	6.73
Coverage	169	19	8.89
Josephson	140	12	11.67
WSJ Porter_100	**Total order score**	**no of queries**	**Average**
Baseline1	180	10	18.00
Baseline2	258	14	18.43
Baseline3	295	14	21.07
Coverage	487	20	24.35
Josephson	284	16	17.75

Table F.33: Average retrieval score (order) for queries whose average precision was improved by the greatest amount by query modification techniques when using the Porter weighting scheme

AP F_4_25	Total order score	no of queries	Average	SJM F_4_25	Total order score	no of queries	Average
Baseline1	138	20	6.90	**Baseline1**	91	13	7.00
Baseline2	66	9	7.33	**Baseline2**	22	5	4.40
Baseline3	70	13	5.38	**Baseline3**	103	10	10.30
Coverage	111	17	6.53	**Coverage**	115	13	8.85
Josephson	71	14	5.07	**Josephson**	121	15	8.07
AP F_4_50	**Total order score**	**no of queries**	**Average**	**SJM F_4_50**	**Total order score**	**no of queries**	**Average**
Baseline1	210	15	14.00	**Baseline1**	109	9	12.11
Baseline2	72	8	9.00	**Baseline2**	93	6	15.50
Baseline3	146	13	11.23	**Baseline3**	187	12	15.58
Coverage	219	17	12.88	**Coverage**	263	16	16.44
Josephson	216	17	12.71	**Josephson**	234	15	15.60
AP F_4_100	**Total order score**	**no of queries**	**Average**	**SJM F_4_100**	**Total order score**	**no of queries**	**Average**
Baseline1	268	17	15.76	**Baseline1**	320	9	35.56
Baseline2	68	7	9.71	**Baseline2**	214	6	35.67
Baseline3	167	13	12.85	**Baseline3**	480	14	34.29
Coverage	472	24	19.67	**Coverage**	661	18	36.72
Josephson	373	17	21.94	**Josephson**	576	16	36.00

WSJ F_4_25	Total order score	no of queries	Average
Baseline1	89	17	5.24
Baseline2	34	10	3.40
Baseline3	23	9	2.56
Coverage	48	13	3.69
Josephson	33	11	3.00
WSJ F_4_50	**Total order score**	**no of queries**	**Average**
Baseline1	120	17	7.06
Baseline2	78	12	6.50
Baseline3	74	9	8.22
Coverage	129	15	8.60
Josephson	169	14	12.07
WSJ F_4_100	**Total order score**	**no of queries**	**Average**
Baseline1	397	21	18.90
Baseline2	285	14	20.36
Baseline3	379	14	27.07
Coverage	490	21	23.33
Josephson	329	15	21.93

Table F.34: Average retrieval score (order) for queries whose average precision was improved by the greatest amount by query modification techniques when using the F_4 weighting scheme

AP *wpq*_25	Total order score	no of queries	Average	SJM *wpq*_25	Total order score	no of queries	Average
Baseline1	152	21	7.24	**Baseline1**	213	26	8.19
Baseline2	156	19	8.21	**Baseline2**	149	15	9.93
Baseline3	49	11	4.45	**Baseline3**	70	7	10.00
Coverage	68	13	5.23	**Coverage**	59	6	9.83
Josephson	86	15	5.73	**Josephson**	55	7	7.86
AP *wpq*_50	**Total order score**	**no of queries**	**Average**	**SJM *wpq*_50**	**Total order score**	**no of queries**	**Average**
Baseline1	452	27	16.74	**Baseline1**	442	26	17.00
Baseline2	209	15	13.93	**Baseline2**	355	19	18.68
Baseline3	63	10	6.30	**Baseline3**	129	7	18.43
Coverage	88	11	8.00	**Coverage**	117	9	13.00
Josephson	153	13	11.77	**Josephson**	185	12	15.42
AP *wpq*_100	**Total order score**	**no of queries**	**Average**	**SJM *wpq*_100**	**Total order score**	**no of queries**	**Average**
Baseline1	628	28	22.43	**Baseline1**	1008	27	37.33
Baseline2	326	15	21.73	**Baseline2**	791	21	37.67
Baseline3	127	11	11.55	**Baseline3**	118	5	23.60
Coverage	231	15	15.40	**Coverage**	162	7	23.14
Josephson	261	15	17.40	**Josephson**	283	10	28.30

WSJ *wpq*_25	Total order score	no of queries	Average
Baseline1	136	23	5.91
Baseline2	24	9	2.67
Baseline3	23	9	2.56
Coverage	40	11	3.64
Josephson	11	10	1.10
WSJ *wpq*_50	**Total order score**	**no of queries**	**Average**
Baseline1	308	28	11.00
Baseline2	38	10	3.80
Baseline3	119	10	11.90
Coverage	75	11	6.82
Josephson	54	10	5.40
WSJ *wpq*_100	**Total order score**	**no of queries**	**Average**
Baseline1	581	29	20.03
Baseline2	274	13	21.08
Baseline3	346	13	26.62
Coverage	370	15	24.67
Josephson	286	14	20.43

Table F.35: Average retrieval score (order) for queries whose average precision was improved by the greatest amount by query modification techniques when using the *wpq* weighting scheme

AP Porter_25	Total terms	no of queries	Average	SJM Porter_25	Total terms	no of queries	Average
Baseline1	2087.5	16	130.47	**Baseline1**	1328.02	11	120.73
Baseline2	810.5	7	115.79	**Baseline2**	120.6	3	40.20
Baseline3	1305.13	7	186.45	**Baseline3**	2533.09	16	158.32
Coverage	2646.57	21	126.03	**Coverage**	2723.47	20	136.17
Josephson	1195.3	14	85.38	**Josephson**	2320.03	18	128.89
AP Porter_50	**Total terms**	**no of queries**	**Average**	**SJM Porter_50**	**Total terms**	**no of queries**	**Average**
Baseline1	1862.24	14	133.02	**Baseline1**	399.622	6	66.60
Baseline2	961.975	7	137.43	**Baseline2**	0	2	0.00
Baseline3	1492.17	13	114.78	**Baseline3**	2556.98	17	150.41
Coverage	2188.97	19	115.21	**Coverage**	3029.28	21	144.25
Josephson	1714.01	15	114.27	**Josephson**	1114.58	9	123.84
AP Porter_100	**Total terms**	**no of queries**	**Average**	**SJM Porter_100**	**Total terms**	**no of queries**	**Average**
Baseline1	1153.93	14	82.42	**Baseline1**	611.956	5	122.39
Baseline2	1748.92	8	218.62	**Baseline2**	126	2	63.00
Baseline3	1057.36	13	81.34	**Baseline3**	1967.05	14	140.50
Coverage	1263.18	26	48.58	**Coverage**	2674.45	20	133.72
Josephson	2878.19	16	179.89	**Josephson**	1745.07	17	102.65

WSJ Porter_25	Total terms	no of queries	Average
Baseline1	2643.5	13	203.35
Baseline2	1880.75	9	208.97
Baseline3	1527.5	9	169.72
Coverage	3432.37	14	245.17
Josephson	2283.84	13	175.68
WSJ Porter_50	**Total terms**	**no of queries**	**Average**
Baseline1	2672.4	14	190.89
Baseline2	2279.32	11	207.21
Baseline3	2076.62	11	188.78
Coverage	3870.15	19	203.69
Josephson	2594.76	12	216.23
WSJ Porter_100	**Total terms**	**no of queries**	**Average**
Baseline1	1802.12	10	180.21
Baseline2	2799.96	14	200.00
Baseline3	3052.9	14	218.06
Coverage	4398.16	20	219.91
Josephson	2815.4	16	175.96

Table F.36: Average similarity of relevant documents for queries whose average precision was improved by the greatest amount by query modification techniques when using the Porter weighting scheme

where Total terms = number of discriminatory terms in relevant documents

AP F_4_25	Total terms	no of queries	Average	SJM F_4_25	Total terms	no of queries	Average
Baseline1	2741.62	20	137.08	**Baseline1**	1772.85	13	136.37
Baseline2	1066.75	9	118.53	**Baseline2**	454.7	5	90.94
Baseline3	1189.67	13	91.51	**Baseline3**	1574.83	10	157.48
Coverage	1944.07	17	114.36	**Coverage**	1592.5	13	122.50
Josephson	1187.89	14	84.85	**Josephson**	1747	15	116.47
AP F_4_50	**Total terms**	**no of queries**	**Average**	**SJM F_4_50**	**Total terms**	**no of queries**	**Average**
Baseline1	2001.89	15	133.46	**Baseline1**	702.839	9	78.09
Baseline2	1125.72	8	140.72	**Baseline2**	394.667	6	65.78
Baseline3	1492.17	13	114.78	**Baseline3**	1581.51	12	131.79
Coverage	1992.12	17	117.18	**Coverage**	2323.25	16	145.20
Josephson	1894.28	17	111.43	**Josephson**	2469.64	15	164.64
AP F_4_100	**Total terms**	**no of queries**	**Average**	**SJM F_4_100**	**Total terms**	**no of queries**	**Average**
Baseline1	2060.6	17	121.21	**Baseline1**	1107.8	9	123.09
Baseline2	961.975	7	137.43	**Baseline2**	475.363	6	79.23
Baseline3	1263.18	13	97.17	**Baseline3**	2048.42	14	146.32
Coverage	2721.38	24	113.39	**Coverage**	2419	18	134.39
Josephson	1766.79	17	103.93	**Josephson**	1797.52	16	112.35

WSJ F_4_25	Total terms	no of queries	Average
Baseline1	3571.64	17	210.10
Baseline2	2147.15	10	214.72
Baseline3	1527.5	9	169.72
Coverage	3164.37	13	243.41
Josephson	2215.7	11	201.43
WSJ F_4_50	**Total terms**	**no of queries**	**Average**
Baseline1	3171.47	17	186.56
Baseline2	2545.72	12	212.14
Baseline3	2076.62	9	230.74
Coverage	3224.73	15	214.98
Josephson	2970.67	14	212.19
WSJ F_4_100	**Total terms**	**no of queries**	**Average**
Baseline1	4112.71	21	195.84
Baseline2	2480.22	14	177.16
Baseline3	3287.07	14	234.79
Coverage	4710.32	21	224.30
Josephson	3035.85	15	202.39

Table F.37: Average similarity of relevant documents for queries whose average precision was improved by the greatest amount by query modification techniques when using the F_4 weighting scheme

where Total terms = number of discriminatory terms in relevant documents

AP *wpq*_25	**Total order score**	**no of queries**	**Average**	**SJM *wpq*_25**	**Total order score**	**no of queries**	**Average**
Baseline1	2731	21	130.05	**Baseline1**	3437	26	132.19
Baseline2	2588	19	136.21	**Baseline2**	2040	15	136.00
Baseline3	978	11	88.91	**Baseline3**	1049	7	149.86
Coverage	1265	13	97.31	**Coverage**	605	6	100.83
Josephson	1358	15	90.53	**Josephson**	638	7	91.14
AP *wpq*_50	**Total order score**	**no of queries**	**Average**	**SJM *wpq*_50**	**Total order score**	**no of queries**	**Average**
Baseline1	3837	27	142.11	**Baseline1**	3146	26	121.00
Baseline2	1858	15	123.87	**Baseline2**	2151	19	113.21
Baseline3	928	10	92.80	**Baseline3**	1183	7	169.00
Coverage	1020	11	92.73	**Coverage**	1169	9	129.89
Josephson	1486	13	114.31	**Josephson**	1740	12	145.00
AP *wpq*_100	**Total order score**	**no of queries**	**Average**	**SJM *wpq*_100**	**Total order score**	**no of queries**	**Average**
Baseline1	3410	28	121.79	**Baseline1**	3184	27	117.93
Baseline2	1899	15	126.60	**Baseline2**	2135	21	101.67
Baseline3	1058	11	96.18	**Baseline3**	577	5	115.40
Coverage	1538	15	102.53	**Coverage**	813	7	116.14
Josephson	1410	15	94.00	**Josephson**	1352	10	135.20

WSJ *wpq*_25	Total order score	no of queries	Average
Baseline1	5527	23	240.30
Baseline2	1669	9	185.44
Baseline3	1527	9	169.67
Coverage	2071	11	188.27
Josephson	1680	10	168.00
WSJ *wpq*_50	**Total order score**	**no of queries**	**Average**
Baseline1	6276	28	224.14
Baseline2	1896	10	189.60
Baseline3	2473	10	247.30
Coverage	1957	11	177.91
Josephson	1868	10	186.80
WSJ *wpq*_100	**Total order score**	**no of queries**	**Average**
Baseline1	6403	29	220.79
Baseline2	2325	13	178.85
Baseline3	3111	13	239.31
Coverage	3178	15	211.87
Josephson	2881	14	205.79

Table F.38: Average similarity of relevant documents for queries whose average precision was improved by the greatest amount by query modification techniques when using the *wpq* weighting scheme

where Total terms = number of discriminatory terms in relevant documents

Appendix G

Experimental system

G.1 Introduction

This Appendix describes the architecture of the system used in the user experiments. The system is composed of a three-layer architecture, shown in Figure G.1. Several versions of the interface and retrieval system components were devised for the interactive experiments described in Chapter Twelve. The data files were constant for all experiments.

Details of the experiments themselves, such as the data collection and search topics, are not described here but are presented in Chapter Twelve. In this Appendix I also do not discuss the indexing components of the overall system; algorithms to calculate the term and document characteristic values were implemented according to the equations described in Chapter Three.

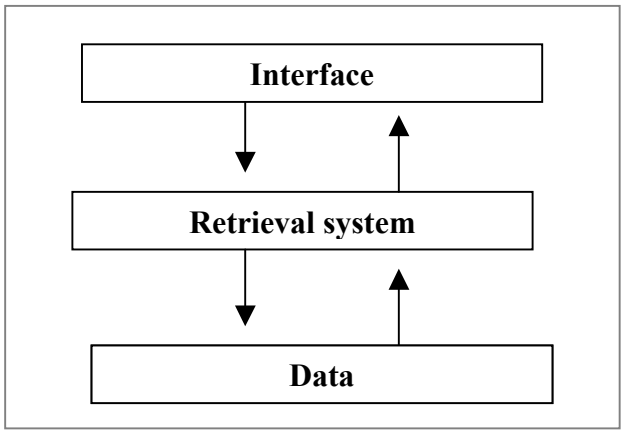

Figure G.1: System architecture

In section G.2, I shall give a brief description of the data files used, in section G.3 I shall describe implementation of the retrieval system, and in section G.4 I shall describe the interfaces used in the experiments. Finally, in section G.5, I shall describe the logging components of the system. The logging features store information on the interactive aspects of the search such as how many queries a user has submitted and which documents a searcher has assessed as relevant.

G.2 Data files

The data files correspond to the lowest level of the system. There are two types of data files: *static* files and *dynamic* files. The static files are created at indexing time and are constant for all queries. The dynamic files are continuously modified throughout searching.

G.2.1 Static data files

The static files are grouped into 3 types: *access* files, *index* files and *feedback* files.

G.2.1.1 Access files

The access files are used to manipulate data that is shown to the user, such as titles and the full-texts of retrieved documents. There are four access files; the first two files are used to extract the full document text of the retrieved documents:

- *data*. This file contains the original document collection in SGML format.

- *offset_docs*. This file is a list of how many bytes should be read from the start of the data file to reach the start of each document, e.g.

  0000000000 bytes should be read to reach the start of document 1
  0000001389 bytes should be read to reach the start of document 2
  0000003072 bytes should be read to reach the start of document 3
  0000004185 bytes should be read to reach the start of document 4

  The *offset_docs* file allows fast access to the full-text of each document.

The second set of files allows fast extraction of the titles of the retrieved documents.

- *titles*. This is a list of the titles of each document in the collection. If a document does not have a title, then the text 'THIS DOCUMENT HAS NO TITLE' is entered in place of the title.

- *offset_titles*. This file is analogous to the *offset_docs* file and operates on the *titles* data file. This file is used to extract the titles of the retrieved documents.

G.2.1.2 Index files

These files contain the weights of the term and document characteristics.

• *info_noise* and *specificity*. These files contain the *specificity* and *information_noise* values for each document.

• *dictionary*. This file contains information on each term in the collection. The format of the file is shown in Table G.1.

term name	***idf***	***noise***	***occurrences***	***offset***
book	33	10	13853	128304358
zebra	50	09	80	783511742

Table G.1: Format of dictionary file

where *term name* is the term as indexed, *idf* and *noise* are the values of the *idf* and *noise* characteristics of each term. *occurrences* is the number of documents in which the term appears and *offset* is the offset into the postings file which stores the *tf* and *theme* values of each term (see below).

The dictionary file is used by the retrieval and RF algorithms to obtain the *idf* and *noise* values for query terms. The dictionary file is also used to access the *postings* file.

• *postings*. This file is lists the documents in which a term appears and the *tf* and *theme* value of the term in each document. The format of the file is a stream of triples of the form shown in Figure G.2

docID *tf* *theme*

Figure G.2: Format of postings file triples

where *docID* is a unique numerical identifier for each document, and *tf* and *theme* are the *tf* and *theme* characteristics of a term in the document *docID*.

The occurrences entry from the dictionary file tells the system how many triples to read (how many documents contain each query term) and the offset value tells the system at what position (in bytes) the triples should be read from. For example, from Table G.1, if the query contains the term *book*, the retrieval system should start reading triples at 128304358 bytes from the start of the postings file, and should read 13853 triples. Thus only documents containing the term *book* receive a retrieval score.

G.2.1.3 Relevance feedback files

These files are used to generate the list of expansion terms in relevance feedback.

• *documents_vectors*. This file contains information on which terms are contained within each document. This is necessary for the RF algorithms to quickly construct a list of possible expansion terms based on a list of relevant documents. The format of the *document_vectors* file is shown in Figure G.3,

docID termID termID termID
docID termID termID termID

Figure G.3: Format of document_vectors file

where *docID* and *termID* are unique numerical identifiers for documents and terms.

• *vectors_offset*. This file contains information that allows quick access to the *document_vectors* file. Each line consists of a triple of the form shown in Table G.2.

docID	***number of terms***	***offset***
12321	22	4636
54543	101	643463

Table G.2: Format of *vectors_offset* file triples

where *docID* is a continuous set of numerical identifiers, *number of terms* is the number of terms in document *docID*, and *offset* is the number of bytes to be read from the start of the *document_vectors* file to reach the correct line for document numbered *docID*.

The access, index and feedback files are constant for all retrieval systems, queries and feedback iterations. The dynamic data files, outlined in the next section, are modified throughout an information-seeking session.

G.2.2 Dynamic data files

There are three groups of files in the dynamic group. These are sub-divided into those files that are controlled by the interface, section G.2.2.1, those controlled by the retrieval system, section G.2.2.2, and those that are jointly controlled, section G.2.2.3. By control, I mean which component of the system has the permission to change the contents of the file.

G.2.2.1 Files controlled by the interface

The only file over which the interface has complete control is the *rels* file.

• *rels*. This file contains a list of the documents that the user has marked as useful[130] in the current search. It is empty at the start of a new search, and is cleared if the user requests a new search rather than an RF iteration (section G.4). Table G.3 shows the format of the *rels* file.

docID	***relevance score***	***iteration***
282848	10	1
34328	7	2
4328739	9	2

Table G.3: *rels* file format

docID is the numerical identifier of a relevant document, *relevance score* is the score the user has given the document (section G.4) and *iteration* indicates in which search iteration (1st, 2nd, 3rd, etc), the document was marked relevant. Only the retrieval systems (see Chapter Twelve) that use ostensive weighting store the *iteration* information. A new search always has an iteration value of 1, corresponding to the first search iteration. An iteration of RF will increase the value of the iteration variable by 1.

G.2.2.2 Files controlled by the retrieval system

All files in this section are generated and *written* to by the retrieval component alone. All files are *read* by the interface to present the results of a retrieval to the user.

• *results*. This file contains a list of the top thirty documents retrieved for each query, each document is represented by its numerical identifier.

• *retrieved_docs*. This file contains the text of the retrieved documents. These documents are formatted by the retrieval system to remove SGML tags for presentation to the user. Subsequent formatting, for example the highlighting of query terms, is handled by the interface.

• *retrieved_titles*. This file contains the titles of the retrieved documents.

• *retrieved_offsets*. This file contains the offsets (in bytes) of the start of each of the retrieved documents in the *retrieved_docs* file. This allows the interface to split the documents contained in *retrieved_docs* into individual documents. An example is shown in Figure G.4. To access the content of the first retrieved document, the system starts reading at 1 byte from

[130] The interfaces ask users to assess documents as *useful* rather than *relevant* to their search, section G.4.

the start of the *retrieved_docs* file and reads until position 4154; to access the content of the second document the system starts reading at byte position 4155 and reads until byte position 7276, and so on.

1 4155 7277 10849 13069 13581 16764 20911 25048 29693 34050 39651
41317 43817 46077 48985 50287 53293 56309 56793 57872 60800 63394
65969 69587 72967 78448 104471 108831 113437 116633

Figure G.4: Example of *document_offsets* file

• *retrieved*. This file contains a list of the documents that have been previously retrieved in the search, i.e. from the point where the user last initiated a new search. In some of the experimental retrieval systems this file is used to eliminate documents from the list presented to the user – only the top thirty previously unretrieved documents are displayed to the user after RF. If the user requests a new search then this file is emptied.

• *explanation*. This file contains an explanation of the current search, section G.4. It is empty after a new search and only contains data after RF.

G.2.2.3 Files that are controlled jointly by the retrieval system and interface

The files in this section can be written to either by the retrieval system or the interface.

• *query*. This file contains the current query. It is created or modified in one of two ways:

i. *by the interface.* The query terms the user enters at the interface (section G.4) are written to this file to perform a new retrieval.

ii. *by the retrieval system.* If the user requests RF, the retrieval system will perform an iteration of RF and create a new query that will be written to the *query* file.

• *log.* This file contains a log of the user's current search. The log file is created when the interface is started and is continuously written to by the interface and retrieval system. Section G.5 explains the format of the log file.

G.3 Retrieval system

The retrieval system is written in AINSI C. I shall not give a detailed account of the algorithms for retrieval and feedback contained in the retrieval system as they correspond to the theoretical work described in Chapters Ten and Twelve.

G.4 Interfaces

The system interfaces were built using the Smalltalk VisualWorks environment[131], running on Unix. The interfaces control the interaction with the user. All interfaces described in this chapter have four main functions:

i. *connection to retrieval system*. The interface connects to the underlying retrieval system and starts the retrieval programs. It also reads in the results of the completed search.

ii. *logging*. The interface logs those documents a user has assessed as relevant. It also logs certain aspects of the user interaction such as which documents a user has viewed.

iii. *provides a interactive search environment*. The main function of the interface is to facilitate interactive searching.

Four interfaces were developed for the experiments discussed in this thesis. In sections G.4.1 – G.4.4 I describe each of the interfaces. The interfaces are labelled Interface One, Interface Two, Interface Three and Interface Four. In Chapter Twelve I discuss the specific variations of the retrieval and relevance feedback algorithms that underlie each interface and the experiments carried out on each interface. In this chapter I simply describe the basic interface and the variations in look and feel between the interfaces.

G.4.1 Interface One

Interface One is the most basic interface, the remaining three interfaces are extensions of this interface. Figure G.5 shows a schematic sketch of Interface One, Figure G.6 shows a screen dump of the interface.

[131] Smalltalk is an object-oriented programming language, VisualWorks is an application that facilitates the construction of Smalltalk interfaces. The interface is written in Smalltalk, and issues retrieval commands to the underlying retrieval system, written in C.

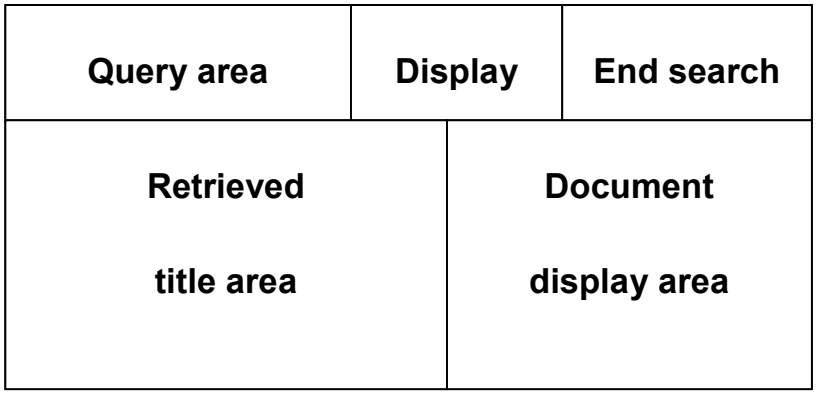

Figure G.5: Interface One – schematic sketch

Interface One has five main areas:

i. *query area.* This area contains a large box into which users can enter query terms. One button is present, the *New search* button, which the user clicks to initiate a search. In Figure G.6, the user has entered the search terms '*lady'*, '*macbeth'*, '*murder'*, '*duncan"*, '*glasms'* and '*shakespeare'*.

ii. *display area.* This area displays messages to the user. These messages are of two types: *status* messages and an *error* message. The error message tells the user that s/he has entered a query term that is not found in the document collection. In Figure G.6, the term '*glasms*' has not been found. Status messages, e.g. *storing new query*, *retrieving new documents*, are displayed when the user initiates a new query. These are intended to reassure the user that the system is functioning.

iii. *end search button.* After clicking the *End search* button the interface initiates a C program on the underlying Unix system. This program appends the user's final relevance assessments to the user log.

iv. *retrieved title area*. This area displays the titles of the retrieved documents. Each search retrieves 30 documents, these are displayed 10 titles at a time. If less than 30 documents contain a query term, then randomly selected documents are chosen to increase the retrieved set to 30. The user can move within the retrieved set by selecting the *Prev 10* or *Next 10* buttons. Check boxes next to the document titles signify that the document has been assessed as useful to the users search. In Figure

G.5, the 1st, 3rd, 5th and 7th listed documents have been assessed useful[132]. The user cannot click the checkboxes directly, they are controlled by the assessment slider (see **v.**)

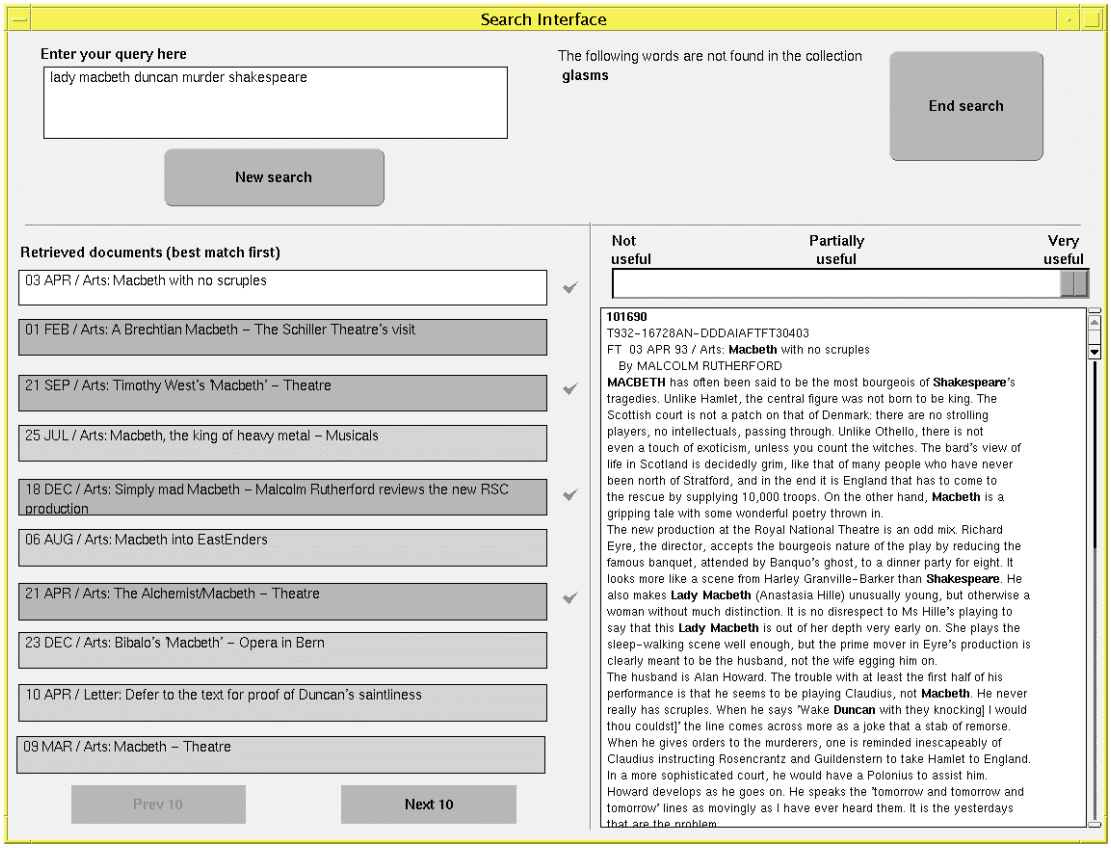

Figure G.6: Interface One

The colours of the titles are used to indicate the status of the corresponding documents. The default background colour of the document titles is light grey. The background colour of the currently selected document is white and the background colour of viewed documents becomes dark grey. For example in Figure G.6, the user has viewed the 3rd to the 7th displayed documents, is currently viewing the 1st displayed document and has not viewed the remaining documents[133]. The system will remember the viewed documents within searches, i.e. if the user issues a new query, the system will remember viewed documents that were retrieved by the previous query.

---

[132] The checkboxes only signify that the document has been assessed as useful, and do not signify the degree of usefulness that the user has assigned.

[133] This use of colour was introduced after pilot testing of the interfaces. The experimental subjects in the pilot test reported confusion as to which documents they were viewing and which they had already viewed.

v. *document display area*. This area shows the full text of the currently selected document and allows the user to assess the usefulness of the displayed document. Each time the user clicks a document title, the full-text of the corresponding document is displayed in the full-text area. The query terms are highlighted in the full-text area to make it easier for the user to locate relevant material within the document. The background of the selected document title is changed to white (rather than the default light gray) to make it obvious which document is being viewed. Figure G.6 shows the interface after the user has clicked on the first document title.

After a new search, or when the user clicks on the *prev 10* or *next 10* buttons, the first document in the list of 10 is highlighted and its full text is displayed.

The assessment slider, Figure G.7, allows the user to give a value to the usefulness of the displayed document. The slider is labelled from '*Not useful'* to '*Very useful'*. The middle of the slider is labelled '*Partially useful'* to indicate that the document contains some useful information. The slider corresponds to an 11-point scale, ranging from 0 (the default value signifying not useful/relevant) to 10 (signifying very useful/relevant). In Figure G.6 the user has assessed the first document as being very useful to his search. In the experiments no specific indication was given to the user of how to interpret *useful*; the searchers were encouraged to decide for themselves what constituted useful information.

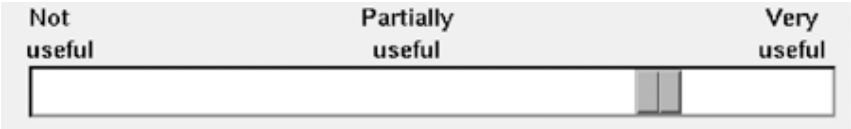

Figure G.7: Assessment slider

When a user moves the relevance slider the value is stored and a tick appears next to the displayed document's title. If the user moves the slider back to 0 ('not useful') the tick disappears.

G.4.2 Interface Two

Interface Two has the same components as Interface One with the addition of a RF button, Figure G.8. This button, *Improve search*, is inactive (switched off) until the user assesses at least one document as containing useful information, Figure G.9. This is so that the user cannot request RF without having supplied any relevance information. Clicking on the button before making relevance assessments will have no affect.

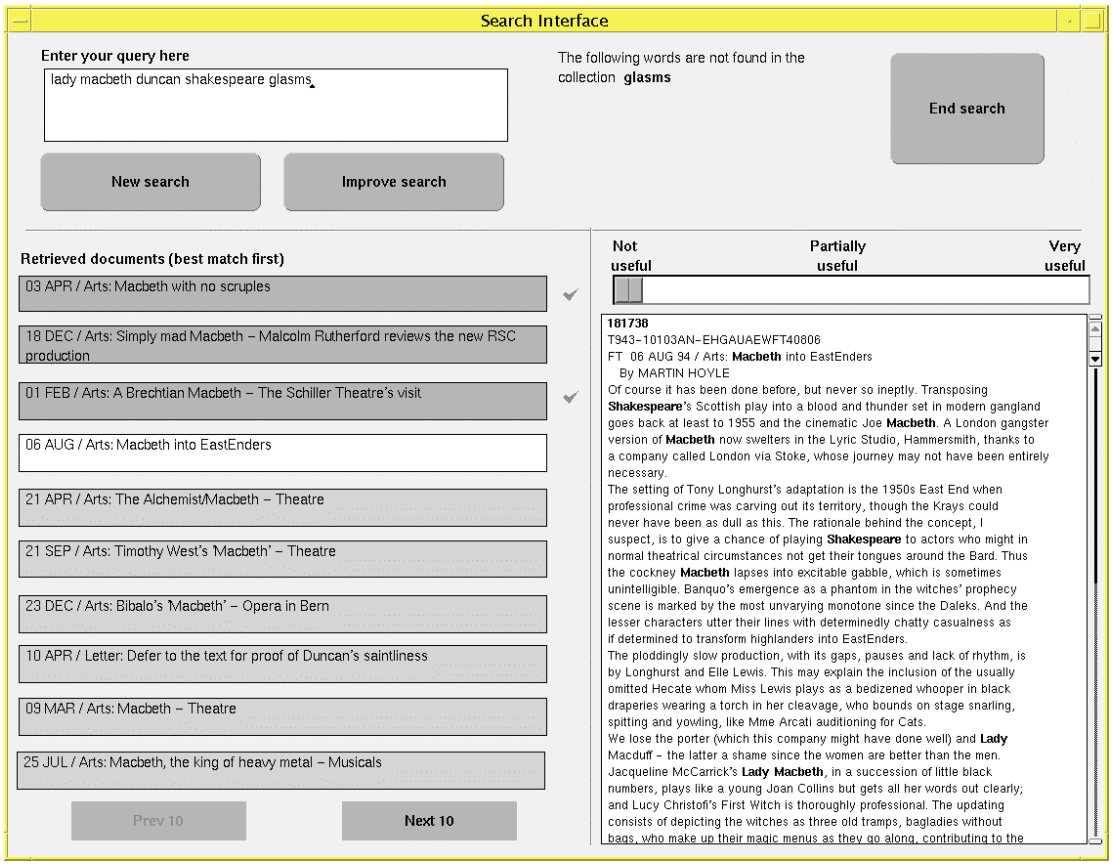

Figure G.8: Interface Two

a. **b.**

Figure G.9: a. Switched-off button **b.** Switched-on button

G.4.3 Interface Three

Interface Three is designed specifically for interactive query expansion, Figure G.10. The display area has been shortened to allow the presentation of suggested expansion terms and the *End search* button has changed shaped. The *Improve search* button is replaced by a '*Suggest terms* button. As with the *Improve search* button, the *Suggest terms* button is inactive until at least one relevance assessment has been made.

After clicking the *Suggest terms* button the system will display the top twenty expansion terms on the top right-hand corner of the interface. Each expansion term is associated with a button: clicking on the term will add the term to the user's query. The updated query is

displayed in the query area (top-left corner). This interface only supports query expansion: if users wish to *remove* a term, they must do this manually. The expansion terms are sorted alphabetically (from top left to bottom right).

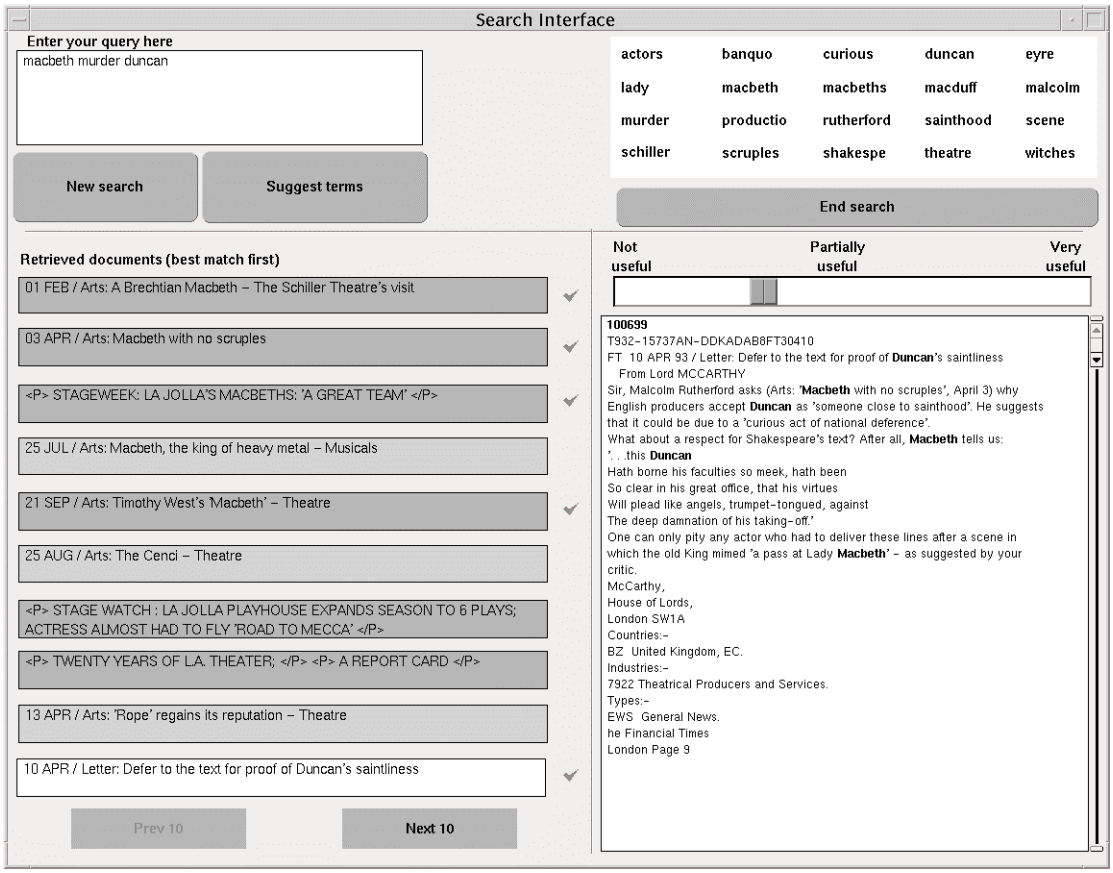

Figure G.10: Interface Three

G.4.4 Interface Four

Interface Four is based on Interface Two with the addition of a summarised explanation of the process of RF. This replaces the display area of Interface One. After the user clicks on the *Improve search* button, the system performs an iteration of RF, and displays a short summary, in the explanation box, of the effect of RF on the user's search.

In Figure G.11 I show the results of an improved search. The explanation presented at the interface corresponds to the type of explanation selected for RF. There are five possible explanations that can be presented to the user, these correspond to the explanation types presented in Chapter Ten. The explanation also contains a direction as to how the modification should be treated, i.e. the system will suggest that the user can add terms that are similar to useful ones added by the system or remove terms that do not appear useful to the search. This is an attempt to persuade the user to interact with the results of RF.

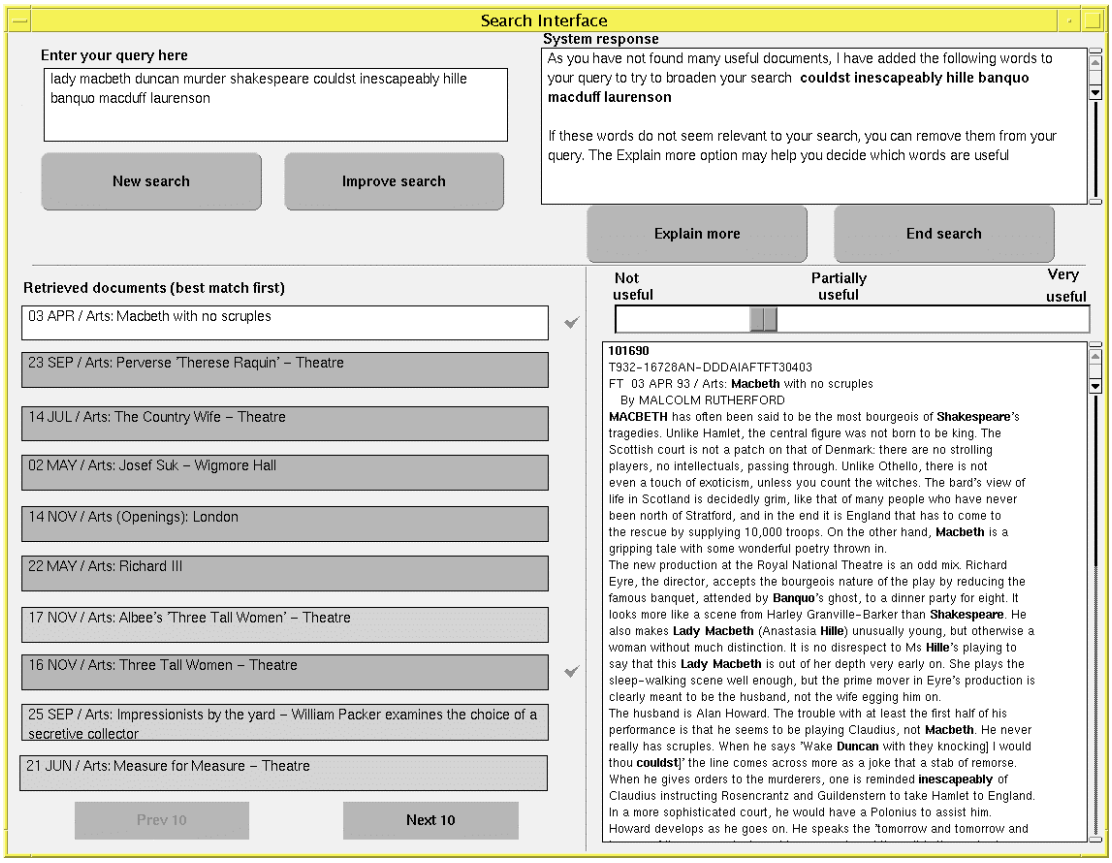

Figure G.11: Interface Four

The five types of possible explanation are:

i. *expansion explanation*. In this case the user has marked few documents relevant and the system attempts to broaden the user's search by adding more search terms. The system lists the terms that it has added and displays the message *'As you not found many useful documents, I have added the following words to try to broaden your search* ***couldst inescapeably hillle banquo macduff laurenson****'*. In this example, ***couldst inescapeably hillle banquo macduff laurenson*** are the top six expansion terms.

ii. *coverage explanation*. In this case the system will present the user with an explanation like this *'I have added the word* ***macduff banquo*** *to your query as they appear in most of the documents you have marked useful*. This type of explanation emphasises the search terms that make the user's documents similar to each other.

iii. *josephson explanation*. In this case the system will present the user with an explanation like this '*I have added the word **macduff banquo** to your query as they appear to be important to your search'*. This type of explanation emphasises search terms that are good discriminators of relevance.

iv. *no expansion explanation.* In this case the system will not add any search terms to the user's query but instead will concentrate on improving the weighting of the search terms – selecting good term and document characteristics. The explanation presented at the interface is '*Based on the documents you have marked useful, I will treat **macbeth** as the most important word in your search and try to retrieve more documents containing this word'*. In this example ***macbeth*** is the term for which there are most characteristics selected.

v. *don't know explanation*. If the system cannot choose one good explanation – all votes are split between different explanation types for example – then the system will tell the user it cannot decide what kind of documents the user requires. It will show the user a message suggesting the user provides more evidence. For example, ''*I am not sure what kind of documents you want – perhaps you could mark some more documents as useful or add some more words to your query. Here are some examples that may be useful **banquo theatre macduff king scene arts**'*. As in the expansion explanation, **i.**, the terms ***banquo theatre macduff king scene arts*** are the top-ranked expansion terms.

The user can request more information on the RF process by clicking the *Explain more* button. This option will expand the information contained within the explanation box with information on how terms are used to select the new set of retrieved documents. In Figure G.12, the user has selected this option and the system gives more information on the role of the query terms in the new search.

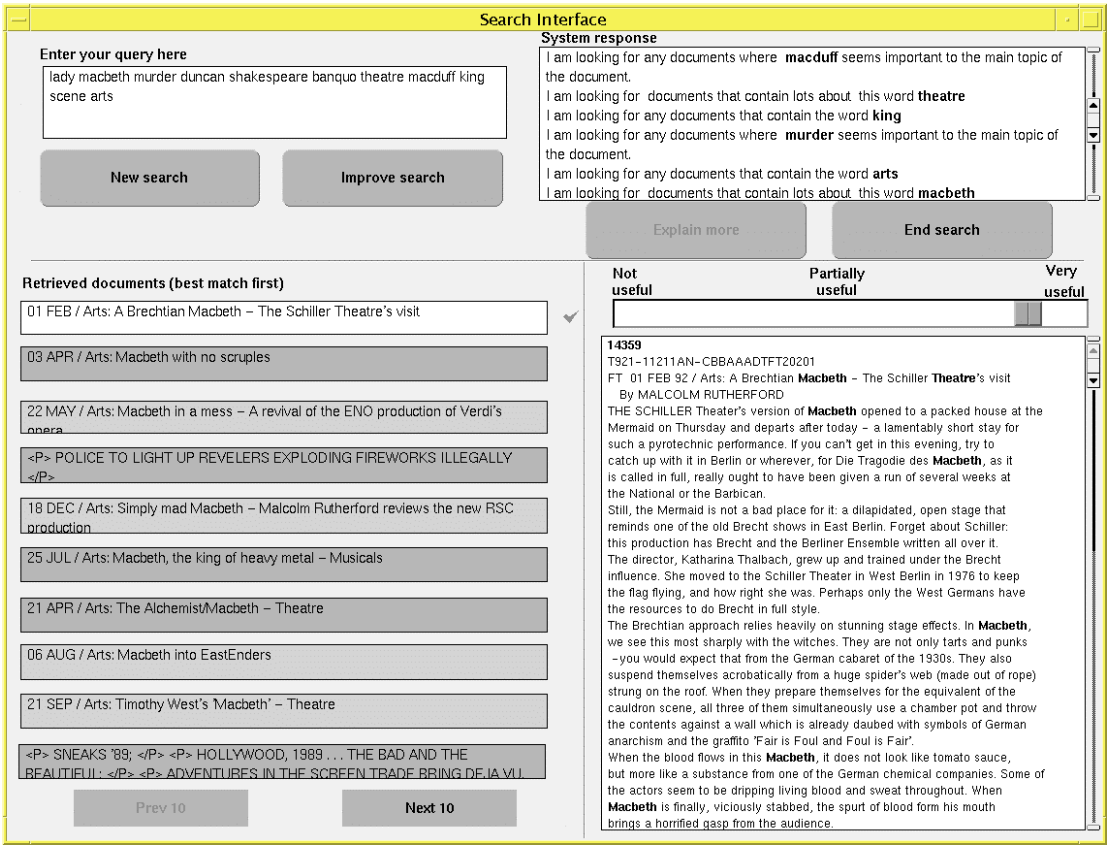

Figure G.12: Interface Four after selection of *Explain more* option

The *Explain more* option can give three types of information.

i. It can tell the user which terms are being treated as important to the main topic of the document. This corresponds to terms for which the *theme* characteristic has been selected as important. The system presents a message like '*I am looking for documents where* ***macduff*** *seems important to the main topic of the document'*, Figure G.12.

ii. It can tell the user which terms should appear often in retrieved documents. This corresponds to selection of the *tf* characteristic. In this case the system will present a message like '*I am looking for documents that contain lots about* ***macduff***'.

iii. If the *noise* or *idf* of the term is important, i.e. either of these characteristics have been selected for a term, the system will simply tell the user that these terms are important, e.g. '*I am looking for any documents that contain the word* ***macduff***'.

G.5 Logging

The system maintains a continuous log of the user interaction. A sample log is given in section G.6, Figure G.13.

The log file stores 11 types of information, for example the query terms entered by the user, those documents viewed by the user and the documents retrieved in response to the user's query. Each line of the log file starts with a tag that denotes the *type* of the log entry; the remainder of the line contains the data associated with the tag. Tale G.4 gives a listing of the tags, sample data, and an explanation of each tag.

All interfaces use the same set of tags, however the individual interfaces may not use the complete set of tags. The tags used in the logs for each interface varied according to the functionality offered by the interface. For example, Interface One only allows users to issue new queries. The logs generated by this interface, then, will not contain any tags relating to relevance feedback actions. The standard format of the log files permitted automatic analysis of the logs to obtain search statistics for each user search.

tag	***sample***	***description***
FEEDBACK	-	This tag is not associated with any additional data. It is used to signify that the user has issued an RF request by clicking the improve search button.
INFO_NOISE USED VALUE	INFO_NOISE USED 0 VALUE 0.000000	This set of tags stores whether the info_noise characteristic was used in RF and the RF value of the characteristic. 1 indicates that the characteristic was used, 0 indicates that the characteristic was not used.
LOGGED QUERY TERM	LOGGED QUERY TERM macbeth	This tag is associated with the query that is actually run by the retrieval system, and is output to the query file after RF. The difference between entries of this tag and the QUERY TERM tag are used to indicate those terms that the user or system has added or removed from the query.
NEW QUERY	-	This tag is not associated with any additional data. It is used to signify that the user has issued a new query

		by clicking the new search button.
QUERY TERM	QUERY TERM lady	This tag is associated with the query terms stored in the query term filG. Each tag is associated with one query term. This is the query as it is give to the retrieval system.
RETRIEVED	RETRIEVED 300038	This tag is associated with a retrieved document. In the example, document number 300038 has been retrieved in response to the user's query.
RELEVANT DOC DEGREE ITERATION	RELEVANT DOC 101690 DEGREE 9 ITERATION 4	This set of tags is used to store the relevance assessments given by the user. Entries are read as follows: document 101690 was given the relevance score 9 by the user during iteration 4. An initial search is iteration 1.
RF	RF macbeth 1 2.250000 1 2.153374 1 3.282353 1 2.250000 8.666667 0.115385	This tag is associated with the performance of the RF algorithms. The line stores a set of numbers for each term used in RF. The data stored includes which characteristics were selected for the term, the RF values for each term, and the ostensive and partial evidence weights for each term (Chapter 8).
SPECIFICITY USED VALUE	SPECIFICITY USED 0 VALUE 0.000000	This set of tags stores whether the specificity characteristic was used in RF and the RF value of the characteristic. 1 indicates that the characteristic was used, 0 indicates that the characteristic was not used.
SUGGESTED TERM	SUGGESTED TERM banquo	This set of tags denote the possible expansion terms suggested by the retrieval system. This set of tags is only used by Interface two.
TERM EXPLAINS	TERM macbeth EXPLAINS 247915	This set of tags denotes which query terms have been used to explain which relevant documents. In this example query term macbeth has been used to explain the (relevant) document 247915.
VIEWED	VIEWED 132902	The document number of each

		document that the user selects to read (by clicking on the document title) is stored.

Table G.4: Tags used in log files

G.6 Sample log

```
FEEDBACK
QUERY TERM art
QUERY TERM crime
GENERIC USED 1 VALUE 1.000000
INFO_NOISE USED 1 VALUE 1.000000
RETRIEVED 257875
RETRIEVED 238130
RETRIEVED 268997
RETRIEVED 278199
RETRIEVED 273470
RETRIEVED 311190
RETRIEVED 224975
RETRIEVED 237470
RETRIEVED 197106
RETRIEVED 304932
RETRIEVED 129180
RETRIEVED 265599
RETRIEVED 252541
RETRIEVED 257858
RETRIEVED 211334
RETRIEVED 42887
RETRIEVED 282265
RETRIEVED 227656
RETRIEVED 250558
RETRIEVED 52056
RETRIEVED 81630
RETRIEVED 281806
RETRIEVED 249934
RETRIEVED 249687
RETRIEVED 287794
RETRIEVED 221132
RETRIEVED 209046
RETRIEVED 209253
RETRIEVED 277930
RETRIEVED 296708
VIEWED 257875
VIEWED 238130
VIEWED 268997
VIEWED 278199
VIEWED 273470
VIEWED 311190
VIEWED 224975
VIEWED 129180
VIEWED 81630
VIEWED 249934
VIEWED 296708
FEEDBACK
RELEVANT DOC 238130 DEGREE 10 ITERATION 2
RELEVANT DOC 278199 DEGREE 7 ITERATION 2
QUERY TERM art
```

```
QUERY TERM fraud
TERM antiquities EXPLAINS 238130
TERM antiquities EXPLAINS 278199
RF antiquities 1 1399.999878 1 209.999985 1 560.000000 1 111.999992
8.500000 0.117647
 RF art 1 895.999939 1 658.000000 1 923.999939 0 1.000000 8.500000
0.117647
 RF antiquities 1 1399.999878 1 209.999985 1 560.000000 1 111.999992
8.500000 0.117647
 GENERIC USED 0 VALUE 0.000000
INFO_NOISE USED 0 VALUE 0.000000
RETRIEVED 238130
RETRIEVED 330986
RETRIEVED 141231
RETRIEVED 324816
RETRIEVED 300109
RETRIEVED 150645
RETRIEVED 85693
RETRIEVED 39208
RETRIEVED 256030
RETRIEVED 277521
RETRIEVED 175566
RETRIEVED 78056
RETRIEVED 181792
RETRIEVED 285962
RETRIEVED 9629
RETRIEVED 235755
RETRIEVED 188998
RETRIEVED 235604
RETRIEVED 58762
RETRIEVED 339381
RETRIEVED 150340
RETRIEVED 71933
RETRIEVED 304984
RETRIEVED 339889
RETRIEVED 17014
RETRIEVED 278680
RETRIEVED 101694
RETRIEVED 330699
RETRIEVED 240983
RETRIEVED 83921
VIEWED 238130
VIEWED 324816
VIEWED 256030
VIEWED 277521
VIEWED 175566
VIEWED 339381
VIEWED 150340
FEEDBACK
RELEVANT DOC 238130 DEGREE 10 ITERATION 2
RELEVANT DOC 278199 DEGREE 7 ITERATION 2
RELEVANT DOC 256030 DEGREE 5 ITERATION 3
QUERY TERM glasgow
QUERY TERM museum
QUERY TERM art
QUERY TERM crime
TERM antiquities EXPLAINS 238130
TERM antiquities EXPLAINS 278199
TERM antiquities EXPLAINS 256030
RF antiquities 1 2850.000000 1 474.999969 1 1520.000000 1 228.000000
7.333333 0.106061
 RF museum 1 1596.000000 1 949.999939 1 1178.000000 0 1.000000
5.666667 0.078431
```

```
 RF art 1 1824.000000 1 1178.000000 1 1995.000000 0 1.000000 7.333333
0.106061
 RF crime 1 2223.000000 1 418.000000 1 1539.000000 0 1.000000
7.333333 0.106061
 RF antiquities 1 2850.000000 1 474.999969 1 1520.000000 1 228.000000
7.333333 0.106061
 GENERIC USED 0 VALUE 0.000000
INFO_NOISE USED 0 VALUE 0.000000
RETRIEVED 238130
RETRIEVED 278199
RETRIEVED 256030
RETRIEVED 147016
RETRIEVED 277521
RETRIEVED 300109
RETRIEVED 324816
RETRIEVED 78056
RETRIEVED 150645
RETRIEVED 39208
RETRIEVED 175566
RETRIEVED 235604
RETRIEVED 235755
RETRIEVED 285962
RETRIEVED 188998
RETRIEVED 181792
RETRIEVED 85693
RETRIEVED 9629
RETRIEVED 58762
RETRIEVED 339381
RETRIEVED 339889
RETRIEVED 304984
RETRIEVED 150340
RETRIEVED 245150
RETRIEVED 326656
RETRIEVED 240983
RETRIEVED 278680
RETRIEVED 259914
RETRIEVED 245299
RETRIEVED 83921
VIEWED 238130
VIEWED 175566
VIEWED 339889
VIEWED 245299
VIEWED 175566
VIEWED 238130
VIEWED 256030
VIEWED 277521
VIEWED 300109
VIEWED 175566
VIEWED 235604
```

Figure G.13: Sample log file

Appendix H

Details on user evaluation

H.1 Topics used in experiments

In this section I shall describe the topics that were used in the user evaluation, Chapter Twelve. For each topic I shall present the original INTTREC topic, the simulated situation derived from the topic, the relationship between my topic and the INTTREC topic and the relation to Borlund's simulated situations.

H.1.1 Topic 303i

H.1.1.1 Original TREC Topic

Number: 303i

Title: Hubble Telescope Achievements

Description:

Identify positive accomplishments of the Hubble telescope since it was launched in 1991.

Narrative:

Documents are relevant that show the Hubble telescope has produced new data, better quality data than previously available, data that has increased human knowledge of the universe, or data that has led to disproving previously existing theories or hypotheses. Documents limited to the shortcomings of the telescope would be irrelevant. Details of repairs or modifications to the telescope without reference to positive achievements would not be relevant.

H.1.1.2 Simulated situation

At a recent party you overhear a discussion about whether science funding gives value for money. One person claimed that many expensive projects, such as the Hubble Telescope, do not produce significant positive advances. You are not sure how true this statement is, and

would like to find more information on the positive achievements of the Hubble Telescope since it was launched in 1991.

Tailoring

No specific tailoring to the likely subject population was included in this simulated situation.

Topical relevance

At the time of searching, the Hubble Telescope is not a current news event, and the specific area of searching – positive achievements of the telescope is considered to be of special interest to the subject population.

Semantic openness

The semantic openness of this reduced by the topical restriction to search for positive achievements on the Hubble Telescope rather than astronomy, or science, in general. The subjects, however, do have freedom to define what constitutes an achievement and, in particular, what constitutes a positive achievement. This simulated situation has a relatively narrow semantic openness, as the topic is restricted and neither the topic, nor the tailoring, is specific to the subjects.

H.1.1.3 Relation to TREC search

The topic does not include the specific indications for relevance outlined in the TREC narrative such as better quality or new datH. The topic instead relates the overall gist of the TREC topic – positive achievements of the telescope.

H.1.1.4 Relation to Borlund

Simulated work task situation

The other night you were at a party where the Hubble Telescope was discussed as one of the other guests knew quite a lot about this subject. Now you want to improve your own knowledge of this topic and more specifically you want to know about the Telescope's technical drawbacks and scientific achievements.

Borlund used this task as a training example. She classified her simulated work task situation as having low semantic openness due to similar factors as we have identified (low tailoring, limited topicality) and also due to the particular context – being present at a party at which this topic was discussed. However, this was one of the topics that her subjects found to be the most stimulating, mostly due to their unexpected interest in the topic.

We have retained the basic components of Borlund's work task situation but made two alterations. First, we have reduced Borlund's relatively strict search *indication 'technical drawbacks and scientific achievements*', replacing it with the less indicative '*positive achievements*'. Second, Borlund created a relatively neutral basis for the origin of the search – the subject wanted general background information. By framing the need for information within a general discussion about science funding we have tried to create a situation that has a stronger connection to the topic of the search. The intention here is to provide a motivation for the search that is more realistic but which does not form part of the specific search topic.

H.1.2 Topic 307i

H.1.2.1 Original TREC Topic

Number: 307i

Title: New Hydroelectric Projects

Description

Identify hydroelectric projects proposed or under construction by country and location. Detailed description of nature, extent, purpose, problems, and consequences is desirable.

Narrative

Relevant documents would contain as a minimum a clear statement that a hydroelectric project is planned or construction is under way and the location of the project. Renovation of existing facilities would be judged not relevant unless plans call for a significant increase in acre-feet or reservoir or a marked change in the environmental impact of the project. Arguments for and against proposed projects are relevant as long as they are supported by specifics, including as a minimum the name or location of the project. A statement that an individual or organization is for or against such projects in general would not be relevant. Proposals or projects underway to dismantle existing facilities or drain existing reservoirs are not relevant, nor are articles reporting a decision to drop a proposed plan.

H.1.2.2 Simulated situation

The new Scottish Parliament is considering planning permission for a series of large hydroelectric projects. These projects will use water power to produce electricity for a large area of Scotland. Supporters of the projects claim that they will give cheaper electricity and reduce global-warming, opponents argue that the projects may cause environmental damage and harm tourism. The Parliament has decided to hold a vote for all Scottish residents to decide if these projects should go ahead. You have little independent information upon which to base your decision, and would like information on similar projects.

Tailoring

Some tailoring on this topic relates to the siting of the hydroelectric projects in Scotland – where all subjects live. Further, the situation refers to a vote for residents rather than Scottish citizens as several of the subjects may be non-UK citizens.

Topical relevance

The references to global warming and the Scottish Parliament – both of which are current news items in the Scottish media – are attempts to make this topic more relevant to the subject group. The specific topic – hydroelectric projects – is not a current news item as the simulated situation is fictious. Even though the situation is not genuine we feel that it is realistic.

Semantic openness

As with topic 303 the subject of the topic – hydroelectric projects – is narrow in that there is little room for subjective interpretation regarding the definition of a hydroelectric project. However the semantic openness regarding what information has been left open to interpretation. Clues to possible aspects such as environmental change are intended to broaden semantic openness.

H.1.2.3 Relation to TREC search

We have retained the core topic – hydroelectric projects – but have not stressed the TREC distinction between new projects and existing projects. Neither have we asked the subjects to ignore the closure of hydrolectric projects. The TREC target of identifying locations of similar projects has not been included directly – we have not asked subjects to find locations specifically.

H.1.2.4 Relation to Borlund

This topic was not used in Borlund's experiment.

H.1.3 Topic 321

H.1.3.1 Original TREC Topic

Number: 321

Title: Women in Parliaments

Description

Pertinent documents will reflect the fact that women continue to be poorly represented in parliaments across the world, and the gap in political power between the sexes is very wide, particularly in the Third World.

Narrative

Pertinent documents relating to this issue will discuss the lack of representation by women, the countries that mandate the inclusion of a certain percentage of women in their legislatures, decreases if any in female representation in legislatures, and those countries in which there is no representation of women.

H.1.3.2 Simulated situation

It is likely that a British General Election will be held in May this year. In the last General Election, one of the main issues was the relatively low number of female members of parliament. This prompted one party to introduce special measures to increase the number of female candidates in the election. Other politicians argue that poor representation of women in parliament is not a specific feature of British politics. As the poor representation is likely to be a major issue in the forthcoming election, you would like to be more informed about the representation of women in politics.

Tailoring

No specific tailoring to university students has been used in this simulated situation towards university students. However, all subjects will be resident in the UK at the time of the election, which does create a topical news interest for searching.

Topical relevance

The topic is unlikely to be of particular interest to university students as an individual group. However, as with the tailoring aspect, there is likely to be a current news interest in this topic.

Semantic openness

The topical relevance does broaden the semantic openness somewhat as the issue is of current national interest at the time of searching. We avoided restricting the topic

specifically to the poor representation of women as this would have narrowed the semantic openness of the sitution, however we have hinted at this in the situation. We believe that this is one of the situations with a broader semantic openness.

H.1.3.3 Relation to TREC search

The main difference between our situation and the TREC topic description was that they did not stress that the subjects should search for documents on the *poor* representation of women. However, by stressing the cause of the situation – the poor representation of women in British parliament – and the fact the poor female representation is the case in most countries, we believe that most documents will be on this topic.

H.1.3.4 Relation to Borlund

This topic was not used in Borlund's experiment.

H.1.3.5 Update to topic

Some of the user experiments were carried out after the British General Election (experiments Three, Four and Five). Consequently the simulated situation was changed to the one below

During the previous General Election, in 1997, one of the main issues was the relatively low number of female members of parliament. This prompted one party to introduce special measures to increase the number of female candidates in the election. Other politicians argued that poor representation of women in parliament was not a specific feature of British politics.

In General Election that took place did in June this year, the poor representation of women was not a major issue but the Labour Party was criticised for its male-dominated election campaign. You wonder whether the poor representation of women is an international feature of politics.

H.1.4 Topic 322i

H.1.4.1 Original TREC Topic

Number: 322i

Title: International Art Crime

Description

Isolate instances of fraud or embezzlement in the international art trade.

Narrative

A relevant document is any report that identifies an instance of fraud or embezzlement in the international buying or selling of art objects. Objects include paintings, jewellery, sculptures and any other valuable works of art. Specific instances must be identified for a document to be relevant; generalities are not relevant.

H.1.4.2 Simulated situation

Several valuable paintings and other works of art in a local Glasgow museum have been discovered to be fakes. The museum's spokesman claims that art crime – in particular fraud – is becoming more common. He also claims that is difficult to distinguish deliberate crime from genuine mistakes made by people selling works of art. You wonder if he is correct or whether these are excuses. You think more information on art crime, and on genuine cases of art fraud, can help you decide if the spokesman is correct.

Tailoring

No specific tailoring to the intended population is made. The reference to Glasgow museums is not considered as tailoring as the search is unlikely to be centred around this particular instance of fraud.

Topical relevance

There is unlikely to be a particular topical interest in this topic.

Semantic openness

The semantic openness in this situation centres around the subject's definition of art crime and what constitutes genuine fraud. As these are left relatively open, this topic shows a relatively broad semantic openness.

H.1.4.3 Relation to TREC search

The original TREC topic specifically mentions embezzlement which we found too difficult to incorporate within the situation. It also specifically asks for particular instances of art crime. We have retained the request for instances of art fraud but have generalised the topic to include the area of art crime in general.

H.1.4.4 Relation to Borlund

Simulated work task situation

> There has been a burglary in your flat. Among the things stolen was an old and unique piece of jewellery with a high value of affect. You called the police, who were not very hopeful of getting the jewellery back. They said that there had been several such burglaries in the areas within the previous few months. You're interested in finding out about similar cases and more specifically the details and the consequences of the crimes.

Borlund heavily tailored this search to allow for a more realistic and personal situation – that of a theft in the subject's flat. The area of fraud in art was translated into the general area of burglaries. There was no specific tailoring to the subject population, nor was the topic felt to be of specific interest to the subjects. This topic was generally less popular with the searchers used by Borlund, even though the topic showed broad semantic openness with reference to the vagueness of the concepts crime, details and consqeuences of crime.

H.1.5 Topic

H.1.5.1 Original TREC Topic

Number: 326i

Title: Ferry Sinkings

Description

Any report of a ferry sinking where 100 or more people lost their lives.

Narrative

To be relevant, a document must identify a ferry that has sunk causing the death of 100 or more humans. It must identify the ferry by name or place where the sinking occurred. Details of the cause of the sinking would be helpful but are not necessary to be relevant. A reference to a ferry sinking without the number of deaths would not be relevant.

H.1.5.2 Simulated situation

You and a friend are trying to choose a holiday for later this summer. One possible holiday destination will mean taking several ferry trips but you have heard rumours that ferries in this area have a poor safety record. You need to book your holiday soon but need more information on the dangers of ferry travel.

Tailoring

There is no tailoring to this particular subject population.

Topical relevance

There is not particular topical relevance to this group.

Semantic openness

We have tried not to make any information on the areas where ferries operate, where ferry travel may be dangerous or the location of the persons intended travel. We have deliberately left open the question of what is meant by dangers posed by ferry travel.

H.1.5.3 Relation to TREC

The main requirement that relevant documents must be about 100 deaths or more has not been incorporated into this search. This was one of the most difficult TREC topics to incorporate into a simulated search.

H.1.5.4 Relation to Borlund

Simulated work task situation: Some friends of yours are about to visit you and as a surprise you are planning a trip for all of you to the Isle of Arran. You have heard rumours that some of the ferries to Arran are less safe than others. In addition to this you have recently seen the movie Titanic. You would therefore like to retrieve information about the causes of safety problems on ferries as well as some information about how to prevent accidents.

As Borlund's experiments took place in Scotland, the Isle of Arran was mentioned to include some topical relevance, although she does not consider this to have narrowed or broadened the semantic openness of the search. This was one to the less popular situations according to her subjects.

H.1.6 Topic 347i

H.1.6.1 Original TREC Topic

Number: 347i

Title: Wildlife Extinction

Description

The spotted owl episode in America highlighted U.S. efforts to prevent the extinction of wildlife species. What is not well known is the effort of other countries to prevent the demise of species native to their countries. What other countries have begun efforts to prevent such declines?

Narrative

A relevant item will specify the country, the involved species, and steps taken to save the species.

H.1.6.2 Simulated situation

Your best friend is an active member of a major wildlife preservation group. She is working on a project to build an electronic database of wildlife species that are in danger of extinction and the steps that different countries have taken to protect these species. She has asked you for help in providing information on international attempts to save native species, and the causes of wildlife extinction.

Tailoring

No tailoring to the subject group was possible.

Topical relevance

This topic is not especially relevant to the subject group.

Semantic openness

This situation is similar to one used by Borlund [Bo01], who reports a narrow semantic openness for this situation. The original situation asked the subject to imagine that s/he was responsible for creating the database. We have tried to increase the semantic openness by reducing the subject's responsibility to simply finding information.

H.1.6.3 Relation to TREC

We have tried to maintain the core aim of the topic, the only aspect which have not specifically highlighted was naming countries that have adopted special measures to prevent wildlife extinction.

H.1.6.4 Relation to Borlund

Simulated work task situation

> You have got a new student job with a local branch of one of the wildlife protection organisations. Your responsibility is to maintain and update the web pages of the organisation. You have been informed that the organisation's next big campaign will be on how to prevent the decline of wildlife species, focusing on the situation in Europe. As a new member of staff you feel you need some basic background information so you have decided to investigate the European situation with particular reference to problems caused by environmental and climate changes.

As discussed above we have retained the core elements of Borlund's situation but have shifted the emphasis of the searcher to finding information rather than creating the web site. We have also reduce the detail of the background need – *'particular reference to problems caused by environmental and climate changes*'.

H.2 Student topics

In this section I describe the simulated situations that were used in the pilot test. These were designed specifically for the student subjects.

H.2.1 Simulated situation 1

After graduation you will be looking for a job in industry. You want information to help you focus your future job seeking. You know it pays to know the market. You would like to find some information about employment patterns in industry and what kind of qualifications employers will be looking for from future employees.

H.2.2 Simulated situation 2

You have just moved into a shared flat with three friends. Your landlord is the father of one of your potential flatmates, who has bought the flat for his daughter whilst she is at university. You haven't been given a rent book or signed a lease as your landlord doesn't see himself as a professional landlord. You are concerned about what rights you have as a tenant in this situation but don't want to fall out with your flatmate. Perhaps you could find more information on the rights of tenants and the responsibilities of landlords before you raise the issue with your new landlord.

H.2.3 Simulated situation 3

Your credit card balance is becoming larger and credit card supplier is becoming less and less sympathetic. You are considering changing your credit card but you realise that, although it is easy to obtain a new card, there are often hidden charges involved if you go into debt. The credit card suppliers you have examined do not make these charges clear and you would like more information on how to choose a credit card.

H.2.4 Simulated situation 4

Last night you were out for a meal with some friends. One of the main topics of conversation was the potential legalisation of cannabis. Many people were favour of soft drugs, such as cannabis, being legalised but other friends were strongly against this. You are not sure where you stand on this issue. From the conversation, you are aware of some of the arguments but would like more facts about the possible implications of the legalisation of cannabis.

H.2.5 Simulated situation 5

You have been buying a lot of books recently. The price of these books has varied a great deal: some books were sold at a big discount, but other books, especially ones aimed at

students, were very expensive. You are beginning to wonder how publishers and booksellers decide how much to charge for a book (and if you are being ripped off).

H.2.6 Simulated situation 6

Last year there was a major crisis in the marking of Scottish school exams. This resulted in many pupils receiving wrong grades or not receiving any results for exams that they had taken. In England the increase in A level passes raised the question of the quality of marking standards. As you have several friends who are students, the debate about the fairness of exam marking and the consistency of individual markers makes you wonder whether exams are a fair method of assessing a student's performance.

H.3 Welcome questionnaire

INTERACTIVE SEARCHING STUDY
EXPERIMENTAL INSTRUCTIONS

Thank you for agreeing to participate in this experiment.

The goal of this experiment is to determine how well an information retrieval system can help you to find information on a given topic. Only the system is being tested, you are **not** being tested on how well you search.

You will be given a short description of a situation in which you might want to search for information. This is an example of such a situation.

> *You have thinking about buying a flat and are aware that there are several types of mortgage available. You are not sure about what kind of mortgage is best for you. You would like more information on the advantages and disadvantages of the different mortgages available before you make your choice.*

You are asked to imagine that you are the person described in the situation and to search for information.

You will be asked to search on six topics. You will be given fifteen minutes to search on each topic.

You will also be asked to complete several questionnaires:

- Before the experiment
- After each search
- After the experiment

At any point in the experiment you may ask for clarification on the search topic, experimental instructions or on how the system works.

You will be paid £20 for your participation in this experiment. This will be paid at the end of the experiment.

H.4 Background questionnaire

INTERACTIVE SEARCHING STUDY
ENTRY QUESTIONNAIRE

1. What college/university degrees/diplomas do you have (or expect to have)?

_____
degree subject date

_____
degree subject date

_____
degree subject date

2. What is your gender? o Female o Male

3. What is your age? _____ years

4. Have you participated in previous online searching studies? o Yes No

5. Overall, for how many years have you been doing online searching? _____ years

6. Please circle the number closest to your experience.....

How much experience have you had...	**No experience**		**Some experience**		**A great deal of experience**
1. using a point-and-click interface, e.g. Macintosh, Windows	1	2	3	4	5
2. searching on computerised library catalogs either locally (e.g. in your library, or remotely (e.g., Library of Congress)	1	2	3	4	5
3. searching on world wide web search services (e.g. Alta Vista, Excite, Yahoo, HotBot, WebCrawler)	1	2	3	4	5
4. searching on other retrieval systems, please specify the system: _____	1	2	3	4	5

7. Please circle the number that is closest to your searching behaviour....

	Never	**Once or twice a year**	**Once or twice a month**	**Once or twice a week**	**Once or twice a day**
How often do you conduct a search on any kind of system?	1	2	3	4	5

H.5 Pre-search worksheet

INTERACTIVE SEARCHING STUDY
PRE-SEARCH WORKSHEET

Searcher #_____

Condition _____

Topic #_____**6**_____

Your search situation is:

> Your best friend is an active member of a major wildlife preservation group. She is working on a project to build an electronic database of wildlife species that are in danger of extinction and the steps that different countries have taken to protect these species. She has asked you for help in providing information on international attempts to save native species, and the causes of wildlife extinction.

Before you start your search, please indicate how much you think you know about this topic

I know almost nothing about this topic	I have some knowledge but not much	I have general background knowledge	I know more than most people	I am very well-informed about this topic
1	2	3	4	5

H.6 Post-search worksheet experiment one

INTERACTIVE SEARCHING STUDY

POST-SEARCH WORKSHEET

Searcher #_____

Condition _____

Topic # _____

Please answer the following questions, as they relate to the search you have just completed.

	Not at all		**Somewhat**		**Extremely**
1. Was it easy to get started on this search?	1	2	3	4	5
2. Was it easy to do the search on this topic?	1	2	3	4	5
3. Are you satisfied with your search results?	1	2	3	4	5
4. Did you have enough time to do an effective search?	1	2	3	4	5
5. Was the search task realistic?	1	2	3	4	5
6. How interested were you in the topic of the search task?					
7. How enjoyable was this search?	1	2	3	4	5

If you used the *improve search* option,
how useful do you think the option was to your search?

Don't know/ didn't use improve search		**Not at all**		**Somewhat**		**Extremely**
0		1	2	3	4	5

How **easy** was it to judge how useful a document was to the search

Not at all		**Somewhat**		**Extremely**
1	2	3	4	5

H.7 Post-search worksheet experiment two

INTERACTIVE SEARCHING STUDY

POST-SEARCH WORKSHEET

Searcher #_____

Condition _____

Topic # _____

Please answer the following questions, as they relate to the search you have just completed.

	Not at all		**Somewhat**		**Extremely**
1. Was it easy to get started on this search?	1	2	3	4	5
2. Was it easy to do the search on this topic?	1	2	3	4	5
3. Are you satisfied with your search results?	1	2	3	4	5
4. Did you have enough time to do an effective search?	1	2	3	4	5
5. Was the search task realistic?	1	2	3	4	5
6. How interested were you in the topic of the search task?					
7. How enjoyable was this search?	1	2	3	4	5

How **easy** was it to judge how useful a document was to the search

Not at all		**Somewhat**		**Extremely**
1	2	3	4	5

H.8 Post-search worksheet experiment three

INTERACTIVE SEARCHING STUDY

POST-SEARCH WORKSHEET

Searcher #_____

Condition _____

Topic # _____

Please answer the following questions, as they relate to the search you have just completed.

	Not at all		**Somewhat**		**Extremely**
1. Was it easy to get started on this search?	1	2	3	4	5
2. Was it easy to do the search on this topic?	1	2	3	4	5
3. Are you satisfied with your search results?	1	2	3	4	5
4. Did you have enough time to do an effective search?	1	2	3	4	5
5. Was the search task realistic?	1	2	3	4	5
6. How interested were you in the topic of the search task?					
7. How enjoyable was this search?	1	2	3	4	5

If you used the *suggest terms* option,
how useful do you think the query words, suggested by the system, were to your search?

Don't know/ didn't use suggest terms		**Not at all**		**Somewhat**		**Extremely**
0		1	2	3	4	5

How **easy** was it to judge how useful a document was to the search

Not at all		**Somewhat**		**Extremely**
1	2	3	4	5

H.9 Post-search worksheet experiment five

INTERACTIVE SEARCHING STUDY

POST-SEARCH WORKSHEET

Searcher #_____

Condition _____

Topic # _____

Please answer the following questions, as they relate to the search you have just completed.

	Not at all		**Somewhat**		**Extremely**
1. Was it easy to get started on this search?	1	2	3	4	5
2. Was it easy to do the search on this topic?	1	2	3	4	5
3. Are you satisfied with your search results?	1	2	3	4	5
4. Did you have enough time to do an effective search?	1	2	3	4	5
5. Was the search task realistic?	1	2	3	4	5
6. How interested were you in the topic of the search task?	1	2	3	4	5
7. How enjoyable was this search?	1	2	3	4	5

If you used any of the following options,
how useful do you think the option was to your search?

Improve search

Don't know/ didn't use improve search		**Not at all**		**Somewhat**		**Extremely**
0		1	2	3	4	5

Explain

Don't know/ didn't use explanation		**Not at all**		**Somewhat**		**Extremely**
0		1	2	3	4	5

Explain more

Don't know/ didn't use explain more		**Not at all**		**Somewhat**		**Extremely**
0		1	2	3	4	5

How **easy** was it to judge how useful a document was to the search

Not at all		**Somewhat**		**Extremely**
1	2	3	4	5

H.9 Exit questionnaire experiment two

Searcher #_____

Condition _____

INTERACTIVE SEARCHING STUDY
EXIT QUESTIONNAIRE

Now, please consider the searching experience that you just had.

	Not at all		**Somewhat**		**Extremely**
1. How easy was it to *learn to use* this information system?	1	2	3	4	5
2. How easy was it to *use* this information system?	1	2	3	4	5
3. How well did you *understand how to use* this information system?	1	2	3	4	5
4. How well did you understand the *improve search* option?	1	2	3	4	5
5. How easy was it to assess how useful a document was?	1	2	3	4	5

Of the six searching tasks you were given (please circle one)

1. Which tasks did you find most interesting?	1	2	3	4	5	6	Don't know
2. For which tasks did you find it the most difficult to *start* a search?	1	2	3	4	5	6	Don't know
3. For which tasks did you find the most difficulty in finding useful documents?	1	2	3	4	5	6	Don't know

Please write down any other comments that you have about your searching experience with this information retrieval system. Thank you!

H.10 Exit questionnaire experiment five

Searcher #_____

INTERACTIVE SEARCHING STUDY

EXIT QUESTIONNAIRE

Now, please consider the searching experience that you just had.

	Not at all		**Somewhat**		**Extremely**
1. How easy was it to *learn to use* this information system?	1	2	3	4	5
2. How easy was it to *use* this information system?	1	2	3	4	5
3. How well did you *understand how to use* this information system?	1	2	3	4	5

Of the six searching tasks you were given (please circle as many as you feel appropriate)

1. Which tasks did you find most interesting?	1	2	3	4	5	6	Don't know
2. For which tasks did you find it the most difficult to *start* a search?	1	2	3	4	5	6	Don't know
3. For which tasks did you find the most difficulty in finding useful documents?	1	2	3	4	5	6	Don't know

Of the two systems you used (System A provided explanations on the improve search option)

	System A	**System**	**Both the same**
Which system did you prefer?			

Please write down any other comments that you have about your searching experience with this information retrieval system. Thank you!