# Division of Labour and Sharing of Knowledge for Synchronous Collaborative Information Retrieval

Colum Foley B.Sc. (Hons)

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University
School of Computing
and
Centre for Digital Video Processing

Supervisor: Prof. Alan F. Smeaton

June 2008

# Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed:

ID No:

Date:

# Acknowledgements

Firstly I would like to thank my supervisor Alan Smeaton for his guidance and support over the last three and a half years, for giving me the time and space to develop and investigate my ideas whilst encouraging me to proceed towards my thesis.

Thanks to the Irish Research Council for Science Engineering and Technology (IRCSET), Science Foundation Ireland (SFI) (grant number 03/IN.3/I361) and the DCU Office of the Vice-President for Research (OVPR) who funded this research.

I would like to thank the Text REtrieval Conference (TREC) organisers for making the search logs from interactive experiments available online and I would like to thank those groups whose log data I used in my simulations: University of California, Berkeley, Royal Melbourne Institute of Technology, University of Massachusetts, Amherst, and the University of Toronto.

Thanks to all those in the Centre For Digital Video Processing (CDVP) who supported me with this thesis, in particular I would like to acknowledge the help of:

- Gareth Jones – whose advice and guidance through many meetings, and in particular with the development of the techniques for Collaborative Relevance Feedback, is very much appreciated.

- Paul Ferguson, Neil O' Hare, Pete Wilkins and Cathal Gurrin – whose assurance and advice helped me progress through my experiments at times where I felt I was going around in circles.

Many thanks to those from the CDVP who made my PhD experience a fun time: Sinéad, James, Hyowon, Fabrice, Georgina, Sandra, Gordon, Kevin, Daragh, Kirk, Mike, Kieran, Paul B, Jiamin.

Thanks to my close friends: Declan and Frances, Liam and Sinéad, and Dave and Roisín for the many nights out which allowed me to switch off for a few hours.

Thanks to my family: Avena and David (and Connor and Megan), Tanya and Andrew, and Shane and Emer for putting up with me over the last few years. Often it is those closest to us that have to bear the brunt of our despair.

Finally I wish to thank my parents, Mary and Patrick. I truly would not have been able to achieve what I have without your love and constant support over the years. I could not wish for two better role-models and for this I am forever indebted to you.

# Abstract

Traditional Information Retrieval (IR) research has focussed on a single user interaction modality, where a user searches to satisfy an information need. Recent advances in web technologies and computer hardware have enabled multiple users to collaborate on many computer-supported tasks, therefore there is an increasing opportunity to support two or more users searching together at the same time in order to satisfy a shared information need, which we refer to as *Synchronous Collaborative Information Retrieval*.

Synchronous Collaborative Information Retrieval (SCIR) represents a significant paradigmatic shift from traditional IR systems. In order to support an effective SCIR search new techniques are required to coordinate users' activities. In this thesis we explore the effectiveness of two techniques on SCIR: *division of labour* and *sharing of knowledge*. By implementing an effective *division of labour* policy the search task can be divided across collaborating searchers, thereby avoiding any duplication of effort across the users. In addition, a *sharing of knowledge* policy refers to the process of passing relevance information across users, whereby group members can benefit from the discoveries of their collaborators.

In order to explore these techniques we simulate two users searching together through an incremental relevance feedback system, whereby the ranked lists of documents returned to each user are modified in order to implement various division of labour and sharing of knowledge policies. In order to populate these simulations we extract data from the logs of interactive text search experiments from previous Text REtrieval Conference (TREC) workshops. These experiments represent the first simulations of SCIR to-date and the first use of TREC logs in order to populate simulations.

# Contents

# List of Figures

# List of Tables

# Publications

Foley, C., Gurrin, C., Jones, G., Lee, H., Mc Givney, S., O'Connor, N., Sav, S., Smeaton, A. F., and Wilkins., P. (2005). TRECVid 2005 Experiments at Dublin City University. In *TRECVid 2005*, Gaithersburg, MD. National Institute of Standards and Technology.

Foley, C. and Smeaton, A. F. (2008). Evaluation of Coordination Techniques in Synchronous Collaborative Information Retrieval. In *1st International Workshop on Collaborative Information Retrieval @ JCDL*, Pittsburgh, Pennsylvania, USA.

Foley, C., Smeaton, A. F., and Jones, G. J. F. (2008). *Collaborative and Social Information Retrieval and Access Techniques for Improved User Modelling*, chapter Combining Relevance Information in a Synchronous Collaborative Information Retrieval Environment. IGI Global.

Foley, C., Smeaton, A. F., and Lee., H. (2006). Synchronous collaborative information retrieval with relevance feedback. In *CollaborateCom 2006 - 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–4, Atlanta, Georgia.

Smeaton, A. F., Foley, C., Byrne, D., and Jones., G. (2008). iBingo Mobile Collaborative Search. In *CIVR 2008 - ACM International Conference on Image and Video Retrieval. VideOlympics at CIVR*, Niagara Falls, Canada.

Smeaton, A. F., Foley, C., and Donald., K. M. (2005). Annodex-ing Broadcast TV

News for Semantic Browsing and Retrieval. In *ISWC2005 - 4th International Semantic Web Conference*, Galway, Ireland.

Smeaton, A. F., Foley, C., Gurrin, C., Lee, H., and Givney, S. M. (2006a). Collaborative Searching for Video Using the Físchlár System and a DiamondTouch Table. In *TableTop2006 - The 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 149–156, Adelaide, Australia.

Smeaton, A. F., Lee, H., Foley, C., Givney, S. M., and Gurrin., C. (2006b). Físchlár-DiamondTouch: Collaborative Video Searching on a Table. In *SPIE Electronic Imaging - Multimedia Content Analysis, Management, and Retrieval, SPIE Vol. 6073*, San Jose, CA.

Smeaton, A. F., Lee, H., Foley, C., and Mc Givney., S. (2006c). Collaborative Video Searching on a Tabletop. *Multimedia Systems Journal*, 12(4):375–391.

# Chapter 1

# Introduction

Information retrieval (IR), as defined by Baeza-Yates and Ribeiro-Neto (1999), is concerned with the representation, storage, organisation of and access to information items. The purpose of an information retrieval system is to satisfy a user's information need. Traditionally, information retrieval (IR) research has focussed on a single user interaction model.

*Collaborative Information Retrieval* is a phrase that has been used to refer to many different technologies which support collaboration in the IR process. Much of the early work in collaborative information retrieval has been concerned with *asynchronous*, remote collaboration. Collaborative filtering systems have been developed which attempt to reuse users' interactions with information objects in order to recommend them to others (Goldberg et al., 1992), collaborative re-ranking systems attempt to promote items of interest to a community of likeminded users (Smyth et al., 2005) and collaborative footprinting systems attempt to record the paths of users through an information space so that others may follow (Ahn et al., 2005). Asynchronous collaborative information retrieval supports a passive, implicit form of collaboration where the focus is to improve the search process for *an individual.*

Recent advances in both web technologies, such as the sociable web of Web 2.0, and computer hardware, such as tabletop interface devices, have enabled multiple users to collaborate on many computer-supported tasks such as town planning (Rogers et al., 2006) and design (Kvan, 2000). Due to these advances there is an increasing opportunity to support two or more users searching together at the same time, in order to satisfy a shared information need which we refer to as *Synchronous Collaborative Information Retrieval* (SCIR). Synchronous collaborative information retrieval systems represent a significant paradigmatic shift in information retrieval systems from an individual focus to a *group* focus. Synchronous collaborative information retrieval systems are concerned with the realtime, explicit, collaboration which occurs when multiple users search together to satisfy a shared information need. This collaboration can take place either with the users working remotely, or, in a co-located setting. These systems have gained in popularity and now with the ever-growing popularity of the social web (or *Web 2.0*), and the development of new collaborative computer interfaces, there is a real opportunity to enable support for explicit, synchronous collaborative information retrieval.

## 1.1 Synchronous Collaborative Information Retrieval

Early examples of SCIR systems include GroupWeb (Greenberg and Roseman, 1996) and W4 browser (Gianoutsos and Grundy, 1996). The focus of these early SCIR systems were in increasing *awareness* across collaborating users during a synchronised search, and this was achieved through various cues such as chat facilities, which users could use to communicate with each other, shared whiteboards, for realtime brainstorming, and bookmarking tools, where users could save documents of interest and bring them to the attention of the group. Although these systems allowed for a more engaging, collaborative search experience, providing awareness tools alone does not create effective SCIR. The benefit of allowing multiple users to search together in order to satisfy a shared information need is that it can allow for a *division of labour* and a *sharing of knowledge* across a collaborating group (Zeballos, 1998; Foley et al., 2006). The awareness cues provided in early SCIR systems could allow users to coordinate their activities in order to achieve both a division of labour and a sharing of knowledge. For example, users could use a chat facility to divide the search task, e.g. "You search for information on $X$ and I'll search for information on $Y$", and the shared bookmark facility could enable a sharing of knowledge, as users can see the documents found by others. However, as noted by Adcock et al. (2007), requiring users to coordinate activities may become troublesome as it requires "too much cognitive load to reconcile and integrate one's own activities with the opinions and actions of teammates".

Recently we have seen work which attempts to provide system-mediated coordination of users actions in a collaborative search. In particular, the "Cerchiamo" system of Adcock et al. (2007) was a system for co-located video search which assigned co-searchers complementary roles and coordinated their activities by directing the group towards unexplored areas of the collection, the "SearchTogether" system by Morris and Horvitz (2007) allowed users to divide the results of a search query across group members. Both of these systems represent "first steps" towards effective system-mediated coordination of an SCIR search, however there is much still to explore. In particular, both systems still require a certain amount of user-user coordination or awareness in order to allow for effective search. The Cerchiamo system was built for a co-located setting and required users to be assigned specialised roles for collaboration. While the SearchTogether system would still require some user-user coordination in order to allow for effective division of work, as we will now outline.

### 1.1.1 Thesis Hypothesis

At present, if two users want to search together on an adhoc search task using a state of the art SCIR system, they can either use awareness cues to coordinate their work, as provided by systems such as GroupWeb or W4, or, having decided on a useful query, a user could elect to divide the search results with their search partner using a system such as SearchTogether. Although the ability to divide a single search result in isolation does allow for a division of labour, it is not clear how such a technique would operate over a number of search iterations. For example, if a collaborating user decides to formulate another query and search again, then should the user split the results again? If a user is examining a set of search results and is presented with another set of results, seemingly, at random then, over a number of iterations, a user may become overwhelmed. Of course, users could use a chat facility to coordinate the activity - “Are you ready to search again?” However as discussed above, requiring users to both coordinate the group activity and search at the same time may cause them to suffer from cognitive overload. Notwithstanding these problems, a division of labour policy is an important aspect of an SCIR system, as without it, as users are searching to satisfy the same, shared information need, they will invariably be presented with similar search results, and may waste time reading the same documents.

Sharing of knowledge in SCIR systems can be achieved through an awareness cue such as a shared bookmark facility. What these bookmarks represent are *explicit relevance judgments* made by the users. In traditional, single user, information retrieval, *relevance feedback* is a simple technique that has been shown to improve the performance of a search by reformulating a user’s query in the light of extra relevance information. At present in SCIR systems we believe that these explicit relevance judgments, which are stored in bookmarks, are wasted, as they are not used in the search process itself but, instead, are merely used as awareness cues where users can see the documents found by others. By introducing a relevance feedback mechanism into an SCIR search, users could see the benefit of these bookmarks in their ranked lists. In addition, as these bookmarks are shared across users, the relevance feedback mechanism does not have to be isolated for a single user but, instead, could incorporate the relevance judgments made by each collaborating user into the feedback process. By providing a relevance feedback mechanism and using each user’s bookmarks as input into the process, we are enabling a system mediated sharing of knowledge across users. Users can see the benefit from their search partner's discovery of documents, without having to read these documents themselves.

These observations have led to the establishment of our hypothesis as:

> *If we have an environment whereby multiple people are searching together in a synchronised manner in order to satisfy a shared information need, then we can improve the performance of the search by implementing both division of labour and sharing of knowledge policies*

We believe that there is a need to develop appropriate mechanisms for division of labour in synchronous collaborative information retrieval. An appropriate policy should enable users to search without having to coordinate their activities whilst benefitting from having the search task divided across themselves and their collaborators. We also believe that by using multi-user relevance information, such as bookmarks, within an SCIR search itself through the relevance feedback process, we can further improve the performance of the group search. It is not clear, however, how the relevance feedback operation should be extended to a *collaborative relevance feedback process* that handles multi-user relevance information. Furthermore, collaborative systems, by their very nature, bring together multiple people with different levels of experience and expertise. For collaborative searches, differences in experience and expertise may cause some users to make poor relevance judgments, and therefore there needs to be a mechanism to allow for a biasing of relevance information in favour of the more expert users. Intuitively, a collaborative relevance feedback system, which essentially combines multi-user relevance information, may cause user's search queries to become too similar thereby limiting the *breadth* across users' queries. An alternative use of relevance information in an SCIR search may, therefore, be in a *complementary relevance feedback process.*

### 1.1.2 Research Objectives

Our primary research objectives are to thoroughly investigate the effects of both a division of labour and a sharing of knowledge, through the relevance feedback process, in a synchronous collaborative information retrieval environment. As we will investigate many different SCIR techniques and policies, we will develop simulations with which to evaluate our work. Simulations, in the context of IR research, attempt to model a user's interactions with an information retrieval system, such as querying the search engine and selecting results to view. Although simulations have been used extensively in IR system research, there has been no work to date which has attempted to model a synchronised collaborative search session and therefore our secondary research objectives are to both develop appropriate SCIR simulations, which will model two users searching together through an SCIR system, and to propose a novel evaluation metric for evaluating the performance of a group of collaborating searchers over the course of an SCIR search.

In order to explore our hypothesis, we have identified the following key research questions:

- Does a division of labour policy improve the performance of a group of users searching together?
    - Does implementing a division of labour in an SCIR environment improve the performance of SCIR?
    - Does it improve the performance over a group of users searching independently?
- Does a system-mediated sharing of knowledge policy, through a collaborative relevance feedback process, improve the performance of an SCIR search?
    - How do the techniques operate under perfect relevance information?
    - How do the techniques operate under imperfect relevance information?

  - Can we circumvent some of the problems associated with imperfect relevance information through a user-biased collaborative relevance feedback process?

- Does a collaborative relevance feedback process cause collaborating users' search results to become more similar than by using their own relevance information only?

- Can a complementary feedback mechanism allow a user's search result to remain more distinct than a collaborative relevance feedback process, and does this improve the performance of the search?

## 1.2 Thesis Organisation

This thesis is organised into seven chapters:

**Chapter 2:** In Chapter 2 we will provide a background for our research topic. We will introduce the research area of information retrieval (IR), providing a historical overview of the development of the research area as well as introducing the main concepts of the area. We will then discuss in more detail the retrieval algorithms which underpin much of IR research. Next we will discuss two ways in which IR researchers have attempted to improve tractional IR systems, through *relevance feedback* and *combination of evidence.* We will conclude the chapter with an overview of the process of evaluation in IR, including the metrics commonly used and the Text REtrieval Conference (TREC), an annual workshop for large-scale IR system evaluation.

**Chapter 3:** In Chapter 3 we will provide a comprehensive account of work to date in collaborative information retrieval. We will describe these systems in terms of their position in a 2-dimensional space of time and place, commonly used in Groupware system classification. Much of the work in collaborative IR has been in asynchronous collaborative IR systems such as collaborative filtering (or recommender) systems, but there are other examples of asynchronous collaborative IR systems, and we will outline these. We will also describe work to date in the emerging area of synchronous collaborative information retrieval, describing research in both remote and co-located domains.

**Chapter 4:** In Chapter 4 we will introduce our research proposal. We will motivate the need for an effective system-mediated division of labour and sharing of knowledge. We will describe how a division of labour system can be applied to an SCIR search to allow for seamless division of search results that does not require any user-user coordination. We will also explore how to achieve a sharing of knowledge by sharing relevance information across users in the relevance feedback process. We will propose novel methods by which the relevance feedback process can be extended into both a *Collaborative Relevance Feedback* process and a *Complementary Relevance Feedback* process.

**Chapter 5:** In Chapter 5 we will propose a novel experimental methodology which we will use to explore our hypothesis. We will describe how we plan to build SCIR simulations, involving two users, by mining transcripts of users' interactions with various search engines from TREC submissions. We will also outline how we plan to evaluate our simulated search sessions.

**Chapter 6:** In Chapter 6 we will present the results from our evaluations of the techniques proposed in Chapter 4, using the evaluation methodology outlined in Chapter 5. Firstly, we will evaluate the effects of a system-mediated division of labour on an SCIR search. Then we will evaluate the effects of both a division of labour and a sharing of knowledge policy through relevance feedback. We will examine the effects of collaborative relevance feedback on both perfect and imperfect relevance information and explore the effects of a user-biased collaborative relevance feedback process. Finally we will explore the effects of a complementary relevance feedback process on an SCIR search.

**Chapter 7:** In Chapter 7 we will conclude the thesis. Referring back to our hypothesis and research objectives, we will discuss the conclusions from our evaluations.

# Chapter 2

# Information Retrieval

In this chapter we will provide an overview of information retrieval. In section 2.1 we will discuss its concepts and the components of a typical information retrieval system. In section 2.3 we will describe the three classical models of information retrieval: the Boolean model, the Vector Space model and the Probabilistic model. Following that, in section 2.4 we will outline the process of iterative query reformulation known as relevance feedback and describe how it can be applied in all three classical retrieval models. We will also look at some extensions to the feedback models. In section 2.5 we will describe another technique used to improve the performance of an initial ranking through combining information from multiple sources of evidence. Finally in section 2.6 we will discuss evaluation in information retrieval, outlining why IR evaluation is difficult before describing the commonly used IR evaluation metrics and methodologies.

## 2.1 Introduction

Information Retrieval (IR), as defined by Baeza-Yates and Ribeiro-Neto (1999), is concerned with the representation, storage, organisation of and access to information items. A distinction is made between *data* and *information* retrieval. Data retrieval refers to retrieval from a structured source that is easily processed by a computer, an example being a relational database. Information retrieval on the other hand refers to retrieval from unstructured raw data such as text documents and it is this type of retrieval that will be the focus of this thesis.

In the middle of the last century, in light of the increasing amount of scientific information being produced after World War II and with the development of the first computer systems, came the notion of using computers to automate the document categorisation and indexing process. In his seminal paper entitled *"As we may think"*, Vannevar Bush illustrated the importance of effective access to these large stores of scientific records in order to ensure that significant work did not get "lost in the mass of the inconsequential" (Bush, 1945).

During the 1950's information retrieval was established as a research field aimed at developing more efficient and effective access to information and several researchers began looking at ways to automate the IR process. In 1957, Luhn proposed the use of document words as index terms in an automatic indexing system (Luhn, 1957), he also proposed the use of overlap of terms as a possible criterion for relevance. The SMART system (Salton, 1971) and Cranfield experiments (Cleverdon, 1967) represented the first experimental IR system and evaluation methodology respectively. Having both an experimental system and methodology, the 1970's and 1980's saw various improvements for indexing and retrieval models being proposed, including the development of two of the most widely used IR retrieval models: the Vector Space model (Salton, 1971) and Probabilistic model (Robertson and Spärck Jones, 1976). In 1992 the Text REtrieval Conference (TREC) was established by the US government under the auspices of the National Institute of Standards and Technology (NIST) in an attempt to promote research in information retrieval. TREC provided research groups with the ability to evaluate and compare systems using a common corpus of documents, topics and metrics. With the introduction of the World Wide Web in the early 90's and the explosion of online authoring and publishing, the need to develop more sophisticated techniques to identify relevant material rather than purely statistical co-occurrence measures was recognised.

## 2.2 A Typical Information Retrieval System

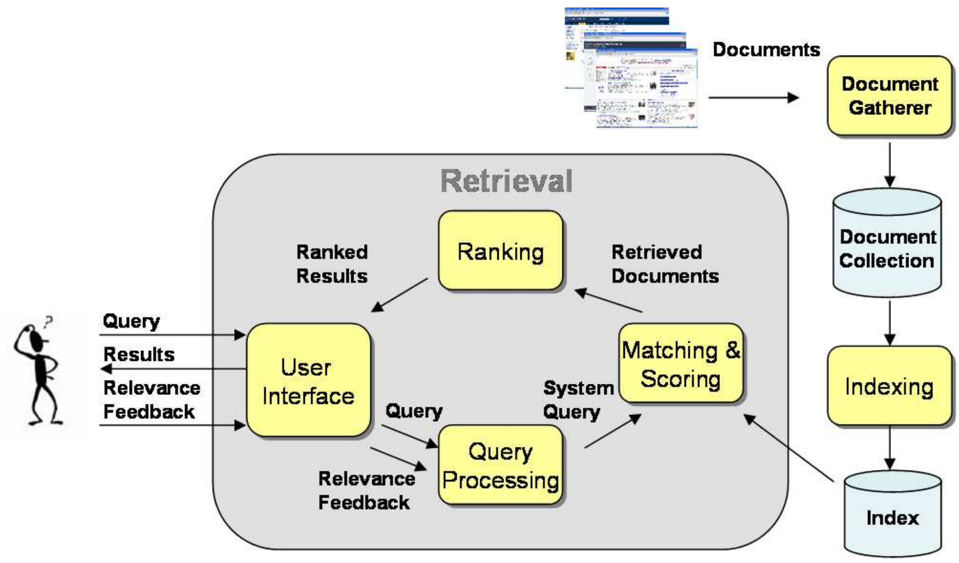


Figure 2.1: Overview of a typical information retrieval system

The purpose of an information retrieval system is to satisfy an information need. Before any user interactions can take place, an IR system must gather its collection. For a web search engine like Google, this collection may consist of the entire web, or for a desktop search engine the collection may consist of the contents of a user's hard drive. In either case, once the search engine's domain has been established, the documents to be searched are crawled and stored in the *Document Collection*. Within an information retrieval system the basic units for search and retrieval are referred to as *terms*, these are words or groups of words which have some meaning, in most cases an index term for document retrieval refers to a single word.

### 2.2.1 Document Processing and Indexing

Before a document collection can be searched, pre-processing techniques are applied to the text. These steps are performed in order to; (1.) reduce the overall size of the collection and thereby make the content amenable to faster access, and (2.) improve the effectiveness of the matching process at retrieval time. The two commonly used pre-processing methods used today are term *stopping* and term *stemming*.

Stopping is a process whereby the most commonly occurring terms are removed from a document, these terms are referred to as *stopwords.* Words which occur commonly in the collection such as “at”, “and”, “of” do not exhibit much discriminatory power when it comes to document matching and ranking, i.e. the presence of one of these terms does not help an IR system when deciding the relevance or non-relevance of a document, and can therefore be removed from the collection without impacting on retrieval performance. Stopping also significantly reduces the space needed to store the collection. Some general stopword lists can be found on the web (Salton, 1971) and these can be expanded or reduced depending on the application domain, for example a medical stop list may contain a stopword like “patient” if this term occurs in many documents of the collection.

*Stemming* is a procedure that may be applied in order to reduce terms to their common root in an attempt to increase matching at retrieval time. Without stemming the terms: “stemmer”, “stemming” , and “stemmed” would all be considered different words, which would create obvious problems at retrieval time due to the small word variations. Reducing words to their stemmed root also has the advantage of reducing the document collection further, in the example given above all three terms are represented in the collection with the single term “stem”. A frequently used stemming algorithm was developed by Porter in the 1980s (Porter, 1980).

Having stopped and stemmed the collection an IR system needs to create an *index* of the collection for fast access at retrieval time. In modern search engines the index of choice is the *inverted index.* Each entry in an inverted index contains a term and a list of those documents which contain the term (commonly referred to as *postings*) using the inverted index the search engine can locate the appropriate terms quickly and see in which documents the terms occur. This index is often supplemented with extra information regarding a term’s occurrence in the document, for example the term’s frequency of occurrence or location within the document may be recorded.

Having performed all pre-processing steps the original documents are reduced to a “bag of words” in the system. We have moved from a user interpretation of the documents to an interpretation more easily manageable to the information retrieval system. By discarding the original document structure, we are assuming that the semantic meaning of the document can be interpreted through its index terms (Baeza-Yates and Ribeiro-Neto, 1999). It is not surprising then that, due to this oversimplification, for some searches, few relevant documents are returned. Some techniques have been proposed to address these issues and we will discuss one of these techniques, known as relevance feedback, in section 2.4.

### 2.2.2 The Retrieval Process

A user's first contact point with a search engine is through its *user interface*, the most successful search engines nowadays present a clean, simple interface where the focus of the screen is on the text entry box. Figure 2.2 shows a screen-shot of the popular Google search engine interface.

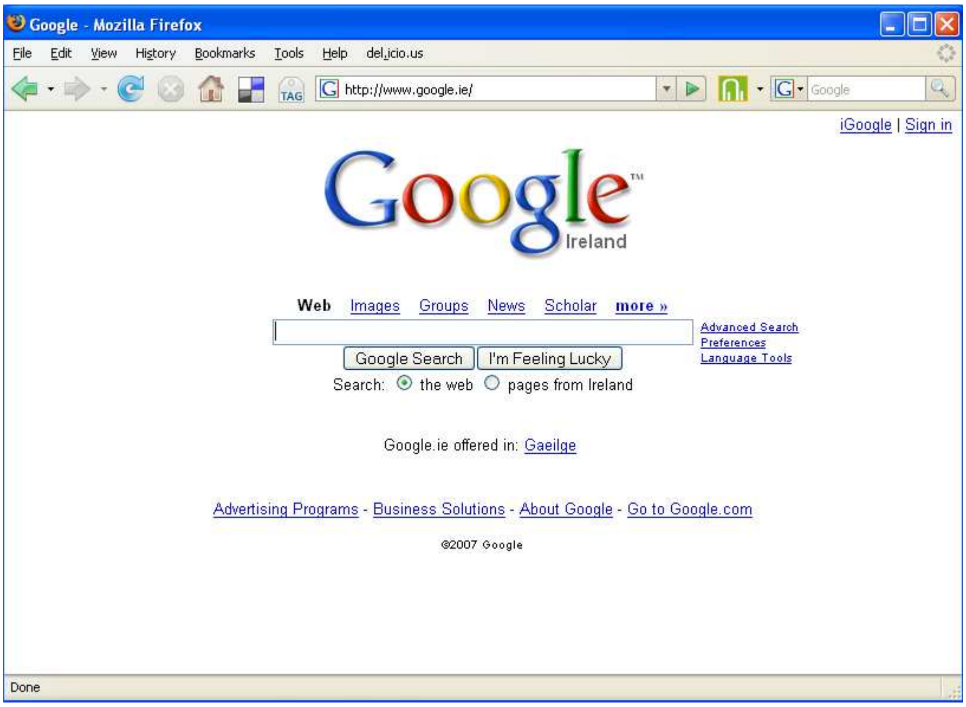

Google - Mozilla Firefox
File Edit View History Bookmarks Tools Help del.icio.us
TAG
http://www.google.ie/
Google
iGoogle | Sign in
Google™
Ireland
Web Images Groups News Scholar more »
Advanced Search
Preferences
Language Tools
Google Search
I'm Feeling Lucky
Search: ● the web ○ pages from Ireland
Google.ie offered in: Gaeilge
Advertising Programs - Business Solutions - About Google - Go to Google.com
©2007 Google
Done

Figure 2.2: The Google search interface

A user begins their interaction with the search engine by providing a query. It is interesting to note that despite the wide text entry box provided by search engine interfaces (see Figure 2.2), most queries nowadays consist of less than three search terms (Spink et al., 2002; Nielsen, 2001).

Having received the query, the search engine performs stopping and stemming on the text in order to transform the query into the form used in the underlying index. This system query is then compared against the inverted index in order to find matching documents in accordance with the *retrieval model* in use and documents are assigned a matching score. Matching documents are then ranked by their associated matching score and formatted before being presented to the user. By default, Google returns a ranked list of documents, 10 per page, and for each matching document, the document title is displayed along with a snippet of text from the document highlighting the matching terms. If a user locates a document of interest and wishes to view the document, the full document is loaded from the *Document Collection* and displayed to the user.

Information retrieval is often an iterative process. A user may begin their search a vague notion of their information need and this can result in a poor query being issued to the search engine. Several techniques have been proposed in order to improve retrieval results, one such approach is known as *relevance feedback*. By using relevance feedback, a user can indicate on an initial ranked list, those items of relevance to them, and a search engine can then use this extra information in order to improve the results returned to the user. This process will be discussed in detail in section 2.4.

### 2.2.3 Ubiquity of Modern Information Retrieval

Up until very recently almost all interactions with an online search engine would have been through an internet browser on a desktop computer or laptop. However in recent times, with the dawn of wireless computing and high speed mobile phone networks, users are able to access information whenever they want and wherever they are (Greenfield, 2006). The ubiquitous nature of modern information retrieval has resulted in a revisiting of research areas and techniques in order to extend IR to this new domain (Crestani et al., 2003). Mobile devices present challenges for user interface designers due to their smaller and more compact screen size and limited input capability (Dunlop and Brewster, 2002).

For example, a search engine's result list page needs to be formatted in order to fit on the small screen of a mobile phone or Personal Digital Assistant (Milic-Frayling et al., 2003). Some emerging research areas in mobile IR include the use of context in the search process, a user searching for "places to eat" may be returned different results depending on their location (Brown and Jones, 2001). The spread of information access beyond the desktop computer has resulted in the IR process becoming much more spontaneous and therefore a user wishing to satisfy an information need no longer has to wait until they are sitting at their desk.

In this section we have provided an overview of information retrieval. For an IR system to operate effectively it needs to return relevant documents in response to a user's query. How a query is matched to a document is determined by the retrieval model, which we will now discuss.

## 2.3 Retrieval Models

The fundamental problem for any retrieval system is to identify documents that are relevant to a particular query from the many that are not. In this section we will describe three classical models of retrieval; the Boolean model, the Vector Space model and Probabilistic model.

### 2.3.1 The Boolean Model

The Boolean model is a retrieval model built on set theory and Boolean algebra. Queries are expressed using a combination of query terms and the operators AND, OR and NOT. Figure 2.3 below shows a query for (Jaguar AND Car) NOT Animal, where the user is trying to retrieve information on the jaguar car brand and not the wild animal. In order to answer a Boolean query, the query is decomposed into its constituent terms then, for each query term, the document collection is queried and a set of documents are returned that contain the term in their text. Finally the Boolean operators are applied on these sets in order to create the final set of documents to be returned.

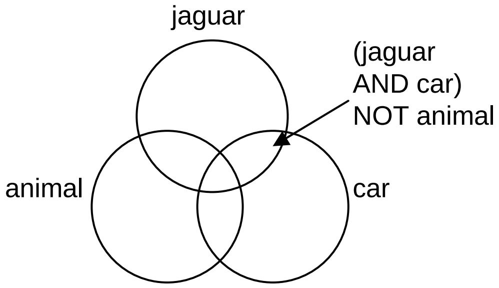


Figure 2.3: Boolean query example

As we can see from the example in Figure 2.3, a Boolean query makes it easy for a searcher to express with a level of precision, often lost in modern internet searches, the nature of their information need. This precise formalism and expressivity made the Boolean model and Boolean querying the interaction mode of choice for trained intermediaries, such as librarians, in the early days of search.

Despite its neat formalism, the model has a number of major drawbacks. Firstly, and perhaps most importantly, the model is built upon a binary decision criterion, a document is deemed to be either relevant or non-relevant there are no term weighting functions to differentiate query terms, no scale of relevance for matching documents and as a result no method for relevance ranking. This “exact-matching” metaphor, in which all documents that match a query are returned, frequently results in either too many or too few documents being retrieved for queries. Although in its most basic form the model is simple, often complex queries need to be formed using a combination of the Boolean algebra in order to separate relevant and non-relevant material and as a result most users of Boolean systems are highly trained individuals.

The need to incorporate a weighing for terms in documents and a ranking for documents led to the development of *best match* retrieval models. In the following two sections we will describe the two most commonly used best-match retrieval models, the Vector Space model and the Probabilistic model.

### 2.3.2 The Vector Space Model

Best match search systems differ from Boolean systems in that documents can be retrieved which only partially match the user's query and, through the provision of term weighting, matched documents can be ranked in order of their perceived relevance.

The vector space model is a best match retrieval model proposed by Salton in 1975 (Salton et al., 1975). In this model, documents and queries are represented as vectors in a t-dimensional space, with t being the number of terms in the document collection.

$$Q = \{q_1, q_2, q_3 ..., q_n\} \tag{2.1}$$

$$D = \{d_1, d_2, d_3 ..., d_n\} \tag{2.2}$$

where $q_i$ and $d_i$ represent the weight of term $i$ in the query vector and document vector respectively. The model attempts to calculate the *degree of similarity* between a document and a query. One often used method for estimating this degree of similarity is through calculating the cosine of the angle ($\theta$) between the document and the query. Figure 2.4 shows an example of the similarity between two documents, $d_1$ and $d_2$, and a query $q$ in the vector space.

Since a document will only contain a small subset of the entire amount of terms in the collection, document vectors are often very sparse.

The cosine correlation of the angle between the query and document vectors can be computed by:

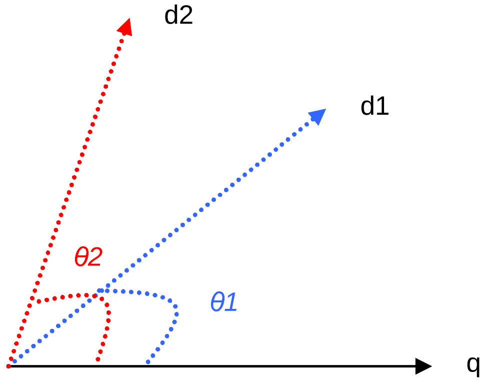


Figure 2.4: Documents and queries represented in a vector space

$$Sim(Q, D_j) = \frac{d_j.q}{||d_j|| \times ||q||} = \frac{\sum_{i=0}^{I-1} w_q(i).w_d(i,j)}{\sqrt{\sum_{i=0}^{I-1} w_q(i)^2}\sqrt{\sum_{i=0}^{I-1} w_d(i,j)^2}} \tag{2.3}$$

Before calculating the correlation, term weights are derived for both query terms ($w_q(i)$), and document terms ($w_d(i,j)$). In its most simplistic form this weight can be a binary value [0,1] indicating the terms appearance or absence in a document. Modern implementations of the vector space model, however, use a weighting function known as *tf-idf* to assign weights. In the next section we describe some common term weighting functions.

#### 2.3.2.1 Term Weighting Functions

Term weighting allows for terms in documents to be assigned weights based on their perceived significance both to the document in which they occur and also within the collection as a whole. The idea behind term weighting is *selectivity*, a good term being one that can select relevant documents from any number of non-relevant documents (Robertson and Spärck Jones, 1997).

The three main components which comprise a modern term weighting function are *term frequency*, *inverse document frequency*, and *document length normalisation*.

**Term Frequency:** Term frequency (or *tf*) is the number of occurrences of a particular term within a document, the rationale behind this weight being that the more often a term occurs within a document the more important the term is to that document. The term frequency for a term $i$ in document $j$ is defined as:

$$tf(i,j) = \text{number of occurrences of term } i \text{ in document } j \tag{2.4}$$

Experiments by the SMART team showed that using raw *tf* as a weight was non-optimal, and therefore proposed the use of certain *dampening* functions for term frequency, such as a logarithmic function (Buckley et al., 1992).

**Inverse document frequency:** Inverse document frequency (or *idf*) weighting was proposed by Spärck Jones (1972) and refers to the number of occurrences of a term across the collection of all indexed documents. The rationale here being that terms which occur in only a few documents within the collection are more selective than those which appear in many documents. The *idf* for a term $i$ is defined as:

$$cfw(i) = log\frac{N}{n_i} \tag{2.5}$$

where

$$N = \text{number of documents in the collection}$$

$$n_i = \text{number of documents in which term } i \text{ occurs}$$

**tf–idf:** A commonly used term weighting scheme is known as tf–idf weighting, which generally refers to any weighting function which incorporates both term frequency and inverse document frequency (Singhal, 2001). In its most simplistic form it can be given by:

$$w(i,j) = tf_{i,j} \times \log\frac{N}{n_i} \tag{2.6}$$

However this method is overly simplistic as it does not take into account the length of a document. It is commonly believed that document relevance should be independent of document length, however without compensating for document length, longer documents will tend to have higher term frequency scores simply because they are long. Several variations of document length compensation have been proposed, including normalising the term frequency component of Equation 2.6 by dividing it by the maximum term frequency in the document (Salton and Buckley, 1988). This method is shown in Equation 2.7, and there have been other proposed methods such as pivoted document normalisation weighting (Singhal et al., 1996).

$$w(i,j) = (\frac{tf_{i,j}}{max_t f}) \times log\frac{N}{n(i)} \tag{2.7}$$

The vector space model is a commonly used retrieval model due to its simplicity and good retrieval performance. In the next section we will outline another retrieval model, the probabilistic model, whose derivation is more formal than the vector space model and is based on probability theory.

### 2.3.3 Probabilistic Model

The probabilistic model for information retrieval is more of a family of models based upon the same probabilistic principle first proposed by Maron and Kuhns (1960), as an attempt to formalise retrieval within the probabilistic framework. The model has since been developed by Robertson and Spärck Jones (1976); van Rijsbergen (1979) and others.

The probabilistic model attempts to calculate the probability that a document ($D$) is relevant ($R$) to a query, *P(R/D)*. This leads to the *Probability Ranking Principle* proposed by Robertson (1977), which states that:

> "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 Probability Ranking Principle can be considered as the starting point for the derivation of the probabilistic model of retrieval. Robertson and Spärck Jones (1976) make use of Bayes' rule and log-odds in order to arrive at an optimal ranking function:

$$sim(D,Q) = \log \frac{P(D/R)}{P(D/\bar{R})} \tag{2.8}$$

It is at this point that the different probabilistic models diverge, based on their underlying assumptions. The *Binary Independence Model* proposed by Robertson and Spärck Jones (1976), is perhaps the most simplistic form of this model. It estimates the probability that a document is relevant by simplifying P(D/R) in terms of its attributes (i.e. terms). It represents document and query terms using binary weights and assumes independence amongst terms, i.e. it assumes that terms occur independently in relevant and non-relevant documents, in order to arrive at a similarity value of:

$$sim(D,Q) = \sum_{ti \in Q,D} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{2.9}$$

where

$p =$ probability that a document contains term t$_i$ given that it is relevant, *P(t$_i$/R)*

$q =$ probability that a document contains term t$_i$ given that it is non-relevant, *P(t$_i$/$\bar{R}$)*

The appropriate substitutions for $p$ and $q$ are the proportions:

$$p = \frac{r_i}{R} \tag{2.10}$$

$$q = \frac{n_i - r_i}{N - R} \tag{2.11}$$

where

$N =$ number of documents in the collection

$n_i =$ number of documents in which term i occurs

$R =$ number of known relevant documents in the collection

$r_i =$ number of known relevant documents in which term i occurs

Substituting the values in Equations 2.10 and 2.11 into Equation 2.9 results in the relevance weighting formula: (Robertson and Spärck Jones, 1976):

$$rw(i) = \log \frac{(\frac{r_i}{R})(1 - \frac{n_i - r_i}{N-R})}{(\frac{n_i - r_i}{N-R})(1 - \frac{r_i}{R})} \tag{2.12}$$

Since at the beginning of the search we do not know the set $R$, the set of relevant documents, we need to estimate values for $p$ and $q$. The method most commonly used, is to leave $r_i$ and $R$ at zero and to add constants to each of the values in equation 2.12 (Robertson and Spärck Jones, 1976).

The Binary Independence Model was implemented by the City University team at TREC-1 (Robertson et al., 1992). However the performance of the model was quite poor, the reasons being that the model made no attempt to model a term's within document frequency (as terms were represented using binary weights) or document length normalisation. Therefore in TREC-2 two ranking models; BM15 and BM11 were introduced in order to experiment with different weighting functions incorporating term frequency and document length normalisation (Robertson et al., 1993). Finally in TREC-3 these two weighting functions were combined into the BM25 weighting function (Robertson et al., 1994), which has become the most frequently used retrieval model for probabilistic retrieval and is shown in equation 2.13 (note the constants added to the relevance weighting proportions to handle the case when no relevance information is available as described above).

$$w(i,j) = \log \frac{(\frac{r_i+0.5}{R+1})(1-\frac{n_i-r_i+0.5}{N-R+1})}{(\frac{n_i-r_i+0.5}{N-R+1})(1-\frac{r_i+0.5}{R+1})} \times \frac{tf(i,j) \times (k1+1)}{k1 \times ((1-b)+(b \times ndl(j))) + tf(i,j)} \tag{2.13}$$

where

$N$ = number of documents in the collection

$n_i$ = number of documents in which term i occurs

$R$ = number of known relevant documents in the collection

$r_i$ = number of known relevant documents in which term i occurs

$tf_{i,j}$ = is the term frequency measure of term i in document j

$ndl_j$ = is the normalised document length (dl) of document j

$= \frac{dl}{\text{avg dl of all docs}}$

$k1$ = constant which determines the influence of tf$_{i,j}$

$b$ = constant which determines the influence of document length normalisation

As we can see from Equation 2.13, by incorporating the relevance weight component, the probabilistic model supports relevance information inherently.

In this section we have discussed three classical models for information retrieval: the Boolean model, the vector space model and the probabilistic model. We have also outlined the term weighting functions on which the best-match models are based. Despite the great success of online search engines, sometimes the ranked lists returned to users are quite poor, and in the following sections we will outline two methods which can be used to improve the quality of ranked lists returned to users.

## 2.4 Relevance Feedback

Despite the success of modern online information retrieval systems, often it is difficult for an information retrieval system to locate any documents relevant to a query. Frequently this can be attributed to a user's poor expression of their information need in the query they issue to the search engine, indeed studies have shown how most web queries are often between 1–3 words in length (Spink et al., 2002; Nielsen, 2001). Relevance feedback is an inexpensive, iterative, query reformulation technique which has been proven to improve the quality of results (Cleverdon and Keen, 1966; Salton, 1989; Salton and Buckley, 1990; Harman, 1992; Haines and Croft, 1993; Buckley et al., 1994).

The motivation behind relevance feedback is that, although a user may not be able to define what they are looking for (in the form of a query), they can recognise relevant material when they encounter it (Ruthven and Lalmas, 2003). By feeding this information back to the system the search engine can generate an improved query formulation and improve retrieval results.

Salton and Buckley (1990) identified the benefits of relevance feedback as:

- It shields the user from details of the query reformulation process, thereby allowing for the construction of improved queries without the need for intimate knowledge of the collection.

- It breaks down the search process into a series of small steps allowing the user to approach the subject area gradually.

- It provides a controlled query reformulation process which can emphasise and deemphasise terms as required.

Users can provide feedback to the system either explicitly, maybe by clicking a button on the user interface, or implicitly, based on their interactions, for example if a user clicks on a document title and views the full document then we may assume this is an indication of relevance (or *relevance judgment*). Another form of relevance feedback, known as *pseudo-relevance feedback* is a completely automatic process (i.e. no user intervention) whereby the search engine assumes that the top documents retrieved from an initial query formulation are relevant and performs a relevance feedback iteration using these documents as relevance judgments.

Regardless of the method used for collecting feedback, the search engine can use these identified relevant documents to improve the initial query formulation in order to make the new query *look* more like relevant material and thereby less like non-relevant material. Relevance feedback improves an initial query formulation in two ways: (1.) by adding important terms from identified relevant documents, through a process known as *query expansion*, and (2.) by attaching a weight to all query terms in order to emphasise important terms and de-emphasise less important terms through *term reweighting.* The largest improvement is generally attributed to query expansion, however the combination of both expansion and reweighting has been shown to provide the best performance (Harman, 1992).

In the following sections, we will outline the relevance feedback process for both the vector space and probabilistic models of retrieval

### 2.4.1 Relevance Feedback in the Vector Space Model

Relevance feedback in the vector space model can be viewed as re-weighting terms in the original query vector. With an initial query vector of

$$Q = \{w_1, w_2, w_3, ..., w_t\} \tag{2.14}$$

Relevance feedback produces

$$Q' = \{w'_1, w'_2, w'_3, ..., w'_t, w'_{t+1}, w'_{t+2}, ..., w'_{t+k}\} \tag{2.15}$$

Term weights have been modified and k new terms have been added to the query. The purpose of the reformulation is to make the original query vector more like the vectors of relevant documents and less like the vectors of non-relevant documents.

The standard model for relevance feedback in Vector Space was introduced by Rocchio (1971). Using the unrealistic scenario of having complete relevance information (i.e. every relevant document in the collection having been identified and used for feedback), Rocchio defined an optimal query as being:

$$Q' = \frac{1}{R}\sum_{i \in rel} \frac{D_i}{|D_i|} - \frac{1}{N-R}\sum_{i \in non-rel} \frac{D_i}{|D_i|} \tag{2.16}$$

where

$Q'$ = new query vector

$D_i$ = document vector i

$R$ = set of relevant documents

$N$ = set of all documents in the collection

Obviously at the start of a search task we do not know the set of relevant documents. Therefore, in the absence of perfect relevance information Rocchio defined an incremental approach to modify the original query vector based on *known* relevant and non-relevant documents (i.e. those relevant and non-relevant documents that have been identified thus far):

$$Q' = \alpha Q + \beta \sum_{i \in \text{known rel}} \frac{D_i}{|D_i|} - \gamma \sum_{i \in \text{known non-rel}} \frac{D_i}{|D_i|} \tag{2.17}$$

$\alpha, \beta, \gamma$ were used to control the effect of the original query vector, the set of known relevant documents, and the set of known non-relevant documents on the modified query respectively. As can be seen, the modified query is based on evidence from both relevant and non-relevant documents. Salton and Buckley (1990) showed that evidence from relevant documents should be considered more important and this can be reflected in Equation 2.17 by setting $\beta > \gamma$.

Ide (1971) extended Rocchio's work and proposed two new models, *Ide Regular* a variant of Rocchio's model which does not normalise for the number of documents used for feedback and *Ide Dec-Hi* which used only the highest ranked non-relevant document for feedback.

The original vector space model formulation expanded the query by including all terms from relevant documents. This massive expansion increases the overhead associated with processing such a query and may increase the amount of non-relevant material being retrieved unless appropriate term weights were used to emphasise certain significant terms. Harman (1992) proposed a selective expansion approach whereby terms were ordered based on a weighting scheme [1] and only the top terms were added to the query. It was shown that selective query expansion can outperform massive expansion and reduce the overheads associated with the latter.

In the vector space model, query expansion and term reweighting are considered as one process. Expanding a query is the same as assigning a positive weight to terms in the query vector with a previous weight of zero. Removing a term from a query is the same as reducing its weight to zero and reweighting is achieved by either increasing or decreasing the weights of terms. As we will see in the next section, in the probabilistic model, these two process are treated separately from the outset.

### 2.4.2 Relevance Feedback in the Probabilistic Model

As outlined by Robertson (1990), the process of query expansion and term reweighting should be treated separately as they attempt to answer different questions.

[1] Several were investigated and the best performing weighting scheme used a combination of a term's inverse document frequency (*idf*) and its term frequency (*tf*) in relevant documents.

Query expansion attempts to answer the question: *How much will adding this term to the query benefit the query's performance?* Term reweighting attempts to answer the question: *How much evidence does the presence of this term provide for the relevance of this document?*

All terms from all known relevant documents are candidates for expanding the original query. Robertson (1990) proposed the offer weight (sometimes refereed to as a *term selection value*) in order to rank terms for expansion:

$$ow_i = r_i \times rw_i \tag{2.18}$$

where

$r_i =$ the number of known relevant documents in which term $i$ occurs

$rw_i =$ the relevance weight of term $i$

Terms are ranked according to the original offer weight and the top $N$ terms are appended to the original query entered by the user (10-20 is considered a reasonable figure for $N$ (Robertson and Spärck Jones, 1997)).

Having expanded the user's original query by adding the top N terms, term reweighting can be applied to all query terms using the relevance weighting formula at retrieval time:

$$rw_i = \log \frac{(\frac{r_i}{R})(1 - \frac{n_i - r_i}{N - R})}{(\frac{n_i - r_i}{N - R})(1 - \frac{r_i}{R})} \tag{2.19}$$

### 2.4.3 Extensions to Relevance Feedback

Relevance Feedback techniques have been extended in a variety of ways in an attempt to improve the performance of a feedback iteration. Ruthven (2003) discusses some of the major extensions which have been investigated, in this section we will summarise these and some other work on extensions to relevance feedback techniques from the literature.

#### 2.4.3.1 Dependency Between Terms

Both the classical vector space model and probabilistic models are built upon an assumption of term independence, i.e. terms occur independently in relevant and non-relevant documents. This oversimplification of the IR problem has caused many researchers to investigate the exploitation of term dependencies in order to improve performance. Various techniques have been proposed for using term dependence (van Rijsbergen et al., 1981; Smeaton and van Rijsbergen, 1983) however these techniques have not produced a significant increase in performance over models built upon the more simplistic independence assumption.

#### 2.4.3.2 Negative Relevance Feedback

The classical models of information retrieval generally do not make provisions for explicit negative relevance information from the user. Instead, negative feedback is approximated using the entire document corpus less the relevant material. Through their TREC experiments the team at Rutgers University implemented techniques for the incorporation of negative feedback into the retrieval process (Belkin et al., 1997, 1998). Negative feedback has generally been problematic due to issues regarding its implementation (how to use non-relevance information) and the difficulty in deciding non-relevance (when is a document non-relevant).

#### 2.4.3.3 Modelling Dynamic Information Needs

Most information retrieval sessions attempt to satisfy a static user information need. In order to model a search environment where a user has a more dynamic, transient, information need, extensions to the relevance feedback process which integrate timing information have been investigated. In particular Campbell (1995) examined the use of *ostensive* relevance weighting whereby *ageing* of relevance judgments was modelled by giving more weighing to recent relevance judgments over older relevance judgments. Experiments with dynamic information needs can be problematic as most reference test collections are built upon the assumption of a fixed information need although some techniques have been proposed (Allan, 1996).

#### 2.4.3.4 Scaled Relevance Judgments

Ruthven et al. (2003) investigated the notion of incorporating a scale of relevance into the standard relevance weighting formula (Robertson and Spärck Jones, 1976). Through an interactive experimentation, users were able to indicate on a scale of 0 - 10 the extent to which a document matched their information need, this scale was then incorporated into the relevance weighting formula by treating the value assigned as part of a relevance judgment (i.e. 1 being treated as $\frac{1}{10}$ th of a relevance judgment and 10 being a complete relevance judgment).

#### 2.4.3.5 Incremental Feedback

Most relevance feedback techniques are evaluated as a batch process, whereby a large set of relevant documents are fed into the system at the same time when performing feedback. The incremental relevance feedback technique introduced by Aalbersberg (1992) is an attempt to model a situation whereby a feedback iteration is performed after each relevance judgment thereby allowing users to develop their query over successive relevance feedback iterations. Spärck Jones (1979) demonstrated that relevance feedback can provide substantial performance improvements after only one or two relevance documents have been judged. Using an incremental feedback system enables users to benefit from their feedback more quickly than having to wait until they have accumulated a certain number of relevance judgments before seeing the benefit from feedback and therefore this approach may better suit a web surfer's interactions with a search engine. Aalbersberg demonstrated that the incremental feedback approach outperformed the classic vector space models. Allan (1996) experimented with the incremental feedback mechanism for information filtering and found that using only a relatively small amount of documents for feedback incrementally achieved similar results to batch processing of much larger amounts of feedback documents. Iwayama (2000) suggests that the incremental approach works for topics that are well specified, i.e. the relevant documents are similar to each other, rather than topics with many loosely related concepts.

In this section we have discussed relevance feedback, an interactive query reformulation technique which can enable an IR system to better understand a user's information need in order to improve the performance of the search process. In the next section we will outline another technique which can improve the performance of an initial ranking through combining multiple sources of evidence about an information need.

## 2.5 Combination of Evidence in Information Retrieval

In any ranked list there are two types of general errors that can occur: assigning a non-relevant document a relatively high rank, or assigning a relevant document a relatively low ranking (Fox and Shaw, 1994). It has also been shown, through comparative performance of different retrieval models, that there is surprisingly little overlap in the ranked lists returned for the different retrieval methods or different variants of the same method even when the effectiveness of retrieved results were quite similar (McGill. et al., 1979; Croft and Harper, 1979; Harman, 1993). This difference can be attributed to the fact that different retrieval strategies may exploit different aspects of the document set when ranking (Bartell et al., 1994). Merging these different strategies may reduce the errors of retrieval. Combination of evidence (or *data fusion*) techniques attempt to exploit differences in retrieval strategies in order to minimise these errors and improve search effectiveness (Croft, 2002).

Combination of evidence has been applied at different levels in the search process (Croft, 2002). *Document representation* level combination refers to the process of searching over multiple document descriptors such as title, abstracts, authors etc., and this approach has been shown to improve performance over searching based on a single representation (Fisher and Elchesen, 1972; Svenonius, 1986). Combination at a *query level* can be seen as the basis for relevance feedback as more terms are added to the query in order to improve the query's representation of the user's information need. Studies have shown that querys can be improved initially by expanding them with citations (Salton and McGill, 1986) or controlled vocabularies (Crouch et al., 1990). Combining at the *ranking level* has also been explored thoroughly and both Belkin et al. (1995) and Rajashekar and Croft (1995) have shown improvements in combining the outputs from multiple retrieval strategies.

Combining evidence improves performance over a single retrieval run by reducing the errors common to retrieval results and combination has been investigated at different stages in the information retrieval cycle. In order to effectively combine this information, several techniques have been proposed and we will now discuss two commonly used approaches.

### 2.5.1 Similarity Merge

Fox and Shaw (1994) performed fusion of results within the vector space model of retrieval. The authors proposed a number of combination schemes for combining the document scores from multiple ranked lists (see Table 2.1), and were able to show an improvement over any single retrieval runs. Overall the best performing were *CombSUM* and *CombMNZ*.

| Name | Combined Similarity = |
|---|---|
| CombMAX | MAX(Individual Similarities) |
| CombMIN | MIN(Individual Similarities) |
| CombSUM | SUM(Individual Similarities) |
| CombANZ | $\frac{SUM(IndividualSimilarities)}{NumberofNonzeroSimilarities}$ |
| CombMNZ | SUM(Individual Similarities) * Number of Nonzero Similarities |
| CombMED | MED(Individual Similarities) |

Table 2.1: Combination formulas proposed by Fox and Shaw (1994)

CombSUM is a simple sum of the document scores across multiple ranked lists.

CombMNZ is an extension to CombSUM in which the sum is also multiplied by the number of non-zero ranked lists in which the document occurs, thereby favouring documents which occur in all ranked lists being combined.

Although this simple sum-of-weights approach has been shown to be effective, there are cases whereby a simple sum can cause a deterioration in performance over a single run, in particular when one of the inputs has very poor performance (Tumer and Ghosh, 1999). Through their experiments in combining query formulations, Belkin et al. (1993) found that a different query formulation can contribute to improved performance in a combination even when the individual query formulation performance was poor. This suggests that a method cannot be simply discounted because it is a poor performer on its own.

### 2.5.2 Linear Combination

By using a weighed approach to combination, the importance or weight of each ranked list can be specified prior to combination thereby reducing the likelihood that a combined score will be polluted by a poor performing ranked list. A common weighting method is referred to as the *Linear Combination* method (Bartell et al., 1994; Vogt and Cottrell, 1998):

$$p(w, x, q) = \sum_{rankedlists} w_i p_i(x, q) \tag{2.20}$$

Several approaches have been investigated into deciding on the weights to assign each ranked list prior to combining. Thompson (1990) investigated the use of assigning each ranked list a weight based on the prior performance of the system. Bartell et al. (1994) used optimisation techniques in order to determine the best weights based on a training phase.

In this section we have outlined the motivation for combining multiple sources of evidence in information retrieval and explained how combination can be performed at different levels in the information retrieval process. We have then explained two commonly used methods for combining evidence from multiple ranked lists, *Similarity Merge* and *Linear Combination*.

## 2.6 Evaluation

The final stage in the development of any retrieval system is its evaluation. The problem of evaluation in information retrieval is a difficult one which has been researched for many decades (Cleverdon and Keen, 1966; van Rijsbergen, 1979), and is still considered a far from solved problem with new techniques being proposed (Ingwersen and Järvelin, 2005).

The fundamental concept in IR is that of *relevance*, the extent to which a document answers a stated information need. Relevance is a term of many definitions and has been studied extensively (Saracevic, 1999; Mizzaro, 1997). In information retrieval the definition of relevance is one of *topicality*, if a document is on-topic then it is deemed relevant. Relevance is an inherently subjective notion, users may differ on their assessment of the relevance or non-relevance of documents for the same information need. As a result information retrieval is an *uncertain* problem domain which poses difficulty when it comes to evaluating an IR system performance. Despite these problems many techniques and methodologies have been implemented and are aiding the development of effective IR strategies.

In order to evaluate an IR system we first need to understand *what* we want to measure. A retrieval system can be measured in terms of its efficiency (e.g. the system response time for a request), in terms of its ability to satisfy users (i.e. help users satisfy some goal) or in terms of its retrieval effectiveness (i.e. a measurement of the quality of returned documents). In evaluating information retrieval systems there is generally some trade-off between realism, the desire to reflect a real world searching scenario, and control of the experimental setting. Historically IR system's evaluation has generally concentrated on improving retrieval effectiveness using a controlled experimental setting, under the assumption that an improved ranked list is better able to satisfy a user and that if an information retrieval system performs well in a controlled setting it will perform well in an operational situation (van Rijsbergen, 1979). This approach to evaluation has a number of advantages, including control over the experimental environment, allowing for direct comparisons across retrieval strategies, and test repeatability. Although this assumption has been criticised (Ingwersen and Järvelin, 2005), the automatic retrieval environment where the user is taken out of the loop and retrieval performance is measured in terms of a ranked list's effectiveness, has become the standard method for retrieval evaluation.

The work carried out by Cyril Cleverdon and his colleagues at Cranfield represents the first attempt at building a standard evaluation methodology for information retrieval (Cleverdon and Keen, 1966). This methodology, known now as the Cranfield model, consisted of a small collection of documents, a set of test queries, and a set of relevance judgments (a set of documents deemed relevant for each query). This methodology was refined and developed over many years (Spärck Jones and van Rijsbergen, 1975; Harman, 1993).

In this section we will outline techniques and methodologies used to measure retrieval effectiveness in terms of the relevance of documents retrieved. First we will outline some commonly used metrics in information retrieval, following that we will discuss the TREC workshop which has developed into the main workshop for comparative evaluation of retrieval methodologies over the last number of years.

## 2.6.1 Measures of Retrieval Effectiveness

In measuring retrieval effectiveness we attempt to estimate the quality of results retrieved in response to a query in terms of the relevance of the documents contained in the ranked list.

### 2.6.1.1 Precision and Recall

Two classic measures of retrieval effectiveness are *Precision* and *Recall*, these are the most frequently used measures of information retrieval effectiveness and the basis of several more complex metrics. As early as 1966 Cyril Cleverdon and his colleagues at Cranfield (Cleverdon and Keen, 1966) recognised their importance, and they are defined as:

- Precision - the proportion of all retrieved documents that are relevant.
- Recall - the proportion of relevant documents that have been retrieved.

The two measures are complementary and measure different aspects of retrieval performance. Empirical studies have shown a trade-off between the two measures, as recall increases precision often decreases, i.e. as we find more relevant material we also encounter more non-relevant material.

Depending on the domain of usage, some IR users may be more concerned with precision than recall or vice-versa. A web searcher is often interested in good precision in order to find relevant documents without having to sift through many non-relevant documents and, due to the vast redundancy on the web, they are often less concerned with recall. On the other hand a user searching through a patent database will be more concerned with ensuring that they retrieve all documents of relevance (i.e. high recall) and less concerned with precision of the list.

As documents retrieved in response to a query are presented in ranked order of their perceived similarity to the query and a user generally progresses through the ranked list by starting at the top, precision and recall are often combined and shown together on a *Precision/Recall curve.* Precision/Recall curves are monotonically decreasing, reflecting the tradeoff between precision and recall, and they show the precision obtained at different recall levels in the ranked list. Precision/Recall curves demonstrate, graphically, the effort required by the user in order to encounter different amounts of relevant documents as they progress down through the ranked list. Figure 2.5 below shows a typical precision recall curve for two retrieval systems. By looking at this graph we can deduce that system S2 is better for retrieval at the lower end of recall, i.e. towards to top of the ranked list, while S1 shows better retrieval performance at higher levels of recall.

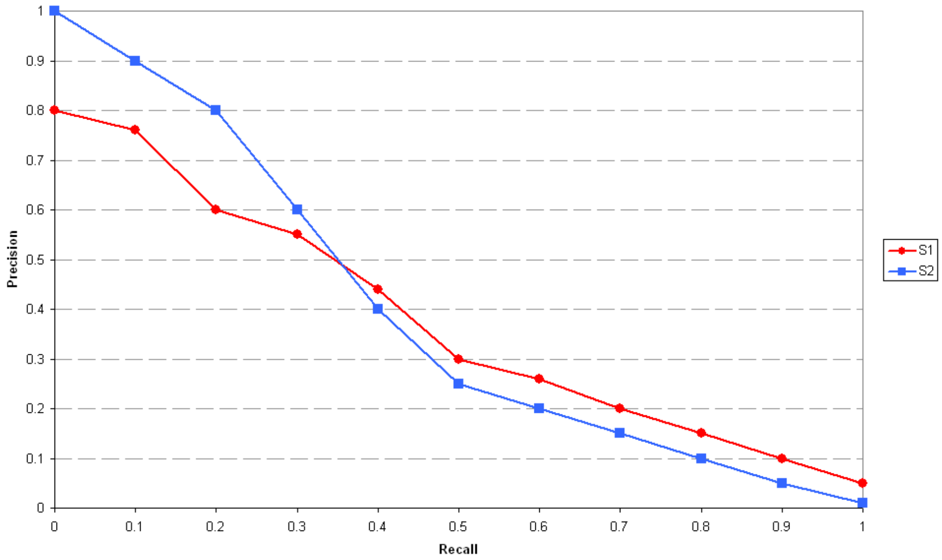


Figure 2.5: Example of a Precision/Recall curve

Experiments in information retrieval are generally run over a set of test questions or *topics* in order to evaluate the system's response to a number of different retrieval scenarios. To facilitate averaging the Precision/Recall graphs across a number of topics, precision values are interpolated to one of 11 standard recall points which range from 0 to 1 in increments of 0.1 and different techniques exist for this interpolation, (van Rijsbergen, 1979).

#### 2.6.1.2 Single Value Measurements of Performance

Although Precision/Recall curves allow us to evaluate the performance of retrieval strategies as we move down the ranked list, it is often desirable to represent a retrieval algorithm's performance using a single figure. Several measures have been proposed which combine the two measures of precision and recall into a single figure (van Rijsbergen, 1979; Baeza-Yates and Ribeiro-Neto, 1999).

Average Precision (or $AP$) is calculated by averaging the precision values at each relevant document encountered in the ranked list:

$$AP = \frac{\sum_{RelDocs} \text{Precision at Rel doc}}{\text{Total No. Of Rel Docs}} \tag{2.21}$$

The measure favours retrieval strategies which retrieve relevant documents high in the ranked list. *Mean Average Precision* or (MAP) is the average of the AP values for each query and is the most commonly used measure in information retrieval evaluation:

$$MAP = \frac{\sum_{Queries} \text{Average Precision for Query}}{No.of Queries} \tag{2.22}$$

Search engine designers may be more interested in the precision of the ranked list at certain points, than the quality of list overall. For this reason, another popular single figure metric is P@X which measures the precision of a ranked list at the rank position, $X$. The value of $X$ may be set to any point in the ranked list, typical IR evaluation uses standard measures such as P@5, P@10, P@30.

### 2.6.2 TREC: The Text Retrieval Conference

The experiments conducted at Cranfield (Cleverdon and Keen, 1966) emphasised the importance of having a standard test collection and in using this collection for comparative evaluation (Harman, 1993). For three decades research in IR was evaluated using the Cranfield Collection, and later the CACM collection (Fox, 1983) and the NPL collection (Spärck Jones and Webster, 1979). However even with the availability of these standard collections, comparative evaluation of search systems across research centres was difficult due to systems being evaluated on different collections and no concerted effort being made to standardise the test methodology. With the growing amount of information being searched in IR systems there was a real need to reflect these changes in scale in the evaluation methodology. The Text REtrieval Conference (TREC) was established in 1992 under the auspices of the National Institute for Standards and Technology (NIST), in an attempt to promote research into information retrieval. TREC extended the Cranfield model in order to address these critical "missing elements" (Harman, 1993) by providing a framework for comparative evaluation of retrieval strategies using a more realistic document collection size.

#### 2.6.2.1 Tracks

For each TREC a set of tasks or *tracks* are devised, each evaluating different aspects of the retrieval process pertinent to contemporary information retrieval. The first TREC consisted of just two tracks, the adhoc track, the standard retrieval environment where a query is issued and the ranked list of results evaluated, and the routing track where a profile is built of a user's information need and documents arriving in a stream are accessed as to their relevance. The most recent TREC, TREC 2007, consisted of 9 tracks including a genomics track (study of retrieval of genomics data) and a SPAM track (study of SPAM filtering approaches) and since its inception TREC has run a total of 17 tracks.

#### 2.6.2.2 Document Collection

The document collections at TREC have been growing steadily since its inception from 2-3 gigabytes initially, to over 400 gigabytes with the GOV2 collection. Document collection sizes vary from track to track depending on the requirements. For example the Terabyte track introduced in TREC-2004 operated over the complete GOV2 collection of 400 gigabytes of text, whilst the interactive tracks from TREC 6 – TREC 8 operated on a collection size of just over 210,000 documents. Documents can come from a variety of sources including newspapers (the *Financial Times*, and *LA Times*), web documents, and emails.

No stopping or stemming of documents in TREC is performed on the collection (see section 2.2.1) and each document is formatted using a standard SGML markup to allow for easy parsing. An example document from the Financial Times collection is shown in Figure 2.6.

```
<DOC>
<DOCNO>FT923-1051</DOCNO>
<PROFILE>_AN-CIYB1AGIFT</PROFILE>
<DATE>920925
</DATE>
<HEADLINE>
FT  25 SEP 92 / UK Company News: Appleyard helped by exceptional
</HEADLINE>
<TEXT>
APPLEYARD Group, the North Yorkshire-based motor dealer, reported pre-tax
profits ahead 16 per cent in the six months to June 30.
The outcome of Pounds 1.75m compared with Pounds 1.51m in the first half of
last year and was only marginally short of the depressed Pounds 1.81m
achieved in the whole of 1991.
However, the latest figure was struck after an exceptional profit of Pounds
1.37m relating to the disposal of freehold property in Leeds, and interest
charges reduced to Pounds 1.85m (Pounds 2.65m).
Turnover fell to Pounds 158.1m (Pounds 218.9m).
Mr Mike Williamson, chairman, said the new car market during the period was
4 per cent below last year, but the group had improved overall margins.
Contract hire and leasing lifted profits by 15 per cent and the commercial
vehicle side performed 'extremely well' in a weaker market.
Referring to sales in August, Mr Williamson said overall volumes were
largely unchanged with the notable exception of Audi/VW where national
volumes were down 23 per cent on August 1991.
The interim dividend is maintained at 2.6p, uncovered by earnings of 2.2p
(2.1p) per share.
</TEXT>
<PUB>The Financial Times
</PUB>
<PAGE>
London Page 22
</PAGE>
</DOC>
```

Figure 2.6: A sample document from the Financial Times collection showing the TREC SGML markup

```
<top>

<num> Number: 301
<title> International Organized Crime

<desc> Description:
Identify organizations that participate in international criminal
activity, the activity, and, if possible, collaborating organizations
and the countries involved.

<narr> Narrative:
A relevant document must as a minimum identify the organization and the
type of illegal activity (e.g., Columbian cartel exporting cocaine).
Vague references to international drug trade without identification of
the organization(s) involved would not be relevant.

</top>
```

Figure 2.7: A sample topic from the TREC collection

#### 2.6.2.3 Topics

Distributed with each test collection is a set of topics. TREC differentiates between topics, which are statements of an information need, and queries, the text string submitted to a search engine, as the translation from statement to query is considered an integral part of the retrieval process (Baeza-Yates and Ribeiro-Neto, 1999). Topics are generally devised by a TREC *assessor* based on a bona fide information need, and the assessor who devises the information need is generally the person who performs the relevance assessments for that topic (this procedure is discussed in the next section). Different numbers of topics are used according to the needs of the track, for example the manual and automatic adhoc track in TREC-6 consisted of 50 topics while the interactive track used just 6.

A TREC topic generally consists of four sections; an identifier, a title, a short description, and a longer narrative. An example topic description is shown in Figure 2.7.

#### 2.6.2.4 Relevance Judgments

An integral part of any test collection is a set of relevance judgments for each topic. These judgments consist of documents whose relevance to a search topic has been assessed and recorded. Generally TREC relevance assessments are binary, i.e. a document is either relevant or not relevant to a topic. Relevance is an inherently subjective and dynamic notion, the perceived relevance of a document can vary across users and a user's notion of relevance can vary through interacting with a document collection (Rees and Saracevic, 1966). This has led many critics to question the validity of IR evaluation methods which use relevance judgments (Harter, 1996; Cuadra and Katter, 1967; Taube, 1965). However despite these issues, even with differences in relevance assessments, the comparative effectiveness of document retrieval strategies has been shown to remain stable (Voorhees, 1998).

Relevance judgments in early text collections were complete, i.e. each document was assessed for relevance for each topic. However, with the growth of collection sizes in TREC collections this procedure became infeasible. Instead, TREC employs a technique known as *pooling*, devised by Spärck Jones and van Rijsbergen (1975), in order to forge a subset of the collection to assess for relevance. The pooling procedure generally proceeds as follows; having completed their TREC experiments each participating group returns their results for each topic in the form of ranked lists of documents (generally to a depth of about 1000 documents). For each topic, the top 100 or so documents are taken from the top of each group's results for that topic and a union of all the documents are taken with duplicate documents removed. This union is the pool of documents that will be assessed by the TREC assessor (the person who composed the topic) for relevance. This method of assessment is based upon two major assumptions; (1.) that the majority of relevant documents will be contained in the pool, and (2.) that documents not in the pool are considered to be non-relevant. Although critics have questioned the validity of the pooling method (Harter, 1996; Blair, 2002) several studies have supported its use (Buckley et al., 1994; Zobel, 1998; Voorhees and Harman, 1999; Keenan et al., 2001).

This set of relevance judgments is known as “qrels” and is made available to groups allowing them to perform their own evaluations on the standard TREC collection.

### 2.6.3 Evaluation of Relevance Feedback Systems

When evaluating a relevance feedback strategy, typically a measure of performance (for example MAP) is taken on the ranked list before and after feedback is performed. However, as relevance feedback generally proceeds by changing the query formulation to *look* more like the relevant documents, this reformulation generally causes the documents used for feedback to be pushed to the top of the ranked list improving precision and recall figures substantially. This is not considered to be a true reflection of the benefit of relevance feedback as pushing documents already seen to the top of the list does not benefit a user’s search (Salton and Buckley, 1990).

Chang et al. (1971) describes three ways to overcome this so called *ranking* effect associated with the evaluation of relevance feedback systems: residual ranking, frozen ranking, and test and control groups.

#### 2.6.3.1 Residual Ranking

Residual ranking operates by measuring the effects of feedback after the documents used for feedback have been removed from the new ranked list. This procedure will overcome the ranking effect from evaluations but it has the disadvantage that the ranked lists before and after feedback are not comparable as the new ranked list will have fewer relevant documents. Salton and Buckley (1990) overcame this problem by removing the feedback documents from the ranked list after feedback and the ranked list before feedback. Another disadvantage of this method is that after a certain number of iterations the score associated with these ranked lists (e.g. their MAP) will reach zero, because all the relevant documents will have been removed. When averaging the performance over a number of queries, certain queries will drop out of the calculations (as they reach zero) and therefore the performance figures in later iterations will be based on fewer queries, reducing stability.

#### 2.6.3.2 Frozen Ranking

In the frozen ranking (or *freezing*) approach to relevance feedback evaluation, when performing feedback, the position of the documents, used for feedback, in the ranked list are frozen before and after the feedback iteration. In this way only the ranks of unseen documents can change during the feedback cycle thus removing the ranking effect. One issue with the frozen ranking method is that as the number of feedback iterations increases, the effect of feedback on the unseen relevant documents can be masked as more of the relevant document set becomes frozen.

#### 2.6.3.3 Test and Control Groups

In this method of evaluation the document collection is split into two, a *test* and *control* group. Relevance feedback is performed on the test group in order to produce a modified query formulation which is then run on the control collection. In this way no attempt has to be made to freeze or remove documents used for feedback as the *ranking effect* is overcome by evaluating the modified query on a separate collection. However care has to be taken when splitting the collection in order to ensure that the control group is representative of the test group. For example errors could occur if the documents used for feedback from the test group were not similar to the relevant documents from the control group.

### 2.6.4 Summary

In this chapter we have provided a background for our work by outlining the concepts and techniques related to information retrieval. We began in section 2.1, by providing a historical overview of IR, highlighting the key milestones. We then outlined the major components of a typical information retrieval system: document preprocessing, the user interface, and the retrieval process. We then detailed the three classical models of retrieval in section 2.3: the Boolean model, the vector space model, and the probabilistic model, and also outlined developments in term weighting schemes which form the basis for best-match retrieval. Following this, in section 2.4 we outlined the relevance feedback process, describing its application in both the vector space and probabilistic models. In section 2.5 we outlined another method for improving retrieval effectiveness, through the combination of multiple sources of evidence. We described the different stages in the IR system whereby evidence combination can occur and the common techniques used to combine multiple ranked list outputs. Finally in section 2.6 we outlined the final stage in the development of a retrieval system, its evaluation. We discussed the problems with IR evaluation and described the techniques used in both standard IR evaluation and relevance feedback evaluation.

In the next chapter we will introduce the area of Collaborative Information Retrieval, an emerging research area which is built upon the foundations of information retrieval research.

# Chapter 3

# Collaborative Information Retrieval

## 3.1 Introduction

Collaborative Information Retrieval is a concept concerned with the human-human collaboration which occurs during the information retrieval (IR) process. Although researchers suggest that information retrieval has always been a social process (Root, 1988; Wilson, 1981; Romano Jr. et al., 1999), the majority of research into information retrieval has focussed on information retrieval for the individual. Romano Jr. et al. (1999) referred to this situation as the *IR-Paradox*. Twidale and Nichols (1998), suggest that systems which actively support formal and informal collaborations are likely to prove more useful and usable to users than current systems whose designs are based upon the principles of single user usage and are often used collaboratively regardless.

Collaborative information retrieval can be seen as an emerging research domain which combines the two more stable research areas of information retrieval and groupware. As defined by Dix et al. (1998), groupware refers to “applications written to support the collaboration of several users”. Groupware systems are generally described in terms of their position in the two-dimensional space of time and place as originally proposed by Ellis et al. (1991), and shown in Figure 3.1. In this chapter we will refer to this categorisation when describing work in collaborative information retrieval. Groupware tools have traditionally been used to support asynchronous communications in the form of email and usenet systems. Recently however we have witnessed the emergence of tools to support more realtime, synchronous collaborative tools in the form of instant messaging, voice over IP and video conferencing.

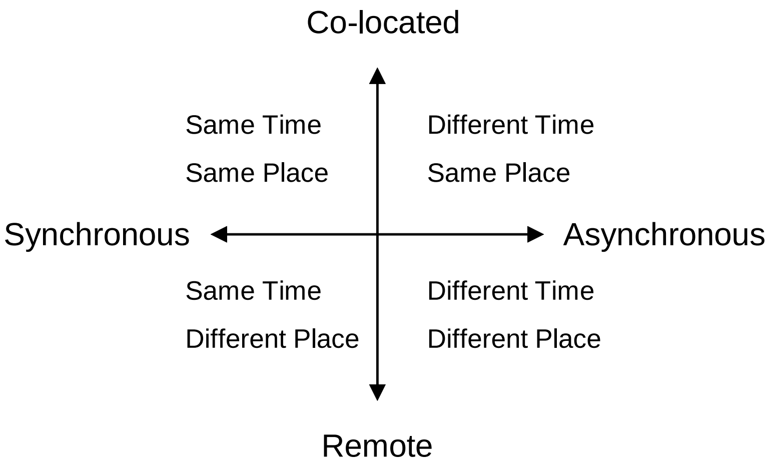


Figure 3.1: The two dimensions of time and place used in Groupware classification

The phrase *Collaborative Information Retrieval* has been used in the past to refer to many different technologies which support collaboration in the IR process. Like traditional groupware tools, much of the early work in collaborative information retrieval has been concerned with asynchronous, remote collaboration via the reuse of search results and processes in collaborative filtering, collaborative re-ranking, and collaborative footprinting systems. These systems support *implicit* collaboration as users are not directly collaborating with each other but are implicitly helping others via their interactions.

Synchronous collaborative information retrieval (SCIR) systems are concerned with the realtime, explicit collaboration which occurs when multiple users search together to satisfy a shared information need. These systems represent a significant paradigmatic shift in information retrieval systems from an individual focus to a group focus. SCIR can take place either remotely, or in a co-located setting. These systems have gained in popularity and now, with the ever-growing popularity of the social web (or *Web 2.0*), support for synchronous collaborative information retrieval is becoming more important than ever.

In this chapter we will provide a comprehensive account of research to-date in collaborative information retrieval. We begin with a look at research studies that have outlined various collaborative activities that occur in information retrieval tasks. These studies can be seen as a motivation for research into collaborative information retrieval. Following this, we will discuss research to-date in asynchronous collaborative information retrieval. We will then describe the emerging field of synchronous collaborative information retrieval. We conclude the chapter with a summary and analysis of existing collaborative information retrieval research.

## 3.2 Information Retrieval: A Social Process

Researchers claim that IR has always been a social process (Root, 1988; Wilson, 1981; Romano Jr. et al., 1999) and studies have shown how collaboration is commonplace throughout the information retrieval process, despite little provision being made for collaborative tools within IR systems (Morris, 2007). The collaborative nature of the information retrieval process has been investigated in a number of domains including: academia, industry, medicine, and the military (Foster, 2006).

Twidale et al. (1997) observed collaboration in activities around 11 OPAC computer terminals in a university library. In the study they found that 10% of all uses of the terminals were of a collaborative nature including: multiple users working around the same terminal in a problem solving task and pointing and gesturing at the screen, users working on adjacent terminals and coordinating their actions, and users asking others questions such as “How do you do that?”. They noted that this rich collaboration occurs despite the fact that talking in the library was discouraged and that terminals and information systems were designed for single usage.

In a study of information sharing behaviour across four academic disciplines, Talja (2002) demonstrated that collaborative information retrieval is as commonplace as individual or solo information retrieval.

Poltrock et al. (2003), studied collaborative information retrieval activities in design teams at Boeing and Microsoft. Based on interviews, observations and records of meetings and emails, they concluded that all stages of the information retrieval process (identifying an information need, query formulation, retrieving information), can be performed individually, or as part of a small adhoc group, or as part of a large group in a meeting environment. They found that the most common source of information was people and that task division in information retrieval was commonplace.

In a study of the activities involved in resolving a patent application, Hansen and Järvelin (2005) observed that collaboration amongst patent engineers was common in most phases of the patent task with information seeking the most common. They noted that overlap was common across patent searches and this led to sharing of information. The types of information sharing included the sharing of documents, queries, and opinions. Hansen and Järvelin (2005) also observed patent engineers cooperating on work tasks, dividing the task amongst colleagues and sharing search strategies.

In a survey of web search usage amongst workers at a large corporation, Morris (2007) found that collaboration was commonplace in web search despite it not being supported in search systems. In the survey, over 97% of all users reported having used some form of collaboration when searching the web. This included: 87.7% of people saying that they had looked over someone's shoulder while they searched and suggested query terms, 30.4% of people who said that they had used instant messaging (IM) to coordinate a real-time search, and 23.5% who had used a large display to perform a web search during a group meeting. For typical group size for collaboration: 80.7% of respondents had collaborated in groups of two, with 19.3% reporting a group size of three or four (no larger group sizes were reported). When multiple users collaboratively searched an information space together using separate devices, Morris (2007) identified two common task division strategies: (1.) *Divide and conquer*, where the coordination of the task involved assigning separate subtasks to each individual, which was used by 19.3% of respondents, and (2.) the *brute-force* approach where no coordination took place and users searched separately was used by 24.8% of respondents. The common types of task for which collaboration took place were purchasing items, booking travel, and researching property.

From these different studies, and others reported in the literature (Foster, 2006), we can conclude that collaboration is common in information retrieval, this collaboration can occur at any stage of the IR process, and that collaboration occurs *in spite of* a lack of support for these activities in most IR systems. In the following sections we will outline research to-date into tools and techniques which support collaboration in the information retrieval process.

## 3.3 Asynchronous Collaborative Information Retrieval

The vast majority of research to date in collaborative information retrieval refers to asynchronous collaboration, where users' search processes and outcomes are reused in order to improve the effectiveness of a new user's search. These systems represent an implicit, passive form of collaboration. The focus of these systems is to improve an individual's search process, based on information garnered from others', often without their knowing. In this section we will review work to date in asynchronous collaborative IR systems, in particular we will describe:

- *Collaborative filtering systems* – which filter documents based on the recommendations of others.
- *Collaborative re-use of search processes* – which inform users of search techniques used previously by similar users.

- *Collaborative footprints* – where users can see the paths taken by previous users through the information space.
- *Incorporating social networks into the retrieval process* – where an existing community of like-minded users is modelled and this model exploited to focus search results.
- *Web 2.0* – where recent advances on the web have facilitated the formation of communities of interest.

We start the review with the most researched and mature collaborative information retrieval systems, collaborative filtering systems.

## 3.3.1 Collaborative Filtering

Collaborative filtering (CF) systems (often referred to as *Recommender systems*) are the most mature and stable of all collaborative information retrieval technologies. CF systems are an extension of traditional, single-user, information filtering systems which have been developed and studied extensively, and are an attempt to circumvent many of the problems these systems face (Hanani et al., 2001).

### 3.3.1.1 Information Filtering

The purpose of an information filtering system is to filter a stream of documents and to find documents relevant to a user according to their *user profile.* Information retrieval and information filtering are two closely related disciplines which use much of the same technologies causing some researchers to consider them as "two sides of the same coin" (Belkin and Croft, 1992). The main difference between IR and IF is the characteristics of the dataset on which they operate. An information retrieval system is characterised by a static document collection and dynamic queries, while an information filtering system is characterised by a dynamic document collection (i.e. a constantly changing stream of new documents like news feeds or email) and long standing user queries (their user profile).

The fundamental component of an information filtering system is a *user profile*. This profile could be a vector of weighted terms, which describe a user's information need gained through the relevance feedback process, it could consist of concepts taken from a structural and hierarchial ontology (Chirita et al., 2005), or a user may possess several profiles, one for each domain of interest (Liu et al., 2002). Once constructed, a user profile can be used to find documents in a stream relevant to a user's interests.

Despite the success of information filtering systems, they can suffer from several problems in their implementation. Practical issues such as a mismatch between terms used in a user's profile and terms in a document due to synonymy and polysemy can result in an IF system filtering out potentially relevant material. IF systems are purely *content-based* systems, which rely on being able to extract content from the items they wish to filter, and, as such, cannot filter items where content analysis is difficult, such as music and videos. Furthermore IF systems' content analysis is shallow and therfore they are not able to filter documents based on aspects such as the quality or authority of a document.

#### 3.3.1.2 Collaborative Filtering: Incorporating the Opinions of Others

Collaborative filtering systems were introduced in an attempt to overcome some of the difficulties associated with information filtering systems by incorporating the opinions of others into the filtering process for a user. As shown in section 3.2, people often develop their views through discussions with others. Collaborative filtering systems are an attempt to automate this "word-of-mouth" (Shardanand and Maes, 1995), by allowing users to recommend items to others. Rather than basing their opinions on a small subset of people in a social circle, collaborative filtering systems enable people to gain opinions from thousands of like-minded people from around the world (Schafer et al., 2007). CF systems are built upon the assumption that users who rated items similarly in the past will rate items similarly in the future (Resnick et al., 1994). Collaborative filtering systems build large user-item ratings matrices, and they compute user-user similarity in order to find the most similar users to a user and then calculate the ratings for the unknown item based on the ratings for this item from these similar users. A rating can be a scalar rating incorporating the degree of support for an item, such as giving a movie 5 stars; it can be a binary judgment, such as agree/disagree; or it can be a unary rating such as a record that a user has purchased an item (Schafer et al., 2007). It could even be a free textual phrase such as "check out this cool article" (Goldberg et al., 1992).

As humans do not share the same problems of synonymy and polysemy as computers (Konstan et al., 1997), items which are semantically similar but may not share the same terms can be recommended. In addition, Herlocker et al. (1999) outlined the advantages of collaborative filtering over a pure content-based filtering approach as:

- It enables filtering for items whose content cannot be easily analysed, like music and movies.
- It allows for rating an item on aspects such as its quality or authority.
- It can generate serendipitous recommendations, unlike pure content-based systems which simply show "more of the same".

The Tapestry system developed at Xerox Palo Alto Research Centre (PARC) (Goldberg et al., 1992) represents the first example of a collaborative filtering system. Tapestry was developed as a replacement for a traditional email client and was motivated by the belief that information filtering can be more effective when humans are involved in the filtering process. Users of Tapestry could recommend documents to others explicitly, by providing annotations on documents (e.g. "excellent reference manual"), or implicitly, by replying to documents. These recommendations could be accessed by other users' mailbox filters (e.g show me "documents replied to by O'Reilly" or "show me documents replied to by O'Reilly and annotated with "excellent""). In order for Tapestry to work, users needed to manually select those users whose opinions they trust and as such these systems were only applicable to small-scale filtering.

In order to develop large-scale filtering systems, researchers began investigating ways to automate the generation of recommendations. Modern collaborative filtering systems automatically generate ratings for items by:

1. Finding users who are most similar to the current user in terms of their previous predictions.

2. Generating a prediction for an item based on the ratings for the item from these similar users.

The most commonly used approach for generating a prediction is called the *nearest neighbour* method which calculates a rating for an item based on the ratings for that item from the most similar users, or "neighbours", to the current user. Rather than simply averaging the ratings for an item across all similar users, a significance value is normally attached to each prediction in the form of a weighting and this weighting is generally the similarity between the user for whom we are generating the prediction and the neighbour who is providing the prediction value.

GroupLens was the first example of an automatic collaborative filtering system which implemented the nearest neighbour approach (Resnick et al., 1994; Konstan et al., 1997). GroupLens was developed as a filter for Usenet news, a document source with a low signal-to-noise ratio (i.e. lots of articles but only some relevant to a user) and therefore a domain which could benefit greatly from the application of effective information filtering systems. As the authors claim "people read news articles and react to them, but those reactions are wasted", GroupLens was an attempt to utilise this feedback in order to generate more meaningful predictions. Feedback in GroupLens was captured through a ratings system, where users were asked to rate items on a scale of 1-5.

Several other measures for calculating the similarity value were introduced and evaluated in Shardanand and Maes (1995) experiments with "Ringo", a music recommender system. Ringo could suggest new artists/albums for users to listen to, list the artists/albums that it predicted users would dislike and generate ratings for specific albums or artists. Unlike the GroupLens system which considered all users' ratings (weighted by their similarity), Shardanand and Maes (1995) experimented with different thresholds for the number of ratings to consider when generating a prediction.

Collaborative filtering systems have gained much popularity since their introduction to the community in a special issue of Communications of the ACM (Resnick and Varian, 1997) and have received much attention in recent times through their use in many e-commerce websites of today. Commercial websites such as Amazon (Amazon, 2007), and MovieFinder (MovieFinder, 2007) recommend items to users based on their purchasing and browsing history. Last.fm (Last.fm, 2007) and MusicStrands (MusicStrands, 2007) record songs played on a user's computer in order to recommend new artists to them.

Experiments with collaborative filtering systems have highlighted some issues with their implementation. These issues relate to both the *sparsity* of ratings and the *scalability* of the systems (Sarwar et al., 2001). Shardanand and Maes (1995) reported that in order for useful recommendations to be generated, a "critical-mass" of ratings was needed. In addition, few users of systems are willing to provide ratings on items. Early adopters of the system may find poor performance (known as the *early-rater problem* (Dieberger et al., 2000)). In order for an item to be recommended it needs to be rated first and in large scale recommender system like those in use in Amazon or CDNOW, users will have rated well under 1% of all items. The computational complexity of neighbourhood-based systems grows as the database of users grows, and therefore in a web based system a nearest neighbour approach is infeasible.

Researchers have worked to extended the basic principles of collaborative filtering systems. The early techniques of generating predictions based on the nearest neighbour approach are referred to as *memory based* collaborative filtering systems, in order to compute a prediction the system needs to store all previous predictions. *Model based* collaborative filtering mechanisms, instead, model the process of generating predictions in a probabilistic framework and try to compute the expected value of a user's predictions given their previous ratings on other items. In order to construct these models, various machine-learning techniques are used (Sarwar et al., 2001). By computing these models off-line, model based collaborative filtering systems can overcome much of the problem of scale associated with memory-based systems.

Techniques have been proposed to deal with the sparsity of ratings. Konstan et al. (1997) and Morita and Shinoda (1994), proposed the use of implicit ratings such as time taken to read an article, and Terveen et al. (1997) used the appearance of a URL in usenet postings as an implicit recommendation. As a means of gaining more judgments, other sites use incentives for users to provide ratings (Schafer et al., 2007).

Fab (Balabanović, 1997; Balabanović and Shoham, 1997), was a system which combined content-based and collaborative filtering in an attempt to gain the advantages of each while "inheriting the disadvantages of neither". Unlike collaborative filtering systems which usually represent items by an identifier, Fab represented text documents by their words modelled in a vector space. A user's profile was also represented as a vector of words from documents rated by the user. The user profile served both as a means to find relevant documents and as a means to find similar users through a proxy known as a *collection agent*. Users were matched to collection agents based on the correlation of their profile with the collection agent's profile and this enabled many users to get recommendations from a single collection agent. Items recommended to a user were further filtered by their own *selection agent* which contained their own profile. Several other approaches for combining content based and collaborative approaches have been proposed in the literature (Pazzani, 1999; Melville et al., 2002; Basilico and Hofmann, 2004; Utiyama and Yamamoto, 2006).

### 3.3.2 Collaborative Re-use of Search Processes

In this section we will describe research into the re-use of users' search techniques and processes in order to inform a new user's search. As a search process is an expensive operation, reusing search processes enables users to benefit from the efforts of others (Hust, 2004).

Based upon their observations of information retrieval in a traditional library setting, Twidale and Nichols (1998) designed Ariadne, a tool which allowed for the visualisation of a search process in order to facilitate communication between searchers and librarians. In Ariadne a search process was divided into several levels: database selection, queries, and results. Each of these processes was represented by a card. Cards could be annotated by the searcher and could be grouped together and this enabled an expert, for example a librarian, to quickly spot rookie mistakes such as using inappropriate query terms.

AntWorld (Kantor et al., 2000; Menkov et al., 2000) was a "browsing assistant" which attempted to match a user's information need against similar search histories in the system database. When using the system, users were required to provide "explicit statements of their information need" (Kantor et al., 2000), which could be a few short keywords describing their need or, a longer narrative. This information, along with any documents deemed relevant (feedback was obtained via a "Judge" option displayed on a document) was captured in a *quest profile*. The system was built upon the vector space model and operated by finding quests that are similar to the user's quest and suggested documents to the user that had been judged relevant for these similar quests. Ranked lists returned to users were modified and "ant-mark" symbols were displayed beside recommended pages. These recommended pages were also shown in a list, ordered by perceived relevance, in a drop-down menu.

Glance (2001) described the Community Search Assistant for web search. The system used graphs to illustrate related queries by using links between them according to the amount of document overlap in their search results; two queries were related if they shared a common URL in the top 10 results. A user searching the web using the community search assistant was presented with the queries most similar to their query, ordered by the amount of result overlap. Clicking on a query displayed the results associated with this search and another set of related queries. By following links in this way it was proposed that users could navigate the query graph and find more relevant material.

Unlike the methods above, which use previous queries to recommend potentially relevant material to a new user, both Fitzpatrick and Dent (1997) and Hust (2004) proposed methods to use previous queries directly in the search process in order to improve query performance. Both use the TREC corpus in order to 'simulate' previous queries and were based on the vector space model of retrieval. Fitzpatrick and Dent (1997) performed experiments with the TREC corpus where those queries similar to a particular query were identified based on the overlap of documents across ranked lists. These similar queries were used as a basis for query expansion by extracting significant terms from the top documents returned for each query. Hust (2004), in an attempt to improve the performance of ad hoc retrieval, proposed a series of query expansion methods based on reusing previous similar queries and relevance judgments from other users. Experiments with several test collections showed performance gains when compared with a pseudo relevance feedback approach.

Lately we have begun to see systems exploiting the re-use of search histories in order to promote the ranking of documents deemed relevant to a community of like-minded users, under the title of "social search". The I-Spy community search system (Smyth et al., 2005), exploited "query repetition" and "selection regularity" in users' search behaviour in order to build a system which employed collaborative re-ranking of search results based on the interactions of similar users with the search engine. Their work showed that the amount of term-overlap in queries to a general search engine is high and that the amount of overlap increases when the domain of search becomes more specialised (e.g. a medical search engine). A collaborative re-ranking mechanism was used to promote documents returned from a search engine based on the number of times a document was visited for similar previous queries.

In I-Spy, in order to generate a community ranking, a user needed to explicitly join a group of like-minded individuals. PERCIRS (PERsonalised Collaborative IR System) from Naderi et al. (2007) was an attempt to automate this process of finding similar users in the search process. When a user submitted a query to PERCIRS their user profile was compared with the user profiles of all other users, and several methods were proposed for this calculation. The documents deemed relevant for queries from these users were given a collaborative rank based on the similarity of the users, their queries, and the degree of relevance of the document to the query.

### 3.3.3 Collaborative Footprints

The motivation behind collaborative footprint systems is in the observation of the social phenomenon that people like to follow the crowd. For example, during a vacation it is common for tourists to view a restaurant, bustling with customers, as an indication of its quality. When browsing bookshelves in a library, dog-ears on book pages show that the book is popular. In this section we will describe systems which attempt to leverage this "trail of breadcrumbs" left by previous searchers in order to provide cues to aid a user's search and browsing experience. As early as 1945, Vannevar Bush in his famous article "As we may think" (Bush, 1945) proposed that people might share their trails through an information space. In this way the history of users' searches is not used to recommend content but instead is used to contextualise a search (Wexelblat and Maes, 1999).

Footprints (Wexelblat and Maes, 1999) was an attempt to capture the rich interaction history of real world objects in the digital domain. Footprints recorded traces of users' navigation in an information space and used *maps*, *paths* and *signposts* in order to provide different visualisations of the browsing history of several users. Maps visualised traffic through a website, pages that had been viewed by other users were displayed as nodes and navigation between pages shown as links between these pages. Paths showed the routes followed by others in the past. Annotations showed the percentage of users who follow each link and signposts were comments which users could enter either on individual queries or on a page.

Ahn et al. (2005) developed an application which provided social cues for a community of searchers on a standard search results page. For each search result the conventional search result display was enhanced with social cues indicating the amount of traffic on the page, the annotations on a page, and a thermometer indicating the like or dislike for a page from the community. The system made strong use of colours with darker colours indicating higher traffic, and foreground and background colours were used to differentiate between a user's navigation history, and the navigation history of the community.

The community re-ranking technology of I-Spy (Smyth et al., 2005) and group navigation support of Knowledge Sea II (Brusilovsky et al., 2004) were combined into ASSIST, a system to support "social space traversal" (Farzan et al., 2007). The ASSIST technology was used to alter the results returned by the ACM search engine through community re-ranking and by providing social navigation cues. The social cues provided beside each document on the ranked list were:

- *Popularity* – an icon indicating the number of times the page was selected for a particular query.
- *Related query* – a list of other queries that were used to retrieve a document.
- *Recency* – an icon indicating the last time the document was selected.
- *Footprint* – an icon indicating the browsing popularity of an article (the most viewed article for a community would be 100% filled).
- *Annotation* – an icon indicating the presence of annotations on a document.

Social cues were also provided on links to articles when users elected to browse the directory structure of the ACM rather than entering a direct search. Through a study of university students using the system for a literature review, the authors concluded that the community features provided important support for the students seeking information.

### 3.3.4 Incorporating Social Networks into the Retrieval Process

Researchers have attempted to incorporate an explicit model of a user's social network into the information retrieval process. The motivation behind these systems is in the observation that social relationships already exist and these systems attempt to leverage this community of users in order to improve the retrieval process. These social networks are generally viewed as graphs with nodes representing users and links between them representing their associations. There are several ways of building these graphs. Users can explicitly enter a list of their closest colleagues, or their email logs can be mined, or these graphs can be constructed based on mining publicly available web documents for co-occurrence of names. Once the graph is built IR systems can exploit this information in the search process.

Referral Web (Kautz et al., 1997) was an example of a system which attempted to model a user's social network in order to make search engine results more focussed. In Referral Web, when a user joined the system, a search was made on their name in order to find documents in which their name occurs. Co-occuring names on these pages were then extracted and a further search was made. In this way a graph was automatically built up, based on implicit knowledge publicly available on the web. This graph could then be used in order to guide searches for both people and documents. An example query to Referral Web could be of the form: "What colleagues of mine know about the Java programming language?".

In an attempt to combine social networks and information retrieval, Kirsch et al. (2006) applied the commonly used PageRank graph based technique to a social network. As PageRank uses linkage analysis on web documents to assess the quality of a webpage, applying PageRank to a social network allows for an assessment of an author's standing in a social network. This "social standing" can then be applied to documents authored by this user in a retrieval process in the same way that PageRank is used in a web based scenario. As outlined by Kirsch et al. (2006), in order for such techniques to work there needs to be a social network present in the domain or one which can be readily inferred. One such domain was the co-authorship of documents from 25 years of SIGIR (Special Interest Group on Information Retrieval) conference proceedings. Applying PageRank to this domain, the authors were able to calculate an authority score for each author. In order to evaluate the effectiveness of incorporating social networks into the information retrieval process, a study was conducted using a mailing list archive, from this list a network was constructed where users were associated with messages and replies to messages were modelled as links between users. Using an evaluation based on *known-item retrieval*, where users are searching for a specific known document, the authors conclude that "social network analysis is an important tool for information retrieval".

### 3.3.5 Web 2.0: Harnessing Collective Intelligence on the Social Web

Web 2.0 is a controversial and sometimes over-used concept first introduced during a conference brainstorming session in 2004 between Tim O' Reilly of O'Reilly publishing and Dale Dougherty of MediaLive international (O'Reilly, 2005). It essentially refers to a series of internet technologies where the user is put at the heart of the system. Most of these technologies are not new but are often the result of a *mashup* of existing concepts or the application of these concepts to a novel domain. In its most abstract form Web 2.0 refers to the notion of individuals sharing, participating and collaborating. Terms such as tags, folksonomy, blogs, wikis, RSS, Ajax and social networks have become Web 2.0 nomenclature (Digimind, 2007).

Social bookmarking websites such as Del.icio.us (Del.icio.us, 2007) and Stumble-

Upon (StumbleUpon, 2007) enable users to access their bookmarks from anywhere rather than storing them on a single computer as is common with traditional bookmarks. Users electing to bookmark a website also have the option to tag URL's with their own free text. Users can also create their own structure to store these bookmarks. The real competitive advantage of social bookmarking however, is the ability to share bookmarks with others. For example, users of Del.icio.us can search and browse other users' bookmarks. In this way social bookmarking sites can create communities of users; a user interested in a particular area can search and browse tags in the topic and also locate users who have tagged these documents. In this way users are able to identify other users with interests or intelligence in certain topics.

"Collaborative search engines" have begun to appear on the web in recent times. Search Wikia is an emerging project from the co-founder of Wikipedia, Jimmy Wales, and aims to improve the search process by relying on "human intelligence to do what algorithms can't" (Search Wikia, 2007). Decipho (Decipho, 2007) is a web search engine which attempts to focus the results of a search based on user supplied tags. Users can tag search results they receive from Decipho and these tags can then be used by subsequent users to filter their results. Yoono (Yoono, 2007) is an "instant people-rated web" which can be added as an extension to the popular Mozilla Firefox browser. When visiting a web page, Yoono lists pages which are similar to the current page and displays users who have bookmarked the current page, in an integrated sidebar. Fooxx (Fooxx, 2007) is a search engine based upon a "Personal Rank" technology. Fooxx records a user's browsing activities in order to recommend sites and ranks websites using a combination of content and user interactions where frequently visited sites receive a higher ranking than those less visited.

In this section we have described research to date in asynchronous collaborative information retrieval. These systems represent the majority of research work to-date in collaborative information retrieval. They can be characterised as implicit, passive forms of collaboration as users are not explicitly collaborating with other users but instead are benefitting through users' previous search activities. In the next section we will outline a more explicit and synchronous form of collaboration in information retrieval.

## 3.4 Synchronous Collaborative Information Retrieval

Synchronous collaborative information retrieval (SCIR) is the study of systems to support two or more people searching together in order to satisfy a shared information need. These systems represent a significant paradigmatic shift in motivation over both traditional information retrieval systems and asynchronous collaborative information retrieval systems described earlier. Whereas the systems described in this chapter thus far are focussed on improving an individual's search by incorporating information from previous searches, these systems aim to improve the effectiveness of a *group* of users searching together to satisfy a shared information need. As such, these systems represent a more explicit, active form of collaboration, where users are aware that they are collaborating with others.

With the ever increasing proliferation of information on the internet, along with the improvements in communication technologies, and the development of a more sociable web came the notion of allowing users to explore this vast information space together. As noted by Gross (1999), it is still difficult to get information in the right quantity and quality and one way of improving this would be through direct communication with others as humans are still the best information providers.

The advantages of having two or more people searching together at the same time in order to satisfy a shared information need is that it can enable both a *division of labour* and a *sharing of knowledge* across the collaborating group (Zeballos, 1998; Foley et al., 2006). Division of labour means that each member of a collaborating group can explore a subset of the information space. Some methods proposed for division of labour include: increasing the awareness amongst users of each collaborative searcher's progress (Diamadis and Polyzos, 2004; Smeaton et al., 2006) and system-mediated splitting of the search task (Foley et al., 2006; Adcock et al., 2007). The ability to effectively share information is the foundation of any group activity (Yao et al., 1999). Sharing of knowledge across group members involved in a collaborative search can occur by providing awareness of other searchers' progress through the search, and this can be achieved by enabling direct chat facilities (Gianoutsos and Grundy, 1996; Gross, 1999; Krishnappa, 2005), or group blackboards (Gianoutsos and Grundy, 1996; Cabri et al., 1999) so that brainstorming activities can be facilitated.

The first examples of synchronous collaborative information retrieval tools were built using a distributed architecture where software enabled communication across groups of remote users. Recently, the development of new computing devices has facilitated the development of co-located collaborative information retrieval tools. We will now outline research to date in each of these areas.

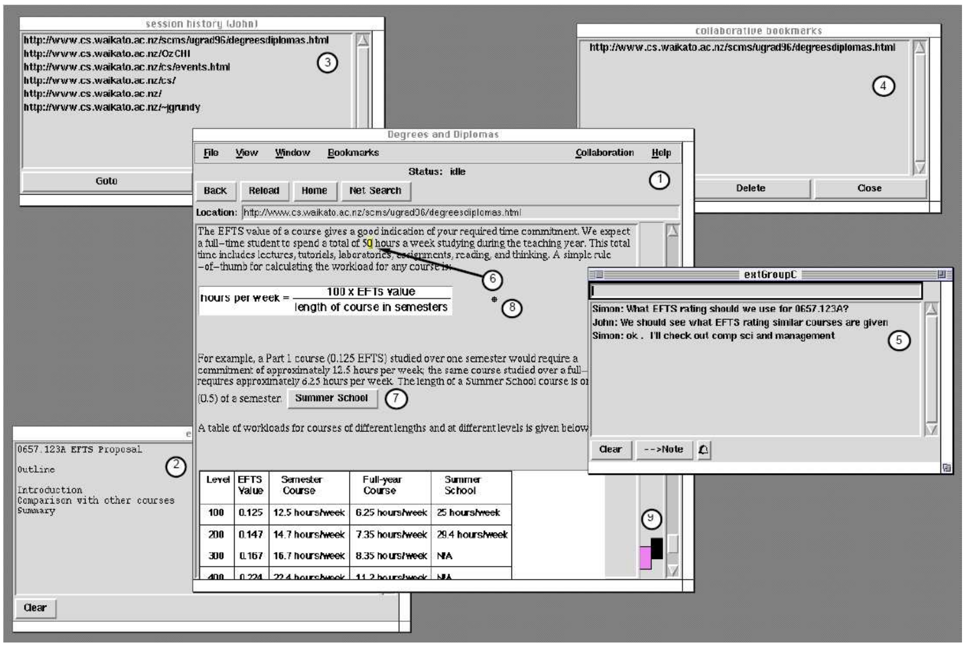
### 3.4.1 Synchronous Remote Collaborative Information Retrieval

Remote synchronous collaborative information retrieval systems enable distributed users to search and browse the web together. These systems often require users to log-in to a particular service or may require the use of particular applications in order to facilitate collaboration.

GroupWeb (Greenberg and Roseman, 1996), represents an early collaborative browsing environment and was built upon the GroupKit groupware toolkit (Roseman and Greenberg, 1996). In GroupWeb, several users could log onto a collaborative browsing session and the web browser was used as a group "presentation tool". A master browser (or "presenter") selected a page and this page was displayed to each group member using a form of "What You See Is What I See" (WYSIWIS).

The system supported synchronous scrolling and independent scrolling on a web page. With synchronous scrolling the presenter scrolled through the page and each collaborator's page scrolled in a synchronised manner; in independent scrolling each group member had control over the scrolling on their own browser. GroupWeb also supported the use of telepointers (showing others' mouse pointers on the page) which allowed users to focus the attention of the group and enact gestures. GroupWeb provided an annotation window where groups could attach shared annotations to pages, and these annotations could be viewed by all group members. In the GroupWeb system, group members were tightly coupled. Synchronising the pages that group members see can increase awareness across group members, but can be an inefficient technique for exploring the vastness of the web.

The W4 browser (Gianoutsos and Grundy, 1996), shown in Figure 3.2, extended the GroupWeb system to allow users to browse the web independently while synchronising their progress. In W4, a user could see all pages viewed by other users, they could chat with each other, share bookmarks (i.e. documents deemed relevant), and use a shared WYSIWIS white-board to brainstorm. Users could also embed chat sessions, links, and annotations directly into a web-page.



http://www.cs.waikato.ac.nz/scms/ugrad96/degreesdiplomas.html
http://www.cs.waikato.ac.nz/OzCHI
http://www.cs.waikato.ac.nz/cs/events.html
http://www.cs.waikato.ac.nz/cs/
http://www.cs.waikato.ac.nz/
http://www.cs.waikato.ac.nz/~jgrundy

http://www.cs.waikato.ac.nz/scms/ugrad96/degreesdiplomas.html

http://www.cs.waikato.ac.nz/scms/ugrad96/degreesdiplomas.html

The EFTS value of a course gives a good indication of your required time commitment. We expect a full–time student to spend a total of 50 hours a week studying during the teaching year. This total time includes lectures, tutorials, laboratories, assignments, reading, and thinking. A simple rule–of–thumb for calculating the workload for any course is:

hours per week = $\frac{\text{100 x EFTs value}}{\text{length of course in semesters}}$

For example, a Part 1 course (0.125 EFTS) studied over one semester would require a commitment of approximately 12.5 hours per week; the same course studied over a full–
requires approximately 6.25 hours per week. The length of a Summer School course is o
(0.5) of a semester.

A table of workloads for courses of different lengths and at different levels is given below

| Level | EFTS Value | Semester Course | Full-year Course | Summer School |
|---|---|---|---|---|
| 100 | 0.125 | 12.5 hours/week | 6.25 hours/week | 25 hours/week |
| 200 | 0.147 | 14.7 hours/week | 7.35 hours/week | 29.4 hours/week |
| 300 | 0.167 | 16.7 hours/week | 8.35 hours/week | N/A |
| 400 | 0.224 | 22.4 hours/week | 11.2 hours/week | N/A |

Simon: What EFTS rating should we use for 0657.123A?
John: We should see what EFTS rating similar courses are given
Simon: ok . I'll check out comp sci and management

0657.123A EFTS Proposal

Outline

Introduction
Comparison with other courses
Summary

Figure 3.2: The W4 browser (from Gianoutsos and Grundy (1996))

A similar approach was employed in Cabri et al. (1999), which used a proxy server to record documents viewed by users. These documents were then displayed to each collaborating user in a separate browser window. The system made users aware of the documents viewed by others by editing the HTML markup in pages returned to each collaborating searcher (links to pages already viewed by other users in a session were indicated using different colours). The CSCW3 application (Gross, 1999) used a room metaphor to show users those other users who were logged onto the system and currently viewing the same web page. Users in the same room could chat and couple their browsers in order to support synchronised browsing. The MUSE system (Krishnappa, 2005) employed a similar approach whereby two users could explore the web and share results and chat using separate windows.

The above systems all required users to explicitly log onto a service to support collaborative searching. SCIR systems have been developed in order to make users who are browsing the web aware of others who may be *nearby* in order to facilitate serendipitous collaboration. The motivation behind these systems is that due to the huge number of people using the internet nowadays, when searching the web for information, there is a high probability that another user is searching for the same information at the same time and providing users with an awareness of others searching for the same information enables a spontaneous collaborative searching session which can benefit both users. Donath and Robertson (1994) developed a tool which enabled people to see others currently viewing the same web page as themselves. The system also allowed users to interact with these people and coordinate their activities in order to travel around the web as a group. Sidler et al. (1997) extended this approach in order to allow users to identify other searchers within their *neighbourhood* to enable spontaneous collaboration. A logical vicinity metric defined the distance between two users in a hyperlinked document space and consisted of: (1.) a *space metric*, which represented the distance between users, (2.) a *semantic metric* based upon the contents of the documents both users were looking at, (3.) a *time metric* based on when users were browsing, and (4.) a *user-interest* *metric* based on user profiles as represented by a set of keywords. Once discovered, neighbours could chat and explore an information space together.

Laurillau and Nigay (2000) identified four types of navigational support in a collaborative browsing system. These are:

1. Guided tour - the guide navigates the web and the other members of the group follow synchronously.

2. Relaxed navigation - an open group without a leader, where each member explores independently.

3. Coordinated navigation - no leader but with each member being given a subset of the information space to explore.

4. Cooperative navigation - the leader decides on partitioning the information space. Group members work independently and at the end of the session the group leader coordinates the results.

The systems described above can be classified into the first two of these navigational types. Laurillau and Nigay (2000) developed the Co-Vitesse system to support all four types of navigation and a chat facility was also included to support communication.

The Web Collaborative Search Assistant (Diamadis and Polyzos, 2004) attempted to improved the efficiency of a search by providing for division of labour through *group member URL traversal awareness (GMUTA)*, the motivation being that if users knew that another person had visited a page then they could jump to another page and therefore save time.

SearchTogether (Morris and Horvitz, 2007), was a prototype system which incorporated many synchronous and asynchronous tools to enable a small group of remote users to work together to satisfy a shared information need. SearchTogether was built to support *awareness* of others, *division of labour*, and *persistence* of the search process. Awareness of others was achieved by representing each group member with a screen name and photo. Whenever a team member performed a new search the query terms were displayed in a list underneath their photo. By clicking on a search query a user could see the results returned for this query and this reduced the duplication of effort across users. When visiting a page, users could also see which users had visited that page previously. Users could also provide ratings for pages using a thumbs-up or thumbs-down facility. Support for division of labour was achieved through an embedded text chat facility, a recommendation mechanism, and a split search and multi-search facility. Using split search a user could divide the results of their search with a collaborating searcher and, using multi-search, a search query could be submitted to different search engines, each associated with different users.

The Adaptive Web Search (AWS) system proposed by Dalal (2007), represented a combination of personalised, social and collaborative search. The system was a type of meta-search system in which users' could search using multiple search engines and maintain a preference vector for a particular engine based on their long and short term search contexts, user goals and geographic location. Users could perform social searching by having their preference vector influenced by others depending on a level of trust.

A commercial application of synchronous collaborative IR is available in the popular Windows Live Messenger, an instant messaging service. During a chat session, users can search together by having the results from a search displayed to each user (Windows Live Messenger, 2007). Netscape Conferencer (Netscape Conferencer, 2007), allows multiple users to browse the web together using WYSIWIS where one user controls the navigation and chat facilities and whiteboards are implemented to facilitate communication.

As we can see most of the work in synchronous collaborative information retrieval has focussed on improving group effectiveness through providing awareness of other searchers' activities. This enables collaborating searchers to coordinate their activities in order to support a division of labour and sharing of search knowledge amongst collaborators. Division of labour in these systems is generally achieved by either showing the pages visited or bookmarked by other users. The sharing of knowledge in these systems is generally supported by providing facilities for communications like chat systems and shared whiteboards for brainstorming.

### 3.4.2 Synchronous Co-located Collaborative Information Retrieval

Recent advances in ubiquitous computing devices such as mobile phones and PDAs have led researches to begin exploring techniques for spontaneous collaborative search. In addition, advances in single display groupware (SDG) technology (Stewart et al., 1999), have enabled the development of collaborative search systems for the co-located environment.

By bringing users together in a face-to-face environment, these systems improve the awareness across collaborating searchers. Increased awareness can enable a more effective division of labour and a greater sharing of knowledge across the collaborating group.

Maekawa et al. (2006) developed a system for collaborative web browsing on mobile phones and PDAs. In this system, a web page was divided into several components and these components were distributed across the devices of collaborating users. In order to effectively divide a page across users, the system considered factors such as a user's profile (a set of keywords representing the user's interests) and their device's screen size. As users are physically close to one another they are aware of each others progress and can discuss and coordinate their activities. WebSplitter (Han et al., 2000) was a similar system for providing partial views to web pages across a number of users and across a number of devices available to a user (e.g. laptop, PDA).

Let's Browse (Lieberman et al., 1999), was a co-located web browsing agent which enabled multiple users standing in front of a screen (projected display onto a wall) to browse the web together based on their user profiles. A user profile in the system consisted of a set of weighted keywords of their interests and was built automatically by extracting keywords from both the user's homepage and those pages around it. Users wore electronic badges so that they could be identified as they approached the screen. A collaborating group of users using Let's Browse were shown a set of recommended links to follow from the current page, ordered by their similarity to the aggregated users' profiles.

The tangible interface system developed by Blackwell et al. (2004), allowed a group of users to perform "Query-By-Argument" whereby a series of physical tokens with RFID transmitters could be arranged on a table to develop a team's query. A team received a list of documents in response to a query and each member chose documents related to their interests. Users could highlight parts of the documents that were relevant and this relevance feedback was used to modify term weights for query expansion using Robertson's offer weight (see section 2.4.2 of Chapter 2). In this way the process of information retrieval became a by-product of the interactions amongst users.

The TeamSearch system developed by Morris et al. (2006), enabled a group of users collaborating around an electronic tabletop to sift through a stack of pictures using collaborative Boolean query formulation. The system enabled users to locate relevant pictures from a stack by placing query tokens on special widgets which corresponded to metadata categories for the collection. A collaborative query could be constructed by, for example, one user placing a query token on the location "New York" and another user placing a token on the metadata tag "2007", to retrieve all photos taken in New York in 2007.

The TeamSearch system used, as its input device, a DiamondTouch electronic tabletop system developed by Mitsubishi Electric Research Labs (MERL) (Dietz and Leigh, 2001). The DiamondTouch is a multi-user touch-sensitive tabletop interface device which enables multiple users to sit around the table and interact with objects projected onto the table from an overhead projector using their fingers. DiamondSpin (Shen et al., 2004) is an interface toolkit which enables development of applications on the DiamondTouch (or another tabletop device) and allows for objects on the screen to be moved, resized and rotated.

Físchlár-DiamondTouch (shown in Figure 3.3) was a multi-user video search application developed by the author and others at the Centre For Digital Video Processing at Dublin City University (Smeaton et al., 2006). Físchlár-DiamondTouch was developed on the DiamondTouch table and DiamondSpin toolkit from MERL and allowed two users to collaborate in a face-to-face manner in order to interact with a state of the art video retrieval application, Físchlár (Smeaton et al., 2001). Using the system, users could enter a free text query using an on-screen keyboard. This query was then issued to the search engine and a list of the 20 top ranked keyframes (an image from a video chosen as a representative of a particular video shot) were displayed upon the screen (the most relevant in the middle and decreasing in relevance as the images spiralled out). Keyframes were rotated to the nearest user in order to provide for an implicit division of labour. Two versions of Físchlár-DiamondTouch were evaluated in TRECVid 2005 (Foley et al., 2005), one which provided for increased awareness amongst users and one which was designed for improved group efficiency. This work represented the first time any group had performed collaborative search in any TREC or TRECVid workshop.

Collaboration in Físchlár-DiamondTouch was front-loaded, i.e. collaboration was supported at the interface level through various awareness widgets, however the system still communicated with a standard single-user search engine. In an effort to improve collaborative search effectiveness through “algorithmically-mediated collaboration”, the “Cerchiamo” system was developed by the 2007 FXPAL TRECVid team (Adcock et al., 2007). Cerchiamo is designed to support two users working together to find relevant shots of video. Two users worked under predefined roles of “prospector” and “miner”. The role of the prospector was to locate avenues for further exploration while the role of the miner was to explore these avenues. The system used information from users’ interactions to determine the next shots to display on-screen and provide a list of suggested query terms.

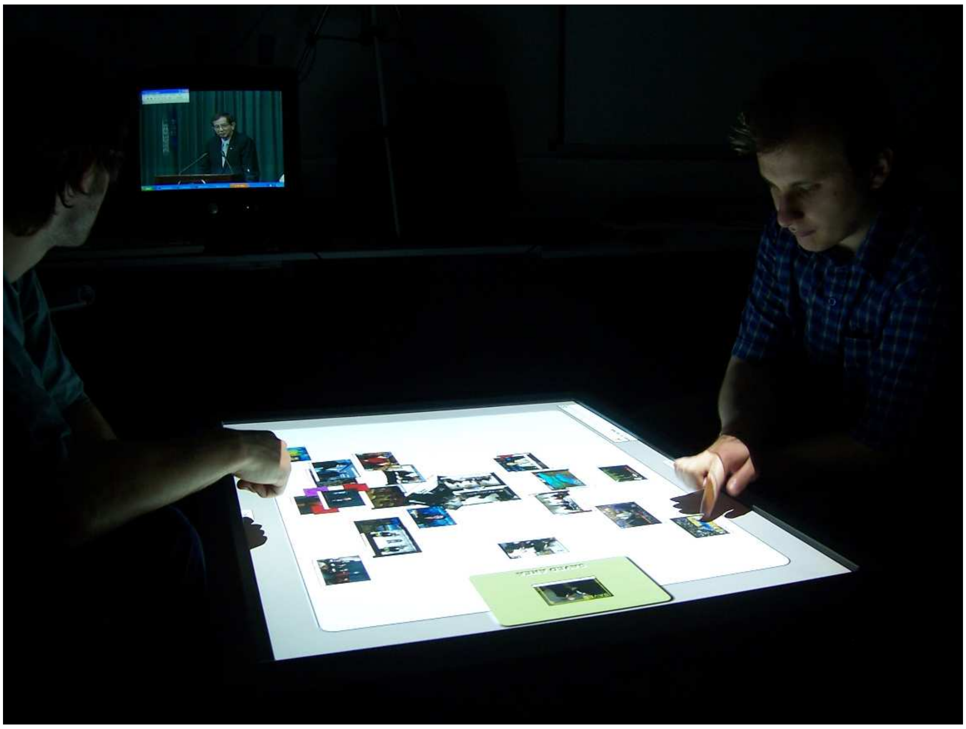


Figure 3.3: Físchlár-DiamondTouch

Single display groupware systems are gaining in popular and recently Microsoft have developed a tabletop system labelled “Surface” (Microsoft Surface, 2007) which will surely promote further exploration into this research area.

In this section we have described techniques and systems developed in order to support synchronous collaborative information retrieval. This represents a more explicit, active, form of collaboration then those systems supporting asynchronous collaboration. As information and communications technology becomes omnipresent in our daily lives our meetings with others can become richer because if we need to search for information during the course of a meeting (for example information on a topic of conversation) then we can search for this information collaboratively. In the next section we will summarise and analyse existing research in collaborative information retrieval.

## 3.5 Analysis of Existing Collaborative Information Retrieval Research

In section 3.2 we outlined how collaboration is prevalent in all parts of the information retrieval process, and that this collaboration occurs despite a lack of support for these activities in most IR systems. We then outlined the major work to-date in the area of collaborative information retrieval, an emerging research domain that combines research from the two more established fields of information retrieval and groupware. To categorise these systems we used the two dimensional time and place categorisation often used in groupware.

Asynchronous collaborative information retrieval tools are designed to help an individual's search activity by reusing other users' search outcomes and processes. These tools represent the most mature collaborative information retrieval technologies. Techniques in asynchronous collaborative information retrieval support a passive, implicit form of collaboration where the focus is to improve the search process for an *individual*.

Synchronous collaborative information retrieval is an emerging form of collaborative IR in which *a group* of two or more users are explicitly collaborating in a synchronised manner in order to satisfy a shared information need. The motivation behind these systems is related to both the ever-growing corpus of human knowledge on the web, the improvement of social awareness on the internet today, and the development of novel computer interface devices. These collaborative information retrieval systems represent a significant paradigmatic shift in focus and motivation compared with both traditional information retrieval systems and asynchronous collaborative information retrieval systems. In order for synchronous collaborative information retrieval to be effective there needs to be both an appropriate *division of labour*, and an effective *sharing of knowledge* across collaborating searchers (Zeballos, 1998; Foley et al., 2006). Division of labour enables each collaborating group member to explore a subset of a document collection in order to reduce the redundancy associated with multiple people viewing the same documents. Sharing of knowledge enables collaborating users to benefit from the activities and discoveries of their collaborators. Early SCIR systems provided various awareness cues such as chat windows, shared whiteboards, and shared bookmarks. By providing these cues, these systems enabled collaborating searchers to coordinate their own activities in order to achieve a division of labour and sharing of knowledge. However, coordinating the activities of multiple users can be troublesome, requiring too much of users' cognitive load (Adcock et al., 2007). Recently we have seen systems to support a more system mediated division of labour by dividing the results of a search query across searchers (Morris and Horvitz, 2007), or defining searcher roles in a co-located environment (Adcock et al., 2007).

However, there has been no work to date which addresses the system-mediated coordination of multiple user's activities over an entire synchronous collaborative information retrieval session which can be either co-located or remote. Suppose two users are searching together to satisfy a shared information need using a state-of-the-art SCIR system, such as those described in this chapter. A current state-of-the-art SCIR system would simply allow an initial search result to be divided across collaborators (Morris and Horvitz, 2007). However, the coordination of an entire SCIR session may be problematic with such a system. In particular, if one user decides to issue another search, it is not clear how to coordinate this search. For example, should the results be split again? Or should the user ask permission first before providing results to their search partner? By splitting the results again, the user who receives the list is expected to move their attention onto another ranked list, as the number of independent search results increases this may lead to users becoming overwhelmed with results. On the other hand, coordinating the activity through a chat facility may also be too demanding of the users. Effective system-mediated *division of labour* strategies need to be developed to allow users to explore subsections of the collection seamlessly and with minimal user intervention.

Also, at present in most SCIR systems, as users find documents of relevance to a search task they are saved by users to a bookmarked area. These bookmarks represent *explicit relevance judgments* made by users. In traditional, single-user, information retrieval, these relevance judgments are often used in a *relevance feedback* process in order to improve the quality of a user's query by reformulating it based on this relevance information. Over a number of relevance feedback iterations, an IR system can build a short term profile of the user's information need. At present, SCIR systems do not use this new relevance information directly in the search process to re-formulate a user's query, instead it is used simply as a bookmark and therefore we believe that this information is wasted. No attempt is made to utilise this relevance information during the course of an SCIR search to improve the performance of a collaborating group of users. As a consequence, the collaborating group does not see the benefit of this relevance information in their ranked lists. Effective *sharing of knowledge* techniques should allow users to benefit from their search partners' relevance judgments (i.e. bookmarks) in realtime.

The recently proposed approach by Adcock et al. (2007) attempts to arrive at a more system-mediated coordination of users' activities in a search. The system divides the searching task for two co-located users into two specialised and complementary roles. Feedback from one user is used to influence results passed to their search partner. However the focus of the work in Adcock et al. (2007) is a co-located environment and roles need to be assigned to each collaborator, a process which may be problematic and cumbersome especially when we move to a distributed environment. Furthermore in Adcock et al. (2007), the relevance assessments are not used directly in the search process but instead are used as a means to order results for presentation and for suggesting possible query terms.

In the next chapter we will describe effective division of labour and sharing of knowledge techniques in order to allow for effective synchronous collaborative information retrieval.

# Chapter 4

# Division of Labour and Sharing of Knowledge for Synchronous Collaborative Information Retrieval

## 4.1 Introduction

In this chapter we explore techniques for effective system-mediated synchronous collaborative information retrieval. We will discuss how a dynamic division of labour policy can be implemented to reduce the redundancy across users' ranked lists and allow users to find more relevant material over the course of an SCIR search, without requiring any user-user coordination. We will also outline how relevance judgments, present in most state-of-the-art SCIR systems, can be used directly in the group search. We propose novel methods to extend the standard relevance feedback approach into both a collaborative and complementary process.

## 4.2 Motivation

At present, if two or more users want to search together in either a remote or co-located setting, the onus for coordinating the group search activity is placed on the users.

When users are searching to satisfy the same information need, without an effective division of labour, a significant amount of time may be wasted by having two or more users read the same documents. Users may attempt to divide the search tasks themselves, however this may lead to users suffering cognitive overload, by having to switch focus between searching and communicating or synchronising.

Currently, in order to support a rudimentary sharing of knowledge, most SCIR systems employ the use of some kind of a shared bookmark facility as a means to make other searchers aware of relevant documents found by others. As users find relevant material they can save them to a shared bookmark area. These bookmarks represent *explicit relevance feedback* from a user regarding the information that a user deems relevant to the search at hand. However at present this information is wasted in the sense that it is simply used as an awareness cue where users can save their results during a search. No attempt is made to utilise this relevance information during the course of an SCIR search to improve the performance of a collaborating group of users. As a consequence, the collaborating group does not see the benefit of this relevance information in their ranked lists.

### 4.2.1 Thesis Hypothesis

This gap in the current state-of-the-art, led to the establishment of our main research question, namely: *can we develop techniques to automate the division of labour and sharing of knowledge across a collaborating group of users searching together synchronously ?*

Figure 4.1 provides a conceptual overview of a single user's interactions with a search engine and a relevance feedback mechanism over the duration of a search.

In this figure, time runs from left-to-right along the timeline $t$. The user begins the search by providing an initial query to the search engine ($Q$). The response of the search engine is a ranked list of documents; the user can then browse through the list of documents and may select documents to view that seem relevant to their search. If a user finds a relevant document they can indicate this to the system by providing a relevance judgment (RJ). Over the duration of the search, users may issue several queries to the search engine ($Q$'), either through reformulating the query themselves, or using a relevance feedback mechanism to expand and reweight their original query based on all relevance judgments made by the user.

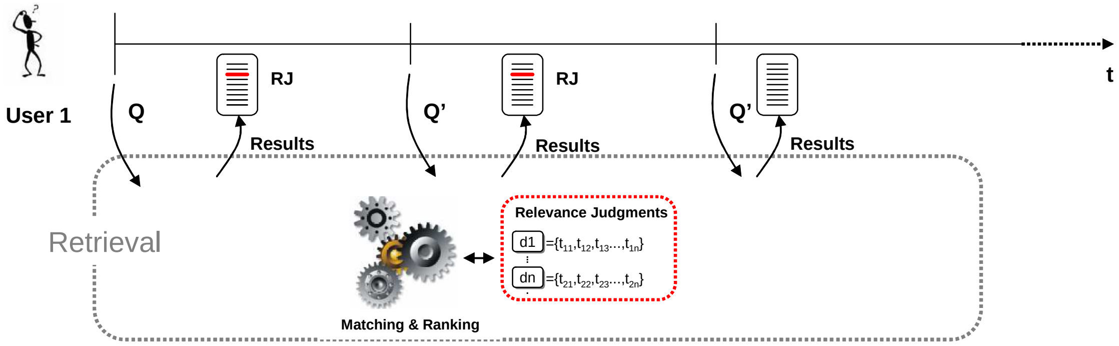


Figure 4.1: A searcher's interactions with an IR system with relevance feedback

Let's now consider a synchronous collaborative information retrieval session. Suppose that we have two collaborating users searching for information together. Figure 4.2 extends Figure 4.1 to show conceptually how two users could collaborate using a synchronous collaborative information retrieval system. This collaboration could take place either remotely or in a face-to-face manner. The interactions from these two users with the SCIR system are shown along the timelines for each user. One important point here is that as we are concerned with a synchronous session, these two timelines represent the same moment in time. In Figure 4.2 we have separated the timelines for each user for illustrative purposes, but conceptually both user's interactions could be plotted on the same line.

When two or more users come together to search in an SCIR environment, there are several ways in which the collaborative search could be initiated. For example,

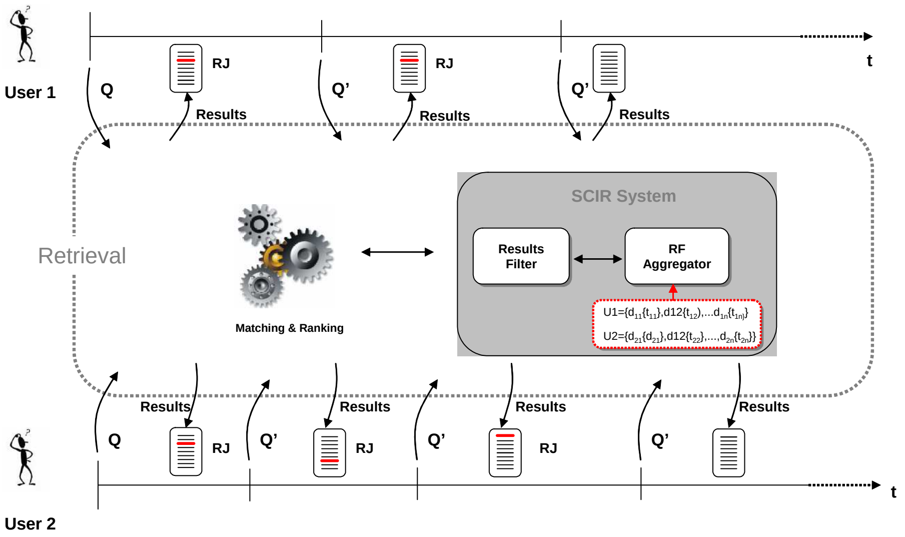


Figure 4.2: A synchronous collaborative information retrieval session involving two users

users may each decide to formulate their own search query, or users may decide on a shared, group query. In either case, unless the SCIR system coordinates the results presented back to users through a division of labour policy, then the results presented to either user may contain many of the same documents. Obviously, this overlap will be more pronounced if the group search begins with a shared query (as there will be complete overlap). If users enter their own independent queries there may more distinct sets of results across users, however, as users are searching to satisfy the same, shared information need, they may enter very similar queries, and therefore their ranked lists may still contain many of the same documents.

As the search task proceeds, each user can examine their ranked list and may decide to view documents that seem relevant to the search task. Over the course of an SCIR search, users may open and read many documents related to the search task. If users decide to issue another query later in the search, either through reformulating the query manually or automatically via a relevance feedback process, then, before returning a new ranked list to the user we have an opportunity to filter this list to automatically remove any documents that have already been examined by any group member. Without implementing such a policy, users may find themselves reading documents that have been seen by others in the group, which could impact on the effectiveness of the group search.

As users examine documents and find those relevant to the search, they may provide relevance judgments, perhaps through a bookmarking facility. If a relevance feedback mechanism is supplied in the SCIR system then, when a user initiates a relevance feedback operation, if their search partner has provided relevance judgments to the system, then we can incorporate both users' relevance judgments into the feedback process. This could enable better quality results to be returned as users are benefitting from the "shared knowledge" of their collaborators automatically. Furthermore such a sharing of knowledge policy allows users to benefit from the relevance of a document without having to view the contents of the document.

These observations led us to establish our hypothesis as:

> *If we have an environment whereby multiple people are searching together in a synchronised manner in order to satisfy a shared information need, then we can improve the performance of the search by implementing both division of labour and sharing of knowledge policies*

In order to explore our hypothesis, we have identified two key research objectives. Our main objective is to investigate our hypothesis by developing effective division of labour and sharing of knowledge techniques, and this will be the focus of this chapter. In order to evaluate the effects of these techniques, we need to develop an effective evaluation methodology for SCIR and this will be our secondary research objective, and will be the focus of Chapter 5.

In order to explore our hypothesis thoroughly, we need to investigate how to implement an appropriate division of labour policy for an SCIR search session. We need to examine how to enable an automated sharing of knowledge by extending the traditional relevance feedback process in order to incorporate multiple users' relevance information.

## 4.3 Division of Labour

Having generated an initial ranked list for an SCIR search, either as a result of a shared query or a separate query for each collaborator, these results can be divided across users using a simple *round-robin* strategy. Where, for a collaborative search involving two users with a shared initial query, user 1 would receive the first document in the ranked list, user 2 would receive the second ranked document, user 1 the third, and so on until all results are distributed across the users. This is the approach proposed by Morris and Horvitz (2007), and at the beginning of an SCIR search, this approach can ensure complete division of labour across users.

During the course of the search any member of the collaborating group may decide to issue a new search query, either by formulating a new query themselves, or by using a relevance feedback mechanism. As outlined in the previous chapter, using a simple strategy of dividing a single result across multiple users each time one member of the collaborating group issues a search, may become overwhelming for users as the number of searches increases. Instead, we believe a more reasonable approach is to allow users to search and receive a new ranked list for themselves only. In this situation, when receiving a new query from a collaborating searcher, the aim of the SCIR system is not to generate multiple ranked lists, one for each user, instead the system needs to return one list to the individual searcher who issued the search, and therefore a simple round-robin division is not applicable in this situation.

When returning a new ranked list to a searcher, however, we do have an opportunity to filter this list in order to maintain a division of labour. In particular, at any point in the search, each user may have read numerous documents and may be in the process of examining a ranked list. The SCIR system can use this information in order to remove from a user's ranked list:

1. Those documents already seen by another user.

2. Those documents contained in other users' ranked lists that we assume they

will examine.

By maintaining a list of all documents viewed during the search an SCIR system can implement *1* by filtering all lists returned to all users to ensure that documents seen by one group member are never returned to another. In order to implement *2*, we need to decide on the number of documents to assume a user is going to examine. This choice is dependent on system implementation. For example, if users are searching together remotely using separate PC's and are presented with lists of 10 documents per page, we may assume that the user takes responsibility for these 10 documents and therefore filter these documents from their search partners' ranked lists. On a large shared tabletop device we may be able to present more results to users and therefore we may assume that users will examine 20, 30, or 40 documents. On a mobile device on the other hand a smaller number of results may be more appropriate.

Of course we can never be sure that users will examine all documents that we assume they have responsibility for. Users may decide to issue another query or perform relevance feedback midway through examining the list. However, as users are searching within the same space in the document collection (i.e. searching within a topic) these unseen documents may be returned to them again as a result of their reformulated query. Failing that, when a new list is returned to the user, the documents that we assume the user is examining will also be changed, and therefore any documents that were unseen from the previous list may be returned to their search partners in subsequent iterations.

By implementing such a dynamic division of labour, and removing these documents from users' ranked lists, we can improve the SCIR search in two ways. Firstly, we can ensure that users will not spend time examining documents that have already been viewed by their co-searchers. Furthermore, this division can allow for more unique relevant documents to be pushed up to higher ranking positions in the user's ranked list, and allow them to find new relevant material faster.

## 4.4 Sharing of Knowledge

One of the common features of most state-of-the-art SCIR systems is a shared bookmark facility, which allows searchers to save items of relevance to the group search task. As outlined in Chapter 2, such *relevance judgments* are used in the information retrieval process of relevance feedback in order to reformulate a user's query and improve the quality of a user's search. In this section we will examine techniques to extend the traditional relevance feedback mechanism in order to support relevance information from multiple users.

### 4.4.1 Collaborative Relevance Feedback

If we have an environment where two or more users are providing relevance judgments to an SCIR system, then, when performing relevance feedback for a user, the SCIR system has an opportunity to incorporate each user's relevance judgments into a *collaborative relevance feedback* process, which should improve the quality of the ranked list returned to the user. It is not clear however how multi-user relevance information should be incorporated into the relevance feedback process.

One of the the simplest ways to incorporate multi-user relevance information into a feedback process is to assume that one user has provided all the relevance judgments made by all users and then initiate a standard, single-user, relevance feedback process.

However, as an SCIR session involves many users searching together, it may be desirable to allow for a more user-biased combination of relevance information within the feedback process, i.e. the favouring of one user's relevance information over another's. One example scenario will be discussed later in section 4.4.1.4, but first we will outline how the RF process can be extended to allow for a weighted combination of multi-user relevance information. In Chapter 2 we discussed the combination of evidence in information retrieval, in our work we are interested in investigating the combination of multi-user relevance information within the relevance feedback process. We use the probabilistic model for retrieval which is both theoretically motivated, and proven to be successful in controlled TREC experiments (Robertson et al., 1992, 1993, 1994). In the probabilistic retrieval model the relevance feedback processes of *Query Expansion* and *Term Reweighting* are treated separately (Robertson, 1990). Figure 4.3 presents a conceptual overview of a collaborative relevance feedback process for two users who are searching together. When the relevance feedback process is initiated, user 1 has provided 3 relevance judgments and user 2 has provided 4. As we can see, we have a choice as to what stage in the relevance feedback process we can combine this information.

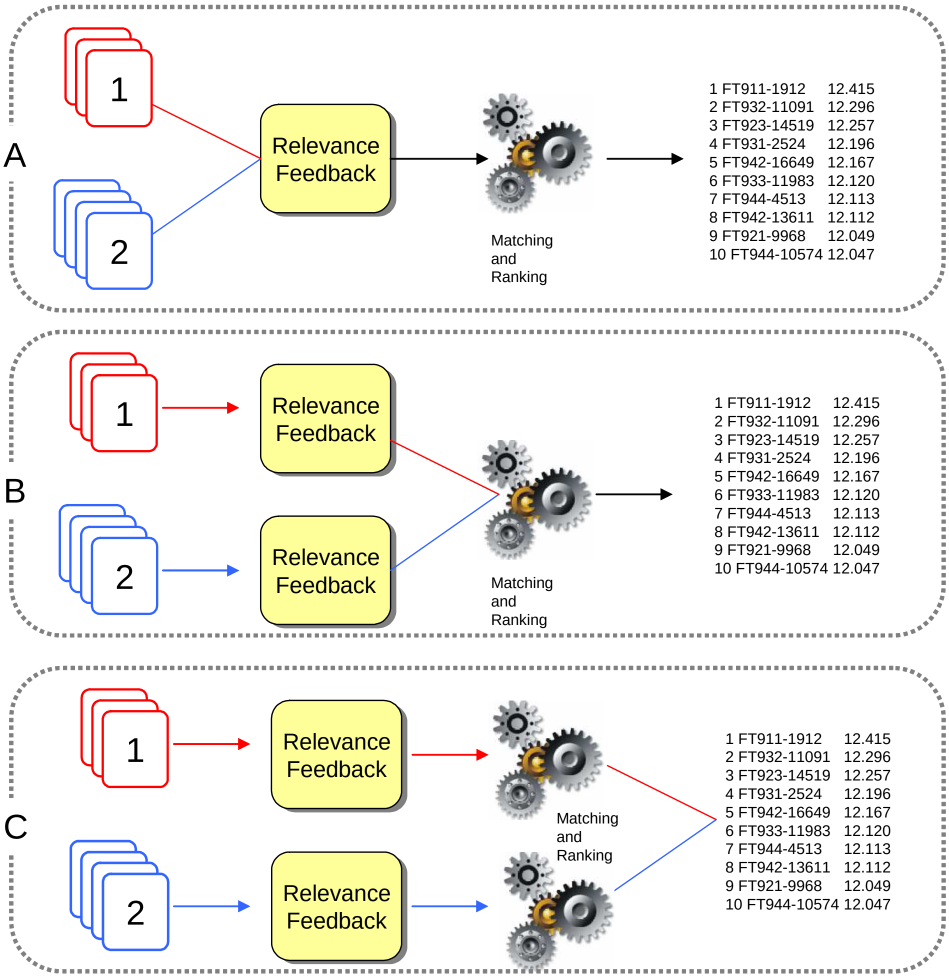


Figure 4.3: Three methods for combining multi-user relevance information

In particular we have identified three stages in the process at which we can combine relevance information, and these are:

- (A) Combine inputs to the relevance feedback process – the earliest combination. In this approach, relevance information from multiple searchers is combined before applying relevance feedback in order to provide a better prediction of term relevance for both query expansion and term reweighting.

- (B) Combine after applying the relevance feedback formulae – in this method, relevance feedback is performed for each individual searcher separately using their own relevance judgments. The outputs from each individual's RF processes are combined to produce a combined relevance weight and offer weight.

- (C) Combine after ranking – the latest stage of combination. In this method, each searcher generates a new relevance feedback query based on their own relevance judgments only. This query is then issued to the search engine, and the results for each user's query are combined, using standard ranked list fusion techniques, to produce a combined ranked list.

Options A and B operate on the relevance feedback processes directly, i.e. on a *term* level, in order to improve the processes of query expansion and term reweighting. Option C, on the other hand, operates on a document level and represents the standard approaches to evidence combination from the literature. By evaluating which approach works best we can understand at what stage in the RF process multi-user relevance information should be combined.

#### 4.4.1.1 Combining Inputs to the Relevance Feedback Process (A)

The relevance feedback process uses all available relevance information for a term in order to assign it a score for both query expansion and term reweighting. If we have relevance information from multiple co-searchers, combining this information before performing relevance feedback should result in an improved combined measure of relevance for these terms. This is the rationale behind this novel method for combining relevance information, which we refer to as *partial-user weighting*, as the evidence for relevance or non-relevance of a term is composed of the combined partial evidence from multiple users. We will now outline the derivation for the partial-user relevance weight and partial-user offer weight. From Robertson and Spärck Jones (1976), we can see that the probability of relevance of a term is defined as:

$$w(i) = \log \frac{p(1-q)}{q(1-p)} \tag{4.1}$$

where

$p =$ probability that a document contains term $i$ given that it is relevant

$q =$ probability that a document contains term $i$ given that it is non-relevant

Where the appropriate substitutions for $p$ and $q$ are the proportions:

$$p = \frac{r_i}{R} \tag{4.2}$$

$$q = \frac{n_i - r_i}{N - R} \tag{4.3}$$

where

$r_i =$ Number of relevant documents in which term $i$ occurs

$R =$ Number of identified relevant documents

$n_i =$ Number of documents in the collection in which term $i$ occurs

$N =$ Number of documents in the collection

The probability that a document contains term $i$ given that it is relevant, $p$, is equal to the proportion of all relevant documents in which the term $i$ occurs. The probability that a document contains term $i$ given that it is non-relevant, $q$, is equal to the proportion of all non-relevant documents that contain the term. Applying these substitutions to equation 4.1 we get the standard relevance weighting formula:

$$rw(i) = \log \frac{(\frac{r_i}{R})(1 - \frac{n_i - r_i}{N-R})}{(\frac{n_i - r_i}{N-R})(1 - \frac{r_i}{R})} \tag{4.4}$$

If we assume that in a collaborative search session we have $U$ collaborating users searching, then the proportions for $p$ and $q$, in equations 4.2 and 4.3 respectively, can be extended as follows:

$$p = \sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}}{R_u} \tag{4.5}$$

$$q = \sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}}{N - R_u} \tag{4.6}$$

where

n$_i$ and N are as before

$r_{ui}$ = Number of relevant documents identified by user $u$ in which term $i$ occurs

$R_u$ = Number of relevant documents identified by user $u$

$\alpha_u$ = Determines the impact of user $u$'s proportions on the final term weight, and $\sum_{u=0}^{U-1} \alpha_u = 1$

Therefore we have extended the proportions using a linear combination (Chapter 2, section 2.5.2) of each user's relevance statistics. Using this approach, the probability that a document contains term $i$, given that it is relevant, is equal to the sum of the proportions for relevance from each user. The probability that a document contains term $i$, given that it is not relevant, is equal to the sum of the proportions of non-relevance. Each of these values is multiplied by a scalar constant $\alpha_u$, which can be used to vary the effect of each user's proportion in the final calculation, and a default value of $\frac{1}{U}$ can be used to consider all users equally.

One important consideration when combining multi-user relevance information is what to do when a term had not been encountered by a user (i.e. the term is not contained in their relevance judgments). There are two choices here, we can either allow the user who has not encountered the term to still contribute to the shared weight, or we can choose to assign a weight to a term based solely on the relevance and non-relevance proportions of users that have actually encountered the term.

If we wish to incorporate a user's proportions for a term regardless of whether the term appears in any of the user's relevance judgments, then the term will receive a relevance proportion, $p = 0, (\frac{0}{R})$ and a non-relevance proportion, $q = \frac{n}{N-R}$ from a user who has not encountered the term (as $\mathrm{r}_i = 0$ for that user).

If we do not wish to incorporate a user's proportion for a term if they have not encountered the term, then the shared relevance and non-relevance proportions of $p$ and $q$ in Equation 4.5 and Equation 4.6 respectively will be composed of the proportions from users who have encountered the term only, by setting the $\alpha_u$ value to zero for those users who have not encountered the term.

Allowing a user to contribute to a term's weight even if s/he has not encountered it causes a term to receive a lower weight than it would by not allowing a contribution. The reason for this is that both techniques will cause a relevance proportion of 0 to be assigned for a user that has not encountered the term. The "no contribution" technique will result in a non-relevance proportion of 0 for the user who has not encountered the term, whereas the "contribution" technique will give a value greater than zero (i.e. a higher non-relevance proportion)

Applying the extended proportions of $p$ and $q$, in Equations 4.5 and 4.6 respectively, to the probability of relevance from Equation 4.1, results in our *partial-user relevance weight* (purw):

$$purw(i) = \log \frac{(\sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}}{R_u})(1 - \sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}}{N-R_u})}{(\sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}}{N-R_u})(1 - \sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}}{R_u})} \tag{4.7}$$

For practical implementation of the standard relevance weighting formula (equation

4.4), and to limit the errors associated with zeros such as dividing by zero, a simple extension is commonly used that adds a constant to the values in the proportions. Applying the proportions suggested in Robertson and Spärck Jones (1976), known as the Jeffrey prior, to equation 4.7, results in:

$$purw(i) = \log \frac{(\sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}+0.5}{R_u+1})(1 - \sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}+0.5}{N-R_u+1})}{(\sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}+0.5}{N-R_u+1})(1 - \sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}+0.5}{R_u+1})} \tag{4.8}$$

So far we have shown how the partial-user method can be applied to the standard relevance weighting formula. Now we will consider applying the scheme to the offer weighting formula of:

$$ow_i = r_i \times rw_i \tag{4.9}$$

Using a linear combination, approach the $r_i$ value from Equation 4.9 can be extended to include a weighted combination of each collaborating user's $r_i$ value, to produce a *partial-user offer weight* (puow):

$$puow(i) = (\sum_{u=0}^{U-1} \alpha_u r_{ui}) \times purw(i) \tag{4.10}$$

where

$r_{ui}$ = Number of relevant documents identified by user $u$ in which term $i$ occurs

$purw(i)$ = The term's partial-user relevance weight

$\alpha_u$ = Determines the impact of each users r$_i$ value on the final value, and

$$\sum_{u=0}^{U-1} \alpha_u = 1$$

Table 4.1 illustrates the result of applying the partial-user relevance weighting and partial-user offer weighting to nine example terms for a relevance feedback process involving two users. In this simple example both users have provided 2 relevance judgments (i.e. $R = 2$ for both users), the number of documents in the collection, *N*, is 100, and each term's $n_i$ value is the same (i.e. $n_i = 10$ for all terms) and $\alpha = 0.5$ for both users. The terms are ordered by the total number of relevance judgments in which they occur. The result of applying the standard relevance weighting formula and offer weighting formula for both users is shown for each term, as is the result of applying the partial-user weighting scheme (*purw*) and partial-user offer weighting scheme (*puow*). In this example, the partial-user technique is operating in the contribution mode, which allows users who have not encountered the term to contribute to it's shared weight. The relevance weighting and offer weighting values, assigned to terms are different for both users depending on the user's *r* value for the term. For example, user 1 ranks term *t2* higher than *t3* whereas for user 2 this ranking is reversed. When combining relevance information in the partial-user formulae the term that occurs in each relevance judgment from both users, *t1*, receives the highest score. For terms where the users *agree*, i.e. the term occurs in the same number of relevance judgments from both users (*t1*, *t4*, *t9*), the formulae produces the same result as per the standard formulae. For terms where the users *disagree* (*t2*, *t3*, *t5*, *t6*, *t7*, *t8*), the formula produces an estimate based on a combination of the proportions.

| Term | $r_{1i}$ | $r_{2i}$ | $rw_1$ | $ow_1$ | $rw_2$ | $ow_2$ | purw | puow |
|---|---|---|---|---|---|---|---|---|
| t1 | 2 | 2 | 3.97 | 7.95 | 3.97 | 7.95 | 3.97 | 7.95 |
| t2 | 2 | 1 | 3.97 | 7.95 | 2.24 | 2.24 | 3 | 4.5 |
| t3 | 1 | 2 | 2.24 | 2.24 | 3.97 | 7.95 | 3 | 4.5 |
| t4 | 1 | 1 | 2.24 | 2.24 | 2.24 | 2.24 | 2.24 | 2.24 |
| t5 | 2 | 0 | 3.97 | 7.95 | 0.52 | 0 | 2.24 | 2.24 |
| t6 | 0 | 2 | 0.52 | 0 | 3.97 | 7.95 | 2.24 | 2.24 |
| t7 | 1 | 0 | 2.24 | 2.24 | 0.52 | 0 | 1.49 | 0.75 |
| t8 | 0 | 1 | 0.52 | 0 | 2.24 | 2.24 | 1.49 | 0.75 |
| t9 | 0 | 0 | 0.52 | 0 | 0.52 | 0 | 0.52 | 0 |

Table 4.1: Partial-user relevance weighting example, N = 100, $n_i$ = 10 and R = 2 for both users

#### 4.4.1.2 Combining Outputs of the Relevance Feedback Process (B)

This method of combination operates by treating the relevance process as a black box and combines the outputs of its processes for multiple users in order to produce a combined score. For each user, relevance weighting and offer weighting are calculated separately using a searcher's own relevance statistics. The outputs from these processes (i.e. the scores) are combined to produce a *combined relevance weight*, in the case of relevance weighting, or *combined offer weight*, in the case of offer weighting. Combination is therefore performed at a later stage in the relevance feedback process than the method proposed in the previous section.

For relevance weighting, we calculate the combined relevance weight using a linear combination of relevance weight scores from all users ($crw$):

$$crw(i) = \sum_{u=0}^{U-1} \alpha_u \times rw(u, i) \tag{4.11}$$

For offer weighting we can follow the same approach, by calculating the offer weight separately for each user and then combining afterwards to produce a combined offer weight ($cow$):

$$cow(i) = \sum_{u=0}^{U-1} \alpha_u \times ow(u, i) \tag{4.12}$$

where

$\alpha_u$ = Determines the impact of each user's contribution on the final score, and

$\sum_{u=0}^{U-1} \alpha_u = 1$

As with the partial-user method, we can either include or leave out a user's contribution to either the combined relevance weight of combined offer weight if they have not encountered the term in their own set of relevance judgments. In either case, using this method a term which has not been encountered by a user will receive an offer weight of 0 for that user. The difference between the "contribution" and "no contribution" variants in this method are only in the calculation of the combined relevance weighting. Once again the alpha variable can be used to control the impact of each user's evidence on the combination and a default value can be set to $\frac{1}{U}$ for all users to consider all users equally.

#### 4.4.1.3 Combining Outputs of the Ranking Process (C)

This stage of combination operates at a higher level of granularity than either of the previous methods, as here we treat the entire search engine as a black box and combination is performed at the ranked list, or document level.

Combining the outputs from multiple ranking algorithms has become a standard method for improving the performance of IR systems' ranking (Croft, 2002). These methods allow the combination of multiple sources of information into a single ranked list.

In our work, the ranked lists we wish to combine are the results of each user's separate relevance feedback query that is formulated using their own relevance information. In order to produce a combined ranked list, a reformulated query is generated for each collaborating user, and these relevance feedback queries are then submitted to the search engine in order to produce separate ranked lists, one for each user. These ranked lists are then combined in order to produce a combined ranked list.

Combination at the document level can be achieved, as before, by performing a linear combination, to provide a combined document score ($cds$):

$$cds(d, q) = \sum_{u=0}^{U-1} \alpha_u \times s(u, d) \tag{4.13}$$

where

$s(u, d)$ = the relevance score for document $d$ in relation to user $u$'s query

$\alpha_u$ = determines the impact of each user's contribution on the final document score, and $\sum_{u=0}^{U-1} \alpha_u = 1$

In this section we have outlined how a system-mediated sharing of knowledge can be achieved in an SCIR search, by incorporating each group member's relevance judgments into a collaborative relevance feedback process. We have proposed three methods by which the standard relevance feedback formula can be extended into a collaborative relevance feedback process. These techniques represent ways to aggregate the evidence from multiple users and the advantages of these techniques over the standard relevance feedback formula is that they can allow for a user-biased combination of relevance information, which we will now discuss.

#### 4.4.1.4 Authority Weighting

Synchronous collaborative information retrieval systems, by their very nature, bring together multiple collaborating users, each with a certain level of expertise and experience. Some users may be more familiar with a topic than others and this may be reflected in the quality of their relevance judgments during the group search activity. For example, a novice user may not understand the search topic entirely and therefore may be mistaken in their relevance assessments. Poor relevance assessments, unless recognised and dealt with, may pollute the collaborative relevance feedback process and degrade results.

We propose to attach an *authority weight* to each user's relevance information and incorporate this authority information into the collaborative relevance feedback process. An authority weight can be applied to all collaborative RF techniques outlined earlier in this section, by adjusting the $\alpha$ value associated with each user's relevance information. In Table 4.2 we examine the application of an authority weighting scheme (labelled "auth") on the partial-user weighting method. In this example we assume that user 1 has been assigned an authority value of 0.75, and user 2 a value of 0.25. The results from the un-biased partial-user method are also shown (*purw*, *puow*) for comparative purposes.

| Term | $r_{1i}$ | $r_{2i}$ | $rw_1$ | $ow_1$ | $rw_2$ | $ow_2$ | purw | puow | purw-auth | puow-auth |
|---|---|---|---|---|---|---|---|---|---|---|
| t1 | 2 | 2 | 3.97 | 7.95 | 3.97 | 7.95 | 3.97 | 7.95 | 3.97 | 7.95 |
| t2 | 2 | 1 | 3.97 | 7.95 | 2.24 | 2.24 | 3 | 4.5 | 3.43 | 6.01 |
| t3 | 1 | 2 | 2.24 | 2.24 | 3.97 | 7.95 | 3 | 4.5 | 2.61 | 3.26 |
| t4 | 1 | 1 | 2.24 | 2.24 | 2.24 | 2.24 | 2.24 | 2.24 | 2.24 | 2.24 |
| t5 | 2 | 0 | 3.97 | 7.95 | 0.52 | 0 | 2.24 | 2.24 | 3 | 4.49 |
| t6 | 0 | 2 | 0.52 | 0 | 3.97 | 7.95 | 2.24 | 2.24 | 1.49 | 0.75 |
| t7 | 1 | 0 | 2.24 | 2.24 | 0.52 | 0 | 1.49 | 0.75 | 1.88 | 1.41 |
| t8 | 0 | 1 | 0.52 | 0 | 2.24 | 2.24 | 1.49 | 0.75 | 1.06 | 0.26 |
| t9 | 0 | 0 | 0.52 | 0 | 0.52 | 0 | 0.52 | 0 | 0.52 | 0 |

Table 4.2: Partial-user weighting example with authority weighting

From Table 4.2 we can see that for terms where users agree on the weighting, (*t1*, *t4*, *t9*), the formulae produce the same result as per the standard, unbiased, partial-user formulae. For terms where the users *disagree* (*t2*, *t3*, *t5*, *t6*, *t7*, *t8*) the formulae produce an estimate based on a weighted combination, where the combined values are closer to user A's estimates than user B's. Allowing for a user-biased authority weighting of the collaborative RF process may allow us to limit the influence of poor relevance assessments on the outputs of the process.

### 4.4.2 Complementary Relevance Feedback

One of the great benefits of having multiple users tackle a search problem together is that each user can be assigned a subsection of the search space to explore, and this is the motivation behind the division of labour techniques as discussed in section 6.2. The collaborative relevance feedback techniques discussed in section 4.4.1 attempt to aggregate all collaborating users relevance information in the relevance feedback process. The effect of this aggregation, however, may be working against the principles of division of labour and collaborative RF techniques may be bringing users *too close together*. Although, through an explicit division of labour policy, we can ensure that no two users will examine the same documents, the collaborative relevance feedback techniques may still cause a reduction in the *breadth* across users' queries and therefore may result in a smaller number of unique relevant documents being found across the entire group.

An alternative way of using relevance information in a synchronous collaborative search is to implement a *complementary relevance feedback process*. Figure 6.10, provides a conceptual comparison of the effects of a collaborative relevance feedback technique (those discussed in the previous section) and the complementary techniques which we will outline in this section, on users' queries. While the collaborative relevance feedback techniques attempt to make users' queries more similar, complementary approaches attempt to make them more distinct.

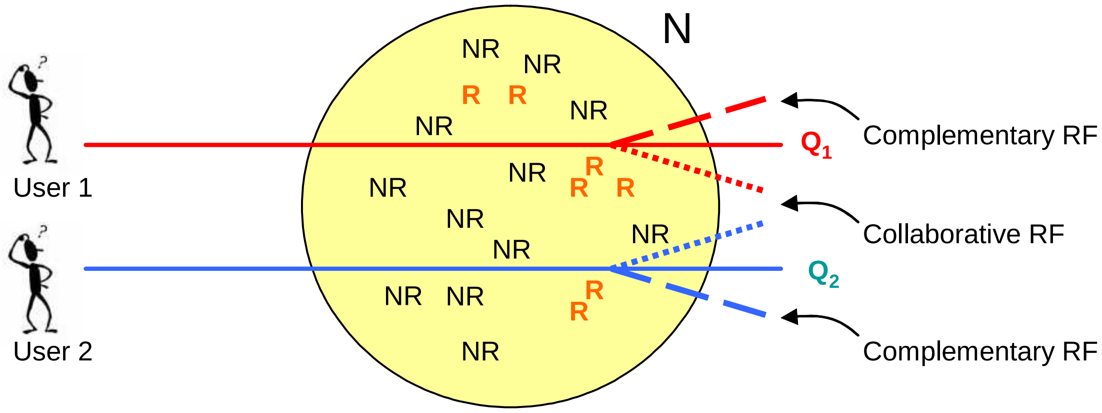


Figure 4.4: Conceptual comparison of the effects of collaborative and complementary relevance feedback on users' queries

When performing relevance feedback for a user, by considering the relevance information of other searchers and reformulating a user's query in such a way that makes it as distinctive as possible from the other searchers in a group, a complementary relevance feedback mechanism can maintain diversity across the users' results and this may allow for the retrieval of more relevant material across the group. We have identified two ways in which complementary relevance feedback can operate in an SCIR search.

#### 4.4.2.1 Complementary Query Expansion

One simple way of maintaining diversity across users through the RF process is by ensuring that the users' queries are only expanded with terms that are not contained in other searchers' queries, a process we refer to as *complementary query expansion.*

In complementary query expansion, when performing relevance feedback for a user, the SCIR system can use a standard relevance feedback process over a user's own relevance judgments. Then, when choosing the top N terms for query expansion, the SCIR system will not consider a term for expansion if it is contained in the current query of their co-searchers. The result of this process is that duplicate terms will be replaced with unique terms, which may enable the discovery of more unique relevant material.

#### 4.4.2.2 Clustering for Complementary Relevance Feedback

At any stage in an SCIR search, there may be a number of relevance judgments made by different users. These documents and the terms contained within them, although related by being relevant to the same topic, may also be quite different in the aspects of the topic that they are related to. For example, a search topic "Hydroelectric Projects" may have relevant documents which discuss the development of hydroelectric dams, while others may discuss government financing for these projects. A collaborative relevance feedback technique, such as those discussed in section 4.4.1 may *aggregate out* this uniqueness.

Clustering is a technique used to organise objects into groups whose members are similar in some way (Kaufman and Rousseeuw, 1990). The use of clustering in information retrieval has been investigated for many years (Willett, 1988). In our work we are interested in using clustering in a complementary relevance feedback process in order to group related material together so that we can assign distinct clusters to each user in a collaborative search session.

Grouping related material together through a clustering process, should enable the SCIR system to maintain diversity across users' queries. This diversity is maintained, not by the removal of terms from a user's query (as outlined in section 4.4.2.1 above), but by assigning each user a subset of the relevant document space to explore.

Clustering in SCIR can operate on both a *document* and a *term* level. In our work, we investigate the use of the *k-means* clustering algorithm (Macqueen, 1967) for clustering relevant material in SCIR. K-means is one of the simplest and most popular clustering algorithms and operates by partitioning objects into $k$ clusters so that objects within one cluster are as close to each other as possible, while as far as possible away from objects in other clusters. It does this by trying to minimise an objective function based on total intra-cluster distances. Prior to performing k-means, the number of clusters, $k$, needs to be defined as an input parameter. For our work, as we are attempting to cluster items into distinct sets across users, $k$ is defined as the number of users searching together.

## 4.5 Summary

In this chapter we have motivated and developed our hypothesis, that a system mediated division of labour and sharing of knowledge can improve the performance of a a group of users searching together synchronously.

We discussed how to implement an effective, dynamic, division of labour over the duration of an SCIR search, by removing those documents seen by users and those which we assume the user has responsibility for.

We then discussed how relevance judgments, which are often made in SCIR systems through a bookmarking facility, can be used directly within the search itself in order to improve the effectiveness of the search process. We proposed novel collaborative relevance feedback techniques which allow for a user-biased combination of multi-user relevance information in the relevance feedback process. We discussed how this user-biasing could enable an authority weighted collaborative relevance feedback process. We then discussed an alternative use of multi-user relevance information in an SCIR session through complementary relevance feedback. In the next chapter we will discuss our evaluation methodology, developed to allow for the evaluation of the techniques proposed in this chapter.

# Chapter 5

# Evaluation Methodology

## 5.1 Introduction

In this chapter we will describe the methodology used to explore our hypothesis as outlined in the previous chapter. This methodology is based upon building simulations of two users searching together with a synchronous collaborative information retrieval system which implements an incremental relevance feedback mechanism. In order to build these simulations we mine data from previous TREC interactive experiments and simulate two users searching together who had previously completed a search topic separately as part of a submission to the TREC interactive search task. Using this approach we can evaluate how these users perform when searching using an SCIR system implementing the various division of labour and sharing of knowledge techniques outlined in the previous chapter. We outline the total amount of simulations used in our evaluation and the test collection used. Finally we conclude the chapter with a detailed explanation of the approach used to evaluate the performance of a group of users searching together.

## 5.2 Simulations in Information Retrieval

Simulations are used in information retrieval in an attempt to model a user's interactions with an IR system. A simulated user's interactions with a system can be controlled by using a parameterised user model and these models can vary in complexity based on the systems they are evaluating and the interactions they are attempting to simulate. Simulations are an attempt to bridge the gap of realism in information retrieval experimentation, between fully automatic experiments, where the user is taken out of the loop completely, and fully interactive experiments, where real users interact with an IR system. Simulations are useful for rapid experimentation with system variants in an environment where a full user experiment is infeasible, perhaps due to time and implementation costs, but some level of user interaction is required for an effective evaluation. According to White et al. (2005), the benefits of simulations are:

- They are less costly and time consuming, compared to real user experiments.
- They allow for the evaluation of IR techniques in many different retrieval scenarios.
- The experimental setup can be controlled by the system designer.

As simulations can be run more quickly and can be repeated easier than a real user experiment, they allow for a thorough analysis of the performance of algorithms under various parameters controlling their process. They allow designers to model the effects of different types of users on the system performance, by setting parameters controlling their operation. Having complete control over the simulated users actions enables system designers to make more meaningful inferences from the results of simulated experiments.

User simulations have been used extensively in information retrieval evaluations as a means to model a user's interactions with a relevance feedback mechanism (Rocchio, 1971; Harman, 1988; Buckley et al., 1994; Magennis and van Rijsbergen, 1998; Ruthven, 2003; White et al., 2005; Keskustalo et al., 2006).

Early work into relevance feedback used simulations in order to investigate the effects of the number of documents provided for feedback and the number of terms used to expand the query (Harman, 1988; Buckley et al., 1994).

White et al. (2005) implemented user simulations in order to evaluate the effectiveness of various implicit feedback mechanisms, where a relevance judgment was inferred from a user's interactions with a document using a novel user interface. Simulations were devised using various *relevance paths*, which determined how a simulated user interacted with the documents using the interface. The system used pre-modelled paths and a user could exhibit "wandering behaviour", where they viewed both relevant and non-relevant material, in order to examine how each implicit feedback model performed in an operational environment.

Keskustalo et al. (2006) examined the effects of different relevance thresholds and user behaviour on the performance of relevance feedback using user simulations. A user model was constructed which controlled the relevance threshold of a user (i.e. stringent, regular, or liberal), the number of documents the user was willing to browse, and the number of feedback documents the user was willing to provide.

As we can see, researchers have made use of simulations in information retrieval in order to evaluate the performance of relevance feedback techniques. These simulations can be constructed in order to model different user interactions from simple techniques, such as choosing the number of documents to provide to the relevance feedback mechanism, to more complex techniques such as determining a user's path through a novel implicit feedback system. In our work we use searcher simulations, but as we will now outline the novel domain of SCIR requires new simulation techniques to be developed.

## 5.3 Synchronous Collaborative Information Retrieval Simulations

In our work, we are attempting to evaluate how a system-mediated SCIR search can improve the performance of a group of users searching together over the duration of an entire search session. In order to explore our hypothesis, we are evaluating many different approaches to division of labour and to sharing of knowledge. It would have been infeasible to evaluate each of these approaches thoroughly using real user experiments. Instead, by using simulations we can evaluate our proposed approaches effectively while ensuring that our evaluations remain realistic.

Previous information retrieval experiments which have used user simulations have focused on a single user's interactions with an IR system. Here we are attempting to simulate a synchronous collaborative information retrieval environment, a dynamic, collaborative simulation. In order to model such an environment we have developed novel simulation techniques as we will now describe.

### 5.3.1 Requirements Analysis

In our work we are interested in developing effective system-mediated collaborative techniques which could be deployed in either a remote or co-located environment, and therefore, we were conscious that the simulations we used should be realistic of future systems in any device or interface which could support SCIR search, i.e. desktop search, tabletop search, PDA or Apple iPhone search etc.

Our SCIR simulations will simulate a search involving *two* collaborating users. Recent studies on the collaborative nature of search have shown how the majority of synchronous collaborative search sessions involve a collaborating group of two users (Morris and Horvitz, 2007) and therefore we believe that this group size is the most appropriate to model, though the techniques proposed could scale to larger group sizes.

One of the important considerations for any SCIR system is how to initiate the collaborative search. As outlined in the previous chapter, each user could be allowed to enter in their own query, or users could decide on a shared query, or a group leader may choose the initial query. For the experiments reported in this thesis each simulation begins by assuming that one initial query has been formulated by the group of two users. How we construct this query will be discussed in section 5.3.2.1. In a real system, this query could be formulated by one user or by both users collaboratively. We feel that by only requiring one query from the set of users, we can limit the interactions needed by users with the search system. Although querying may be easy using the standard keyboard and mouse combination, interactions with phones or other handheld devices can be difficult. One potential disadvantage of having only one shared query is that we are limiting the diversity across users, which may be introduced through having individual queries, however as users are searching to satisfy the same information need, users' queries will often contain many of the same terms. In fact, our analysis of the user-user pairs used in our simulations (described later in section 5.4.1) shows that over 94 % of users' initial search queries shared at least 1 query term in common. Considering that the average number of terms-per-query in the data used in our simulations is 3.01 terms, this shows a substantial overlap across users. Therefore the benefit of allowing each user to enter their own query, in terms of the diversity across queries, may not outweigh the cost in terms of the extra effort required by each user to enter their own query.

Further to this point, in our simulations, users do not manually reformulate their queries during the search, instead, in order to receive new ranked lists during the search, users use the relevance feedback mechanism. In any SCIR search, users may provide multiple relevance judgments over the course of the search session, and therefore, in our simulations, we have a choice as to when to initiate a relevance feedback operation. For example, we could choose to perform feedback after a user provides one relevance judgment, or after the user has provided a certain number of relevance judgments. For the purposes of the experiments reported in this thesis, our SCIR simulations operate by initiating a relevance feedback operation for a user each time they provide a relevance judgment, thereby returning a new ranked list of documents to the user. This approach is known as *Incremental Relevance Feedback*, a method first proposed by Aalbersberg (1992). Using the Incremental RF approach, a user is provided with a new ranked list of documents after each relevance judgment, rather than accumulating a series of relevance judgments together and issuing them in batch to the RF process. For several reasons we believe that this level of feedback granularity will be the most suitable for a synchronous collaborative information retrieval system. Firstly, studies have shown that applying feedback after only one or two documents have been identified can substantially improve performance over an initial query (Spärck Jones, 1979). Secondly, as searchers can be particularly impatient individuals, often only examining a few pages per result and rarely proceeding beyond the first page of results (Jansen and Spink, 2006), it is unreasonable to assume that a searcher would be willing to provide feedback on a large number of documents before seeing the benefit in their ranked lists. By using an incremental feedback approach, users can see the benefit of their relevance information immediately. Furthermore, as we are attempting to model a collaborative environment, where users are trying to maximise the output from their shared time together, performing a RF iteration after each relevance judgment allows for a greater crossover of relevance information between the searchers. A similar approach to the incremental feedback method used in our simulations is used in the popular Google online search engine through its "Similar Pages" option. By clicking a Similar Pages link beside a result, Google will return a new list of documents to a user containing documents that are similar to a given web page. Unlike the Google Similar Pages operation, which uses only one relevance judgment for feedback, our incremental feedback system uses all relevance judgments made in the search process so far in the feedback process.

Another choice for any SCIR system, is whether to present a user with a new ranked list only when they interact with the search engine themselves or when they *or* their search partner interacts with the search engine. Presenting users with a new ranked list when their search partner performs a search may allow users to benefit more quickly from their partner's progress. However deployment of such an intensive SCIR system would require designers to develop novel interface techniques in order to allow for the updating of ranked lists displayed on users' screens in such a way that it minimises disruption for the user. Furthermore, users searching with such an intensive system may suffer from cognitive overload by seeing their ranked list changing, seemingly, at random and having to re-adjust their focus to new documents. Due to these issues, in our experiments we will mainly simulate SCIR sessions where users receive new ranked lists only when they interact with the search engine themselves, through the relevance feedback process. However, we will also experiment with a more intensive environment, where users are presented with new ranked lists when their search partner performs relevance feedback, in order to explore the effects of a such an environment on the performance of a group of searchers.

Figure 5.1 presents a conceptual overview of two users collaborating using the SCIR system described thus far.

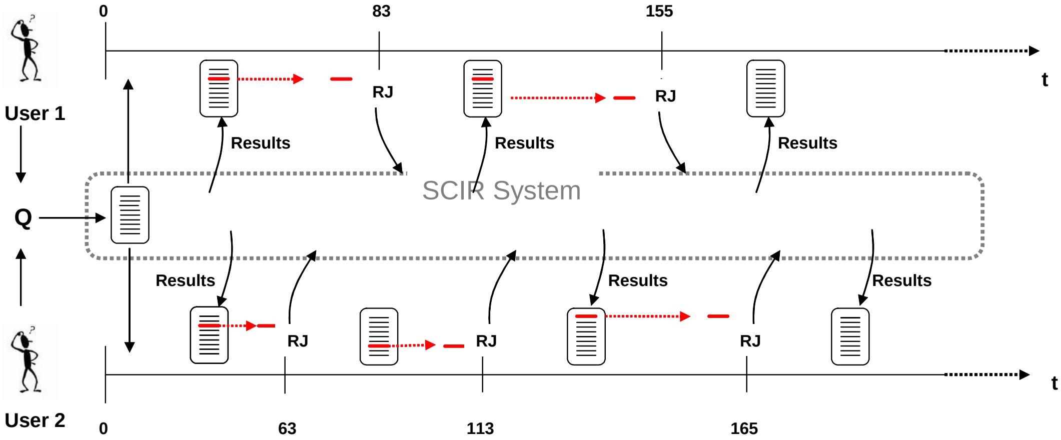


Figure 5.1: A simulated SCIR session

Referring to this figure, the data required to populate our SCIR simulations is:

- An initial query (Q) – as outlined above, the simulated SCIR session begins with one initial query entered by the users. This query could be entered by one user or collaboratively formulated by both users. Figure 5.1 represents two users contributing to the shared query.

- Series of relevance judgments (RJ) – these are explicit indications of relevance made by a user on a particular document. State-of-the-art SCIR systems allow for relevance judgments to be made in the form of bookmarks. In our work we are interested in evaluating how best to utilise these relevance judgments in an SCIR search session.
- Timing information – this represents the time, in seconds, relative to the start of the search session, at which relevance judgments are made. This timing information is used to order relevance judgments in an SCIR simulation and allows us to model SCIR sessions in which collaborating searchers are providing relevance information at distinct times and at different rates in the process.

This simulated environment will allow us to explore the effects of both a division of labour and sharing of knowledge policy on a collaborating group. The incremental relevance feedback process will provide multiple iterations of ranked lists being returned to users, which will allow us to explore the effects of a division of labour over the course of an entire search. The effects of knowledge sharing can also be explored through implementing the techniques outlined in the previous chapter on the users' relevance judgments.

### 5.3.2 Methodology

Having outlined the requirements for an SCIR simulation, we will now describe how we populated our simulations using data from previous TREC interactive search experiments.

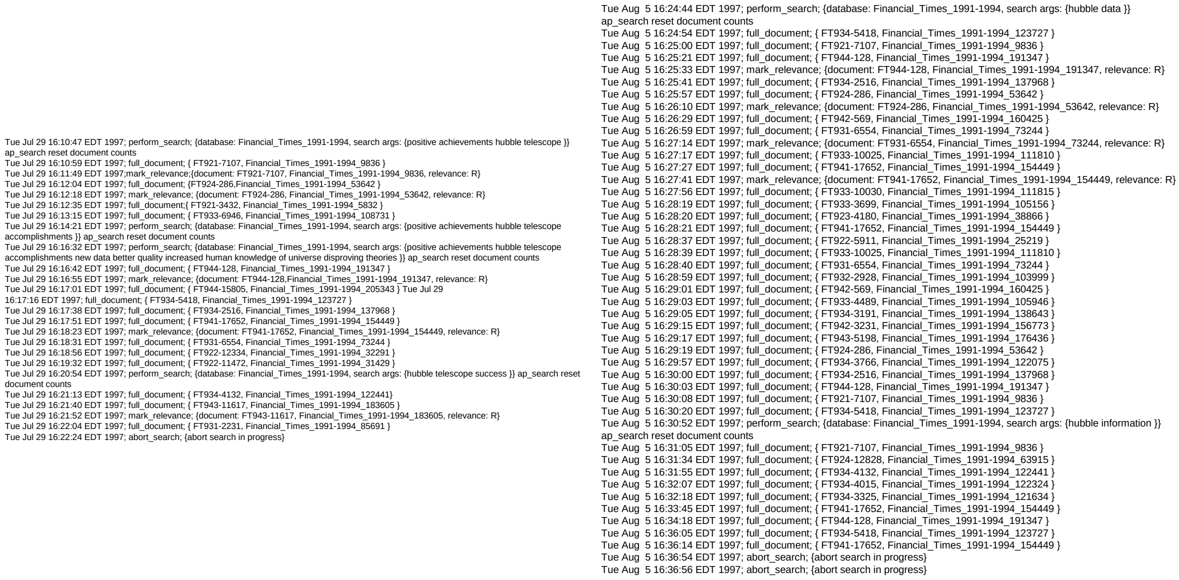
#### 5.3.2.1 Populating Simulations using TREC Rich Format Data

The purpose of the TREC (Text REtreival Conference, see Chapter 2 for details) interactive search task is for a searcher to locate documents of relevance to a stated information need (a search "topic") using a search engine and to save them. The interactive search track at TREC 6, TREC 7 and TREC 8 was an investigation of searching as an interactive task by examining the "process as well as the outcome" (NIST, 1997). Therefore, each participating group that submitted results for evaluation was required to also include *rich format data* with their submission. This data consisted of transcripts of a searcher's significant events during a search, such as their initial query, the documents saved (i.e. relevance judgments), and their timing information.

Figure 5.2 shows the rich format data from two users who completed topic 303i, entitled "Hubble Telescope Achievements", as part of the University of Massachusetts TREC 6 submission. We can identify events such as queries (*perform_search*), relevance judgments (*mark_relevance*), and timing information (*16:22:24*). Here we can see that the user on the left of Figure 5.2 ("user 1") began their search by entering the query *"positive achievements hubble telescope"*. After 62 seconds the user indicated that document *FT921-7107* was relevant and the search continued with the user providing a further 4 relevance judgments until the search session finished after 697 seconds. From the rich format data on the right-hand side of Figure 5.2 , we can see that this user ("user 2") began their search by querying "hubble data", they made their first relevance judgment on document *FT944-128* after 49 seconds and made a further 3 relevance judgments until their search finished after 368 seconds.

Originally, these users would have performed these topic searches independently as part of their group's TREC submission, but for our evaluations we simulate these two users searching together. In order to simulate these users searching at the same time, we synchronise their session start times by aligning the times for their initial query. We then arrange the relevance judgments of the two users in time-order using the timing offsets from each user's data. In order to formulate an initial search query for the group we concatenate all unique terms from the users' original querys.

Using the notation used earlier, Figure 5.3 shows an SCIR session involving the



```
Tue Jul 29 16:10:47 EDT 1997; perform_search; {database: Financial_Times_1991-1994, search args: {positive achievements hubble telescope }}
ap_search reset document counts
Tue Jul 29 16:10:59 EDT 1997; full_document; { FT921-7107, Financial_Times_1991-1994_9836 }
Tue Jul 29 16:11:49 EDT 1997;mark_relevance;{document: FT921-7107, Financial_Times_1991-1994_9836, relevance: R}
Tue Jul 29 16:12:04 EDT 1997; full_document; {FT924-286,Financial_Times_1991-1994_53642 }
Tue Jul 29 16:12:18 EDT 1997; mark_relevance; {document: FT924-286, Financial_Times_1991-1994_53642, relevance: R}
Tue Jul 29 16:12:35 EDT 1997; full_document;{ FT921-3432, Financial_Times_1991-1994_5832 }
Tue Jul 29 16:13:15 EDT 1997; full_document; { FT933-6946, Financial_Times_1991-1994_108731 }
Tue Jul 29 16:14:21 EDT 1997; perform_search; {database: Financial_Times_1991-1994, search args: {positive achievements hubble telescope
accomplishments }} ap_search reset document counts
Tue Jul 29 16:16:32 EDT 1997; perform_search; {database: Financial_Times_1991-1994, search args: {positive achievements hubble telescope
accomplishments new data better quality increased human knowledge of universe disproving theories }} ap_search reset document counts
Tue Jul 29 16:16:42 EDT 1997; full_document; { FT944-128, Financial_Times_1991-1994_191347 }
Tue Jul 29 16:16:55 EDT 1997; mark_relevance; {document: FT944-128,Financial_Times_1991-1994_191347, relevance: R}
Tue Jul 29 16:17:01 EDT 1997; full_document; { FT944-15805, Financial_Times_1991-1994_205343 } Tue Jul 29
16:17:16 EDT 1997; full_document; { FT934-5418, Financial_Times_1991-1994_123727 }
Tue Jul 29 16:17:38 EDT 1997; full_document; { FT934-2516, Financial_Times_1991-1994_137968 }
Tue Jul 29 16:17:51 EDT 1997; full_document; { FT941-17652, Financial_Times_1991-1994_154449 }
Tue Jul 29 16:18:23 EDT 1997; mark_relevance; {document: FT941-17652, Financial_Times_1991-1994_154449, relevance: R}
Tue Jul 29 16:18:31 EDT 1997; full_document; { FT931-6554, Financial_Times_1991-1994_73244 }
Tue Jul 29 16:18:56 EDT 1997; full_document; { FT922-12334, Financial_Times_1991-1994_32291 }
Tue Jul 29 16:19:32 EDT 1997; full_document; { FT922-11472, Financial_Times_1991-1994_31429 }
Tue Jul 29 16:20:54 EDT 1997; perform_search; {database: Financial_Times_1991-1994, search args: {hubble telescope success }} ap_search reset
document counts
Tue Jul 29 16:21:13 EDT 1997; full_document; { FT934-4132, Financial_Times_1991-1994_122441}
Tue Jul 29 16:21:40 EDT 1997; full_document; { FT943-11617, Financial_Times_1991-1994_183605 }
Tue Jul 29 16:21:52 EDT 1997; mark_relevance; {document: FT943-11617, Financial_Times_1991-1994_183605, relevance: R}
Tue Jul 29 16:22:04 EDT 1997; full_document; { FT931-2231, Financial_Times_1991-1994_85691 }
Tue Jul 29 16:22:24 EDT 1997; abort_search; {abort search in progress}
```

```
Tue Aug  5 16:24:44 EDT 1997; perform_search; {database: Financial_Times_1991-1994, search args: {hubble data }}
ap_search reset document counts
Tue Aug  5 16:24:54 EDT 1997; full_document; { FT934-5418, Financial_Times_1991-1994_123727 }
Tue Aug  5 16:25:00 EDT 1997; full_document; { FT921-7107, Financial_Times_1991-1994_9836 }
Tue Aug  5 16:25:21 EDT 1997; full_document; { FT944-128, Financial_Times_1991-1994_191347 }
Tue Aug  5 16:25:33 EDT 1997; mark_relevance; {document: FT944-128, Financial_Times_1991-1994_191347, relevance: R}
Tue Aug  5 16:25:41 EDT 1997; full_document; { FT934-2516, Financial_Times_1991-1994_137968 }
Tue Aug  5 16:25:57 EDT 1997; full_document; { FT924-286, Financial_Times_1991-1994_53642 }
Tue Aug  5 16:26:10 EDT 1997; mark_relevance; {document: FT924-286, Financial_Times_1991-1994_53642, relevance: R}
Tue Aug  5 16:26:29 EDT 1997; full_document; { FT942-569, Financial_Times_1991-1994_160425 }
Tue Aug  5 16:26:59 EDT 1997; full_document; { FT931-6554, Financial_Times_1991-1994_73244 }
Tue Aug  5 16:27:14 EDT 1997; mark_relevance; {document: FT931-6554, Financial_Times_1991-1994_73244, relevance: R}
Tue Aug  5 16:27:17 EDT 1997; full_document; { FT933-10025, Financial_Times_1991-1994_111810 }
Tue Aug  5 16:27:27 EDT 1997; full_document; { FT941-17652, Financial_Times_1991-1994_154449 }
Tue Aug  5 16:27:41 EDT 1997; mark_relevance; {document: FT941-17652, Financial_Times_1991-1994_154449, relevance: R}
Tue Aug  5 16:27:56 EDT 1997; full_document; { FT933-10030, Financial_Times_1991-1994_111815 }
Tue Aug  5 16:28:19 EDT 1997; full_document; { FT933-3699, Financial_Times_1991-1994_105156 }
Tue Aug  5 16:28:20 EDT 1997; full_document; { FT923-4180, Financial_Times_1991-1994_38866 }
Tue Aug  5 16:28:21 EDT 1997; full_document; { FT941-17652, Financial_Times_1991-1994_154449 }
Tue Aug  5 16:28:37 EDT 1997; full_document; { FT922-5911, Financial_Times_1991-1994_25219 }
Tue Aug  5 16:28:39 EDT 1997; full_document; { FT933-10025, Financial_Times_1991-1994_111810 }
Tue Aug  5 16:28:40 EDT 1997; full_document; { FT931-6554, Financial_Times_1991-1994_73244 }
Tue Aug  5 16:28:59 EDT 1997; full_document; { FT932-2928, Financial_Times_1991-1994_103999 }
Tue Aug  5 16:29:01 EDT 1997; full_document; { FT942-569, Financial_Times_1991-1994_160425 }
Tue Aug  5 16:29:03 EDT 1997; full_document; { FT933-4489, Financial_Times_1991-1994_105946 }
Tue Aug  5 16:29:05 EDT 1997; full_document; { FT934-3191, Financial_Times_1991-1994_138643 }
Tue Aug  5 16:29:15 EDT 1997; full_document; { FT942-3231, Financial_Times_1991-1994_156773 }
Tue Aug  5 16:29:17 EDT 1997; full_document; { FT943-5198, Financial_Times_1991-1994_176436 }
Tue Aug  5 16:29:19 EDT 1997; full_document; { FT924-286, Financial_Times_1991-1994_53642 }
Tue Aug  5 16:29:57 EDT 1997; full_document; { FT934-3766, Financial_Times_1991-1994_122075 }
Tue Aug  5 16:30:00 EDT 1997; full_document; { FT934-2516, Financial_Times_1991-1994_137968 }
Tue Aug  5 16:30:03 EDT 1997; full_document; { FT944-128, Financial_Times_1991-1994_191347 }
Tue Aug  5 16:30:08 EDT 1997; full_document; { FT921-7107, Financial_Times_1991-1994_9836 }
Tue Aug  5 16:30:20 EDT 1997; full_document; { FT934-5418, Financial_Times_1991-1994_123727 }
Tue Aug  5 16:30:52 EDT 1997; perform_search; {database: Financial_Times_1991-1994, search args: {hubble information }}
ap_search reset document counts
Tue Aug  5 16:31:05 EDT 1997; full_document; { FT921-7107, Financial_Times_1991-1994_9836 }
Tue Aug  5 16:31:34 EDT 1997; full_document; { FT924-12828, Financial_Times_1991-1994_63915 }
Tue Aug  5 16:31:55 EDT 1997; full_document; { FT934-4132, Financial_Times_1991-1994_122441 }
Tue Aug  5 16:32:07 EDT 1997; full_document; { FT934-4015, Financial_Times_1991-1994_122324 }
Tue Aug  5 16:32:18 EDT 1997; full_document; { FT934-3325, Financial_Times_1991-1994_121634 }
Tue Aug  5 16:33:45 EDT 1997; full_document; { FT941-17652, Financial_Times_1991-1994_154449 }
Tue Aug  5 16:34:18 EDT 1997; full_document; { FT944-128, Financial_Times_1991-1994_191347 }
Tue Aug  5 16:36:05 EDT 1997; full_document; { FT934-5418, Financial_Times_1991-1994_123727 }
Tue Aug  5 16:36:14 EDT 1997; full_document; { FT941-17652, Financial_Times_1991-1994_154449 }
Tue Aug  5 16:36:54 EDT 1997; abort_search; {abort search in progress}
Tue Aug  5 16:36:56 EDT 1997; abort_search; {abort search in progress}
```

Figure 5.2: Rich format output from the University of Massachusetts TREC 6 submission for two users

two users whose rich format data is shown in Figure 5.2, searching on TREC topic 303i. From Figure 5.3, we can see that the search begins with the group query "positive achievements hubble telescope data", which is the concatenation of both users' original querys. By the time user 1 provides their first relevance judgment on document *FT921-7107*, user 2 has already provided a relevance judgment, on document *FT944-128*. By the time user 2 makes their second relevance judgment on document *FT924-286*, user 1 has made their first relevance judgment on *FT921-7107*.

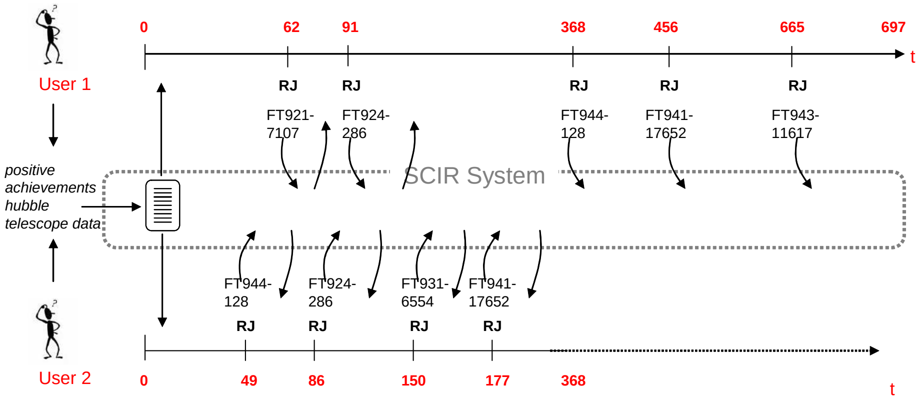


Figure 5.3: Conceptual overview of two searchers searching together

By extracting rich format data associated with different users' interactions on a search topic, we can construct multiple heterogenous simulations, where the data populating our simulations is from real users searching to satisfy the same information need on a standardised corpus. By aligning these users interactions, we can simulate a synchronised search session where an SCIR system can coordinate the ranked lists returned to these users' in order to explore division of labour and sharing of knowledge.

#### 5.3.2.2 Dynamic Relevance Judgments

The SCIR simulations proposed thus far are based on taking *static* rich format data, which records a users' previous interactions with a particular search engine, and imposing our *dynamic* SCIR simulated environment on this data. The data used to populate our simulations come from a variety of group's TREC submissions (as will be outlined in section 5.4.1), each of which would have implemented an individual search system and interface in their experiments. In our simulations we have applied this data to our own SCIR system, we implement a back-end search engine (as will be described in section 5.4.4) and we simulate two users searching together through an SCIR system which modifies results returned to users in order to implement division of labour and sharing of knowledge policies.

By imposing our own simulated environment on this rich format data, we cannot assume that users would have saved the same documents as they did during their original search, as recorded in the rich format data. Before we can proceed with our simulations we need to replace these static relevance judgments with dynamic relevance judgments based on the ranked lists that simulated users are presented with. Although in any simulation we can never predict with absolute certainty the actions of a user, it is important that the simulated relevance judgments are a reasonable approximation of real user behaviour.

Our solution is to simulate the user providing a relevance judgment on the first relevant document, i.e. highest ranked, on their current ranked list, where the relevance of the documents is judged according to the TREC relevance assessments for the topic ("qrels"). Although we can never be fully certain that a user will always save the *first* relevant document that they encounter on a ranked list (i.e. rather than the second or third), recent studies have shown that users tend to examine search results from top to bottom, "deciding to click each result before moving to the next" (Craswell et al., 2008). Therefore we believe that this approximation of a real user's action is reasonable.

Assuming that users will only provide relevance judgments on relevant documents is modelling a *best case scenario*, where users always recognise relevant material in the search. As outlined in the previous chapter, in SCIR search, users may make mistakes in their relevance judgments. Our analysis of the rich format data used in our simulations reveals that approximately 37% of all documents saved by real users in the original TREC submissions were non-relevant. Although these figures of user-incompetence are probably exaggerated due to the fact that users were searching TREC topics rather than searching to satisfy their own information need, in an SCIR search some users may be more familiar with a topic than others and this may cause poor relevance assessments to be made. We believe that it is important to model such an environment in order to explore the effects of poor relevance judgments on the performance of a collaborating group. Therefore we will also experiment with introducing noise into the relevance assessments, by simulating users providing relevance judgments on non-relevant documents. One way to introduce this noise would be a simple extension to the best-case scenario and to simulate the user saving either the first relevant or non-relevant document in the ranked list, whichever comes first. However, it may be overly simplistic to assume that a user will save any non-relevant document providing it comes above a relevant document in the list. What we do instead is simulate users making a relevance judgment on the first relevant or *perceived relevant* document in the ranked list, whichever comes first. These perceived relevant documents are non-relevant documents that were saved by at least two real users in the original TREC rich format data used in our simulations. The reason for using these documents, over any arbitrary non-relevant document occurring in the ranked list, is that as these documents were originally saved by at least two real users during their TREC experiments, they may resemble relevant material from a user's perspective, unlike non-relevant material at the top of a list which may match a system query but does not resemble relevant material from a user's perspective.

Before finalising our simulations, we also need to enforce an upper limit on the number of documents a simulated user will examine in order to locate a document on which to provide a relevance judgment. For example, it would be unreasonable to assume that a user would look down as far as rank *900* in the ranked list in order to find a relevant document. Instead, we limit the number of documents that a simulated user will examine to the top 30 documents in their ranked list. Although in a real world system, users may be willing to examine more or fewer documents according to the device they are using for searching (e.g. personal computer, tabletop device), we feel that 30 is a reasonable figure for most SCIR simulations. If a simulated user does not find a document on which to provide a relevance judgment in the top 30, they do not provide a relevance judgment and are instead returned the next 30 documents to examine, this can allow documents that were beyond the top 30 in one iteration to be found in later iterations.

After performing relevance feedback, the relevance judgments made by a user are never returned to them again for the duration of the search. Depending on the division of labour policy these documents may also be removed from their search partner's ranked lists, and we will investigate this in the next chapter.

In this section we have described how we propose to simulate a synchronous collaborative information retrieval environment by populating a collaborative search session with data from TREC rich format data. In the next section we will describe the total number of simulations used in our evaluations, we will also describe the search topics used and the underlying search system implementation that supported our evaluations.

## 5.4 Test Environment

In order to perform a comprehensive evaluation of the effects of a system-mediated coordination of users in an SCIR search, a large and diverse set of simulated sessions were needed which captured different types of users searching on several example search topics.

### 5.4.1 Rich Format Data Extraction and Analysis

As outlined earlier, our simulations are populated using rich format data from previous TREC interactive experiments. In particular, we extract data from various group's submissions to TREC 6 to TREC 8. Unfortunately, it was not possible to extract data for our simulations from each participating group's submissions to these TRECs. This was due to a variety of reasons but typically these groups had either failed to submit rich format data or the data they submitted was not complete as it lacked some of the data needed to build our simulations. Despite groups using different schemas for recording a searcher's interactions[1] across groups the data roughly followed the same outline of an initial query followed by multiple relevance judgments and potentially some intermediate query reformulations.

The breakdown for all rich format data extracted and used in simulations described in this thesis can be seen in Table 5.1. TREC 6, 7 and 8 used 6, 8, and 6 topics respectively and we will describe these topics in greater detail in section 5.4.2. In Table 5.1, the figures in each cell correspond to the number of single-user interactive search sessions per topic performed by a group in the original TREC submission. For example, in the Berkeley TREC 6 (BRK_T6) submission each topic was searched twice.

As we can see from Figure 5.1, the rich format data used in our simulations is particularly noisy, with topics having an uneven number of runs and some topics, those from TREC 8, being completed by one group. Therefore special care was

[1] some phrases used for recording relevance judgments included: "Record Selected as Relevant" (BRK_T6) or "mark_relevance" (UMASS_T6_Z)

| | TREC 6 | | | | | | TREC 7 | | | | | | | | TREC 8 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Submission Id/Topic | 303 | 307 | 322 | 326 | 339 | 347 | 352 | 353 | 357 | 362 | 365 | 366 | 387 | 392 | 408 | 414 | 428 | 431 | 438 | 446 |
| BRK_T6 | 2 | 2 | 2 | 2 | 2 | 2 | | | | | | | | | | | | | | |
| RMIT_T6_MG | 2 | 2 | 2 | 2 | 2 | 2 | | | | | | | | | | | | | | |
| RMIT_T6_Z | 2 | 2 | 2 | 2 | 2 | 2 | | | | | | | | | | | | | | |
| UMASS_T6_AI | 6 | 6 | 6 | 6 | 6 | 6 | | | | | | | | | | | | | | |
| UMASS_T6_AIP | 6 | 6 | 6 | 6 | 6 | 6 | | | | | | | | | | | | | | |
| UMASS_T6_Z | 8 | 8 | 8 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| BRK_T7 | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | | | | | | |
| Toronto_T7_E | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | | | | | | |
| Toronto_T7_C | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | | | | | | |
| BRK_T8 | | | | | | | | | | | | | | | 7 | 4 | 4 | 4 | 7 | 7 |
| Total | 26 | 26 | 26 | 26 | 26 | 26 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 7 | 4 | 4 | 4 | 7 | 7 |

Table 5.1: Rich format data used in simulations

needed when averaging across results to ensure that no group or topic bias was introduced, and this will be explained later (in section 5.5.3).

In order to simulate an SCIR session involving two users, for each topic from each group, we performed a pairwise combination of the rich format data. For example, referring to Table 5.1, we see that 8 users completed topic 303i for the UMASS_T6_Z submission. In order to simulate SCIR sessions involving each of these eight users searching with each other (i.e. user 1 with user 2, user 1 with user 3 etc.), 28 SCIR sessions were created by combining the users' rich format data.

From the extracted data used in our simulations Table 5.2 shows, for each group, the mean session time and standard deviation, and the mean number of relevance judgments made per search and standard deviation. As we can see from these figures we have a wide spread of values for both session time and number of relevance judgments both within and across the groups. Having such a wide spread of data allows us to evaluate the performance of our systems in a number of different operational situations.

### 5.4.2 Topics

The interactive track of TREC 6 to TREC 8 used twenty search topics in total. Table 5.3 shows the topic id, title, and number of relevant documents for each topic. These topics represent a wide range of search themes, with differing numbers of relevant documents and search difficulty.

These interactive tracks encouraged searchers to find as many different "aspects"

| Group | Mean Session Time | Std. Dev | Mean # Relevance Judgments | Std. Dev |
|---|---|---|---|---|
| BRK_T6 | 951.67 | 141.76 | 7.5 | 4.15 |
| RMIT_T6_MG | 919.67 | 261.78 | 4.92 | 3.42 |
| RMIT_T6_Z | 894.92 | 180.15 | 7.17 | 5.08 |
| UMASS_T6_AI | 914 | 236.01 | 6.31 | 4.39 |
| UMASS_T6_AIP | 818.64 | 287.22 | 4.81 | 3.18 |
| UMASS_T6_Z | 742.27 | 238.06 | 5.58 | 3.86 |
| BRK_T7 | 647.44 | 199.21 | 5.69 | 4.62 |
| Toronto_T7_E | 728.16 | 146.04 | 5.56 | 3.14 |
| Toronto_T7_C | 767 | 110.77 | 4.63 | 1.84 |
| BRK_T8 | 960.58 | 306.15 | 9.33 | 5.38 |

Table 5.2: Mean and standard deviation of session times and number of relevance judgments per session for each group

(TREC 6) or "instances" (TREC 7, 8) of a topic as possible. The topic description for each of the topics therefore included a disclaimer that indicated to searchers that a document need only be saved if it contained a new aspect or instance not already saved. For example, the full topic description from topic 303i from TREC 6 is shown in Figure 5.4. This results in a smaller set of relevance judgments per search and consequently a smaller number in our simulations, than would be expected if users were told to save all documents that are relevant. We believe that this smaller number of relevant judgments more closely resembles a real world search environment. As discussed in section 5.3.1, it would be unreasonable to assume that users will provide many relevance assessments during a search.

### 5.4.3 Document Collection

All experiments for TREC 6 to TREC 8 were carried out on the TREC Financial Times of London 1991-1994 Collection (TREC disk 4), a subset of the adhoc search collection, therefore this was the search corpus used in our evaluations. The collection consists of 210,158 documents (approximately 560 mbs in size) pertaining to newspaper stories, the median number of terms per doc is 316, and mean is 412.7. The collection was installed directly from a cd-rom distributed from NIST.

```
<num> Number: 303i

<title> Hubble Telescope Achievements

<desc> Description: Identify positive accomplishments of the
Hubble telescope since it was launched in 1991.

<narr> 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.

<aspects> Aspects: Please save at least one RELEVANT document that
identifies EACH DIFFERENT positive accomplishment of the sort
described above. If one document discusses several such
accomplishments, then you need not save other documents that repeat
those aspects, since your goal is to identify different positive
accomplishments of the sort described above.
```

Figure 5.4: Topic description of TREC interactive topic 303i

| Topic Id | Title | Number of Relevant Documents |
|---|---|---|
| TREC 6 | | |
| 303i | Hubble Telescope Achievements | 6 |
| 307i | New Hydroelectric Projects | 81 |
| 322i | International Art Crime | 9 |
| 326i | Ferry Sinkings | 45 |
| 339i | Alzheimer's Drug Treatment | 6 |
| 347i | Wildlife Extinction | 50 |
| TREC 7 | | |
| 352i | British Chunnel impact | 246 |
| 353i | Antartica Exploration | 122 |
| 357i | territorial waters dispute | 270 |
| 362i | human smuggling | 3 |
| 365i | El Nino | 35 |
| 366i | commercial cyanide uses | 99 |
| 387i | radioactive waste | 85 |
| 392i | robotics | 105 |
| TREC 8 | | |
| 408i | tropical storms | 37 |
| 414i | Cuba, sugar, exports | 14 |
| 428i | declining birth rates | 50 |
| 431i | robotic technology | 49 |
| 438i | tourism, increase | 89 |
| 446i | tourists, violence | 68 |

Table 5.3: Topics used in evaluations

### 5.4.4 System Implementation and Back-End Search Engine

The rich-text parser and simulator with collaborative SCIR system were developed in Java. For a back-end search system our simulation used the Okapi Basic Search System (BSS), part of the "Okapi Pack" developed at City University London. The BSS system implements the BM25 weighting model as described in Chapter 2 (section 2.3.3). The Okapi team have participated in TREC since the original TREC 1 workshop, and the BSS system has been used in all of the team's TREC experiments (Robertson et al., 1992, 1993, 1994, 1995; Hancock-Beaulieu et al., 1996; Walker et al., 1997; Robertson et al., 1998; Robertson and Walker, 1999). For our experiments both the simulator and BSS system run on the Ubuntu operating system. The BSS handles all search queries on the document collection and all calls from the simulator to the BSS system are made through a series of perl scripts.

### 5.4.5 Parameter Tuning

Before using the Okapi BSS system for simulations, the BM25 parameter values of k1 and b were tuned together in order to find their optimal values on the collection. The tuning procedure followed was to use the title field of the 100 topics from the TREC 6 and 7 adhoc track as queries to the BSS system with certain combinations of values for k1 and b. The values tested for k1 were between 0 – 2 in 0.2 increments and the values of b tested were between 0 – 1 in 0.1 increments. Having executed a query, the returned ranked list was evaluated using the "trec_eval" evaluation software from NIST (trec_eval, 2008). The trec_eval software produces a series of measurements on a ranked list of documents when supplied with a ground truth file containing the documents judged as relevant to the topic by the TREC topic accessors (known as *qrels*). For our tuning runs the measure we examined was average precision (AP). Having executed all 50 queries, the AP figures were averaged across all queries to arrive at final values for the k1 and b combination. Overall, values of 0.8 and 0.2 for k1 and b respectively were found to give the best performance in terms of average precision, and were therefore chosen as the BM25 parameters for our experiments.

We also tuned the relevance feedback operation on the test collection. In accordance with common usage of query expansion Robertson and Spärck Jones (1997), in all our experiments the original query terms (i.e. those entered by the users) are up-weighted to reflect their relative importance in a relevance feedback query, as being the only terms entered by the users directly. Through tuning we also needed to determine the optimal value for the number of terms with which to expand a user's query. Therefore we tuned the relevance feedback operation in order to find the best combination of the number of terms to add to the query and the number of times to up-weight the original query terms. The parameter values tested were, for query up-weighting: 1, 2, 3, 4, 5, 5.5, and, for number of terms to add: 5, 10, 15, 20, 25, 30, 40, 60. To tune the relevance feedback parameters, we took the same 100 topic titles, as used for the k1 and b parameter tuning runs, as initial queries to the BSS. Then, on the returned list of documents we simulated a user looking down through the ranked list and providing a relevance judgment on the first relevant document they encounter, i.e. we simulate an *incremental feedback session* operating under the best case relevance scenario. A relevance feedback iteration was then initiated with new terms added to the query whilst up-weighting the original title query terms according to the parameter values under evaluation. This RF query was issued to the search engine and a new ranked list was returned to the simulated user. Before evaluating the new ranked list, we first performed frozen ranking (see Chapter 2, section 2.6.3) in order to remove the "ranking effect", associated with relevance feedback evaluation by keeping the position of the document used for feedback in the same position it was in the original ranked list before the relevance feedback operation. This frozen list was then evaluated using the trec_eval software and the list's average precision was measured. The tuning simulation proceeded in this manner, with the simulated user looking down through each new ranked list returned to them and providing a relevance judgment on the first new relevant document they encounter until either the simulated user had used all of the relevant documents for feedback or the list contained no new relevant documents on which to perform feedback. The average precision of the ranked list was then evaluated and plotted after each RF iteration in order to show the performance over the course of the incremental feedback search. Using this approach for each combination of the parameter pairings under investigation we found that a value of 20 for the number of terms to add to the query and a value of 5 for the number of times to up-weight the original query provided the best average precision score at each RF iteration.

So far, we have described how our SCIR simulations are implemented, by using rich format data from TREC submissions, and simulating two users searching together at the same time and making relevance judgments on documents. We have also described the total amount of rich format data used in our simulations. Next we will describe how we evaluate the performance of a group of users searching together synchronously.

## 5.5 Evaluation Procedure

The novel SCIR environment presents challenges for developing effective evaluation techniques. Firstly, we need to decide on how to measure the performance of a group of users at any point in the search. Then, as we are investigating the effects of various division of labour and sharing of knowledge policies over the duration of an entire search, we need to decide on the particular points in the search at which to extract a measurement of performance and the frequency of this measurement. In this section we will explore these issues.

### 5.5.1 Generating a Group Score

At each stage in an SCIR session, each collaborating user will have associated with him/her a ranked list of documents. In our simulations, as described in this chapter, this list could have been returned to a simulated user either as a result of the initial query or after performing relevance feedback. In traditional, single-user, information retrieval the accuracy of ranked lists can be evaluated using standard IR measurements such as average precision (AP). In our work we are concerned with the performance of a *group* of users and therefore we need to be able assign a score to the collaborating group at any particular point in the search process.

One potential method for generating this *group score*, would be to evaluate the quality of each collaborating searcher's list using a standard IR measure like AP then average these values across group members to get the average score for the group. For example, if at a particular point in a collaborating group consisting of two users, one user had, associated with them, a ranked list with an average precision of 0.3 and the other user had a ranked list with an average precision of 0.5 and then averaging these two figures, would result in a group score of 0.4. Unfortunately, this approach of generating a group score does not adequately measure the group's performance as no attempt is made to examine the contents of the users' ranked lists and, in particular, the amount of overlap between them. To illustrate this further, if two separate collaborating groups of users have the same associated group score, arrived at by averaging the AP of each group member's ranked list, but the members of the first group had ranked lists which contained many of the same documents, while the second group had ranked lists with a greater diversity of relevant documents, then the performance of the second collaborating group should be considered better than the first as, across the group, the total amount of relevant material found across collaborating users' lists is greater in the second group. By simply averaging each individual's AP scores, however, this information is lost.

What we need instead is a measure which captures the quality and diversity across collaborating users' ranked lists. Our solution is to measure the *total number of unique relevant documents across user's ranked lists* at a certain cutoff and use this figure as our *group score*. In our simulations as we assume that users will examine the top 30 documents in the ranked list (see section 5.3.2.2), our measure of quality is taken at a cutoff of 30 documents from each user's list. This performance measure will enable us to capture both the quality and diversity across collaborating users' ranked lists and in particular the parts of the list that they will examine.

As described earlier, in our simulations, before returning a new ranked list to a searcher, all relevance judgments made by the searcher are removed. For the purposes of calculating the group score we also include these saved documents in the calculation.

### 5.5.2 Measurement Granularity

In order to capture the performance of a group over the entire search, we need to measure the group score after each significant event in the search which causes a new ranked list to be returned to either searcher. In our simulations of SCIR search, a user receives a new ranked list after the initial group query and then after they make a relevance judgment. Taking a measurement of the group performance after each of these events allows us to capture the change in group performance over the course of a search.

Figure 5.5 illustrates the procedure followed in our evaluations in order to calculate the performance of a group over the duration of an SCIR simulation. The SCIR simulation begins with a shared query, at this point we measure the total number of relevant documents in the top 60 positions of this list (top 30 for each user). As this figure represents the initial group score before any relevance feedback is provided to the system, it is plotted at position 0 on the x-axis of the graph at the bottom of Figure 5.5. The first relevance feedback iteration is initiated after user 2 provides a relevance judgment after 63 seconds. At this point, user 2's current list is updated as a result of a feedback iteration, however user 1 is still viewing the results of the initial query. We calculate the group score at this point by counting the total number of unique relevant documents across these two ranked lists (labelled "GS"). As this is the first relevance feedback iteration in the SCIR session, this group score is plotted at position 1 on the graph. The measurement proceeds in this manner, by calculating the group score after each relevance judgment in order to show the group's performance over the course of the entire search.

### 5.5.3 Averaging Group Scores

The evaluation procedure described thus far enables us to plot the performance of a single group of users over the over the course of a search. In order to evaluate how these techniques perform when applied to many different groups of users, such as the entire set of simulations used in our evaluations, we need to average together the group scores from multiple groups of users.

Figure 5.6 illustrates the different levels of averaging performed during our evaluations. Each black arrowed line corresponds to an individual simulated SCIR session (i.e. "1"). Each dash on these lines represents a relevance feedback iteration with a corresponding group score calculated as described above. In order to calculate the average performance of all runs from a group submission (e.g. "BRK_T6"), we average the group scores at each iteration for all simulations in this topic from the group submission. Referring to Figure 5.6, we average down through each black line

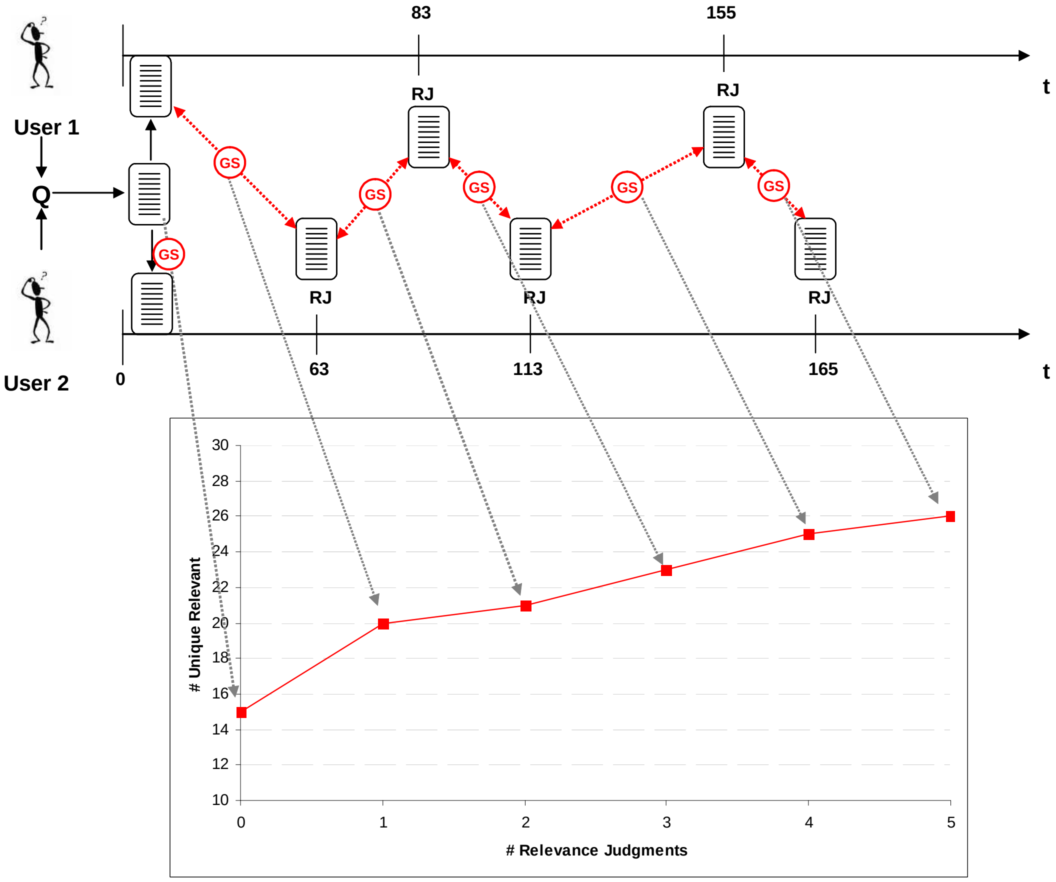


Figure 5.5: Measurement granularity used in experiments

to arrive at the dotted blue line.

To get the performance for a topic, represented by a red line with crosses, we average together the average values of all group submissions (i.e. all dotted blue lines). Finally, to arrive at an overall score (i.e. the orange line) we average together all scores from all topics (i.e. all red lines with crosses).

By using this approach to averaging values, we can ensure that no one group or topic biases the overall results for topic scores, or overall scores. However, as each simulated SCIR session will contain different numbers of relevance judgments (represented in Figure 5.6 by the different lengths of the arrowed lines), scores from later relevance feedback iterations may be calculated based on a much smaller number of runs than those of earlier iterations and therefore may not be representative of an overall trend in the results.

For this reason, along with plotting the performance of a group over the duration of a search, in our evaluations, we also calculate a single figure performance measure for each collaborative session. This single figure is arrived at by averaging the group score at each relevance feedback iteration into one value. Referring to Figure 5.6 this single performance figure is calculated for a particular run (i.e. the black lines) by averaging horizontally across the run. This value for a pair of users can then be averaged across all groups and topics. This value allows us, in a single figure, to measure how an SCIR system performs across an *entire* SCIR search.

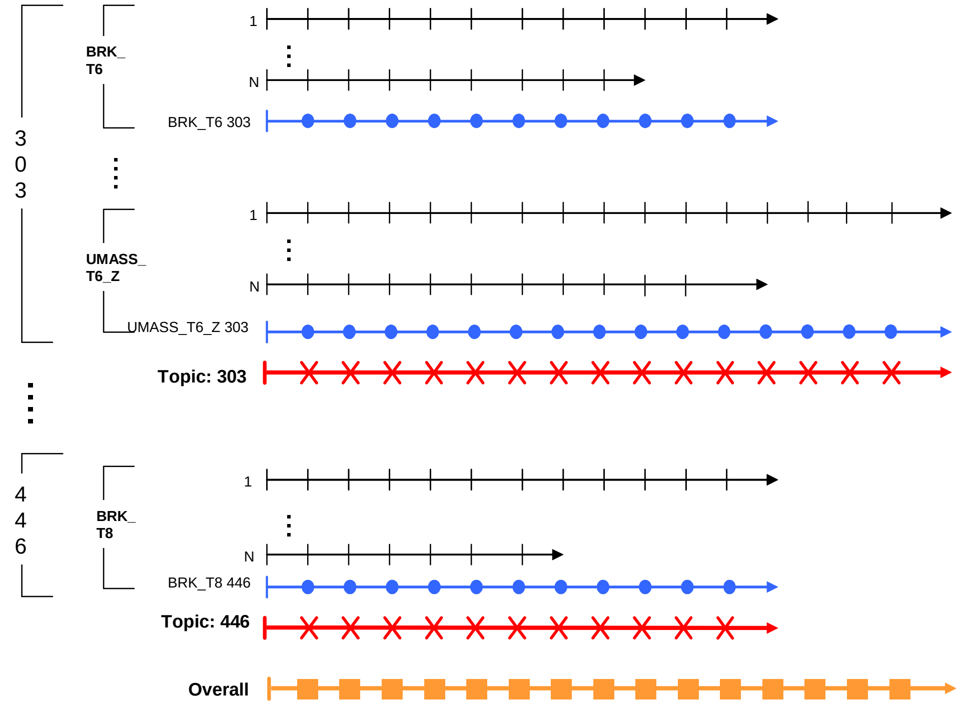


Figure 5.6: Averaging group scores

## 5.6 Summary

In this chapter we have proposed a novel methodology for evaluating the performance of synchronous collaborative information retrieval which will be used in exploring our hypothesis. We have described the simulated SCIR environment used in our experiments, including how we model users making dynamic relevance judgments on the ranked lists they are presented with. We outlined how we extracted rich format data from previous TREC interactive experiments in order to populate these simulations. Finally, we demonstrated how to evaluate the performance of a group of collaborating searchers, by measuring the number of unique relevant documents across users' ranked lists after each significant event in the search. We described how we can visualise the progress of a group of searchers after each significant event and how these figures can be averaged into overall scores at different levels of granularity.

In the next chapter we will describe the results of our evaluations using the methodology described in this chapter.

# Chapter 6

# Evaluation

## 6.1 Introduction

In this chapter we explore our hypothesis by evaluating the effects of various division of labour and sharing of knowledge techniques as proposed in Chapter 4 using the simulated SCIR sessions described in Chapter 5.

We will follow the structure as set out in Chapter 4. Firstly, in section 6.2 we will explore the effects of a division of labour policy on the performance of a group of searchers. Then in section 6.3 we will explore the effects of implementing a collaborative relevance feedback process alongside a division of labour policy in SCIR. In section 6.4, we will examine the influence of noise, in terms of non-relevance judgments, on the collaborative relevance feedback process, before evaluating a user-biased authority weighting of the process. Finally, in section 6.5, we will explore the effectiveness of a complementary relevance feedback process on an SCIR session.

## 6.2 Division of Labour

One of the great advantages of having multiple users searching together at the same time in order to satisfy a shared information need is that the search task can be divided across all collaborating searchers, often referred to as a *division of labour*.

In this thesis, we are interested in investigating how a division of labour policy can operate in a dynamic, interactive, collaborative search session. If we have an environment whereby users are searching together, then when the SCIR system returns a new ranked list to a user during the course of the search it can filter this ranked list in order to remove documents that the user's search partner has viewed or may view in the future. By removing these documents, we can reduce the amount of document overlap across group member's ranked lists. This can allow new relevant documents not seen by the either searcher to be pushed up in the ranked lists returned to each user thereby improving the performance of the group.

In order to investigate the effect of such a division of labour policy on the performance of a group of synchronous searchers, we simulate two users searching together using the SCIR incremental relevance feedback system as outlined in the previous chapter, with different types of division of labour policies. These four different variants are illustrated in Figure 6.1 and are described as:

1. No Division – a basic SCIR system in which two users come together to search with no attempt being made to limit the amount of overlap across searchers. This will result in both users from the collaborating group being presented with the same ranked lists and therefore causes much duplication of effort across the users.

2. Initial Search Result Division – the results presented to each user are filtered in order to ensure that, initially, each user sees a unique document list. The initial ranked list returned as a result of the shared query is distributed across users in a round-robin manner (i.e. first document to user 1, second document to user 2, third document to user 1 etc.). However the coordination amongst users begins and ends with the initial ranked list as no further attempts are made to limit document overlap beyond the users' initial query.

3. Initial Search Result Division and Removal of Documents Seen – initial search results are divided as in 2 above. In addition to this, when returning a new

ranked list to either user, we filter the list to remove documents that have been seen by either user. These seen documents consist of those documents *explicitly* marked as relevant, i.e. the relevance judgments, and also all documents that were located above the explicit relevance judgment in the ranked list on which the user made the judgment.

4. Full Division – as above but with an additional filtering to remove, from each user's ranked list, all documents that we assume will be examined by the user on their current ranked list. In our simulations, as described in the previous chapter, we assume that a user will look down through at most 30 documents at each iteration, we therefore filter each user's list to remove the top 30 documents in their search partner's current list. In this way we can ensure a complete division of labour, users are guaranteed to examine different documents. In our simulations a user will stop examining the list when they encounter the first relevant document, and therefore they may not examine all documents in the top 30. As a result there is the potential of loosing some relevant documents in the list beyond the document used for feedback. However these documents may be returned again to the user as a result of the relevance feedback process, and if not, then their search partner may be returned these documents in subsequent iterations.

Alongside our comparisons of the performance of these SCIR systems with increasing levels of result division, we also compare the performance of these collaborative systems with two baseline systems showing users searching independently without any collaboration in terms of division of labour. The *Independent Group* baseline will evaluate how the group of users perform without any collaboration in terms of the initial query or dividing of search results whilst the second baseline system, *Best Individual*, will show how, for each pair of users searching, the best user performs when searching on their own, using their own initial query and the incremental feedback system. Figure 6.2 shows an example of two users, *02* and

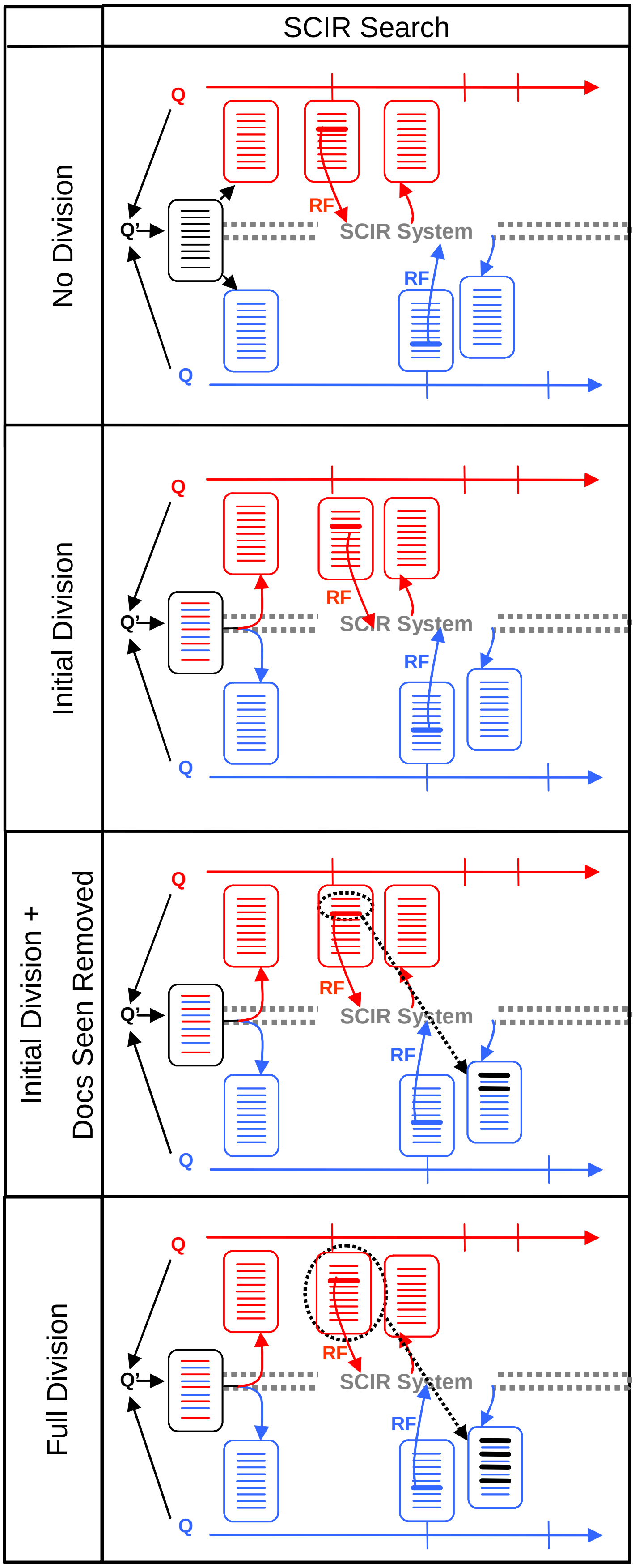
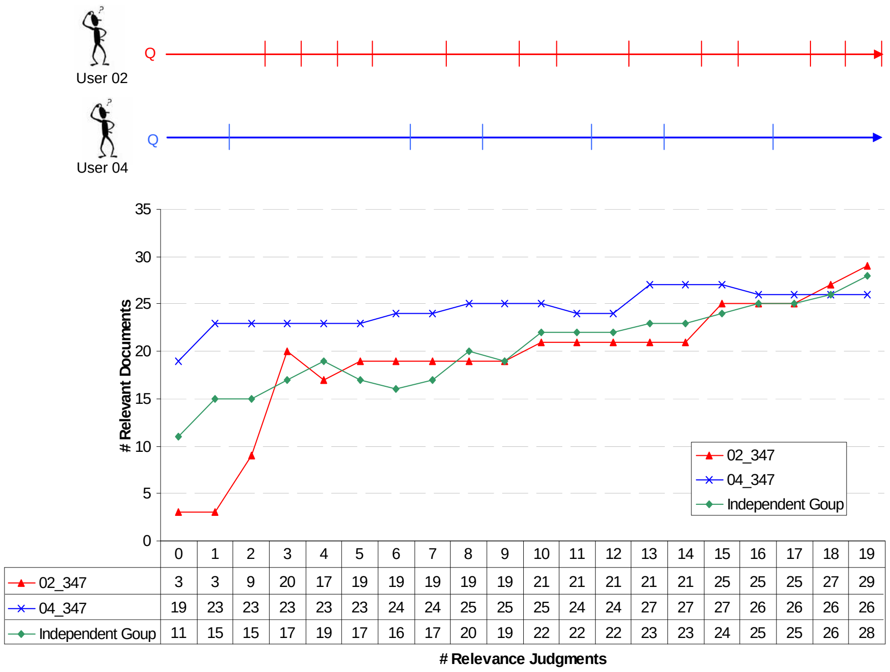


Figure 6.1: Division of Labour evaluated systems



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 02_347 | 3 | 3 | 9 | 20 | 17 | 19 | 19 | 19 | 19 | 19 | 21 | 21 | 21 | 21 | 21 | 25 | 25 | 25 | 27 | 29 |
| 04_347 | 19 | 23 | 23 | 23 | 23 | 23 | 24 | 24 | 25 | 25 | 25 | 24 | 24 | 27 | 27 | 27 | 26 | 26 | 26 | 26 |
| Independent Goup | 11 | 15 | 15 | 17 | 19 | 17 | 16 | 17 | 20 | 19 | 22 | 22 | 22 | 23 | 23 | 24 | 25 | 25 | 26 | 28 |

Figure 6.2: Baseline systems used in evaluations

*04*, searching on topic *347* in order to illustrate how the two baseline systems are constructed and evaluated.

In the *Independent Group* baseline simulation, each simulated user uses their *own* initial query (extracted from the rich-format data) in order to begin their search. They then proceed to provide a series of relevance judgments to the system which the system uses to reformulate the user's own query and return a new ranked list to the user. We then evaluate the performance of the group in the same way as per the SCIR runs, by evaluating the number of unique relevant documents across the top 30 documents in both user's ranked lists.

We now describe how we construct and evaluate the second baseline system, *Best Individual*. Firstly we need to decide on the metric for evaluating the quality of a single user's ranked list. As our evaluation metric for the SCIR runs is the number of unique relevant documents across the top 30 documents from both users, a reasonable comparison with an individual user would be the number of relevant documents in the top 60 documents on the user's ranked list. Within any collaborating group, over the course of a search each individual user will contribute a subset of the relevance judgments. Referring to the top part of Figure 6.2 we can see that user *02* has contributed 13 of the relevance judgements, whilst user *04* has contributed 6. In order to allow for the comparison of a single user searching alone versus the group over the duration of the search, we need to assign a score, i.e. number of relevant documents in top 60, for each individual searcher for relevance feedback iterations which belong to their search partner. This score should be representative of the performance of the user at a particular point in time in the search process and therefore our solution is to keep the user's score static between feedback iterations belonging to their search partner. Having decided on how to assign a score for each user at each point in the search, in order to generate our *Best Individual* baseline we choose as the best user, the user with the best average score over the duration of the entire group search. Referring again to Figure 6.2, from this group of users, user *04* would be considered as the better user, with an average of 24.5 relevant documents over the entire search versus 19.15 for user *02*.

### 6.2.1 Results

Figure 6.3 presents the results of our evaluations with the four SCIR systems implementing different division of labour policies, along with the two baseline systems of users searching independently, in terms of the total number of unique relevant documents contained across the top 30 documents from each group member's ranked list. These scores are plotted over each relevance feedback iteration, with iteration "0" representing the results of the initial search query. Firstly, as discussed in Chapter 5 (section 5.5.3), it should be noted that as the calculation of this overall figure involves averaging across many simulated runs with differing numbers of relevance judgments per search, the results at later iterations may not be as representative of an overall trend as those in earlier iterations. In Figure 6.4 we plot the number of runs, on the left, and topics, on the right, contributing to the score at each relevance feedback iteration. As we can see from the left of Figure 6.4, beyond 11 feedback iterations the score is calculated based on less than half of the total number of simulated runs and may therefore be less representative of an overall trend in performance. For this reason, in Table 6.1 we plot the single figure group score per topic. This value shows us how each system performs across an *entire* SCIR search, i.e. averaged over each relevance feedback iteration. We will use these overall figures in order to test for significant differences between systems.

In this thesis we use randomisation testing (Kempthorne and Doerfler, 1969), to test for statistical significance and use a significance threshold of $p < 0.05$. All results with p values less than this threshold are considered as *significant*.

Comparing the results from the four different SCIR systems first, we see that the SCIR system with full division of labour is the best performer, providing substantially more relevant documents than any of the other SCIR systems. Performing significance testing on the overall single figures from Table 6.1, we find that the SCIR system with full division significantly outperforms all other SCIR systems across topics, providing an average of 20.79 relevant documents across user's lists compared to 18.29 for the SCIR system with removal of documents seen, 16.69 for the SCIR system with just an initial division, and 16.04 for the basic SCIR system with no division of labour whatsoever. Comparing results across the other SCIR systems we find that the SCIR system with documents seen removed significantly outperforms both the SCIR system with an initial result division and the basic SCIR system with no division, while the SCIR system with initial result division significantly outperforms the basic SCIR system with no division. These results are not surprising and show a clear increase in performance with increasing levels of division of labour for the SCIR systems.

Next we compare the performance of the SCIR systems with the two independent baseline systems. Firstly, we find that both the SCIR system with full division and the SCIR system with just a removal of documents seen significantly outperform two users searching independently (independent group). However, the independent group baseline significantly outperforms both the basic SCIR system with no division and the SCIR system with an initial division.

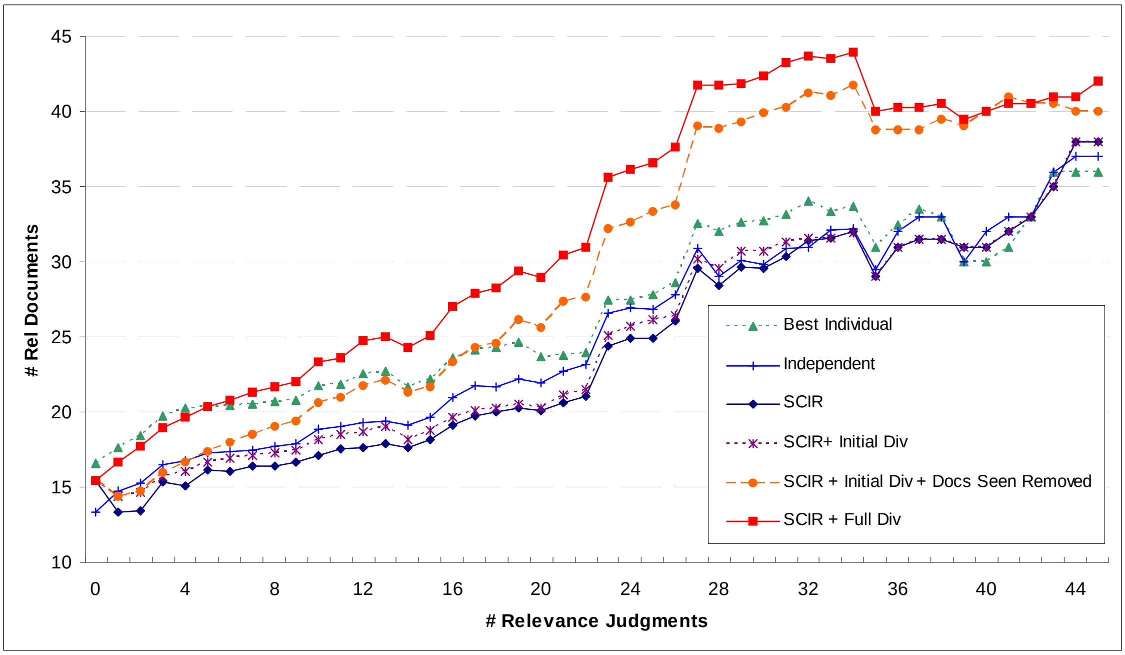


Figure 6.3: Comparison of SCIR systems with division of labour and two baseline systems

From Figure 6.3, we see the best user's initial query provides them with a better staring point to the search, allowing them to achieve better results over the first few iterations. However, beyond five iterations we find that the SCIR system with full division begins to outperform the best individual. Running significance tests over the single figure group scores we find that the best individual outperforms all SCIR systems except the SCIR system with full division. With the SCIR system with full division significantly outperforming the best individual searching alone over the entire search.

### 6.2.2 Discussion

In this section we have explored the question of how a division of labour policy effects the performance of a group of users searching together synchronously. We hypothesised that a division of labour policy could allow a group of users to search together more effectively by allowing more unique documents to be found across

| Topic # | Best Individual | Independent | SCIR | SCIR + Initial Div | SCIR + Initial Div + Docs Seen Removed | SCIR + Full Div |
|---|---|---|---|---|---|---|
| **303** | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| **307** | 36.46 | 31.73 | 27.81 | 28.60 | 32.57 | 35.75 |
| **322** | 3.49 | 3.30 | 2.56 | 2.55 | 2.54 | 2.99 |
| **326** | 33.44 | 29.15 | 28.16 | 29.96 | 31.43 | 35.43 |
| **339** | 5.95 | 5.89 | 5.89 | 5.91 | 5.93 | 5.93 |
| **347** | 21.38 | 19.94 | 18.64 | 19.03 | 21.74 | 25.69 |
| **352** | 31.87 | 24.70 | 22.30 | 24.26 | 27.31 | 31.88 |
| **353** | 31.38 | 24.65 | 23.48 | 24.96 | 26.57 | 30.82 |
| **357** | 24.64 | 23.26 | 19.45 | 21.06 | 22.99 | 26.69 |
| **362** | 2.99 | 2.99 | 2.97 | 2.96 | 2.95 | 2.96 |
| **365** | 11.00 | 11.00 | 10.95 | 10.97 | 10.97 | 10.97 |
| **366** | 18.42 | 18.23 | 17.83 | 18.04 | 18.10 | 17.98 |
| **387** | 8.87 | 7.48 | 7.47 | 7.82 | 8.01 | 8.99 |
| **392** | 31.56 | 25.45 | 23.75 | 24.08 | 26.31 | 31.10 |
| **408** | 12.23 | 11.14 | 10.95 | 11.23 | 13.61 | 15.67 |
| **414** | 9.26 | 8.59 | 7.79 | 8.02 | 8.24 | 9.63 |
| **428** | 23.76 | 19.00 | 18.51 | 18.51 | 21.15 | 25.80 |
| **431** | 25.47 | 21.49 | 20.70 | 21.50 | 26.19 | 29.77 |
| **438** | 29.21 | 24.56 | 21.89 | 23.84 | 27.01 | 31.87 |
| **446** | 30.22 | 25.29 | 23.73 | 24.54 | 26.22 | 29.79 |
| **Overall** | **19.88** | **17.19** | **16.04** | **16.69** | **18.29** | **20.79** |

Table 6.1: Average number of relevant documents across users' ranked lists over an entire search session

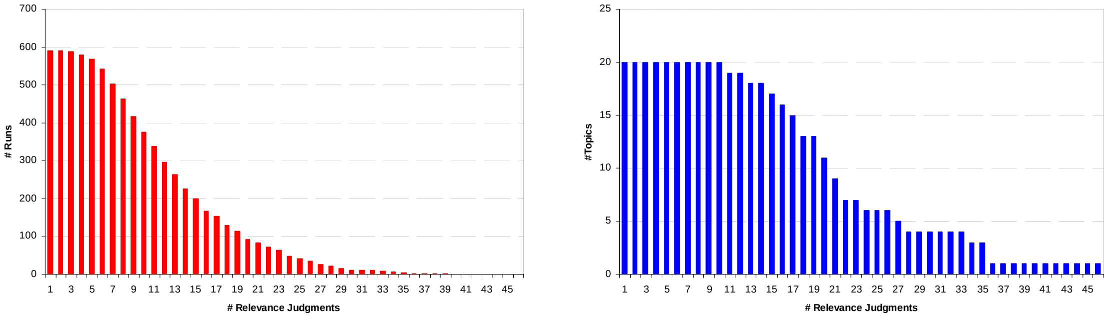


Figure 6.4: The number of runs (left) and topics (right) contributing to the overall group score at each iteration

users' ranked lists. We also examined how the performance of an SCIR system with an effective division of labour policy would compare with the performance of two users searching independently with no collaboration whatsoever, or with the best searcher from the group of two searching alone.

Comparing the performance of all SCIR systems, our results show that a division of labour policy can indeed significantly increase the performance of a group of users searching together through an SCIR system. A significant improvement is shown between systems with increasing levels of result division.

Indeed, our results show that a division of labour policy is an important facet of a state-of-the-art synchronous collaborative information retrieval system as the SCIR systems with no division and with just an initial division perform significantly worse than a group of users searching independently or the best individual searching alone. It is only by filtering the results returned to searchers, through the removal of either documents seen, or those in the current list of their search partner, that we achieve significant performance increases over the baseline of a group of users searching independently. Although the performance of the best searcher searching alone is best over the first few iterations, over an entire search our results show that an SCIR system implementing a full division of labour policy can outperform even a good user searching alone.

## 6.3 Sharing of Knowledge for Collaborative Relevance Feedback

In the previous section we showed how a division of labour policy can improve the performance of a group of searchers searching together through an SCIR system. Another advantage of having multiple users searching together through an SCIR system is that if these users are supplying relevance information to the system, as is common in state-of-the-art SCIR systems through a bookmark service, then we have an opportunity to improve the ranked lists returned to searchers, through implementing a *collaborative relevance feedback* process. Such a process can incorporate the relevance information of each collaborating searcher into each RF iteration. Passing relevance information among users in this way could enable users to benefit from the documents judged by their search partner without having to view and judge these documents themselves, thereby enabling an automatic sharing of knowledge across users.

As outlined in Chapter 4 there are a number of different techniques by which multi-user relevance information can be combined in the relevance feedback process, and we refer to these as *pseudo user*, *partial-user weighting*, *combined weighting* and *document fusion*. In this section we will evaluate each of these techniques in order to investigate which works best. We will also investigate if passing relevance information between searchers in the feedback process can improve the performance of SCIR. We will extend the best performing SCIR system from the previous section, the SCIR system with full division of labour, by introducing a collaborative relevance feedback process. We will also compare the performance of the SCIR systems with both a division of labour and a sharing of knowledge policy alongside the baselines of users searching independently.

To recap, the techniques proposed for combining relevance information in a collaborative relevance feedback process operate as follows:

1. *Pseudo User* – this represents the simplest possible combination where we assume that we have one user searching who has provided *all* the relevance judgments from all users and then run standard, single-user, relevance feedback over this pseudo user's relevance judgments.

2. *Partial-User Weighting* – In this technique the relevance and non-relevance proportions for terms are averaged over all users.

3. *Combined Weighting* – In this method, each user computes a relevance weight and offer weight for a term based on their own relevance information and these scores are averaged over all users.

4. *Document Fusion* – This method represents combination at the latest stage, where a feedback query is generated for each user using their own relevance information and this query is submitted to the retrieval engine and the results are combined using a standard ranked list fusion technique.

For these experiments, techniques 2 – 4 operate by using a simple linear combination strategy across all users, where the weight ($\alpha$) associated with each user's contribution to a term (techniques 2, 3) or a document (technique 4) is 0.5 to consider all users equally. Techniques 1 – 3 operate on a term level, whereby each user contributes to the relevance weight and offer weight of a term and how this contribution is combined is specific to the combination technique. As outlined in Chapter 4, one important consideration for techniques 2 and 3 is what action to take when a term has been encountered by one user only. In particular we have a choice as to whether to allow the user who has not encountered the term to provide a *contribution* to the term's weight or not. In these experiments we will run two variations of techniques 2 and 3, one which allows a user who has not encountered a term to contribute to its relevance and offer weight (*contr*), and another which calculates the weighting for a term based on the relevance information from the user who has encountered it only (*no contr*).

As outlined in the previous chapter, one other important consideration for any SCIR system is when to provide users with the relevance information from their search partner. In particular, should we present a user with a new list only when they themselves make relevance judgments or should we allow for a more dynamic, intensive search environment, whereby a user's ranked list changes as soon as they or their search partner makes a relevance judgment? Although this dynamic environment would allow users to benefit immediately from their search partner's relevance information, as discussed in the previous chapter, such an intensive environment raises issues related to both system implementation and user's cognitive load. Despite these concerns, in these experiments we evaluate all collaborative relevance feedback techniques in both a standard, *static*, interaction environment along with a more intensive *dynamic* environment for comparative purposes.

### 6.3.1 Results

Figure 6.5 plots the performance of all combination techniques for both the static SCIR environment and the dynamic SCIR environment along with the two baseline runs of users searching independently. The graph at the top shows the performance of the techniques over the entire search, while the graph at the bottom shows the performance of systems over the first few iterations only. Table 6.2 shows the single figure group score per topic.

Firstly, as with the results in the previous section, we find that the SCIR systems with collaborative relevance feedback for both the static and dynamic environment significantly outperform both baselines of users searching independently and the best individual searching alone.

As we can see, from Table 6.2, all collaborative relevance feedback techniques, except the document fusion techniques, provide small improvements in performance over the SCIR + Full Div system. Running significance testing over the single performance figures, however, reveals no significant difference between any SCIR system implementing a collaborative relevance feedback process for either feedback environment (i.e. static or dynamic), and the SCIR system with no combination of relevance information (SCIR + Full Div). When we relax the significance threshold, we find, in the static environment, that the combined weighting (contr) method and the pseudo user method outperform the SCIR + Full Div system at significance values of p = 0.165 and p = 0.186 respectively. While in the dynamic environment, the partial user (no contr) and combined weighting technique outperform the SCIR + Full Div system at significance values of p = 0.186, and p = 0.169 respectively.

Comparing the performance across collaborative RF techniques, from Figure 6.5 and Table 6.2, it does appear that the document fusion technique for both the static and dynamic environment, does not perform as well as the term-based techniques. Significance tests reveal that the dynamic collaborative RF techniques of pseudo

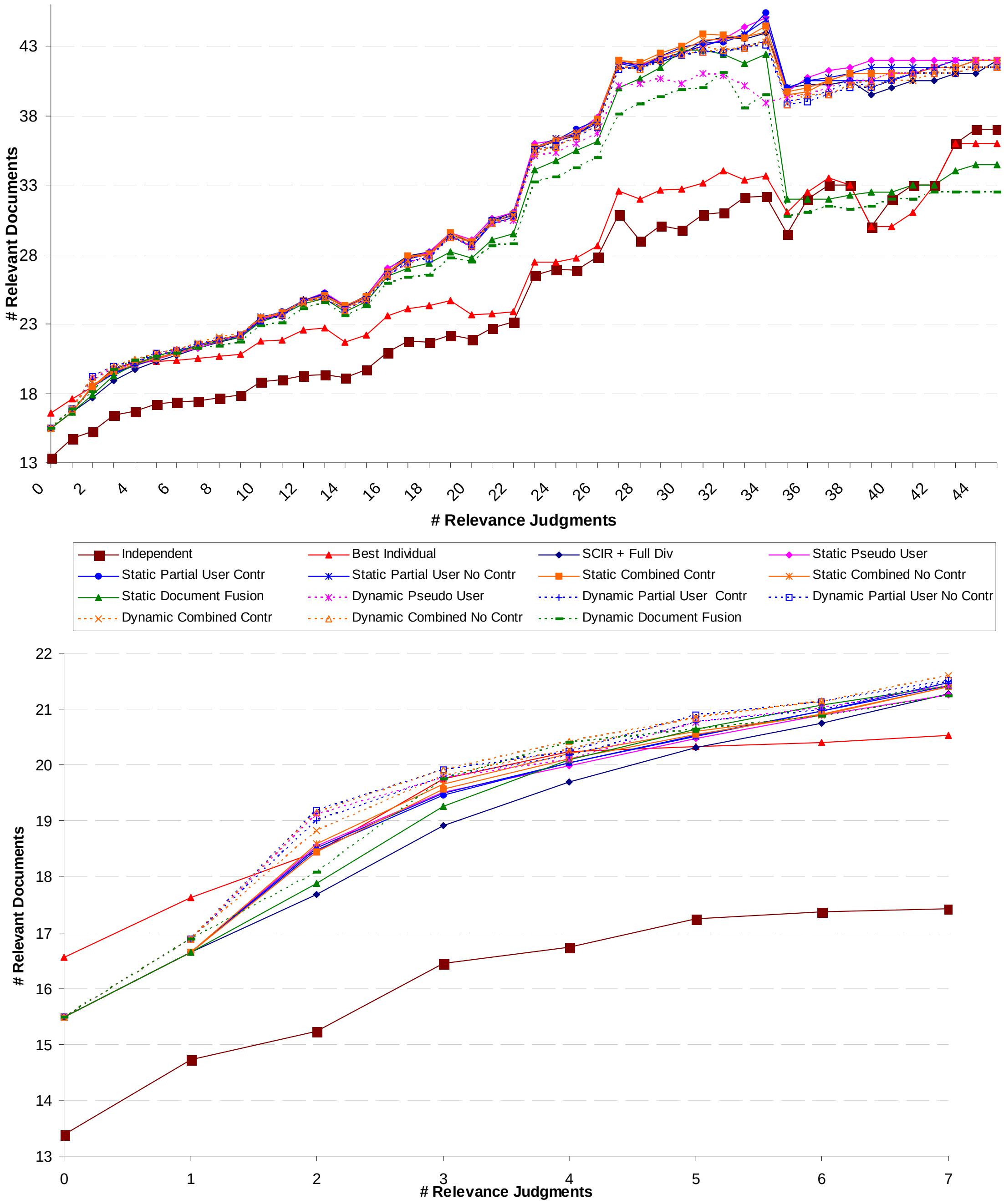


Figure 6.5: Comparison of collaborative relevance feedback techniques under both static and dynamic environments

user, partial-user, and combined weighting all significantly outperform the dynamic document fusion technique. However, no difference could be found at the significance threshold between any static term-based technique and the static document fusion technique. When we relax our significance threshold to a threshold of p $<$ 0.1, we do find that the techniques of pseudo user, partial-user, and combined weighting perform better than the static document fusion technique. Although not strictly significant according to our threshold, these p values, suggest that the results are unlikely due to chance.

Comparing the overall performance of the contribution versus no contribution techniques for both partial-user and combined weighting, we find no significant difference.

Examining the bottom graph in Figure 6.5, we can see that the combination of relevance information techniques do provide a more substantive increase in performance over the SCIR system for the first few iterations. Our significance tests confirm that for iterations 2 – 5, all collaborative RF techniques, for both static and dynamic environments, significantly outperform the SCIR system with full division.

Next we compare the performance of static versus dynamic feedback environments. From Figure 6.5 and Table 6.2, it appears that the SCIR systems operating in a dynamic feedback environment provide a modest increase in performance over their static counterparts. Our significance tests reveal that only the partial-user technique shows any significant difference between the running of the technique in static versus dynamic mode and no significant improvement could be found between any dynamic collaborative relevance feedback technique and the static combined weighting technique.

### 6.3.2 Discussion

In this section, we have explored the effects of a collaborative relevance feedback technique operating alongside an explicit division of labour policy in a synchronous collaborative information retrieval system. We hypothesised that by allowing users

| | **Best** | | **SCIR +** | **Static** | **Static** | **Static** | **Static** | **Static** | **Static** | **Dynamic** | **Dynamic** | **Dynamic** | **Dynamic** | **Dynamic** | **Dynamic** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Topic #** | **Individual** | **Independent** | **Full Div** | **Pseudo** | **Partial User** | **Partial User** | **Combined** | **Combined** | **Document** | **Pseudo** | **Partial User** | **Partial User** | **Combined** | **Combined** | **Document** |
| | | | | **User** | **Contr** | **No Contr** | **Contr** | **No Contr** | **Fusion** | **User** | **Contr** | **No Contr** | **Contr** | **No Contr** | **Fusion** |
| **303** | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| **307** | 36.46 | 31.73 | 35.75 | 35.92 | 36.32 | 36.32 | 36.17 | 36.46 | 35.37 | 35.90 | 36.42 | 36.67 | 36.55 | 36.41 | 35.88 |
| **322** | 3.49 | 3.30 | 2.99 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.99 | 2.97 | 2.97 | 2.99 | 2.98 | 3.01 | 3.07 |
| **326** | 33.44 | 29.15 | 35.43 | 34.64 | 34.70 | 34.74 | 34.63 | 34.60 | 35.27 | 35.20 | 35.16 | 35.26 | 35.09 | 35.34 | 35.59 |
| **339** | 5.95 | 5.89 | 5.93 | 5.94 | 5.94 | 5.94 | 5.94 | 5.95 | 5.94 | 5.93 | 5.93 | 5.93 | 5.94 | 5.94 | 5.93 |
| **347** | 21.38 | 19.94 | 25.69 | 26.29 | 26.04 | 26.02 | 26.11 | 25.99 | 26.00 | 26.01 | 26.10 | 26.10 | 26.05 | 25.89 | 26.26 |
| **352** | 31.87 | 24.70 | 31.88 | 34.51 | 34.83 | 34.89 | 34.40 | 34.62 | 33.90 | 36.64 | 36.39 | 36.89 | 36.18 | 36.72 | 34.91 |
| **353** | 31.38 | 24.65 | 30.82 | 31.03 | 30.66 | 30.68 | 30.80 | 31.27 | 30.91 | 31.71 | 31.40 | 31.63 | 31.52 | 32.01 | 30.87 |
| **357** | 24.64 | 23.26 | 26.69 | 24.84 | 24.74 | 24.80 | 25.20 | 24.65 | 25.73 | 24.26 | 24.18 | 24.22 | 25.14 | 24.28 | 24.83 |
| **362** | 2.99 | 2.99 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 |
| **365** | 11.00 | 11.00 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 |
| **366** | 18.42 | 18.23 | 17.98 | 17.54 | 17.91 | 17.96 | 17.97 | 18.03 | 17.12 | 17.17 | 17.47 | 17.48 | 17.38 | 17.56 | 16.05 |
| **387** | 8.87 | 7.48 | 8.99 | 9.15 | 9.13 | 9.15 | 9.13 | 9.16 | 9.03 | 9.15 | 9.15 | 9.13 | 9.11 | 9.08 | 9.03 |
| **392** | 31.56 | 25.45 | 31.10 | 32.18 | 32.19 | 32.26 | 32.00 | 32.07 | 31.70 | 31.85 | 32.44 | 32.38 | 32.38 | 32.44 | 31.64 |
| **408** | 12.23 | 11.14 | 15.67 | 15.37 | 15.53 | 15.02 | 15.52 | 14.75 | 15.47 | 15.01 | 15.25 | 14.68 | 15.24 | 14.47 | 15.24 |
| **414** | 9.26 | 8.59 | 9.63 | 9.67 | 9.56 | 9.61 | 9.59 | 9.59 | 9.54 | 9.42 | 9.36 | 9.34 | 9.23 | 9.22 | 9.40 |
| **428** | 23.76 | 19.00 | 25.80 | 26.26 | 26.22 | 26.39 | 26.43 | 26.71 | 25.49 | 26.17 | 25.67 | 26.01 | 25.86 | 26.17 | 25.00 |
| **431** | 25.47 | 21.49 | 29.77 | 29.93 | 29.51 | 29.47 | 29.90 | 29.90 | 26.78 | 29.17 | 29.18 | 29.45 | 29.03 | 29.21 | 24.90 |
| **438** | 29.21 | 24.56 | 31.87 | 32.35 | 32.25 | 32.23 | 32.21 | 32.20 | 32.38 | 32.73 | 32.68 | 32.77 | 32.85 | 33.07 | 32.84 |
| **446** | 30.22 | 25.29 | 29.79 | 30.95 | 30.56 | 30.61 | 30.47 | 30.58 | 30.72 | 31.35 | 30.90 | 31.19 | 30.85 | 31.10 | 30.45 |
| Overall | **19.88** | **17.19** | **20.79** | **20.97** | **20.95** | **20.95** | **20.97** | **20.97** | **20.71** | **21.03** | **21.03** | **21.10** | **21.07** | **21.09** | **20.59** |

Table 6.2: Single figure group score comparison of collaborative relevance feedback techniques

to pass relevance information to each other through a collaborative relevance feedback mechanism the performance of the group could be improved. We will now discuss our findings.

Our results show that over the entire search, the collaborative relevance feedback techniques do provide modest increases in performance over the SCIR system with just a division of labour. Although at the significance threshold of $p < 0.05$ no significance can be found, improvements could be found at lower significance thresholds. When we examine the performance of the group over the first few iterations of feedback, we find that the collaborative relevance feedback techniques provide more substantial improvements and that all techniques in both the static and dynamic environments significantly improve the performance over the SCIR + Full Div system over these early iterations. This result is interesting, and suggests that although users may benefit from gaining relevance judgments from their search partner early in the search, after a number of iterations this benefit is reduced. We will explore this issue further in section 6.5.

Comparing the performance of the collaborative relevance feedback techniques, we found that the term-based techniques of pseudo user, partial-user, and combined weighting all outperform the document fusion technique. Although no significant difference could be found at a significance threshold of $p < 0.05$. When we relax the threshold we do find that all term-based techniques outperform the document fusion technique. These results suggest that for both the dynamic and static environments a term-based technique can outperform a document based fusion technique.

Over the entire search, no significant differences could be found across term-based techniques. In particular, the collaborative techniques of partial-user and combined weighting perform similarly to the standard single user relevance feedback process (pseudo user). This result is not surprising as all techniques are attempting to aggregate each user's relevance information. The potential advantages of a collaborative relevance feedback technique will be realised when we need to perform a user-biased combination of relevance information, which will be explored in the next section.

Our results show small improvements can be made over some techniques by implementing an intensive, dynamic environment. However due to the slenderness of these differences and the fact that not all static techniques can be significantly improved upon, it may not be worthwhile implementing such a policy due to the discussed difficulties that such an environment presents for both the system designer and the user. For these reasons we will not consider this environment in subsequent experiments in this chapter.

In this section we have explored the effects of a sharing of knowledge policy in an SCIR environment in which we assume that users will provide perfect relevance information to the system. However, this may not always be the case as users may make mistakes in their relevance assessments. In the next section we will explore how such collaborative relevance feedback techniques operation under an *imperfect* relevance information environment.

## 6.4 Sharing of Knowledge Under Imperfect Relevance Information

As discussed in Chapter 4, due to the nature of synchronous collaborative search, different searchers will have different levels of expertise and familiarity with searching and search topics and this may be reflected in the quality of their relevance assessments. For this reason we want to explore the effects of imperfect relevance information on a synchronous collaborative search. As outlined in the previous chapter, an imperfect simulation will proceed by simulating a user saving the first document on their ranked list that is either relevant *or* is a non-relevant document that was saved by at least two real users during the original TREC interactive experiments, referred to as *perceived relevant* documents.

Before examining the effects of imperfect relevance information on a collabo-

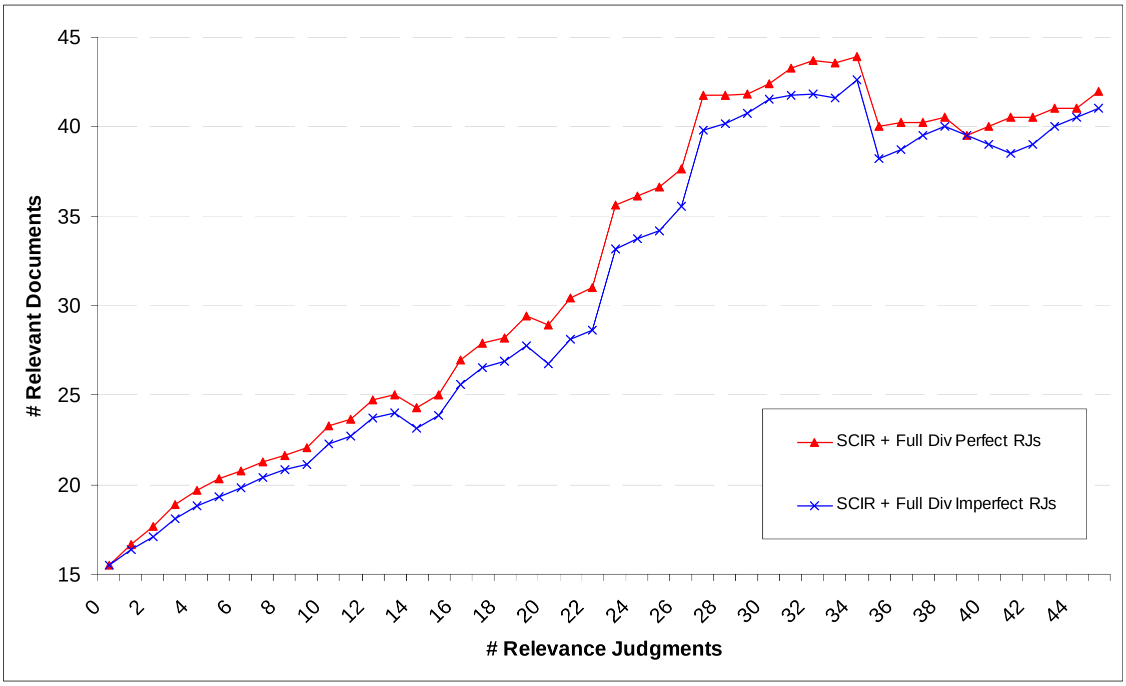


Figure 6.6: Comparison of SCIR + Full Div system under perfect and imperfect relevance information

rative relevance feedback process, it is important to understand the influence of non-relevance information on a standard, single user, relevance feedback process. Therefore, in Figure 6.6 we plot the performance of the SCIR + Full Div system under both perfect and imperfect relevance information. As expected we see that the inclusion of noise through imperfect relevance judgments ("RJs") causes a degradation in the performance of the RF process, and our analysis of the associated single figure measures shows this difference to be significant. The performance gap between the two, however, may not be as big as expected. The reason for this could be due to the fact that these non-relevant, or perceived relevant documents, may, as the name suggests, resemble relevant documents in so-far-as they may contain many terms relevant to the topic without fulfilling the criteria for relevance. In this regard, although these documents may be strictly non-relevant according to the qrels, they may be good relevance feedback documents, for example they may expand the query with good terms.

Next we explore the effects of imperfect relevance information on a collaborative relevance feedback process. In Figure 6.7 and Table 6.3 we show the performance of

| Topic # | SCIR + Full Div | Pseudo User | Partial User Contr | Partial User No Contr | Combined Contr | Combined No Contr | Document Fusion |
|---|---|---|---|---|---|---|---|
| **303** | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| **307** | 35.46 | 34.79 | 35.11 | 35.14 | 35.20 | 35.17 | 35.25 |
| **322** | 2.72 | 2.45 | 2.45 | 2.44 | 2.46 | 2.44 | 2.63 |
| **326** | 35.43 | 34.64 | 34.59 | 34.74 | 34.63 | 34.60 | 35.27 |
| **339** | 5.83 | 5.83 | 5.84 | 5.85 | 5.84 | 5.84 | 5.88 |
| **347** | 23.17 | 23.60 | 23.58 | 23.57 | 23.48 | 23.42 | 23.26 |
| **352** | 28.27 | 31.26 | 31.50 | 31.97 | 31.53 | 31.06 | 30.54 |
| **353** | 29.52 | 30.07 | 29.65 | 29.78 | 29.61 | 30.26 | 29.92 |
| **357** | 24.73 | 23.30 | 23.04 | 23.03 | 23.34 | 23.03 | 23.81 |
| **362** | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 |
| **365** | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 |
| **366** | 18.30 | 17.97 | 18.12 | 18.14 | 18.16 | 18.19 | 17.49 |
| **387** | 8.29 | 8.49 | 8.41 | 8.42 | 8.39 | 8.34 | 8.15 |
| **392** | 29.49 | 31.50 | 31.65 | 31.64 | 31.36 | 31.41 | 30.97 |
| **408** | 13.28 | 12.94 | 12.96 | 12.94 | 13.09 | 13.01 | 12.87 |
| **414** | 9.49 | 9.54 | 9.49 | 9.52 | 9.51 | 9.51 | 9.41 |
| **428** | 25.80 | 26.26 | 26.22 | 26.39 | 26.43 | 26.71 | 25.49 |
| **431** | 28.34 | 28.58 | 28.42 | 28.49 | 28.56 | 28.76 | 25.37 |
| **438** | 29.60 | 30.51 | 30.21 | 30.28 | 30.21 | 30.13 | 30.17 |
| **446** | 29.79 | 30.95 | 30.56 | 30.61 | 30.47 | 30.58 | 30.72 |
| **Overall** | **19.87** | **20.13** | **20.09** | **20.14** | **20.11** | **20.12** | **19.86** |

Table 6.3: Single figure comparison of SCIR + Full Div system and collaborative relevance feedback techniques under imperfect relevance information

each of the collaborative relevance feedback techniques operating under imperfect relevance information. From Table 6.3 we see that the collaborative relevance feedback techniques provide an increase over the SCIR + Full Div system. As we found under the perfect relevance information environment, however, at the significance threshold of $p < 0.05$ we find no significant difference. When we relax the threshold we do find differences between all techniques and the SCIR + Full Div system at a threshold of $p < 0.2$.

Across all systems, the best performing run is the partial-user no contr system When we compare the performance across both partial-user variations (contr versus no contr), we find that the no contr system significantly outperforms the contr system. As outlined in Chapter 4, the effect of allowing a user to contribute to a term's weight (contr) even if they have not encountered it is to promote shared items. As the no contr system outperforms the contr system this suggests that the promotion of shared items degrades the performance of an SCIR search under imperfect relevance information.

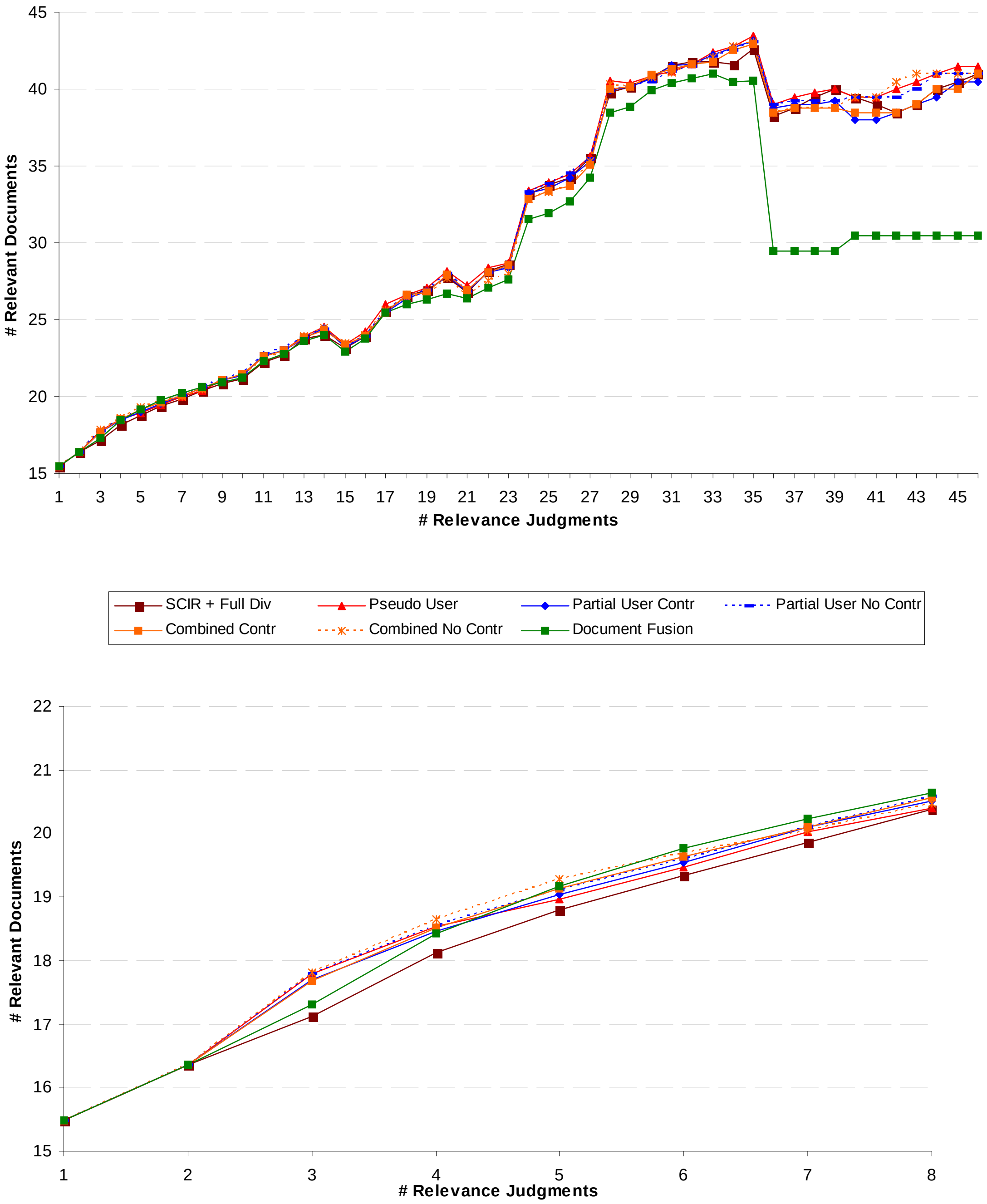


Figure 6.7: Comparison of SCIR + Full Div system and collaborative relevance feedback techniques under imperfect relevance information

As before, we also investigated the performance of the techniques for combining relevance information over the first few iterations of feedback only and these results are plotted on the bottom of Figure 6.7. As we can see, all combination techniques outperform the SCIR + Full Div system over these iterations. However our significance tests reveal that only the document fusion technique and the combined weighting technique with no contribution significantly outperform the SCIR + Full Div system over these iterations. However, the partial user contr (p = 0.097), no contr (p = 0.079), combined weighting contr (p = 0.057), and pseudo-user (p = 0.116) do outperform the SCIR + Full Div system at lower thresholds.

The collaborative relevance feedback techniques proposed thus far attempt to aggregate multi-user relevance information, and this may not always be the best policy, especially in the presence of non-relevance information. In the next section we will investigate ways of extending the collaborative relevance feedback process in order to overcome problems associated with non-relevance information, by performing a user-biased combination through *authority weighting*.

### 6.4.1 Authority Weighting

During any SCIR session, users may make multiple relevance judgments. The quality of these judgments may vary across users due to a variety of factors including the user's expertise with the search topic. By recognising that users can make mistakes in their relevance assessments, we can take steps to counteract the problems using a user-biased combination of relevance information as allowed by the collaborative relevance feedback techniques. In this section we will examine if by attaching an authority weight to users in a collaborative relevance feedback process, we can improve the performance of the group search.

Query performance predication, is an emerging research area which investigates ways to determine the quality of queries dynamically without any ground truth data and several methods have been proposed in the literature (Cronen-Townsend et al., 2002; He and Ounis, 2004; Yom-Tov et al., 2005; Vinay et al., 2006; Zhou and Croft, 2006), for a good overview of these approaches see Lang et al. (2008). These techniques could be used in an SCIR search session in order to assess which user's relevance assessments are the better before performing feedback. In this thesis we will not attempt to predict the performance of either user, instead what we want to investigate is that, if we had a method which allowed us to tell which user is the better, in terms of their relevance assessments, then can we exploit this information in order to perform a user-biased collaborative relevance feedback process?

In order to attach an authority weight to each user we first need a way of deciding on which user is the better during the search process. Our approach was to pre-compute an *Oracle* relevance weighting of terms for all topics using an approach similar to the one employed by Magennis and van Rijsbergen (1998). In order to generate this oracle weighting, for each topic we performed a single iteration of relevance feedback, using all relevant documents (according to the qrels) as the input to this process, we then recorded the relevance weight produced by this process for all terms from these relevant documents. The weights produced by this process represent the best possible weighting of all terms given the complete set of relevant documents. This procedure was followed for all topics in order to produce an oracle weighting of all terms for all topics.

In a simulated SCIR search we can use these oracle weights in order to evaluate the quality of each user's relevance assessments. At any point in the search, each user will have made a certain number of relevance judgments. As outlined earlier, in these imperfect simulations, some of these relevance assessments may be mistaken. In order to calculate which user of the two is the more *authoritative* in terms of their relevance judgments, for each user we calculate the relevance weights of all terms from their relevance judgments. We then calculate the correlation between the oracle relevance weighting of terms and the user's weighting of their terms. Performing this procedure for each user, allows us to see which user's relevance judgments are more closely correlated with the oracle. The user with the higher correlation is considered to have the more authoritative relevance judgments.

Having decided on which user's relevance judgments are the more authoritative, there are different ways in which this information can be incorporated into a user-biased collaborative relevance feedback process. One approach is to set the more authoritative user's $\alpha$ value in the collaborative RF process to an authority weight ($Aw$), assigning the less authoritative user an $\alpha$ value of 1 - $Aw$. This method would keep the authority value constant throughout the search regardless of how big the gap in performance is between the users (although the recipient of this authority weight may change over the course of the search). For example if we decided to give the more authoritative user a 60% bias, then, even if the other searcher's relevance judgments were particularly bad, they would still contribute 40% to the collaborative relevance feedback process. Another approach, would be to use the correlation figures for each user, as a means to dynamically apportion the authority weight across users. In these experiments, we evaluated both approaches.

For the *constant* authority weighting method, we have chosen to implement the popular *Pearson* correlation measure. We investigated authority weights of 0.6 – 1.0 in 0.1 increments, for completeness we also experimented with inverting the authority weight and so evaluated the effects of authority weights of 0 – 0.4 in 0.1 increments.

In the *dynamic* authority weighting approach, the correlation value is more important to the assignment of weights, therefore we experimented with several methods of calculating this correlation value. Firstly, we experimented with both the *Pearson* and *Spearman* correlation measure. One important consideration when calculating a user's correlation with the oracle is what action to take when a user's relevance judgements do not contain a term that is contained in the oracle. This will often be the case as, at any point in the search, users will have only made judgments on a subset of the relevant documents. Common usage of correlations propose two approaches for dealing with sparse data, one is to assign an *average* value to the

|  | **Group 1** |  | **Group 2** |  |
|---|---|---|---|---|
|  | **User 1** | **User 2** | **User 1** | **User 2** |
| **Correlation** | 0.25 | 0.2 | 0.85 | 0.8 |
| **Absolute Aw** | 0.525 | 0.475 | 0.525 | 0.475 |
| **Proportion Aw** | 0.5556 | 0.4444 | 0.5152 | 0.4848 |

Table 6.4: Absolute and proportion authority weighting example

item, a term in our case, that does not appear in the user's judgments, and another is to *skip* terms that are not shared and calculate the correlation based on the shared terms only. In these experiments we experiment with both approaches.

Having generated a user's correlation value, we need to translate this figure into an $\alpha$ value for each user. One method we could use is to simply find the absolute difference between the correlation values and then increase the authoritative user's $\alpha$ value from 0.5 by adding half the absolute difference, decreasing the other user's $\alpha$ value from 0.5 by subtracting half the absolute difference. Another approach we could use would be to normalise the values so that they sum to one, by dividing each user's correlation value by the sum of the correlation values. Table 6.4, shows an example of the effect of each technique on the assignment of weights. As we can see, correlation values of 0.2 vs 0.25 and 0.8 vs 0.85 will be considered the same for the absolute method, but for the proportion method the gap between 0.2 and 0.25 is considered greater than between 0.8 and 0.85 and this is reflected in the authority weights.

#### 6.4.1.1 Results

For these experiments we implemented authority weighting on the best performing collaborative relevance feedback technique under imperfect relevance information, partial-user no contr. Table 6.5 presents the single figure performance values for all methods of authority weighting along with both the un-biased partial-user relevance weighting method which assigns a weighting of 0.5 to both users, and the standard, single user relevance feedback technique (pseudo user).

Overall we can see that all the dynamic authority weighting approaches, and the constant Aw values of 0.6 and 0.7 provide small improvements in performance over both the unbiased partial-user and the pseudo user method, with the best performing method being the dynamic, Pearson correlation (average) method.

Significance tests reveal that the dynamic-absolute-Pearson-avg, dynamic-absolute-Spearman-avg and dynamic-absolute-Spearman-skipped all significantly outperform the unbiased partial-user method ($p < 0.05$) while the dynamic-proportion-Pearson-avg method outperforms the pseudo user method at this threshold. When we relax the significance threshold, we find that the static authority weighting of 0.6 outperforms the unbiased partial-user and the pseudo user method at significance levels of $p = 0.093$ and $p = 0.125$ respectively.

Examining the constant authority weighting results, we see that an authority weight of 0.6 provides the best performance. The performance degrades for values of authority higher than this value. As expected, the inverted authority weights degrade performance substantially with the authority weight of 0, which gives a zero weighting to the more authoritative user, performing the worst. Significance tests reveal that all authority values between $0.6 - 1.0$ are significantly better than the inverted authority values of $0.4 - 0$.

In the dynamic weighting methods we see that for both Pearson and Spearman, the *average* approach to dealing with unique terms performs better than the *skipped* approach. However no significant difference could be found at the significance threshold and only the dynamic-Spearman-raw run showed a difference at a more relaxed threshold ($p = 0.067$). It also appears that using the absolute difference when mapping the correlation value to the $\alpha$ value performs better than the proportion method, however only the dynamic-Spearman-average run shows a significant difference between its raw and proportion variant. Finally it seems that the Pearson correlation method performs better than the Spearman correlation method, and our significance tests reveal that both the dynamic-Pearson-raw-average and the dynamic-Pearson-prop-average perform better than the dynamic-Spearman-raw-average and dynamic-Spearman-prop-average respectively.

In this section we explored the effects of imperfect relevance information on a collaborative relevance feedback process operating in an SCIR environment. Our results show that imperfect relevance information can significantly degrade the performance of a relevance feedback operation in an SCIR search. The comparative effects of collaborative relevance feedback on an SCIR system are similar to those found under perfect relevance information. To overcome the problems associated with imperfect relevance judgments in a collaborative relevance feedback process we experimented with attaching a user-biased authority weight and found that we can significantly improve upon both an unbiased combination of relevance information and the standard, single user, relevance feedback process.

## 6.5 Sharing of Knowledge for Complementary Relevance Feedback

Our experiments in the previous two sections have applied a collaborative relevance feedback process to an SCIR search session. The results have shown that, although the techniques can provide good improvements in performance over the first few iterations of feedback, over an entire search, the improvements are less substantial.

One reason for this may be that the collaborative relevance feedback process of aggregating relevance information is causing the relevance feedback process for each user to become too similar, thereby limiting the *breadth* across collaborating users' reformulated queries. By implementing a full division of labour policy we are ensuring that each user is presented with unique documents across the top 30 positions of their ranked lists, however, we suspect that the aggregation of relevance information may be causing a loss of uniqueness across users. In order to investigate

| | **Unbiased Combination** | | **Constant Authority Weight** | | | | | | | | | | **Dynamic - Absolute** | | | | **Dynamic - Proportion** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Topic #** | **Pseudo** | **Partial** | **0.6** | **0.7** | **0.8** | **0.9** | **1** | **0.4** | **0.3** | **0.2** | **0.1** | **0** | **Pearson** | | **Spearman** | | **Pearson** | | **Spearman** | |
| | **User** | **User** | | | | | | | | | | | **Average** | **Skipped** | **Average** | **Skipped** | **Average** | **Skipped** | **Average** | **Skipped** |
| **303** | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 4.78 | 4.68 | 4.56 | 4.60 | 4.62 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| **307** | 34.79 | 35.14 | 35.25 | 35.38 | 35.36 | 35.11 | 35.09 | 14.45 | 12.42 | 11.51 | 11.43 | 11.39 | 35.34 | 35.35 | 35.42 | 35.43 | 35.42 | 35.44 | 35.53 | 35.32 |
| **322** | 2.45 | 2.44 | 2.45 | 2.47 | 2.49 | 2.50 | 2.46 | 2.29 | 2.45 | 2.44 | 2.44 | 2.46 | 2.45 | 2.45 | 2.45 | 2.44 | 2.42 | 2.41 | 2.45 | 2.44 |
| **326** | 34.64 | 34.74 | 34.73 | 34.66 | 35.00 | 35.10 | 35.02 | 16.71 | 15.77 | 14.54 | 14.30 | 15.03 | 34.84 | 34.80 | 34.78 | 34.76 | 34.90 | 34.75 | 34.77 | 34.74 |
| **339** | 5.83 | 5.85 | 5.83 | 5.85 | 5.86 | 5.85 | 5.83 | 4.16 | 4.00 | 3.86 | 3.81 | 3.80 | 5.83 | 5.85 | 5.84 | 5.85 | 5.83 | 5.85 | 5.84 | 5.85 |
| **347** | 23.60 | 23.57 | 23.50 | 23.60 | 23.43 | 23.33 | 23.35 | 12.32 | 10.32 | 9.27 | 9.15 | 9.14 | 23.64 | 23.62 | 23.62 | 23.64 | 23.71 | 23.60 | 23.68 | 23.62 |
| **352** | 31.26 | 31.97 | 32.09 | 30.96 | 30.43 | 29.62 | 28.88 | 18.02 | 13.44 | 12.41 | 12.34 | 11.88 | 32.60 | 32.37 | 32.42 | 32.34 | 31.58 | 32.26 | 31.51 | 32.17 |
| **353** | 30.07 | 29.78 | 29.83 | 29.81 | 29.89 | 29.80 | 29.87 | 18.82 | 18.45 | 18.11 | 18.00 | 18.10 | 29.85 | 29.82 | 29.85 | 29.78 | 29.85 | 29.79 | 29.91 | 29.76 |
| **357** | 23.30 | 23.03 | 23.88 | 24.06 | 23.94 | 23.95 | 24.05 | 13.54 | 12.67 | 11.32 | 11.22 | 11.02 | 23.49 | 23.27 | 23.47 | 23.20 | 23.65 | 23.22 | 23.54 | 23.22 |
| **362** | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.19 | 2.19 | 2.18 | 2.21 | 2.20 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 |
| **365** | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 6.94 | 6.67 | 6.51 | 6.51 | 6.54 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 |
| **366** | 17.97 | 18.14 | 18.19 | 18.26 | 18.32 | 18.30 | 18.35 | 8.93 | 8.99 | 9.00 | 8.98 | 8.99 | 18.21 | 18.17 | 18.19 | 18.15 | 18.22 | 18.16 | 18.21 | 18.15 |
| **387** | 8.49 | 8.42 | 8.45 | 8.46 | 8.38 | 8.41 | 8.33 | 5.88 | 5.45 | 4.82 | 4.63 | 4.60 | 8.43 | 8.42 | 8.42 | 8.42 | 8.41 | 8.42 | 8.38 | 8.42 |
| **392** | 31.50 | 31.64 | 31.53 | 31.43 | 31.09 | 30.59 | 29.94 | 11.12 | 11.01 | 10.77 | 10.54 | 10.47 | 31.60 | 31.58 | 31.54 | 31.60 | 31.53 | 31.58 | 31.51 | 31.63 |
| **408** | 12.94 | 12.94 | 12.97 | 13.02 | 12.92 | 12.98 | 13.06 | 7.34 | 6.82 | 6.17 | 6.30 | 5.46 | 12.90 | 12.96 | 12.93 | 12.93 | 12.94 | 12.91 | 12.96 | 12.93 |
| **414** | 9.54 | 9.52 | 9.52 | 9.53 | 9.48 | 9.58 | 9.43 | 6.00 | 6.13 | 5.83 | 5.76 | 5.84 | 9.49 | 9.55 | 9.49 | 9.51 | 9.49 | 9.53 | 9.45 | 9.52 |
| **428** | 26.26 | 26.39 | 26.34 | 25.95 | 25.60 | 25.25 | 25.17 | 11.26 | 11.01 | 10.66 | 10.38 | 10.07 | 26.33 | 26.50 | 26.27 | 26.42 | 26.25 | 26.41 | 26.22 | 26.41 |
| **431** | 28.58 | 28.49 | 28.76 | 28.78 | 28.87 | 28.78 | 28.36 | 11.69 | 10.43 | 9.42 | 9.09 | 9.17 | 28.62 | 28.43 | 28.62 | 28.45 | 28.68 | 28.46 | 28.62 | 28.45 |
| **438** | 30.51 | 30.28 | 30.09 | 30.21 | 30.02 | 30.07 | 30.08 | 11.18 | 9.52 | 8.20 | 7.81 | 7.72 | 30.36 | 30.36 | 30.30 | 30.35 | 30.46 | 30.36 | 30.20 | 30.31 |
| **446** | 30.95 | 30.61 | 30.82 | 30.99 | 30.77 | 30.47 | 30.33 | 16.15 | 15.30 | 14.03 | 13.61 | 13.82 | 30.86 | 30.56 | 30.82 | 30.67 | 30.90 | 30.58 | 30.93 | 30.60 |
| **Overall** | **20.13** | **20.14** | **20.21** | **20.17** | **20.09** | **19.98** | **19.88** | **10.19** | **9.39** | **8.78** | **8.66** | **8.62** | **20.24** | **20.20** | **20.22** | **20.19** | **20.21** | **20.18** | **20.18** | **20.17** |

Table 6.5: Single figure comparison for authority runs

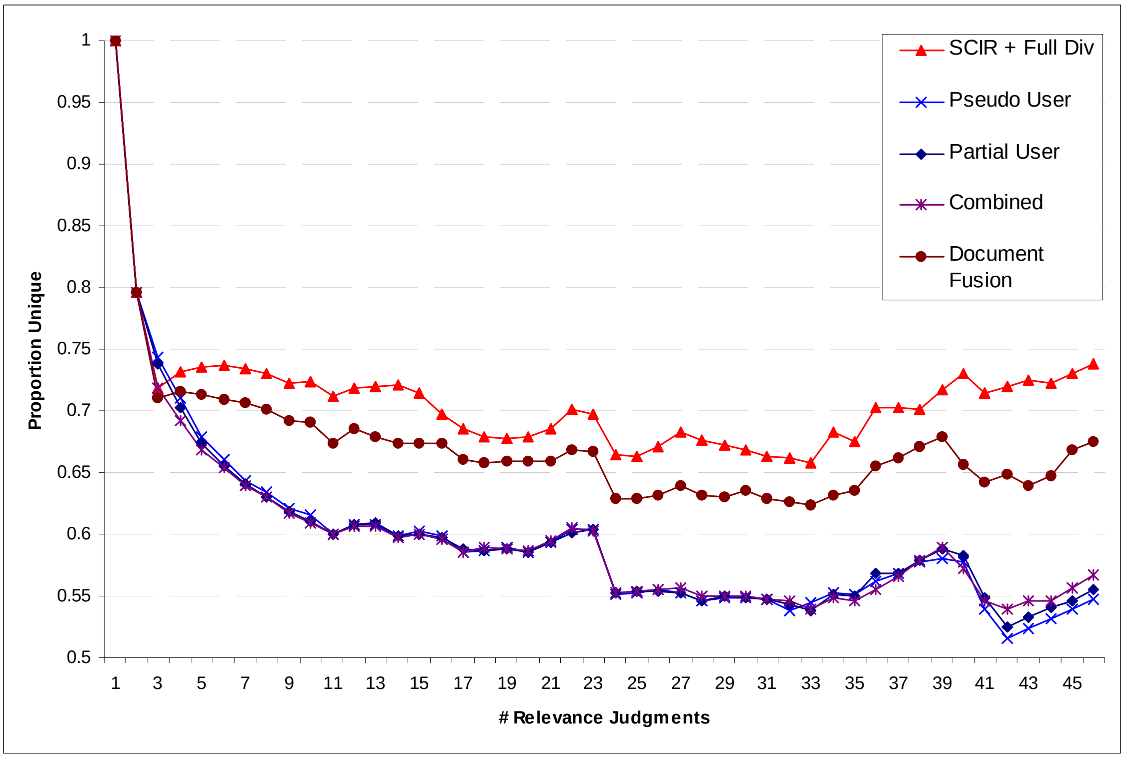


Figure 6.8: Comparison of the proportion of unique documents across both user's ranked lists for SCIR with collaborative RF and without

this hypothesis, in Figure 6.8 we plot the proportion of unique documents across the top 1000 documents of each user's ranked lists for the SCIR system with full division only and all collaborative relevance feedback techniques operating under perfect relevance information (from section 6.3).

As we can see, there is a clear difference in the total number of unique relevant documents found across users' ranked lists between all collaborative relevance feedback systems and the SCIR + Full Div system. This difference is significant for all techniques across all topics. The decrease in the proportion of unique documents in user's lists across all techniques confirms our hypothesis that the collaborative relevance feedback process is causing ranked lists of users to become too similar. This finding is intuitive - one of the great advantages of having multiple users tackle a search task is that it allows the task to be divided across users. However, by implementing a collaborative relevance feedback process in such an environment, where the relevance feedback process for a user uses the relevance information of their search partner, we are causing users to loose this uniqueness. Interestingly however, the gap is less substantial between the SCIR system with full division and the SCIR system implementing document fusion combination. The fact that the document fusion technique provides substantially more unique documents than any of the term based techniques suggests that the term-based techniques are causing the selection of similar terms for expansion between users. The document fusion technique, allowing for a later stage of fusion, does not suffer from this problem. However as the results in section 6.3 have shown, this does not lead to this technique outperforming the others in terms of discovering more unique relevant documents. This does not, of course, mean that the introduction of unique documents degrades performance but that the use of a collaborative relevance feedback mechanism needs to strive to allow for the introduction of more unique *relevant* documents. Therefore, in this section we will explore the use of users' relevance judgments in an SCIR search session in order to implement *complementary relevance feedback* techniques. These techniques will operate in an opposite manner to the collaborative relevance feedback techniques. The motivation of the complementary techniques being that, when performing relevance feedback, we can use the relevance judgments of a user's search partner in order to reformulate a user's query in such a way as to limit the overlap of results, thereby allowing users to explore distinct areas of the document collection. In this way it is hoped that these techniques will increase the number of unique relevant documents across users' ranked lists.

### 6.5.1 Complementary Query Expansion

One way of maintaining diversity across users through the relevance feedback process is by ensuring that the expansion terms assigned to each user are unique. In this section we will investigate the effects of implementing such a *complementary query expansion* technique in an SCIR environment. When performing feedback for a user, the complementary QE technique operates by removing, from a user's expansion terms produced using a standard, single user relevance feedback mechanism over

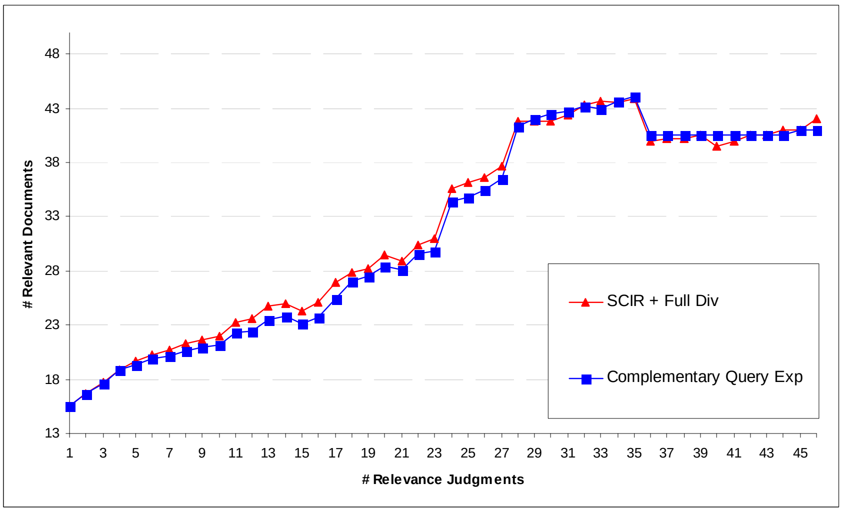


Figure 6.9: Comparison of SCIR + Full Div and complementary query expansion

their own relevance information, those terms that are contained in their search partner's current query. This has the effect of replacing shared terms with unique terms and such a technique should increase the diversity across users' queries and allow for more unique documents to be discovered.

In Figure 6.9 we compare the performance of the SCIR system with full division (SCIR + Full Div), which implements a standard relevance feedback mechanism, with an SCIR system which implements the complementary query expansion technique. As we can see, the complementary expansion approach performs worse than the SCIR + Full Div system. Running significance tests over the associated single figure group scores confirms this result to be significant across topics.

As Figure 6.10 shows, the complementary query expansion technique is indeed introducing more unique documents into user's ranked lists, but due to the poor performance of the technique, this diversity is obviously being achieved at a cost of a significant degradation in the quality of user's lists.

### 6.5.2 Clustering

In the previous section we found that the complementary query expansion technique, while introducing more unique documents across users' ranked lists, also reduced the

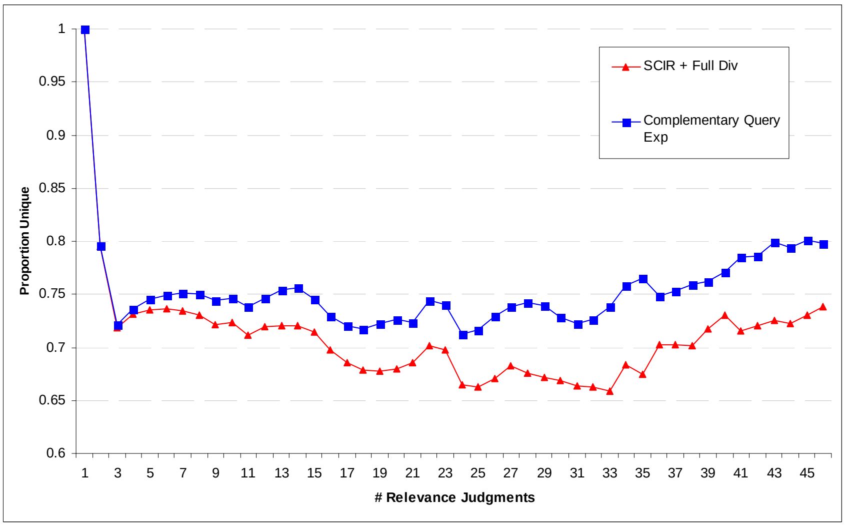


Figure 6.10: Comparison of SCIR + Full Div and complementary query expansion in terms of the proportion of unique documents found across users' ranked lists

quality of user's lists. In this section we will investigate a more sophisticated form of complementary feedback, through the use of clustering.

As outlined in Chapter 4 we want to investigate the use of clustering in order to partition: (1.) the set of relevant documents, and (2.) the terms contained within these documents, into two distinct clusters, one for each user, prior to performing feedback. The motivation for both techniques is that, by partitioning either the document or term space into two, we should generate more distinct relevance feedback queries than is produced by the collaborative relevance feedback techniques of section 6.3, while producing better quality queries than those produced by a simple removal of shared terms as investigated in the previous subsection.

For clustering we use the popular k-means algorithm. An important consideration for k-means is the choice of *distance measure*, which is used as the objective function. For high-dimensional data such as text documents *Cosine Similarity* has been shown to be an effective distance measure (Zhong, 2005) and is therefore used as the distance measure in our implementation. The k-means algorithm requires an initial choice of cluster centre-points or *centroids* before clustering can begin. These points are often chosen at random, from the set of input data. The choice of these initial centroids is important to the outcome of the algorithm, as poor initial selection can lead to the algorithm resting on a local minimum. In order to increase the chances of finding the best possible clusters, the clustering algorithm can use a heuristic (Kaufman and Rousseeuw, 1990) which attempts to find the best possible initial centroids and this is the approach used in these experiments.

Document clustering should allow for the grouping together of similar documents prior to performing relevance feedback. Term clustering will operate over the expansion terms produced by a collaborative relevance feedback technique so that each user is assigned a distinct set of terms, which is used to expand the shared query for this user. We will now describe the operation of clustering for both documents and terms in greater detail.

#### 6.5.2.1 Clustering of Relevant Documents

Our application of document clustering operates by partitioning all identified relevant documents into two distinct clusters. As often used in document clustering (Steinbach et al., 2000), we represent each document by a large feature vector which consists of normalised tf–idf values for each unique term from the set of all documents to be clustered.

In these experiments, clustering of documents is only performed for relevance feedback iterations after each user has made a relevance judgment and one of the users has made two relevance judgments, thereby ensuring that we have at least three relevant documents to cluster and that both users have provided at least one relevance judgment. For all relevance feedback iterations in the search prior to this criteria being met, users are provided with results from a collaborative relevance feedback technique, and for these experiments we use the partial user (no contr) technique.

After the criteria for clustering is met, when a user performs relevance feedback, we perform k-means over all relevance judgments made at this point in the search, in order to produce two clusters. After producing the two distinct clusters, we then need to assign one of these clusters to the user who is performing feedback. The cluster we assign to a user at this point is the cluster that contains the lowest overlap of documents with the cluster assigned to their search partner in their partner's last feedback iteration. In this way we can ensure that the cluster assigned to a user, and subsequently used for relevance feedback, is the most unlike their search partner's current cluster, although we can never be sure that the document assigned to users are completely unique. The documents contained in this cluster are then used to perform feedback for this user using a standard, single-user, relevance feedback process.

#### 6.5.2.2 Clustering of Expansion Terms

When performing relevance feedback the term clustering technique operates by clustering the set of top $T$ expansion terms produced by a collaborative relevance feedback technique (partial-user no contr) into two sets, one for each user. Terms are clustered based on their co-occurrence in documents across the entire collection. In particular, for each term, a feature vector is produced for all documents in the collection, with a "1" indicating the presence of this term in the document and a "0" indicating its absence.

The choice of $T$, the number of terms to use for clustering, is important for the quality of clustering results. A smaller number will cause clusters to be produced with a smaller number of terms, thereby causing user's queries to be expanded with less than 20 terms, which was shown in Chapter 5 to provide the best performance on the test collection. By increasing the number of terms to be clustered we increase the potential for producing clusters with an optimal number of expansion terms, however, the more terms we add to the clustering process the more potential there is for adding noise to the clustering process. For this reason, we will experiment with term clustering over the top 10, 20, 30, 40, and 50 terms as produced by a collaborative relevance feedback process.

As per the document clustering technique, after the k-means algorithm has produced two distinct clusters, a user is assigned one cluster of terms, and this cluster is the cluster that contains the lowest overlap of terms with the cluster assigned to their search partner in their partner's previous relevance feedback iteration. At any point, a query will only be expanded with a maximum of 20 terms, as was found in our tuning experiments to be the optimal number of expansion terms. If a cluster that is assigned to a user contains more than 20 terms, we only expand the user's query with the top 20 terms from this cluster.

#### 6.5.2.3 Results

Table 6.6 presents the overall results of clustering for both document and term clustering with the different values for $T$. As we can see, no clustering technique performs as well as the partial-user collaborative relevance feedback technique, or the SCIR + Full Div system. Significance tests reveal that all term clustering techniques perform significantly worse than the SCIR + Full Div system, the partial-user collaborative RF approach, and the document clustering technique.

In order to examine the effect of clustering in terms of maintaining diversity across users' ranked lists, in Figure 6.11, we plot the proportion of unique documents in the top 1000 positions across users' lists as before. As we can see, all document and term clustering techniques provide substantially more unique documents than the collaborative relevance feedback technique of partial-user. However, as our results confirm, this introduction of unique documents does not improve performance.

Clustering in SCIR is a difficult issue, and the poor results reported here could be due to a number of factors. Firstly, the assignment of clusters may be problematic. The technique we have proposed to assign clusters is based on the overlap of documents between clusters over iterations of feedback. Although such a technique will strive to assign users as distinctive a cluster as possible, it is possible that the nature of the clusters from iteration to iteration may change substantially causing users to be assigned similar clusters. The poor performance of the term clustering technique may be caused by the aforementioned introduction of noise into the pro-

| Topic # | SCIR + Full Div | Partial User | Document Clustering | Term Clustering (T=10) | Term Clustering (T=20) | Term Clustering (T=30) | Term Clustering (T=40) | Term Clustering (T=50) |
|---|---|---|---|---|---|---|---|---|
| **303** | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| **307** | 35.75 | 36.32 | 36.69 | 37.49 | 36.72 | 36.60 | 36.26 | 36.09 |
| **322** | 2.99 | 2.90 | 2.90 | 2.91 | 2.90 | 2.91 | 2.90 | 2.90 |
| **326** | 35.43 | 34.74 | 35.79 | 33.50 | 34.23 | 34.74 | 35.15 | 35.04 |
| **339** | 5.93 | 5.94 | 5.92 | 5.92 | 5.93 | 5.92 | 5.92 | 5.91 |
| **347** | 25.69 | 26.02 | 25.50 | 23.97 | 24.56 | 24.85 | 25.10 | 25.20 |
| **352** | 31.88 | 34.89 | 32.43 | 27.53 | 29.47 | 30.39 | 30.28 | 30.40 |
| **353** | 30.82 | 30.68 | 30.64 | 30.89 | 30.68 | 30.54 | 30.63 | 30.69 |
| **357** | 26.69 | 24.80 | 25.57 | 22.79 | 23.84 | 24.68 | 24.94 | 24.94 |
| **362** | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 | 2.96 |
| **365** | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 | 10.97 |
| **366** | 17.98 | 17.96 | 17.92 | 18.58 | 18.51 | 18.36 | 18.23 | 18.09 |
| **387** | 8.99 | 9.15 | 9.02 | 8.79 | 8.86 | 8.93 | 9.03 | 9.01 |
| **392** | 31.10 | 32.26 | 31.12 | 30.61 | 30.92 | 30.87 | 30.76 | 30.76 |
| **408** | 15.67 | 15.02 | 15.64 | 14.63 | 14.68 | 14.75 | 14.75 | 14.78 |
| **414** | 9.63 | 9.61 | 9.34 | 9.57 | 9.62 | 9.73 | 9.70 | 9.68 |
| **428** | 25.80 | 26.39 | 25.93 | 25.35 | 25.77 | 25.57 | 25.75 | 25.85 |
| **431** | 29.77 | 29.47 | 29.41 | 32.15 | 31.35 | 30.98 | 31.05 | 30.76 |
| **438** | 31.87 | 32.23 | 31.77 | 30.27 | 30.46 | 30.46 | 30.13 | 30.37 |
| **446** | 29.79 | 30.61 | 29.31 | 26.34 | 27.21 | 27.06 | 27.55 | 27.62 |
| Overall | **20.79** | **20.95** | **20.74** | **20.06** | **20.28** | **20.36** | **20.40** | **20.40** |

Table 6.6: Comparison of SCIR + Full Div, collaborative relevance feedback, and clustering techniques

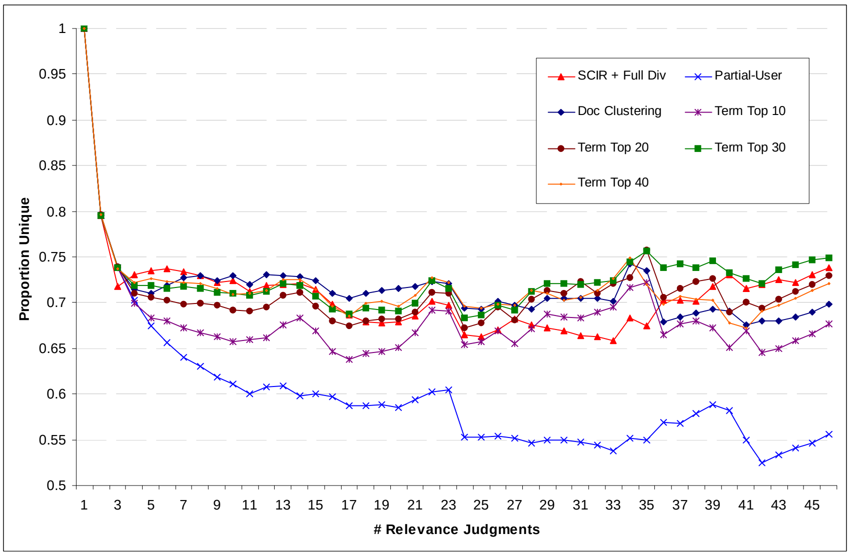


Figure 6.11: Comparison of the proportion of unique documents found across users' ranked lists for clustering techniques

cess. What the term clustering approach may cause, is the clustering of terms into two conceptual sets, *good* and *bad*, where the good terms are those with a higher relevance weighting, and the bad terms those with a lower weighting. In the next chapter we will discuss how we believe the clustering approach could be extended.

In this section we have introduced the notion of complementary feedback for synchronous collaborative information retrieval. The motivation of these techniques is to attempt to make users' queries more distinct than those produced using a collaborative relevance feedback technique. Our results show that the proposed techniques of complementary query expansion and clustering of documents and terms do not improve the performance over a collaborative relevance feedback technique, despite all techniques introducing substantially more diversity across users' ranked lists.

## 6.6 Summary

In this chapter we have investigated our hypothesis experimentally by evaluating the division of labour and sharing of knowledge techniques outlined in Chapter 4 using the evaluation methodology proposed in Chapter 5. Overall our results show that both techniques can improve the performance of a group of users searching together through a synchronous collaborative information retrieval system. In the next chapter we will summarise our conclusions from these evaluations.

# Chapter 7

# Conclusions and Summary

In this thesis we have explored the effects of a system-mediated division of labour and sharing of knowledge on a synchronous collaborative information retrieval environment. The motivation for this work was in the belief that for effective SCIR there needs to be an appropriate division of labour and sharing of knowledge among collaborating searchers. Although most work to date in SCIR had focussed on improving the awareness across users, so that they could coordinate the search activity themselves, requiring users to coordinate the group activity and search may result in users suffering from cognitive overload. We hypothesised that both a system-mediated division of labour and sharing of knowledge could improve the performance of the group search and we proposed techniques that allowed us to investigate the effects of each.

## 7.1 Research Objectives Re-Visited

Our primary research objective was to explore our hypothesis and evaluate the effects of both a division of labour and sharing of knowledge on an SCIR search, but before we could evaluate our hypothesis we needed to develop an effective and re-usable framework for evaluating synchronous collaborative information retrieval, and this became our secondary objective.

In Chapter 5 we outlined our evaluation methodology. This methodology was based upon building *simulations* of a synchronous collaborative information retrieval session involving two users, where the users searched together using a simple incremental feedback search system. We described the requirements for our SCIR simulations, and then outlined how we populated our experiments by mining rich format data from TREC interactive search experiments. Finally, we explored the notion of evaluation in SCIR, before proposing a novel evaluation framework which captures both the quality and diversity of group members' ranked lists over an entire search session. Although user simulations are never a precise method for IR evaluation, and indeed modelling a SCIR search was particularly difficult, simulations do allow for rapid and reusable evaluations based on a more realistic scenario than is possible with fully automatic experiments, whilst being much less expensive than fully interactive experiments. We believe that our evaluation methodology as outlined in Chapter 5 satisfies our research objectives of building an effective SCIR evaluation methodology. Nevertheless, there is always scope for improvement and in section 7.3 we will outline how we believe these simulations could be improved.

In Chapter 6, we explored our hypothesis. We proposed division of labour policies that enabled users to search unimpeded whilst ensuring that users were not presented with documents that their search partner had viewed or that we assumed they would view in the immediate future. For system-mediated sharing of knowledge we proposed using relevance information, inherent in most state-of-the-art SCIR systems through a shared bookmark tool, directly in the search through the relevance feedback process. We proposed novel techniques to extend the standard probabilistic relevance feedback algorithm into a *collaborative relevance feedback process* to allow for the incorporation of multi-user relevance information into both the relevance weighting and the offer weighting formulae. We experimented with all techniques in both a perfect and imperfect relevance scenario. We proposed an extension to the collaborative relevance feedback techniques which exploited the techniques' ability to allow for a user-biased combination of relevance information, by motivating the use of authority weighting and evaluating its effect on an SCIR search which assumed that users could make mistakes in their relevance judgments. Finally we explored the effects of a collaborative relevance feedback process on an SCIR search in terms of the total number of unique documents found across users' ranked lists and proposed a novel alternative to collaborative relevance feedback, *complementary relevance feedback*.

## 7.2 Conclusions

Overall our results showed that both a system-mediated division of labour and sharing of knowledge can improve the performance of an SCIR search. Our results showed that the most substantial improvement in performance is achieved through a division of labour policy. An SCIR system with both a division of labour and a system mediated sharing of knowledge offers modest increases over an SCIR system with just a division of labour policy over the entire search, with more substantive increases achievable over the first few iterations of feedback. This result confirmed our hypothesis that a system-mediated division of labour and sharing of knowledge policy improves performance over a standard SCIR search. We will now discuss the findings of each individual research question in more detail.

### 7.2.1 Division of Labour

Having two or more people searching together to satisfy a shared information need can allow the search task to be divided across searchers. In our work we evaluated the research question – *Does a division of labour policy improve the performance of a group of users searching together?*

In order to investigate this question we simulated users searching together through an SCIR system with various levels of division of labour, alongside two baseline systems of users searching independently and the best user searching alone.

Our experiments confirmed that the performance of an SCIR search can be significantly improved by implementing an explicit division of labour policy. In particular, we found that an aggressive division of labour which removed, from a user's ranked list, both the documents seen by their search partner, and the documents that we assume their search partner will examine, provides the best performance for division of labour systems.

Furthermore, our experiments also confirmed that if we are able to bring together two users in an SCIR environment with an explicit division of labour, then the group's searching performance can be significantly better than either the users searching separately with no coordination, or the best user searching alone.

In order to implement the best performing division of labour policy in a real world system, the number of documents that we assume the user will examine and therefore will not be returned to another user should be modified to suit the deployed system. For example, if we are building a distributed web based SCIR system we could set the number of assumed documents to 20 or 30 but for a PDA or iPhone we may instead want to limit this number to 5 or 10. Although as discussed in the previous chapter, even if users do not examine documents that they are assigned, these documents may be returned to either themselves or their search partner again over the duration of the search.

### 7.2.2 Sharing of Knowledge

One of the common features of state-of-the-art SCIR systems in the literature, is their use of a shared bookmarked facility which allows users to see documents deemed relevant by their co-searchers. In this thesis we attempted to use such explicit relevance judgments in the search task itself in order to improve the performance of the group through a system-mediated sharing of knowledge.

Firstly, we attempted to answer the research question: *Does a system-mediated sharing of knowledge policy, through a collaborative relevance feedback process, improve the performance of an SCIR search?* We investigated the sharing of knowledge through a collaborative relevance feedback process under an assumption of both perfect and imperfect relevance judgments, where imperfect relevance judgments are non-relevant documents that were perceived relevant by real users during the original TREC runs. We experimented with 3 different techniques for combining relevance information alongside the standard, single user, relevance feedback process (*pseudo user*). For two of these techniques, partial-user and combined weighting, we experimented with two variants, one that allowed users who had not encountered a term to provide a contribution towards its weighting (*contr*) and another which did not (*no contr*).

Our results showed that over an entire SCIR search, under either relevance assumption, passing relevance information between users does provide small improvements in the group's performance. Although no significance could be found at the predefined significance threshold of $p < 0.05$, improvements could be found at a significance level of $p = 0.165$ for perfect relevance information (combined weighting contr) and $p = 0.132$ for imperfect relevance information (combined weighting no contr). Although not strictly significant, these p values show a strong confidence that this difference is non-random. Encouragingly, our results confirm that over the first few iterations of feedback, a sharing of knowledge policy can offer a more substantial improvement in performance over an SCIR system which implements just a division of labour policy only and these results were significant at the significance threshold $p < 0.05$.

Under perfect relevance information, we experimented with two different interaction approaches, dynamic and static. The static approach represents a more standard interaction model, where users are presented with new ranked lists only when they interact with the SCIR system themselves, by making relevance judgments. The dynamic interaction approach, on the other hand, allows for a more intensive search environment, where users can benefit from their search partner's relevance judgments immediately by being presented with updated ranked lists, but at a cost on the user's cognitive load. Our results here showed that by implementing a dynamic system, we can get improvements over the static interaction approach, however, this improvement is not significant for all collaborative relevance feedback techniques operating in a static interaction environment. This result is interesting as it suggests that a standard interaction environment can perform as well as a more intensive environment.

Comparing across all collaborative relevance feedback techniques, results showed that the term based techniques outperform the document fusion technique under both relevance assumptions. Although no significance could be found at a significance threshold of p $<$ 0.05, we found that all techniques provide improvements under either relevance assumption at a more relaxed threshold of p $<$ 0.1. No significant difference could be found across the term-based techniques under either relevance assumption. In particular in both the perfect and imperfect relevance information scenario, the proposed term-based collaborative relevance feedback techniques performed as well as the standard single user (pseudo) technique. This result is to be expected as essentially all of these techniques are trying to do the same thing - aggregate relevance information from multiple users. The advantages of the proposed collaborative techniques over a standard relevance feedback technique is that they can allow for a user-biased combination of the relevance information for each user.

Our results showed that the inclusion of imperfect relevance information can significantly degrade the performance of a relevance feedback operation in an SCIR search. In order to explore the research question: *Can we circumvent some of the problems associated with imperfect relevance information through a user-biased collaborative relevance feedback process?* We proposed a method for user-biasing of the collaborative relevance feedback process based on the authority of their relevance judgments. Where a user's authority was based on the correlation between their relevance judgments and an oracle weighting of terms, our results revealed that attaching an authority weight to user's relevance judgments and using this authority weighting in a collaborative relevance feedback formula can improve the performance of the process. It should be noted that the differences in performance due to authority weighting are only minor. Given the slenderness of these increases under an oracle for predicting a user's authority, the obvious question is whether it is worthwhile pursuing research in this area and in particular, whether it is worthwhile attempting to develop automatic techniques for determining the authority of users without relevance information. We believe that there is scope for further research here for a number of reasons. Firstly, the difference in performance between an SCIR system operating under perfect versus imperfect relevance information, in these experiments, is quite small. Therefore, it is not entirely surprising that the authority weighting scheme did not provide large increases. In a real world setting there may be occasions where the differences between good and bad searchers in terms of their relevance judgments are more pronounced. For example, for some particularly difficult search topics a novice searcher may make poor relevance judgments. Alternatively, as some relevant documents may be better relevance feedback documents than others, an expert searcher may recognise good relevance feedback documents over documents that are simply relevant. Exploiting these differences through an effective authority weighting scheme could lead to an improved collaborative relevance feedback process. Furthermore, although the absolute differences, between authority weighting and unbiased collaborative relevance feedback found in our experiments are small, the fact that our significance tests reveal that these differences are non-random suggests that there could be further scope for improvement in the techniques in future work.

We hypothesised that the reason that the collaborative relevance feedback techniques offered only a modest improvement in an SCIR system with just a division of labour over the entire search was due to the aggregation of relevance information causing users to become too similar. We investigated the research question: *Does a collaborative relevance feedback process cause collaborating users' search results to become more similar than by using their own relevance information only?* We compared the number of unique document across the top 1000 documents from each user in a collaborative search, for an SCIR search with no sharing of relevance information (SCIR + Full Div) and for the collaborative relevance feedback techniques. Our results confirmed our intuition that significantly less unique documents were being found across users lists when they used a collaborative relevance feedback process than when they searched using their own relevance judgments only. The benefit of having multiple users searching together in order to satisfy a shared information need is that the search task can be divided across users, allowing users to explore different aspects of the collection. Early in the search, when neither user has found many relevant documents, a collaborative relevance feedback process can benefit a user by supplementing their relevance information. Over an entire search, however, the benefit offered by sharing relevance information is offset by the cost of losing uniqueness across users' ranked lists.

Based on these findings, in section 6.5, we explored an alternative use of multi-user relevance information in an SCIR search, in order to implement a *complementary relevance feedback* process. Unlike collaborative RF a complementary RF process attempts to leverage each user's relevance judgments in order to allow users to explore different areas of the collection. We explored the research question: *Can a complementary feedback mechanism allow user's search results to remain more distinct than a collaborative relevance feedback process, and does this improve the performance of the search?* We proposed two techniques here, complementary query expansion and clustering of documents and terms. Although both techniques do provide substantially more unique documents than the term-based collaborative RF techniques, they failed to improve the performance of the search.

## 7.3 Future Work

Our work and results pose many new research questions, and we will now outline where we feel progress could be made. Firstly we will discuss how improvements could be made to the evaluation methodology before describing how the techniques for division of labour and sharing of knowledge could be extended.

**Evaluation Methodology** We believe that the evaluation methodology, and in particular the simulations, could be improved and extended. In our simulations, we motivated the need to replace the static relevance judgments which were mined from the TREC rich format data, with dynamic relevance judgments based on the ranked lists returned to the simulated user. Although we have attempted to model a dynamic relevance judgment process, our simulations do not attempt to model the time aspect of the relevance judgments dynamically. Our simulations presume that the timing of relevance judgments will remain the same in the SCIR environment. Therefore, what our evaluations do not allow us to capture is how an SCIR search can improve the efficiency of a search, i.e. by dividing the search we are able to finish the search quicker. In our evaluations, the benefit from collaboration is measured in terms of the quality across users' ranked lists. If we developed a more comprehensive user model, which modelled the speed at which the user could read documents, and make relevance judgments for example, we could examine if an SCIR search would allow users to find more relevant documents more quickly.

In this thesis we investigated the notion of imperfect relevance information and its effect on the performance of an SCIR search. Our model operated the same across all users, all simulated users would react the same given the same ranked list. It would be interesting to extend the user model to allow us to model differences across users' expertise, for example modelling an expert searcher searching with a novice, where an expert searcher would always make correct relevance judgments but a novice could make mistakes.

In our evaluations we have modelled a simple incremental feedback system, which assumes that users receive a new ranked list after each relevance judgment is made. It would be interesting to explore the effects of a division of labour and sharing of knowledge policy in a more elaborate search system that allowed users to reformulate their query manually or make several relevance judgments before issuing them in batch to the search system.

In our work we examined a synchronous collaborative search involving two collaborating users. An obvious candidate for future work would be the extension of the simulated group to a group consisting of 3, 4, or 5 users or more and investigate how a division of labour and sharing of knowledge policy would operate in such a setting. For division of labour one issue could be that, as the number of users grow, removing documents that are assumed to be examined by users may result in some users being presented with ranked lists containing no relevant documents. For sharing of knowledge, it would be interesting to examine if the effect of the collaborative relevance feedback process of bringing users closer together would have an exaggerated negative impact as the size of the group grows.

An obvious area for future work is to apply these techniques to a real, interactive search session involving two or more real users. In our previous work we developed an interactive video search system for the TRECVid workshop. We would like to investigate the application of the proposed techniques in both a distributed and co-located domain. We would like to examine if the techniques provide more potential for improving the SCIR search in one domain over another.

Further to the previous point, in our work we have investigated the application of the techniques to the retrieval of textual data. An interesting area for future work would be to apply these techniques to an SCIR search over multi-media data. In particular, it would be interesting to examine the effects of a collaborative relevance feedback mechanism on a content-based video retrieval system, given the notorious difficulties associated with the *semantic gap* in content-based retrieval. Does the ability to aggregate multi-user relevance information allow for greater improvements in video retrieval than was found in these experiments over text data?

**Division of Labour** In this thesis we have explored the notion of authority in relation to the relevance feedback process. For future work it would be interesting to extend this notion of authority, and model cases where users can *skip* relevant material. Modelling such an environment would enable us to investigate how a division of labour policy would operate in an environment where searchers may read a document but fail to recognise it as relevant. At present, the division of labour policy outlined in this thesis would exclude this document from all ranked lists returned to all users for the rest of the search. This would cause the group to miss relevant material due to the actions of a poor searcher. A potential solution to this issue would be to extend the division of labour policy to allow for an approach which reduced the rank of seen documents rather than excluding them, as is currently the case. This "dampening effect" could be weighted by the perceived expertise of a user so that, if a more expert user reads a document without providing a relevance judgment on it, then we can be more confident that the document is non-relevant than if a poor searcher performed the action.

**Sharing of Knowledge** In this thesis we proposed techniques which allowed for a user-biased weighting of relevance judgments, and we investigated one application of a user-biasing through authority weighting. However we believe there are many more applications of a user-biased collaborative relevance feedback process. For example, in a real search system, a user could use a user-biased relevance feedback process to favour their own relevance judgements over their search partners', thereby allowing the results to be tailored to them. Or a user may decide to use an inverted approach and bias their results in the favour of their search partner if they feel they cannot locate any relevance documents on their own. We believe there are many more applications of our proposed collaborative relevance feedback process, which we have not even considered.

In this thesis we introduced the notion of imperfect relevance judgements to synchronous collaborative search. In traditional, single-user information retrieval, the notion of imperfect relevance judgments is less of an issue as, if a user has made a relevance judgment, then it should be considered as relevant for that user. When we move to a synchronous collaborative domain, in which a group of users are searching together, the issue of non-relevance or misunderstanding of the search topic could have a major effect of the performance of an SCIR search. We believe that this also represents an interesting area from future research. We investigated the application of authority weighting, and as discussed in the previous section we believe there is scope for future research here and, in particular, in future work we would like to apply some of the various query performance predictions techniques to the area of SCIR authority weighting.

Although the results of our evaluations of our proposed complementary relevance feedback techniques were not favourable, we believe there is still scope for further research here. In particular we feel that the clustering of documents and terms in a SCIR search may prove useful. In this thesis we attempted to cluster documents and terms into two clusters, one for each user. An interesting avenue for further research would be to try to use clustering to discover unique *concepts* in the search topic. For example a search topic "Wildlife Extinction" may have concepts related to zoos, poachers, animals, etc. By recognising the underlying concepts present in the relevance judgments, we may be able to make a more intelligent division of the clusters across users, whereby each user is assigned a unique concept to search.

## 7.4 Summary

Synchronous collaborative information retrieval is an exciting and emerging research area which is gaining momentum in the research community. The ability to allow two or more people to search together at the same time will become more in demand as novel computer interface devices such as the Microsoft Surface and the Apple iPhone become mainstream and allow users to collaborate on computer-related tasks in a co-located manner. In addition, the more sociable web we see developing on the internet allows remote people to communicate and collaborate easier than ever before. As people begin using computers more collaboratively, the need for effective techniques that allow users to search together synchronously will be demanded.

As such, this thesis represents an important contribution to the development of effective synchronous collaborative information retrieval systems. We have evaluated the effects of a system-mediated division of labour and sharing of knowledge on a collaborating group of searchers and have shown how the effectiveness of SCIR can be improved.

# Bibliography

Aalbersberg, I. J. (1992). Incremental relevance feedback. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–22, Copenhagen, Denmark. ACM Press.

Adcock, J., Pickens, J., Cooper, M., Anthony, L., Chen, F., and Qvarfordt, P. (2007). FXPAL Interactive Search Experiments for TRECVID 2007. In *TRECVid2007 - Text REtrieval Conference TRECVID Workshop*, Gaithersburg, MD, USA.

Ahn, J.-W., Brusilovsky, P., and Farzan, R. (2005). Investigating users' needs and behaviour for social search. In *Proceedings of the Workshop on New Technologies for Personalized Information Access (part of the 10th International Conference on User Modelling (UM '05))*, pages 1–12, Edinburgh, Scotland, UK.

Allan, J. (1996). Incremental relevance feedback for information filtering. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 270–278, Zurich, Switzerland. ACM Press.

Amazon (2007). http://www.amazon.com.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison Wesley.

Balabanović, M. (1997). An adaptive web page recommendation service. In Lewis Johnson, W. and Hayes-Roth, B., editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 378–385, Marina del Rey, California, United States. ACM Press.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72.

Bartell, B. T., Cottrell, G. W., and Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 173–181, Dublin, Ireland. Springer-Verlag New York, Inc.

Basilico, J. and Hofmann, T. (2004). Unifying collaborative and content-based filtering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 9, Banff, Alberta, Canada. ACM.

Belkin, N. J., Carballo, J. P., Cool, C., Kelly, D., jeng Lin, S., Park, S., Rieh, S. Y., Savage-Knepshield, P. A., and Sikora, C. (1998). Rutgers' TREC-7 interactive track experience. In *Text Retrieval Conference (TREC)*, pages 221–229.

Belkin, N. J., Carballo, J. P., Cool, C., Lin, S., Park, S. Y., Rieh, S. Y., Savage, P., Sikora, C., Xie, H., and Allan, J. (1997). Rutgers' TREC-6 interactive track experience. In *Text Retrieval Conference (TREC)*, pages 597–610.

Belkin, N. J., Cool, C., Croft, W. B., and Callan, J. P. (1993). The effect multiple query representations on information retrieval system performance. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, Pittsburgh, Pennsylvania, United States. ACM Press.

Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38.

Belkin, N. J., Kantor, P., Fox, E. A., and Shaw, J. A. (1995). Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448.

Blackwell, A. F., Stringer, M., Toye, E. F., and Rode, J. A. (2004). Tangible interface for collaborative information retrieval. In *CHI '04: extended abstracts on Human factors in computing systems*, pages 1473–1476, Vienna, Austria. ACM.

Blair, D. C. (2002). Some thoughts on the reported results of TREC. *Inf. Process. Manage.*, 38(3):445–451.

Brown, P. J. and Jones, G. J. F. (2001). Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Personal Ubiquitous Comput.*, 5(4):253–263.

Brusilovsky, P., Chavan, G., and Farzan, R. (2004). Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. In Nejdl, W. and De Bra, P., editors, *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH-04*, pages 24–34, Eindhoven, The Netherlands.

Buckley, C., Salton, G., and Allan, J. (1992). Automatic retrieval with locality information using SMART. In *The First Text REtrieval Conference (TREC)*, pages 59–72, National Institute of Standards and Technology, Gaithersburg, MD.

Buckley, C., Salton, G., and Allan, J. (1994). The effect of adding relevance information in a relevance feedback environment. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 292–300, Dublin, Ireland. Springer-Verlag New York, Inc.

Bush, V. (1945). As we may think. *The Atlantic Monthly*, 176(1):101–108.

Cabri, G., Leonardi, L., and Zambonelli, F. (1999). Supporting Cooperative WWW Browsing: a Proxy-based Approach. In *7th Euromicro Workshop on Parallel and Distributed Processing*, pages 138–145, University of Maderia, Funchal, Portugal. IEEE Press.

Campbell, I. (1995). Supporting information needs by ostensive definition in an adaptive information space. In Ruthven, I., editor, *MIRO '95. Elelectronic Workshops in Computing*, Glasgow, Scotland, UK.

Chang, Y. K., Cirillo, C., and Razon, I. (1971). Evaluation of feedback retrieval using modified freezing, residual collection, and test and control groups. *The SMART retrieval system-experiments in automatic document processing*, pages 355–370.

Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using ODP metadata to personalize search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, Salvador, Brazil. ACM Press.

Cleverdon, C. (1967). The Cranfield tests on index language devices. *Aslib Proceedings*, 19(6):173–194.

Cleverdon, C. and Keen, E. (1966). Factors Determining the Performance of Indexing Systems (Volume 1: Design; Volume2: Results). Technical report, Cranfield.

Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of WSDM*, Palo Alto, CA.

Crestani, F., Dunlop, M., and Mizzaro, S., editors (2003). *Mobile and Ubiquitous Information Access*, volume 2954 of *LNCS*. Springer-Verlag, Heidelberg, Germany.

Croft, W. B. (2002). *Advances in Information Retrieval*, volume 7 of *The Information Retrieval Series*, chapter Combining Approaches to Information Retrieval, pages 1–36. Springer Netherlands.

Croft, W. B. and Harper, D. J. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4):285–295.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 299–306, Tampere, Finland.

Crouch, C. J., Crouch, D. B., and Nareddy, K. R. (1990). The automatic generation of extended queries. In *SIGIR '90: Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 369–383, Brussels, Belgium. ACM Press.

Cuadra, C. A. and Katter, R. V. (1967). Opening the black box of relevance. *Journal of Documentation*, 23(4):291–303.

Dalal, M. (2007). Personalized social & real-time collaborative search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1285–1286, Banff, Alberta, Canada. ACM Press.

Decipho (2007). http://www.decipho.com.

Del.icio.us (2007). http://del.icio.us/.

Diamadis, E. T. and Polyzos, G. C. (2004). Efficient cooperative searching on the web: system design and evaluation. *Int. J. Hum.-Comput. Stud.*, 61(5):699–724.

Dieberger, A., Dourish, P., Höök, K., Resnick, P., and Wexelblat, A. (2000). Social navigation: techniques for building more usable systems. *Interactions*, 7(6):36–45.

Dietz, P. and Leigh, D. (2001). DiamondTouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, Orlando, Florida. ACM.

Digimind (2007). White paper: Web 2.0 for market intelligence and information research. http://www.digimind.com.

Dix, A. J., Finlay, J. E., Abowd, G. D., and Beale, R. (1998). *Human-Computer Interaction*. Prentice Hall, 2nd edition.

Donath, J. S. and Robertson, N. (1994). The Sociable Web. In *Proceedings of the Second International WWW Conference*, Chicago, IL.

Dunlop, M. D. and Brewster, S. (2002). The challenge of mobile devices for human computer interaction. *Personal Ubiquitous Comput.*, 6(4):235–236.

Ellis, C. A., Gibbs, S. J., and Rein, G. (1991). Groupware: some issues and experiences. *Commun. ACM*, 34(1):39–58.

Farzan, R., Coyle, M., Freyne, J., Brusilovsky, P., and Smyth, B. (2007). ASSIST: adaptive social support for information space traversal. In *HT '07: Proceedings of the 18th conference on Hypertext and hypermedia*, pages 199–208, Manchester, UK. ACM Press.

Fisher, H. L. and Elchesen, D. R. (1972). Effectiveness of Combining Title Words and Index Terms in Machine Retrieval Searches. *Nature*, 238(5359):109–110.

Fitzpatrick, L. and Dent, M. (1997). Automatic feedback using past queries: social searching? In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 306–313, Philadelphia, Pennsylvania, United States. ACM.

Foley, C., Gurrin, C., Jones, G., Lee, H., Mc Givney, S., O'Connor, N., Sav, S., Smeaton, A. F., and Wilkins., P. (2005). TRECVid 2005 Experiments at Dublin City University. In *TRECVid 2005 - Text REtrieval Conference TRECVID Workshop*, Gaithersburg, MD. National Institute of Standards and Technology.

Foley, C., Smeaton, A. F., and Lee., H. (2006). Synchronous collaborative information retrieval with relevance feedback. In *CollaborateCom 2006 - 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp1-4*, Adelaide, Australia.

Fooxx (2007). http://www.fooxx.com/.

Foster, J. (2006). Collaborative information seeking and retrieval. *Annual Review of Information Science and Technology*, 40(1):329–356.

Fox, E. A. (1983). Characteristics of two new experimental collections in computer and information science con- taining textual and bibliographic concepts. Technical report, Cornell University: Computing Science Department.

Fox, E. A. and Shaw, J. A. (1994). Combination of multiple searches. In *Text REtrieval Conference*, pages 243–249.

Gianoutsos, S. and Grundy, J. (1996). Collaborative work with the World Wide Web: adding CSCW support to a Web browser. In *Procedingss of Oz-CSCW'96, DSTC Technical Workshop Series, University of Queensland*, University of Queensland, Brisbane, Australia.

Glance, N. S. (2001). Community search assistant. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 91–96, Santa Fe, New Mexico, United States. ACM.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70.

Greenberg, S. and Roseman, M. (1996). GroupWeb: a WWW browser as real time groupware. In *CHI '96: Conference companion on Human factors in computing systems*, pages 271–272, Vancouver, British Columbia, Canada. ACM Press.

Greenfield, A. (2006). *Everyware: The Dawning Age of Ubiquitous Computing.* Peachpit Press, Berkeley, CA, USA.

Gross, T. (1999). Supporting awareness and cooperation in digital information environments. In *Basic Research Symposium at the Conference on Human Factors in Computing Systems - CHI'99*, Pittsburgh, Pennsylvania, USA. ACM.

Haines, D. and Croft, W. B. (1993). Relevance feedback and inference networks. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–11, Pittsburgh, Pennsylvania, United States. ACM Press.

Han, R., Perret, V., and Naghshineh, M. (2000). WebSplitter: a unified XML framework for multi-device collaborative Web browsing. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 221–230, Philadelphia, Pennsylvania, United States. ACM.

Hanani, U., Shapira, B., and Shoval, P. (2001). Information Filtering: Overview of Issues, Research and Systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259.

Hancock-Beaulieu, M., Gatford, M., Huang, X., Robertson, S. E., Walker, S., and Williams, P. W. (1996). Okapi at TREC-5. In *Text Retrieval Conference (TREC)*.

Hansen, P. and Järvelin, K. (2005). Collaborative information retrieval in an information-intensive domain. *Inf. Process. Manage.*, 41(5):1101–1119.

Harman, D. (1988). Towards interactive query expansion. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–331, Grenoble, France. ACM Press.

Harman, D. (1992). Relevance feedback revisited. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10, Copenhagen, Denmark. ACM Press.

Harman, D. (1993). Overview of the first trec conference. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 36–47, Pittsburgh, Pennsylvania, United States. ACM Press.

Harter, S. P. (1996). Variations in relevance assessments and the measurement of retrieval effectiveness. *J. Am. Soc. Inf. Sci.*, 47(1):37–49.

He, B. and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *SPIRE*, pages 43–54.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, Berkeley, California, United States. ACM Press.

Hust, A. (2004). Introducing Query Expansion Methods for Collaborative Information Retrieval. In Dengel, A., Junker, M., and Weisbecker, A., editors, *Reading and Learning - Adaptive Content Recognition*, volume 2956 of *Lecture Notes in Computer Science*, pages 252–280, Berlin, Heidelberg, New York. Springer-Verlag.

Ide, E. (1971). *The SMART Retrieval System*, chapter New Experiments in Relevance Feedback, pages 337–354. Prentice-Hall, Inc., Englewood Cliffs, N.J.

Ingwersen, P. and Järvelin, K. (2005). *The Turn: Integration of Information Seeking and Retrieval in Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Iwayama, M. (2000). Relevance feedback with a small number of relevance judgements: incremental relevance feedback vs. document clustering. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–16, Athens, Greece. ACM Press.

Jansen, B. J. and Spink, A. (2006). How are we searching the world wide web?: a comparison of nine search engine transaction logs. *Inf. Process. Manage.*, 42(1):248–263.

Kantor, P. B., Boros, E., Melamed, B., Meñkov, V., Shapira, B., and Neu, D. J. (2000). Capturing human intelligence in the net. *Commun. ACM*, 43(8):112–115.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience.

Kautz, H., Selman, B., and Shah, M. (1997). Referral web: combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65.

Keenan, S., Smeaton, A. F., and Keogh, G. (2001). The effect of pool depth on system evaluation in trec. *J. Am. Soc. Inf. Sci. Technol.*, 52(7):570–574.

Kempthorne, O. and Doerfler, T. E. (1969). The behavior of some significance tests under experimental randomization. *Biometrika*, 56(2):231–248.

Keskustalo, H., Järvelin, K., and Pirkola, A. (2006). The Effects of Relevance Feedback Quality and Quantity in Interactive Relevance Feedback: A Simulation Based on User Modeling. In Lalmas, M., MacFarlane, A., Rüger, S. M., Tombros, A., Tsikrika, T., and Yavlinsky, A., editors, *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006, Proceedings*, volume 3936, pages 191–204.

Kirsch, S., Gnasa, M., and Cremers, A. (2006). Beyond the Web: Retrieval in Social Information Spaces. In *Proceedings of the 28th European Conference on Information Retrieval (ECIR 2006)*, pages 84–95, London, UK. Springer.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM*, 40(3):77–87.

Krishnappa, R. (2005). Multi-user search engine (MUSE): Supporting collaborative information seeking and retrieval. Master's thesis, University of Missouri-Rolla, University of Missouri-Rolla, Rolla.

Kvan, T. (2000). Collaborative design: what is it? *Automation in Construction*, 9(4):409–415.

Lang, H., Wang, B., Jones, G., Li, J., Ding, F., and Liu., Y. (2008). Query performance prediction for information retrieval based on covering topic score. *Journal of Computer Science and Technology (in press)*.

Last.fm (2007). http://www.lastfm.com.

Laurillau, Y. and Nigay, L. (2000). Model of collaborative and synchronous navigation for large information space. In Höök, K., Wexelblat, A., and Munro, A., editors, *Workshop on Social Navigation: A Design Approach?, Extended Abstracts, CHI2000*, page 375, The Hague, Netherlands,. ACM Press.

Lieberman, H., Van Dyke, N. W., and Vivacqua, A. S. (1999). Let's browse: a collaborative web browsing agent. In *IUI '99: Proceedings of the 4th international conference on Intelligent user interfaces*, pages 65–68, Los Angeles, California, United States. ACM Press.

Liu, F., Yu, C., and Meng, W. (2002). Personalized web search by mapping user queries to categories. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 558–565, McLean, Virginia, USA. ACM Press.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 4:309–317.

Macqueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

Maekawa, T., Hara, T., and Nishio, S. (2006). A collaborative web browsing system for multiple mobile users. In *PERCOM '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, pages 22–35, Washington, DC, USA. IEEE Computer Society.

Magennis, M. and van Rijsbergen, C. J. (1998). The potential and actual effectiveness of interactive query expansion. *SIGIR Forum*, 31(SI):324–332.

Maron, M. E. and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7(3):216–244.

McGill., M., Koll, M., and Noreault, T. (1979). An evaluation of factors affecting document ranking by information retrieval systems. In *Final Report for grant BSF-IST-78-10454 to the National Science Foundation*, Syracuse University, Syracuse, New York.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Menkov, V., Neu, D., and Shi, Q. (2000). Antworld: A collaborative web search tool. In *DCW '00: Proceedings of the Third International Workshop on Distributed Communities on the Web*, pages 13–22, London, UK. Springer-Verlag.

Microsoft Surface (2007). http://www.microsoft.com/surface/.

Milic-Frayling, N., Sommerer, R., Rodden, K., and Blackwell, A. F. (2003). Smartview and searchmobil: Providing overview and detail in handheld browsing. In *Mobile HCI Workshop on Mobile and Ubiquitous Information Access*, pages 158–171, Udine, Italy.

Mizzaro, S. (1997). Relevance: The whole history. *Journal of the American Society of Information Science*, 48(9):810–832.

Morita, M. and Shinoda, Y. (1994). Information filtering based on user behavior analysis and best match text retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–281, Dublin, Ireland. Springer-Verlag New York, Inc.

Morris, M. (2007). Collaborating alone and together: Investigating persistent and multi-user web search activities. Technical report, Microsoft Research.

Morris, M. R. and Horvitz, E. (2007). SearchTogether: an interface for collaborative web search. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 3–12, Newport, Rhode Island, USA. ACM Press.

Morris, M. R., Paepcke, A., and Winograd, T. (2006). TeamSearch: Comparing Techniques for Co-Present Collaborative Search of Digital Media. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 97–104, Washington, DC, USA. IEEE Computer Society.

MovieFinder (2007). http://www.moviefinder.com.

MusicStrands (2007). http://www.musicstrands.com.

Naderi, H., Rumpler, B., and Pinon, J.-M. (2007). An Efficient Collaborative Information Retrieval System by Incorporating the User Profile. *Adaptive Multimedia Retrieval: User, Context, and Feedback*, 4398/2007:247–257.

Netscape Conferencer (2007). http://www.aibn.com/help/software/netscape/communicator/co

Nielsen, J. (2001). Search: Visible and simple. Jakob Nielsen's Alertbox, http://www.useit.com/alertbox/20010513.html.

NIST (1997). TREC-6 Interactive Track Specification. http://www-nlpir.nist.gov/projects/t6i/trec6spec.

O'Reilly, T. (2005). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408.

Poltrock, S., Grudin, J., Dumais, S., Fidel, R., Bruce, H., and Pejtersen, A. M. (2003). Information seeking and sharing in design teams. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 239–247, Sanibel Island, Florida, USA. ACM.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Rajashekar, T. B. and Croft, W. B. (1995). Combining automatic and manual index representations in probabilistic retrieval. *Journal of the American Society of Information Science*, 46(4):272–283.

Rees, A. M. and Saracevic, T. (1966). The measurability of relevance. In *Proceedings of the American Documentation Institute*, page 225234, Washington, DC. American Documentation Institute.

Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina. ACM.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Commun. ACM*, 40(3):56–58.

Robertson, S. and Spärck Jones, K. (1997). Simple proven approaches to text retrieval. Technical Report 356, Computer Laboratory, University of Cambridge.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304.

Robertson, S. E. (1990). On term selection for query expansion. *Journal of Documentation*, 46(4):359–364.

Robertson, S. E. and Spärck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146.

Robertson, S. E. and Walker, S. (1999). Okapi/Keenbow at TREC-8. In Voorhees, E. M. and Harman, D., editors, *Proceedings of the Eight Text REtrieval Conference (TREC-8)*, pages 151–162, Gaithersburg, MD. NIST.

Robertson, S. E., Walker, S., and Hancock-Beaulieu, M. (1998). Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive. In Voorhees, E. and Harman, D., editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 199–210, Gaithersburg, MD. NIST.

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gatford, M., and Payne, A. (1995). Okapi at TREC-4. In Harman, D. K., editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 73–96, Gaithersburg, MD. NIST.

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gull, A., and Lau, M. (1992). Okapi at TREC. In Harman, D., editor, *Proceedings of the First Text REtrieval Conference (TREC)*, pages 21–30, Gaithersburg, MD. NIST.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1993). Okapi at TREC-2. In Harman, D. K., editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 21–34, Gaithersburg, MD. NIST.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at TREC-3. In Harman, D. K., editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 109–126, Gaithersburg, MD. NIST.

Rocchio, J. (1971). Relevance feedback in information retrieval. Prentice Hall, Ing. Englewood Cliffs, New Jersey.

Rogers, Y., Youn-Kyung, L., and Hazlewood, W. (2006). Extending tabletops to support flexible collaborative interactions. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, Adelaide, Australia.

Romano Jr., N. C., Roussinov, D., Nunamaker, J. F., and Chen, H. (1999). Collaborative information retrieval environment: Integration of information retrieval with group support systems. In *HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 1*, Maui, Hawaii. IEEE Computer Society.

Root, R. W. (1988). Design of a multi-media vehicle for social browsing. In *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 25–38, Portland, Oregon, United States. ACM Press.

Roseman, M. and Greenberg, S. (1996). Building real-time groupware with GroupKit, a groupware toolkit. *ACM Trans. Comput.-Hum. Interact.*, 3(1):66–106.

Ruthven, I. (2003). Re-examining the potential effectiveness of interactive query expansion. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 213–220, Toronto, Canada. ACM Press.

Ruthven, I. and Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, 18(2):95–145.

Ruthven, I., Lalmas, M., and van Rijsbergen, C. J. (2003). Incorporating user search behavior into relevance feedback. *J. Am. Soc. Inf. Sci. Technol.*, 54(6):529–549.

Salton, G. (1971). *The SMART Retrieval System;Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Saracevic, T. (1999). Information science. *Journal of the American Society of Information Science*, 50(12):1051.

Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, Hong Kong, Hong Kong. ACM Press.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). *The Adaptive Web*, chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer Berlin / Heidelberg.

Search Wikia (2007). http://search.wikia.com/.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating “word of mouth”. In *CHI ’95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, Denver, Colorado, United States. ACM Press/Addison-Wesley Publishing Co.

Shen, C., Vernier, F. D., Forlines, C., and Ringel, M. (2004). DiamondSpin: an extensible toolkit for around-the-table interaction. In *CHI ’04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 167–174, Vienna, Austria. ACM.

Sidler, G., Scott, A., and Wolf, H. (1997). Collaborative Browsing in the World Wide Web. In *Proceedings of the 8th Joint European Networking Conference*, Edinburgh, Scotland. Springer.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

Singhal, A., Buckley, C., and Mitra, M. (1996). Pivoted document length normalization. In *Research and Development in Information Retrieval*, pages 21–29.

Smeaton, A. F., Lee, H., Foley, C., and Mc Givney., S. (2006). Collaborative Video Searching on a Tabletop. *Multimedia Systems Journal*, 12(4):375–391.

Smeaton, A. F., Murphy, N., O’Connor, N., Marlow, S., Lee, H., Donald, K. M., Browne, P., and Ye., J. (2001). The Físchlár Digital Video System: A Digital Library of Broadcast TV Programmes. In *JCDL 2001 - ACM+IEEE Joint Conference on Digital Libraries*, pages 312–313, Roanoke, VA.

Smeaton, A. F. and van Rijsbergen, C. J. (1983). The Retrieval Effects of Query Expansion on a Feedback Document Retrieval System. *The Computer Journal*, 26(3):239–246.

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., and Boydell, O. (2005). Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web Search Engine. *User Modeling and User-Adapted Interaction*, 14(5):383–423.

Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28(1):11–20.

Spärck Jones, K. (1979). Search term relevance weighting given little relevance information. *Journal of Documentation*, 35(1):30–48.

Spärck Jones, K. and van Rijsbergen, C. J. (1975). Report on the need for and provision of an ’ideal’ information retrieval test collection. British Library Research and Development Report 5266, University Computer Laboratory, Cambridge.

Spärck Jones, K. and Webster (1979). Research in relevance weighting. Technical Report 5553, Computer Laboratory, University of Cambridge.

Spink, A., Ozmutlu, S., Ozmutlu, H. C., and Jansen, B. J. (2002). U.S. versus European web searching trends. *SIGIR Forum*, 36(2):32–38.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, Boston, MA, USA.

Stewart, J., Bederson, B. B., and Druin, A. (1999). Single display groupware: a model for co-present collaboration. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 286–293, Pittsburgh, Pennsylvania, United States. ACM.

StumbleUpon (2007). http://www.stumbleupon.com/.

Svenonius, E. (1986). Unanswered questions in the design of controlled vocabularies. *Journal of the American Society for Information Science*, 37(5):331–340.

Talja, S. (2002). Information sharing in academic communities: types and levels of collaboration in information seeking and use. *The New Review of Information Behaviour Research*, 3:143–159.

Taube, M. (1965). A note on the pseudomathematics of relevance. *American Documentation,*, 16(2):69–72.

Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. (1997). PHOAKS: a system for sharing recommendations. *Commun. ACM*, 40(3):59–62.

Thompson, P. (1990). A combination of expert opinion approach to probabilistic information retrieval, part 1: The conceptual model. *Inf. Process. Manage.*, 26(3):371–382.

trec_eval (2008). http://trec.nist.gov/trec_eval/.

Tumer, K. and Ghosh, J. (1999). Linear and order statistics combiners for pattern classification.

Twidale, M. and Nichols, D. M. (1998). Designing interfaces to support collaboration in information retrieval. *Interacting with Computers*, 10(2):177–193.

Twidale, M. B., Nichols, D. M., and Paice, C. D. (1997). Browsing is a collaborative process. *Inf. Process. Manage.*, 33(6):761–783.

Utiyama, M. and Yamamoto, M. (2006). Relevance feedback models for recommendation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 449–456, Sydney, Australia. Association for Computational Linguistics.

van Rijsbergen, C. J. (1979). *Information Retrieval, 2nd edition.* Butterworth-Heinemann, Newton, MA, USA.

van Rijsbergen, C. J., Harper, D. J., and Porter, M. F. (1981). The selection of good search terms. *Information Processing and Management,*, 17:77–91.

Vinay, V., Cox, I. J., Milic-Frayling, N., and Wood, K. (2006). On ranking the effectiveness of searches. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 398–404, Seattle, Washington, USA. ACM Press.

Vogt, C. C. and Cottrell, G. W. (1998). Predicting the performance of linearly combined ir systems. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 190–196, Melbourne, Australia. ACM Press.

Voorhees, E. M. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–323, Melbourne, Australia. ACM Press.

Voorhees, E. M. and Harman, D. (1999). Overview of the Eighth Text REtrieval Conference (TREC-8). In Voorhees, E. M. and Harman, D., editors, *Text Retrieval Conference (TREC)*, Gaithersburg, MD. NIST.

Walker, S., Robertson, S. E., Boughanem, M., Jones, G. J. F., and Jones, K. S. (1997). Okapi at TREC-6 Automatic ad hoc, VLC, routing, filtering and QSDR. In Voorhees, E. and Harman, D., editors, *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, pages 125–136, Gaithersburg, MD. NIST.

Wexelblat, A. and Maes, P. (1999). Footprints: history-rich tools for information foraging. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 270–277, Pittsburgh, Pennsylvania, United States. ACM.

White, R. W., Ruthven, I., Jose, J. M., and Rijsbergen, C. J. V. (2005). Evaluating implicit feedback models using searcher simulations. *ACM Trans. Inf. Syst.*, 23(3):325–361.

Willett, P. (1988). Recent trends in hierarchic document clustering: a critical review. *Inf. Process. Manage.*, 24(5):577–597.

Wilson, T. D. (1981). On user studies and information needs. *Journal of Documentation*, 37(1):3–15.

Windows Live Messenger (2007). http://get.live.com/messenger/overview.

Yao, K.-T., Neches, R., Ko, I.-Y., Eleish, R., and Abhinkar, S. (1999). Synchronous and asynchronous collaborative information space analysis tools. *icppw*, 00:74.

Yom-Tov, E., Fine, S., Carmel, D., and Darlow, A. (2005). Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 512–519, Salvador, Brazil. ACM Press.

Yoono (2007). http://www.yoono.com.

Zeballos, G. S. (1998). Tools for Efficient Collaborative Web Browsing. In *Proceedings of CSCW'98 workshop on Collaborative and co-operative information seeking in digital information environment*, Seattle, Washington, USA.

Zhong, S. (2005). Efficient online spherical k-means clustering. In *IEEE international joint conference on neural networks (IJCNN '05)*, pages 3180–3185, Montreal, Canada.

Zhou, Y. and Croft, W. B. (2006). Ranking robustness: a novel framework to predict query performance. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 567–574, Arlington, Virginia, USA. ACM Press.

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, Melbourne, Australia. ACM Press.