Combining Relevance Information in a Synchronous Collaborative Information Retrieval Environment

Colum Foley, Alan F. Smeaton and Gareth J. F. Jones
Centre for Digital Video Processing and Adaptive Information Cluster
Dublin City University, Glasnevin, Dublin 9, IRELAND
colum.foley@computing.dcu.ie

Abstract

Traditionally information retrieval (IR) research has focussed on a single user interaction modality, where a user searches to satisfy an information need. 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-related tasks. Due to these advances there is an increasing need 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 chapter we explore the effectiveness of a *sharing of knowledge* policy on a collaborating group. Sharing of knowledge refers to the process of passing relevance information across users, if one user finds items of relevance to the search task then the group should benefit in the form of improved ranked lists returned to each searcher.

In order to evaluate the proposed techniques we simulate two users searching together through an incremental feedback system. The simulation assumes that users decide on an initial query with which to begin the collaborative search and proceed through the search by providing relevance judgments to the system and receiving a new ranked list. In order to populate these simulations we extract data from the interaction logs of various experimental IR systems from previous Text REtrieval Conference (TREC) workshops.

1 Introduction

The phrase "*Collaborative Information Retrieval*" has been used in the past to refer to many different technologies which support collaboration in the information retrieval (IR) process. Much of the early work in collaborative information retrieval has been concerned with asynchronous, remote collaboration via the reuse of previous search results and processes in collaborative filtering systems, collaborative re-ranking, and collaborative footprinting systems. Asynchronous collaborative information retrieval supports a passive, implicit form of collaboration where the focus is to improve the search process for *an individual*.

Synchronous collaborative information retrieval (SCIR) 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. SCIR systems represent a significant paradigmatic shift in focus and motivation compared with traditional IR systems and asynchronous collaborative IR systems. The development of new IR techniques is needed to exploit this. In order for collaborative IR 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 knowledge 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 the collaborating searchers to coordinate their activities in order to achieve a division of labour and sharing of knowledge. However, coordinating activities amongst users can be troublesome, requiring too much 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 amongst searchers (Morris and Horvitz, 2007), or defining searcher roles (Adcock et al., 2007). However, there has been no work to date which addresses the system-mediated sharing of knowledge across collaborating searchers. In this chapter we introduce our techniques to allow for effective system-mediated sharing of knowledge. We evaluate how a sharing of knowledge policy affects the performance of a group of users searching together collaboratively. But first, in the next section, we provide a comprehensive account of work to date in synchronous collaborative information retrieval.

2 Synchronous Collaborative Information Retrieval

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 IR system is to satisfy an information need.

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 IR systems from an individual focus to a group focus. As such these systems represent a more explicit, active form of collaboration, where users are aware that they are collaborating with others towards a common, and usually explicitly stated, goal. This collaboration 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 explicit, synchronous collaborative information retrieval is becoming more important than ever.

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. Division of labour means that each member of a collaborating group can explore a subset of information thereby reducing the redundancy associated with two or more people viewing the same documents, and improving the efficiency of a search. Some methods proposed here include increasing the awareness amongst users of each collaborative searcher's progress (Diamadis and Polyzos, 2004; Smeaton et al., 2006) or system-mediated splitting of a 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 SCIR 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 SCIR tools. We will now outline research to date in each of these areas.

2.1 Synchronous Remote Collaborative Information Retrieval

SCIR systems have been developed to enable remote 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 also supported synchronous scrolling and independent scrolling on a web page and supported the use of telepointers (showing others' mouse pointers on the page) in order to allow users to focus the attention of the group and to enact gestures. GroupWeb provided an annotation window where groups could attach shared annotations to pages when viewing and these annotations could be viewed by all group members. In GroupWeb, group members were tightly coupled. Enabling each user to see the same documents increases awareness amongst group members but can be an inefficient technique for exploring the vastness of the web.

The W4 browser (Gianoutsos and Grundy, 1996) extended the GroupWeb system to allow users to browse the web independently whilst synchronising their work. In W4, a user could view all pages viewed by other users, they could chat with each other, share bookmarks (i.e. documents deemed relevant), and see a shared WYSIWIS white-board to brainstorm. Users could also embed chat sessions, links and annotations directly into a web-page. A similar approach was employed by Cabri et al. (1999), which used a proxy server to record documents viewed by others. These documents were then displayed to each user in a separate browser window. The system also made others aware of these viewed documents by editing the HTML mark-up in pages viewed by each collaborating searcher (links to pages already viewed by other users in a session were indicated using different colours).

The above systems all required users to explicitly log onto a service to support collaborative searching. 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 a more spontaneous collaboration. 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 them 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.

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 other users had visited that page previously. Users could also provide ratings for pages using a thumbs-up or thumbs down metaphor. 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, 2001) 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.

2.2 Synchronous Co-located Collaborative Information Retrieval

Recent advances in ubiquitous computing devices such as mobile phones and PDAs have allowed researchers to begin exploring techniques for spontaneous collaborative search.

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. WebSplitter (Han et al., 2000) was a similar system for providing partial views to web pages across a number of users and potentially across a number of devices available to a user (e.g. laptop, PDA).

Advances in single display groupware (SDG) technology (Stewart et al., 1999), have enabled the development of collaborative search systems for the co-located environment. The advantages of such systems are that they improve the awareness of collaborating searchers by bringing them together in a face-to-face environment. Increased awareness can enable both a more effective division of labour and a greater sharing of knowledge.

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 approach 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.

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 predefined metadata categories for the images.

Físchlár-DiamondTouch (shown in Figure 1) was a multi-user video search application developed by the authors and others at the Centre for Digital Video Processing at Dublin City University (Smeaton et al., 2006). Físchlár-DiamondTouch was developed on an interactive table known as DiamondTouch (Dietz and Leigh, 2001) and used the groupware toolkit DiamondSpin (Shen et al., 2004) from Mitsubishi Electric Research Labs (MERL). The system 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). 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 represented the first time any group had performed collaborative search in any TREC or TRECVid workshop.

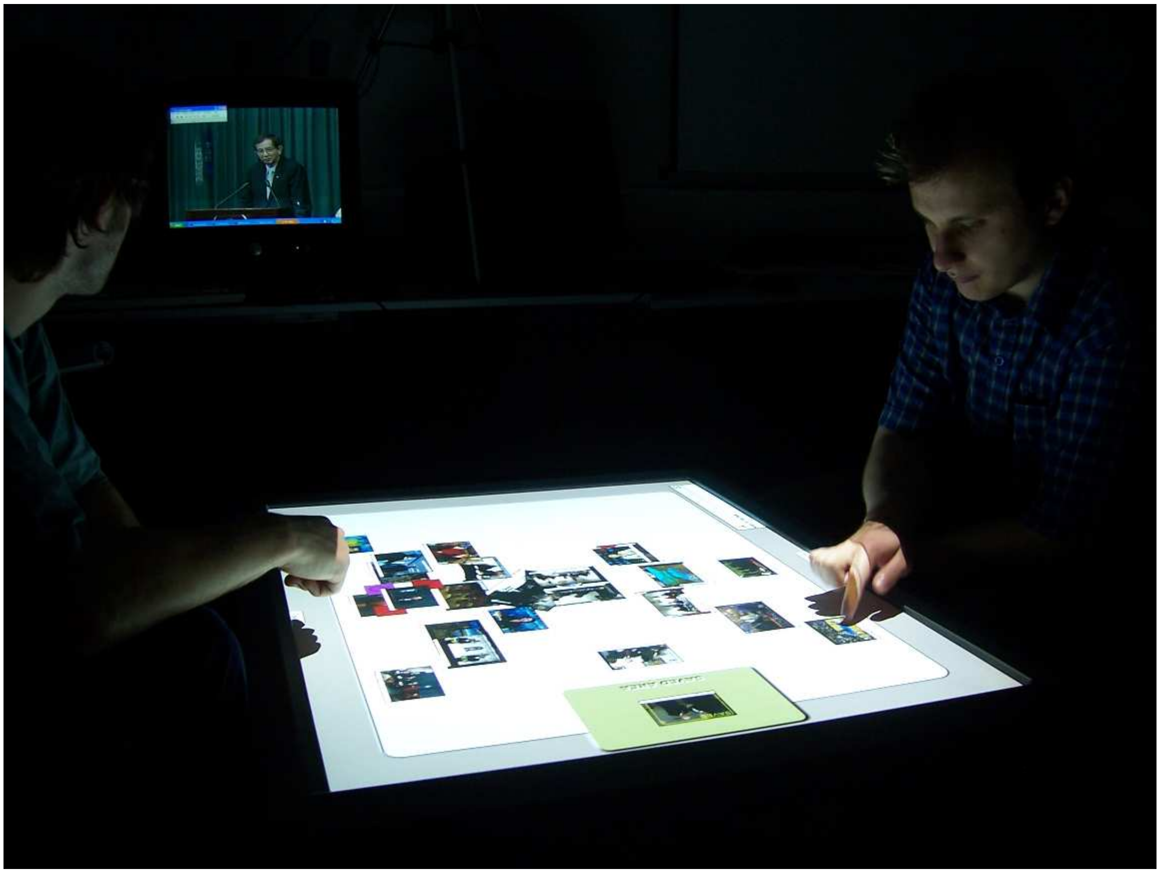


Figure 1: Físchlár-DiamondTouch

In an effort to improve collaborative search effectiveness through “algorithmically-mediated collaboration”, the “Cerchiamo” system was developed by the FXPAL TRECVid team (Adcock et al., 2007). Cerchiamo was designed to support two users working together to find relevant shots of videos. 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.

In this section we have described work to date in synchronous collaborative information retrieval (SCIR). Early SCIR systems focussed on providing tools to increase awareness across collaborating users. Research has suggested that the performance of a group searching in a SCIR system can be improved by allowing for a division of labour and sharing of knowledge. In early SCIR systems, the onus for coordinating the group's activities was placed on the users. Work has been done into allowing for a system-mediated division of labour, but no work, as yet, has attempted to implement a system-mediated sharing of knowledge in order to improve the quality of ranked lists returned to users searching together in an adhoc search task.

3 Sharing of Knowledge in Synchronous Collaborative Information Retrieval

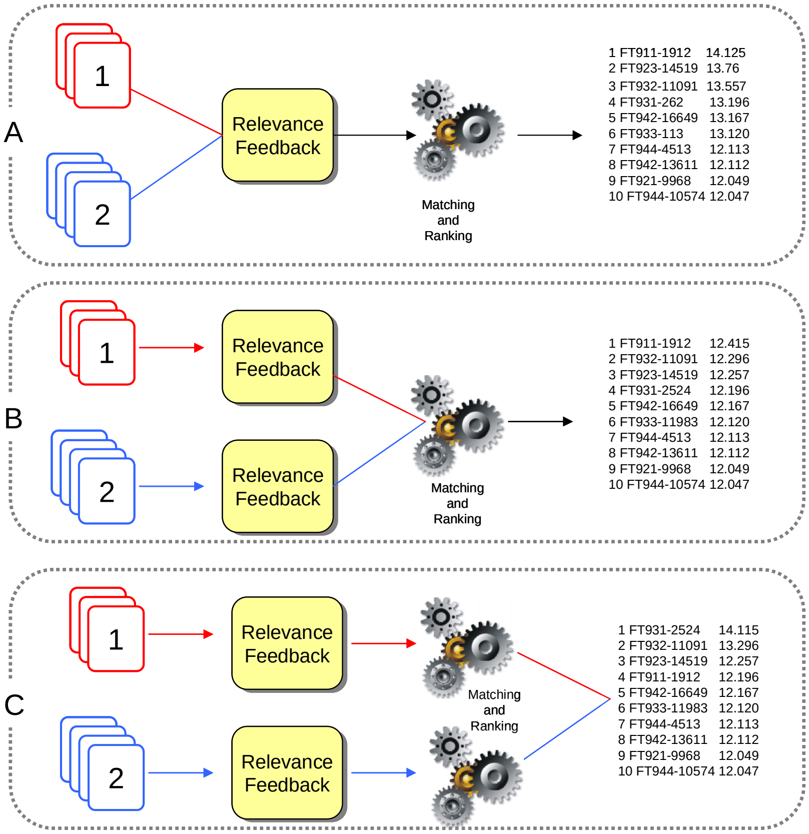
Suppose two users are searching together to satisfy a shared information need using a state-of-the-art SCIR system as described in the previous section. When two or more users come together in an SCIR environment, there are several ways in which to initiate the collaborative search. For example, users may each decide to formulate their own search query, or users may decide on a shared, group query. 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). As users examine documents and find those relevant to the search, they may save them to a "bookmarked" area. What these users are doing is providing *explicit relevance judgments* to the search engine. In traditional, single-user IR, these relevance judgments are often used in a process known as *relevance feedback* 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.

Relevance feedback is an IR technique which has been proven to improve the performance of the IR process through incorporating extra relevance information provided by users (in the form of documents identified as relevant by a user) into an automatic query reformulation process. The basic operation of relevance feedback consists of two steps: (1.) query expansion – whereby significant terms from documents judged relevant are identified and appended to the user's original query, and (2.) relevance weighting – which biases weights of each query term based on this relevance information.

If we provided a relevance feedback mechanism in an SCIR system, then when a member of an SCIR group initiates a relevance feedback operation, if their search partner has provided relevance judgments to the system, we could incorporate both users' relevance judgments into the feedback process. This could enable better quality results to be returned to the user. Furthermore such a sharing of knowledge policy could allow users to benefit from the relevance of a document without having to view the contents of the document. It is not clear, however, how multi-user relevance information should be handled in a relevance feedback process.

3.1 Collaborative Relevance Feedback

One of 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 over these documents. There may, however, be occasions where it is desirable to allow for a user-biased combination of multi-user relevance information. Therefore, we will outline how the RF process can be extended to allow for a weighted combination of multi-user relevance information in a *collaborative relevance feedback process*. Combination of evidence is an established research problem in IR (Croft, 2002), in our work we are interested in investigating the combination of multi-user relevance information within the relevance feedback process. In our work we use the probabilistic model for retrieval which is both theoretically motivated, and proven to be successful in controlled TREC experiments first shown in (Robertson et al., 1992). In the probabilistic retrieval model the relevance feedback processes of *Query Expansion* and *Term Reweighting* are treated separately (Robertson, 1990). Figure 2 presents a conceptual overview of the collaborative relevance feedback process for two users 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.



1	FT911-1912	14.125
2	FT923-14519	13.76
3	FT932-11091	13.557
4	FT931-262	13.196
5	FT942-16649	13.167
6	FT933-113	13.120
7	FT944-4513	12.113
8	FT942-13611	12.112
9	FT921-9968	12.049
10	FT944-10574	12.047

1	FT911-1912	12.415
2	FT932-11091	12.296
3	FT923-14519	12.257
4	FT931-2524	12.196
5	FT942-16649	12.167
6	FT933-11983	12.120
7	FT944-4513	12.113
8	FT942-13611	12.112
9	FT921-9968	12.049
10	FT944-10574	12.047

1	FT931-2524	14.115
2	FT932-11091	13.296
3	FT923-14519	12.257
4	FT911-1912	12.196
5	FT942-16649	12.167
6	FT933-11983	12.120
7	FT944-4513	12.113
8	FT942-13611	12.112
9	FT921-9968	12.049
10	FT944-10574	12.047

Figure 2: Combining relevance information, the 3 choices

In particular we have identified three stages in the process at which we can combine relevance information.

3.1.1 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:

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

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

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

$$p = \frac{r_i}{R} \qquad (2)$$

$$q = \frac{n_i - r_i}{N - R} \qquad (3)$$

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 1 we get the standard relevance weighting formula (Robertson and Spärck Jones (1976)):

$$rw(i) = \log \frac{\left(\frac{r_i}{R}\right)\left(1 - \frac{n_i - r_i}{N - R}\right)}{\left(\frac{n_i - r_i}{N - R}\right)\left(1 - \frac{r_i}{R}\right)} \qquad (4)$$

If we assume that in a collaborative search session we have *U* collaborating users searching. Then the proportions for *p* and *q*, in equations 2 and 3 respectively, can be extended as follows:

$$p = \sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}}{R_u} \tag{5}$$

$$q = \sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}}{N - R_u} \tag{6}$$

where

n_i, 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
α_{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 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 α , which can be used to vary the effect of each user's proportion in the final calculation, and a default value of $\frac{1}{\mathrm{U}}$ can be used to consider all users equally.

One important consideration when combining multi-user relevance information is what to do when a term has not been encountered by a user (i.e. the term is not contained in the user's relevance judgments). There are two choices here, we can either allow the user that 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*, of 0 ($\frac{0}{\mathrm{R}}$) and a non relevance proportion, *q*, of $\frac{\mathrm{n}}{\mathrm{N\text{-}R}}$, from a user who has not encountered the term (as $r_i = 0$ for that user).

If we do not wish to incorporate a user's proportion for a term, in the case that they have not encountered the term, then the shared relevance and non-relevance proportions of *p* and *q* in equation 5 and equation 6 respectively will only be composed of the proportions from users who have encountered the term.

Applying the extended proportions of *p* and *q*, in equations 5 and 6 respectively, to the probability of relevance from equation 1, results in our *partial-user relevance weight* (*purw*):

$$purw(i) = \log \frac{\left(\sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}}{R_u}\right)\left(1 - \sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}}{N - R_u}\right)}{\left(\sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui}}{N - R_u}\right)\left(1 - \sum_{u=0}^{U-1} \alpha_u \frac{r_{ui}}{R_u}\right)} \qquad (7)$$

For practical implementation of the standard relevance weighting formula (equation 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 7, results in:

$$purw(i) = \log \frac{\left(\sum_{u=0}^{U-1} \alpha_u \frac{r_{ui} + 0.5}{R_u + 1}\right)\left(1 - \sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui} + 0.5}{N - R_u + 1}\right)}{\left(\sum_{u=0}^{U-1} \alpha_u \frac{n_i - r_{ui} + 0.5}{N - R_u + 1}\right)\left(1 - \sum_{u=0}^{U-1} \alpha_u \frac{r_{ui} + 0.5}{R_u + 1}\right)} \qquad (8)$$

So far we have shown how the partial-user method can be applied to the standard relevance weighting formula which is used for reweighting terms in the relevance feedback process. Now we will consider applying the scheme to the offer weighting formula (Robertson (1990)), which is used to rank terms for query expansion:

$$ow_i = r_i \times rw_i \qquad (9)$$

Using a linear combination approach the r_i value in equation 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*):

$$puow(i) = \left(\sum_{u=0}^{U-1} \alpha_u r_{ui} \times purw(i) \right) \qquad (10)$$

where

α_u = Determines the impact of each user's r_i value on the final weight, and

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

3.1.2 Combining Outputs of the Relevance Feedback Process (B)

This method of combination operates at a higher level of granularity than the previous method by treating the relevance process as a black box. In this method, for each user, relevance weighting and offer weighting are calculated separately using only a searcher's own relevance statistics (i.e. terms from documents identified as relevant by the user and their distribution in these relevant documents). The outputs from these processes (i.e. the scores) are combined to produce a combined weight. 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 (*crw*) using a linear combination of relevance weight scores from all users:

$$crw(i) = \sum_{u=0}^{U-1} \alpha_u \times rw_{ui} \qquad (11)$$

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*):

$$cow(i) = \sum_{u=0}^{U-1} \alpha_u \times ow_{ui} \qquad (12)$$

where

α_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 or combined offer weight if they have not encountered the term in their own set of relevance judgments. Once again the α 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 give all users the same weighting.

3.1.3 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).

In order to produce a combined ranked list, a reformulated query is generated for each collaborating user, based on their own relevance information, 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 of the document scores produced by the search engine, to arrive at a combined document score (*cds*):

$$cds(d,q) = \sum_{u=0}^{U-1} \alpha_u \times s_{ud} \qquad (13)$$

where

s_{ud} = the relevance score for document d in relation to user u's query

α_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. Evaluating these techniques will allow us to establish how a sharing of knowledge policy, via collaborative relevance feedback, impacts on the performance of a collaborating group. In the next section we will outline how we plan to evaluate these techniques.

3.2 Experimental Setup

In this chapter, we are evaluating many different approaches to combining relevance information. 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 are realistic.

3.2.1 Requirements Analysis

Previous IR experiments that have used user simulations have focussed on a single user's interactions with an IR system. Here we are attempting to simulate a synchronous collaborative IR environment, a dynamic, collaborative simulation. We are conscious that the simulation 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 of the collaborative nature of a search task have shown how the majority of synchronous collaborative search sessions involve a collaborating group of size two (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 groups.

One of the important considerations for any SCIR system is how to begin a collaborative search. For the experiments reported here the search assumes that one initial query has been formulated. In a real system, this query could be formulated by one user or by both users collaboratively. 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.

In these experiments we are interested in evaluating how a system-mediated sharing of knowledge policy can operate alongside an explicit division of labour policy. Therefore, the simulated SCIR system will implement a system-mediated division of labour where the results returned to a searcher at any point in the search are automatically filtered in order to remove:

1. Documents seen by their search partner.
2. Documents assumed seen by their search partner. These are documents that are in their search partner's current ranked list.

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 a simple relevance feedback mechanism. In any SCIR search, users may provide multiple relevance assessments over the course of the search session, and therefore 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 a relevance judgment, or after the user has provided a certain number of relevance judgments. For the purposes of the experiments reported in this chapter, 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. This can enable users to benefit immediately from their relevance judgments. Furthermore, studies have shown that applying feedback after only one or two relevant documents have been identified can substantially improve performance over an initial query (Spärck Jones, 1997).

Another choice for any SCIR system, related to feedback granularity, is whether to present a user with a new ranked list only when they perform feedback themselves or when they *or their search partner* provides feedback. Presenting users with a new ranked list when their search partner performs relevance feedback may allow users to benefit more quickly from their partner’s relevance judgments. However deployment of such an intensive SCIR system would require designers to develop novel interface techniques to allow for the seamless updating of a user’s ranked list. Furthermore, users searching in such an intensive system may suffer from *cognitive overload* by being presented with new ranked lists, seemingly, at random. In this chapter we will evaluate the effects of both of these interaction environments on an SCIR search.

Figure 3, presents a conceptual overview of the SCIR system we will simulate in our evaluations.

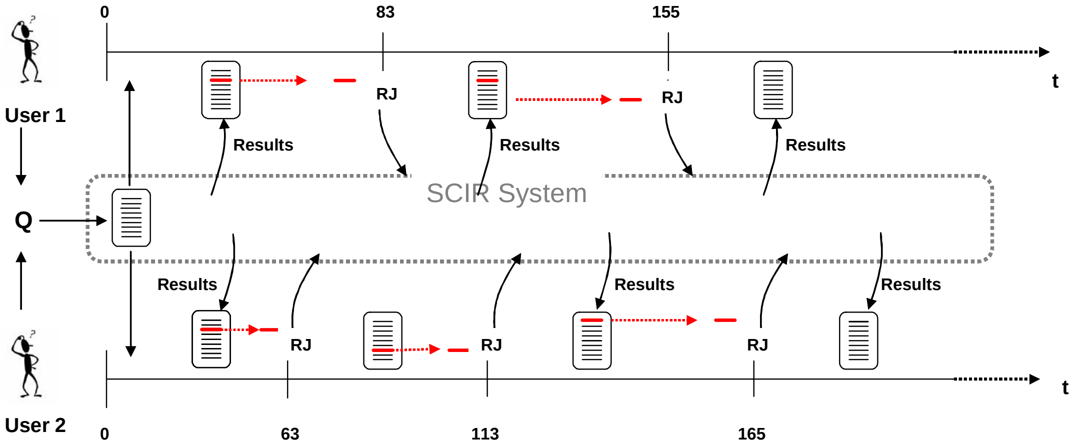


Figure 3: A simulated SCIR session

Referring to Figure 3, the data required to populate our SCIR simulations is:

- An initial query (Q) – as outlined above, the simulated SCIR session begins with an initial query entered by the set of two users.
- Series of relevance judgments (RJ) – these are explicit indications of relevance made by a user on a particular document.
- Timing information – this represents the time, in seconds, relative to the start of the search session at which relevance judgments were 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.

3.2.2 Methodology

In order to populate these simulations we have mined data from previous TREC interactive search experiments. The Text REtrieval Conference (TREC) is an annual workshop established in 1992 under the auspices of the National Institute for Standards and Technology (NIST) in an attempt to promote research into IR. For each TREC, a set of tasks or *tracks* are devised, each evaluating different aspects of the retrieval process. The purpose of the TREC (Text REtreival Conference) interactive 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. For the interactive track at TREC 6 – TREC 8 (1997-1999), 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. For our simulations we extracted data from several groups' submissions to TREC 6 – TREC 8. Unfortunately, it was not possible to extract data for our simulations from all participating groups' submissions to these TRECs. This was due to a variety of reasons but typically some 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. In our simulations we used this extracted data to simulate two users, who originally searched the topic separately as part of their group's submission, searching together with an SCIR system.

Figure 4 shows a conceptual overview of a simulated SCIR session involving two users whose rich format data was extracted from TREC data, searching on a TREC topic entitled "Hubble Telescope Achievements". From Figure 4, we can see that the search begins with the group query "positive achievements hubble telescope data", which is the concatenation of both users' original queries. 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*.

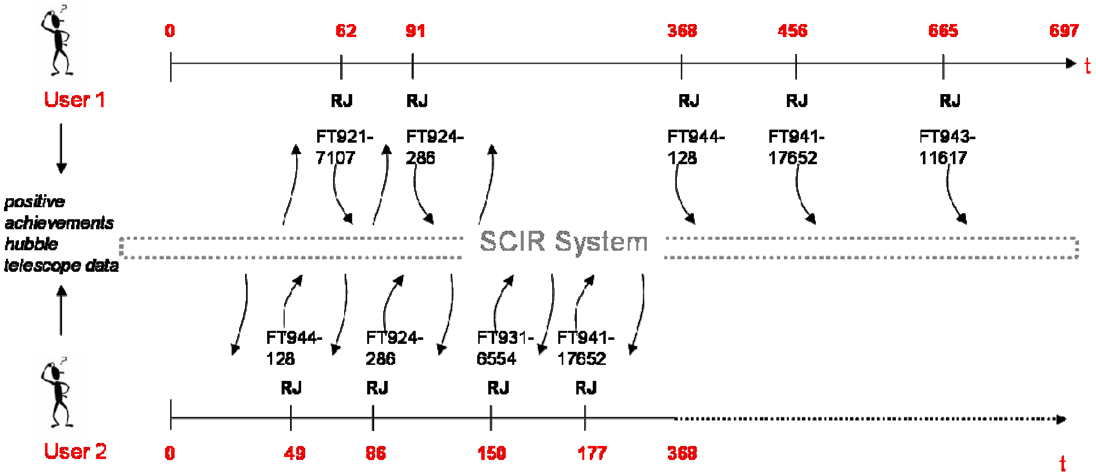


Figure 4: Conceptual overview of two searchers searching together

By extracting rich format data associated with different users' interactions on a search topic, we can acquire multiple heterogeneous simulations, where the data populating our simulations is from real users searching to satisfy the same information need on a standardised corpus. There were a total of 20 search topics used in these TREC workshops, with varying degrees of difficulty and therefore our simulations were evaluated across these 20 topics.

3.2.2.1 Relevance Judgments

The **SCIR simulations** proposed thus far are based on taking *static* rich format data, which records a user's previous interactions with a particular search engine, and imposing our SCIR simulated environment on this data. By imposing our own simulated environment on this rich format data, we cannot assume that users would have saved the same documents as that 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 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, in the simulations used in our experiments it is important that the 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.

Before finalising our simulations, we also need to enforce an upper limit on the number of documents a 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 less documents according to the device they are using for searching, we feel that 30 is a reasonable figure to assume users will examine in any SCIR search.

After performing relevance feedback, the relevance judgments made by a user are never returned to them again for the duration of that search. As our baseline SCIR system will implement a division of labour policy, we also remove these *seen* documents from their search partner's subsequent list. Furthermore, as we assume that users will examine a maximum of 30 documents, our baseline system will also remove these documents from their search partner's ranked list.

3.2.3 Evaluation Metric

At each stage in an SCIR session, each collaborating user will have associated with them a ranked list of documents. This list could have been returned to the user either as a result of the initial query or after performing relevance feedback. In traditional, single-user IR the accuracy of each individual searcher's ranked lists can be evaluated using standard IR measurements such as average precision (AP), a measure which favours systems that rank relevant documents higher in a returned ranked list of documents. In our work we are concerned with the performance of a *group* of users, and therefore we need to be able to assign a score to the collaborating group at any particular point in the search process. What we need is a measure which captures the quality and diversity across collaborating users' ranked lists. Our solution is to calculate the *total number of unique relevant documents across user's ranked lists* at a certain cut-off 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, our measure of quality is taken at a cut-off of 30 documents from each users list (i.e. a total of 60 documents). 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 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.

3.2.4 Measurement Granularity

In our simulations of SCIR search, a user is presented with a new ranked list of documents each time 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 illustrates the procedure followed to calculate the performance of a group over the duration of an SCIR simulation involving two collaborating users. 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. 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") including the one relevance judgment made. As this is the first relevance feedback iteration in the SCIR session, the AP for this merged list is plotted at position 1 on the graph. The measurement proceeds in this manner, by calculating the number of unique relevant documents across each user's current ranked list after each relevance judgment, these figures are plotted after each relevance feedback iteration in order to show the group's performance over the course of the entire search.

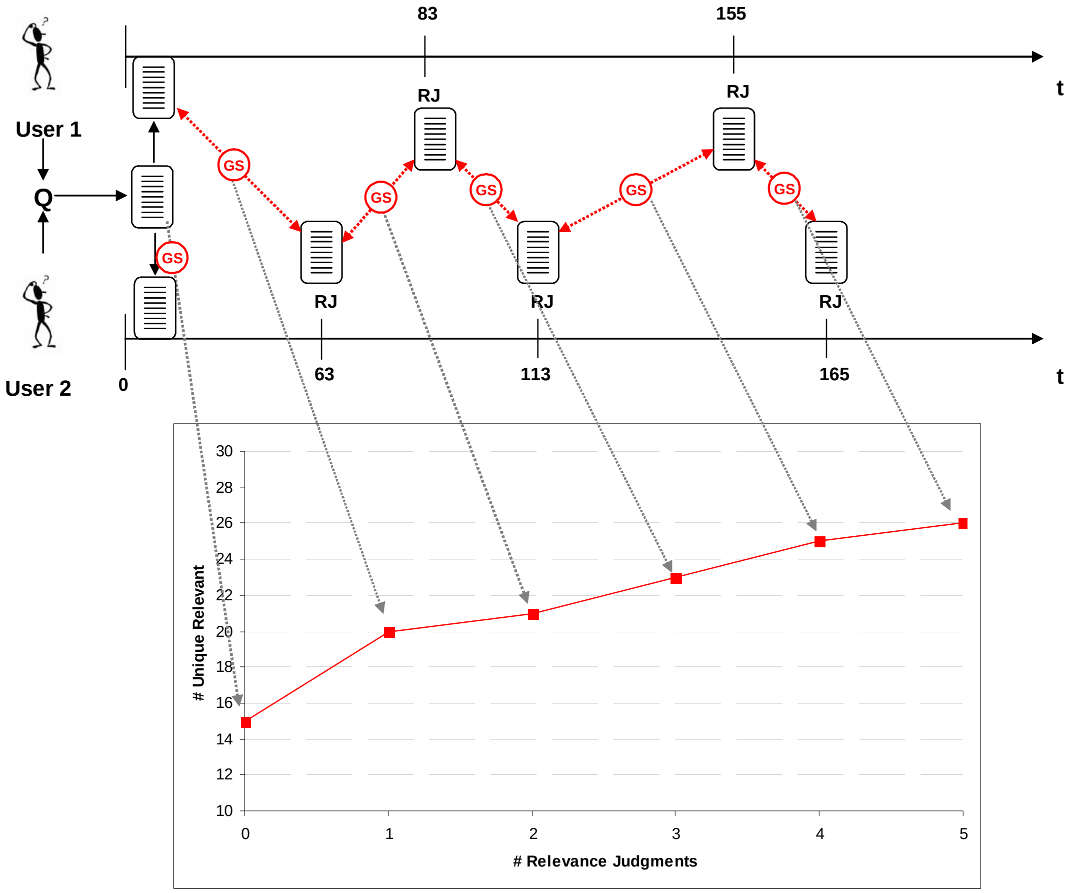


Figure 5: Measurement granularity used in experiments

From this graph, we can also generate a single performance figure for the entire search, by averaging the group score after the initial query and each subsequent feedback iteration.

3.2.5 Significance Testing

We will use this single performance figure in order to test for significance in our results. In this chapter we use randomisation testing (Kempthorne and Doerfler, 1969), a non-parametric significance test, 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*. These tests will allow us to understand whether observed differences in performance are due to chance or point to real differences in system performance. Due to the lack of assumptions made by the randomisation test, it can be applied to data whose underlying distribution would not satisfy the conditions for a parametric test. Furthermore even when the conditions for parametric tests are justified, the randomisation test has been shown to be of similar power to these tests.

In this section we outlined our proposed evaluation methodology. We proposed a novel method by which a group of users can be simulated searching together. We also proposed techniques for measuring the performance of a group of searchers at any point in the search. In the next section we will present the results from our evaluation.

3.3 Results

In this section we present the results from our evaluations of each of the collaborative relevance feedback techniques (Type A, B and C), described in section 3.1. We also evaluate the performance of a standard single-user relevance feedback mechanism, which assumes that all relevance judgments made in the search were made by one *pseudo-user*. Through these experiments we will investigate if passing relevance information across searchers in the feedback process can improve an SCIR search over a baseline SCIR system that implements just a division of labour policy. For both the partial-user (A) and combined weighting technique (B), we will run two variations of the technique, one which allows a user who has not encountered a term in their relevance judgments to contribute to its relevance and offer weight (*Contr*), and another which only considers the weighting for a term from the user that has encountered (*No Contr*).

As outlined in the previous section, an important consideration for any SCIR system is when to provide users with the relevance information from their search partner. In these experiments, we evaluated both a *dynamic* intensive environment, where users' ranked lists are updated when either they or their search partner make relevance judgments, alongside a more *static* environment, where users ranked lists are only updated after they make judgments themselves.

Figure 6 plots the performance of all combination techniques for both the static SCIR environment and the dynamic SCIR environment, along with the baseline system of an SCIR system implementing just a division of labour policy (SCIR + Full Div). 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 3.3 shows the single performance figure across all topics for all runs. When examining Figure 6, it should be noted that as these values are computed across a number of simulated runs with differing numbers of relevance judgments, values at later iterations (i.e. > 11 on x-axis) may not be representative of an overall trend than values for earlier iterations. For example, the sudden drop-off at around 35 relevance judgments is due to this value being calculated based on only one or two simulated runs.

As we can see from Table 3.3, all collaborative relevance feedback techniques, except the document fusion technique (C), provide small improvements in performance over the

SCIR + Full Div system. With the best performing system, the dynamic partial user (A) no contr technique, providing a 1.5% improvement. 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) (B) 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) (A) and combined weighting (B) 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 and Table 3.3, it does appear that the document fusion technique for both the static and dynamic environments does not perform as well as the term-based techniques. Significance tests reveal that the dynamic collaborative RF techniques of pseudo user, partial-user (A), and combined weighting (B) all significantly outperform the dynamic document fusion technique (C). 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 (A) and combined weighting (B), we find no significant difference.

Examining the bottom graph in Figure 6, we can see that the combination of relevance information techniques do provide a more substantive increase in performance over the SCIR system with just a division of labour 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. With the best performing system, the dynamic combined weighting no contr technique (B) providing a 4.8% improvement over the SCIR + Full Div system for these iterations.

Next we compare the performance of static versus dynamic feedback environments. From Figure 6 and Table 3.3, 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 (A) 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.

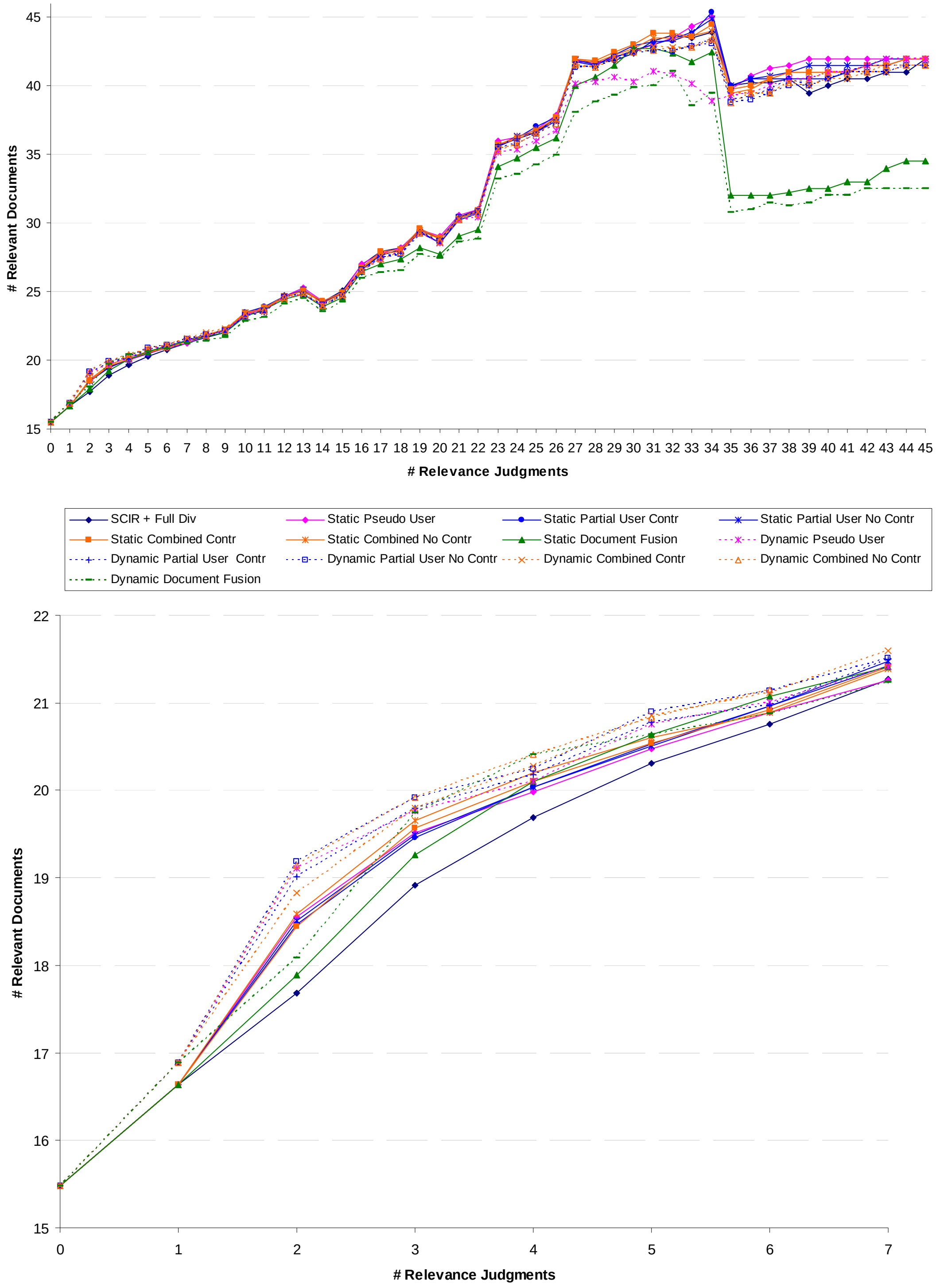


Figure 6: Comparison of collaborative relevance feedback techniques and baseline division of labour system across all topics

Topic #	SCIR + Full Div	Static Pseudo User	Static Partial User Contr	Static Partial User No Contr	Static Combined Contr	Static Combined No Contr	Static Document Fusion	Dynamic Pseudo User	Dynamic Partial User Contr	Dynamic Partial User No Contr	Dynamic Combined Contr	Dynamic Combined No Contr	Dynamic Document 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
307	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	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	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	5.93	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	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	31.88	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	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	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	2.96	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	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	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	8.99	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	31.10	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	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	9.63	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	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	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	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	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	**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 3.3: Single performance figure comparison of collaborative relevance feedback techniques and baseline division of labour system across all topics

3.4 Discussion

We 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 incorporating each user's relevance information into a collaborative relevance feedback mechanism, the ranked lists returned to the searchers could be improved.

Firstly, 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. However 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 the 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 (pseudo-user) method.

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.

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. However when we examine the performance of the group over the first few iterations of feedback, we do find that all the collaborative relevance feedback techniques in both the static and dynamic environments significantly improve the performance over the SCIR + Full Div system. This result is interesting, and suggests that although users may benefit from gaining relevance judgments from their search partner early in the search, that after a number of iterations this benefit is reduced.

We believe that the collaborative relevance feedback process of aggregating relevance information is causing users' ranked lists to become too similar. Although, by implementing the division of labour policy we are ensuring unique documents across the top 30 ranked positions for both users, we feel that the aggregation may be causing a loss of uniqueness across users' lists. In order to investigate this hypothesis, in Figure 7 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 static collaborative relevance feedback techniques.

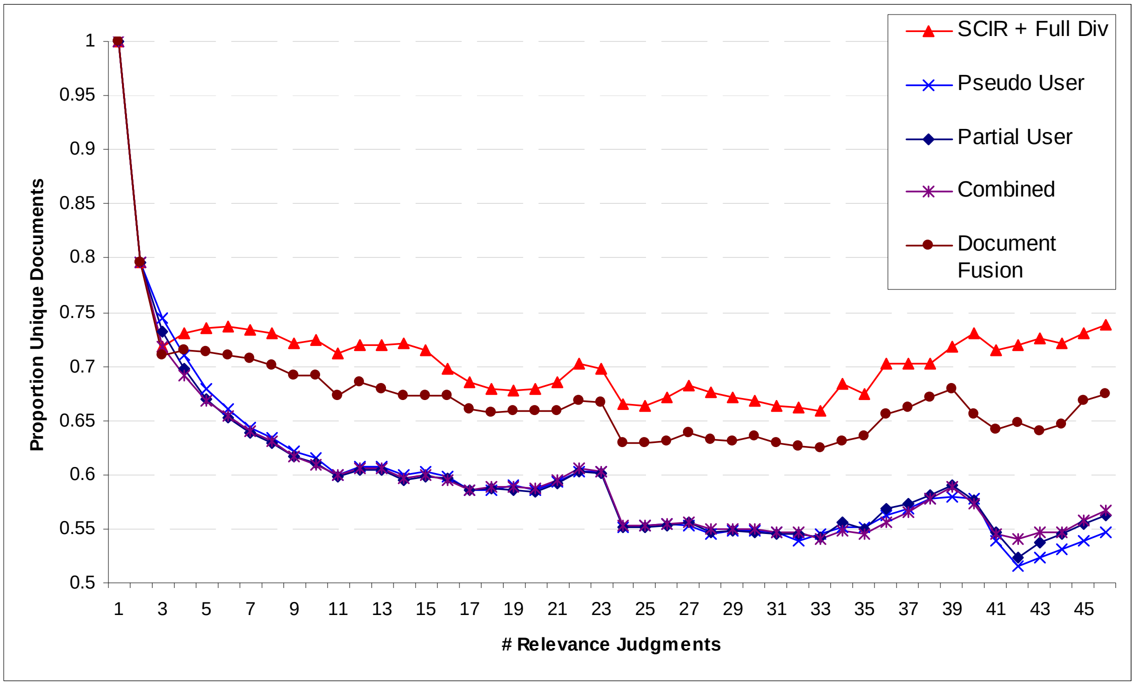


Figure 7: Comparisons of the total amount of unique documents across users' ranked lists for SCIR system with collaborative RF and without, across all topics

As we can see, there is a clear difference in the total number of unique relevant documents across users between the collaborative relevance feedback systems and the SCIR + Full Div system. This difference is significant for all techniques and across all topics. This result confirms our hypothesis, that the collaborative relevance feedback process is causing users' ranked lists to become too similar. This finding is intuitive- one of the great advantages of having multiple users tackle a search task is that the task can be divided. By ensuring a complete division of labour, in the SCIR + Full Division system, we are allowing users to make unique relevance judgments, 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. The fact that the document fusion technique provides substantially more unique documents than all 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 combination does not suffer from this problem, however as our results in this section have shown this does not lead to this technique outperforming the others in terms of discovering relevant unique documents. This does not, of course, mean that the introduction of unique documents degrades performance, but that the collaborative relevance feedback mechanism needs to strive to allow for the introduction of more unique relevant documents.

4 Future Trends

Our results have shown that the proposed, term-based, collaborative relevance feedback techniques perform similarly to the standard single-user relevance feedback formula. This result is not surprising as all techniques are attempting to aggregate each user's relevance information. The advantages of the collaborative techniques over the standard single user technique are that they can allow for a user-biased combination (by changing the α value associated with each user), but in these experiments this α has set to 0.5 to consider all users equally. In a real-world system, an SCIR system could exploit this α value in order to allow users to bias the relevance feedback process in favour of their own relevance judgments or in favour of their search partner. Alternatively an SCIR system could use this α value to enable an SCIR system to bias the collaborative relevance feedback process in favour of expert searchers through an *authority weighting* mechanism.

Our results have shown how a collaborative relevance feedback mechanism causes a loss of uniqueness across collaborating users' ranked lists. An alternative way of using multiple users' relevance judgments in an SCIR search is to implement a *Complementary Relevance Feedback* technique. Whereas the collaborative relevance feedback techniques discussed in this chapter attempt to make user's queries more similar, a complementary technique would make them more distinct.

5 Conclusions

Synchronous collaborative information retrieval refers to an explicit and active form of collaboration whereby users are collaborating in realtime to satisfy a shared information need. The benefits that such a system can provide over users searching independently are that it can enable both a *division of labour* and a *sharing of knowledge* across collaborating searchers. Although there has been some work to date into system-mediated division of labour, there has been no work which has investigated how a system-mediated sharing of knowledge can be realised in an adhoc search which can be either remote or co-located, despite the fact that SCIR systems in the literature have allowed users to make explicit relevance judgments in the form of bookmarks.

In this chapter, we have outlined how to make use of these bookmarks in order to benefit the group. We have proposed several techniques by which the standard relevance feedback formula can be extended to allow for relevance information from multiple users to be combined in the process. We have also outlined a novel evaluation methodology by which a group of users, who had previously searched for a search topic independently as part of a TREC interactive track submission, were simulated searching together.

Our results have shown that over an entire SCIR search, the passing of relevance information between users in an SCIR search does not improve the performance of the group. However, over the first few iterations of feedback only, the combination of relevance information does provide a significant improvement. This is an interesting result and encourages us and others to pursue further work in this area.

Acknowledgments

This work is supported by Science Foundation Ireland under grant 07/CE/I1147.

References

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., & Qvarfordt, P. (2007). FXPAL Interactive Search Experiments for TRECVID 2007. In *TRECVid2007 - Text REtrieval Conference TRECVID Workshop*, Gaithersburg, MD, USA.

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

Blackwell, A. F., Stringer, M., Toye, E. F., & 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 Press.

Cabri, G., Leonardi, L., & 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.

Craswell, N., Zoeter, O., Taylor, M., & Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of WSDM*, pages 87 – 94, Palo Alto, CA, ACM Press.

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.

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.

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

Dietz, P. & 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, USA. ACM Press.

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

Foley, C., Gurrin, C., Jones, G., Lee, H., Mc Givney, S., O’Connor, N., Sav, S., Smeaton, A. F., & 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, MD USA.

Foley, C., Smeaton, A. F., & 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., Adelaide, Australia, IEEE Computer Society.

Gianoutsos, S. & 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*, pages 14-21 University of Queensland, Brisbane, Australia.

Greenberg, S. & 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.

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 Press.

Han, R., Perret, V., & 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, USA. ACM Press.

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

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, USA.

Lieberman, H., Van Dyke, N. W., & 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, USA. ACM Press.

Maekawa, T., Hara, T., & 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.

Morris, M. R. & 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., & 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.

Netscape Conferencer (2001). http://www.aibn.com/help/Software/Netscape/communicator/conference/.

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

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

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

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

Shen, C., Vernier, F. D., Forlines, C., & 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 Press.

Sidler, G., Scott, A., & Wolf, H. (1997). Collaborative Browsing in the World Wide Web. In *Proceedings of the 8th Joint European Networking Conference*, Edinburgh, Scotland. Springer.

Smeaton, A. F., Lee, H., Foley, C., & 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., & 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, USA.

Spärck Jones, K. (1997). Search term relevance weighting given little relevance information. *Readings in information retrieval* pages 329–338, Morgan Kaufmann Publishers Inc.

Stewart, J., Bederson, B. B., & 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, USA. ACM Press.

Windows Live Messenger (2007). http://get.live.com/messenger/overview.

Yao, K.-T., Neches, R., Ko, I.-Y., Eleish, R., & Abhinkar, S. (1999). Synchronous and Asynchronous Collaborative Information Space Analysis Tools. In *ICPP ’99: Proceedings of the 1999 International Workshops on Parallel Processing*, page 74, Washington, DC, USA. IEEE Computer Society.

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*.