Chapter 18
Communities, Collaboration, and Recommender Systems in Personalized Web Search

Barry Smyth, Maurice Coyle and Peter Briggs

Abstract Web search engines are the primary means by which millions of users access information everyday and the sheer scale and success of the leading search engines is a testimony to the scientific and engineering progress that has been made over the last ten years. However, mainstream search engines continue to deliver largely *one-size-fits-all* services to their user-base, ultimately limiting the relevance of their result-lists. In this chapter we will explore recent research that is seeking to make Web search a more personal and collaborative experience as we look towards a new breed of more social search engines.

18.1 Introduction

Web search engines are among the most important and wide-spread information tools in use today. Every month the leading search engines recommend search results to billions of users and, in the process, generate billions of dollars in advertising revenue annually. In all of this Google stands tall as the clear market leader and one would be forgiven for assuming that all of the major web search challenges have by now been addressed, and that all that remains is the need for some minor algorithmic refinements. The reality is very different however, and while Google may have won the current round of search battles, the web search war is far from over.

Recent research has highlighted how even the leading search engines suffer from low success rates when it comes to delivering relevant results to the average searcher. For example, in one study [24] of more than 20,000 search queries researchers found that, on average, Google delivered at least one result worth selecting only 48% of the time; in other words, in 52% of cases, searchers chose to select none of the results returned. In large part this problem is as much due to the searcher as it is the search

Barry Smyth, Maurice Coyle, Peter Briggs
CLARITY: Centre for Sensor Web Technologies, School of Computer Science & Informatics, University College Dublin, Ireland, {firstname.lastname}@ucd.ie

F. Ricci et al. (eds.), *Recommender Systems Handbook*,
DOI 10.1007/978-0-387-85820-3_18,

engine: our search queries tend to be vague and under-specified, and rarely provide a clear indication of our search needs [100, 98, 99, 45, 90]. As frequent searchers we have adapted to these success rates, generally responding to poor result-lists with follow-up or alternative queries. However, at best, this means that web search is far less efficient than it should be — indeed recent studies suggest that among information workers 10% of salary costs are lost due to wasted search time [30] — and at worst a significant proportion of searchers may fail to find the information they need.

Thus, while Google, Yahoo and others continue to provide strong search services for millions of users, there remains plenty of headroom for improvement. In this chapter we will look into the future of web search by reviewing some of most promising research ideas that have the potential to bring game-changing innovation to this exciting technology sector. We will argue that the past is apt to repeat itself, and just as Google's game-changing take on web search led to its relentless rise over the past 10 years, so too will new search technologies emerge to have a similarly disruptive effect on the market over the next 10 years.

Even in their current form, modern search engines can be loosely viewed as a type of recommender system: they respond to users' queries with a set of result page recommendations. But recommendation technologies are set to play an increasingly important role in web search, by helping to address core web search challenges as well as contributing to the solution of a number of secondary search features. For example, recently modern search engines have added *query recommendation* services to supplement core search functionality. As the user enters their query, services like Google Suggest use recommendation techniques to identify, rank and recommend previously successful and relevant queries to the user; see [81]. In this paper, we will focus on two promising and powerful new ideas in web search — personalization and collaboration — that can trace their origins to recent recommender systems research [6, 53, 83, 35, 89, 77] and Chapters 5, 4 and 13. They question the very core assumptions of mainstream web search engines and suggest important adaptations to conventional web search engines. The first assumption concerns the *one-size-fits-all* nature of mainstream web search — two different users with the same query will, more or less, receive the very same result-list, despite their different preferences — and argues that web search needs to become more personalized so that the implicit needs and preferences of searchers can be accommodated. We will describe a number of different approaches to personalizing web search by harnessing different types of user preference and context information to influence the search experience; see for example [19, 23, 33, 97, 2, 48, 49, 108, 22, 69, 86, 14, 31]. The second assumption that will be questioned concerns the *solitary nature* of web search. By and large web search takes the form of a isolated interaction between lone searcher and search engine, however, recent research has suggested that there are many circumstances where the search for information has a distinctly collaborative flavour, with groups of searchers (e.g., friends, colleagues, classmates) cooperating in various ways as they search for and share results. We will describe recent work in the area of *collaborative information retrieval*, which attempts to capitalize on potential for collaboration during a variety of information seeking tasks; see for example, [70, 71, 73, 72, 58, 59, 94, 1].

In addition we will highlight a new breed of search service that combines elements of personalization and collaboration: so-called *social search* services take advantage of the recent evolution of the web as a social medium, one that promotes interaction and collaboration among individuals during search, so that searchers can benefit from the preferences and experiences of other like-minded individuals. Indeed this provides a new source of information for search engines to use during retrieval: interaction and collaboration information. And this information can be used to drive recommendations at search time so that organic search results, based on term-overlap and link connectivity information, are complimented by additional result recommendations that are based on the preferences and activities of searchers. This will represent a coming together of recommendation systems and search systems and, just as the introduction of connectivity information led to its rise to dominance, there is considerable optimism that this new source of interaction and preference information will lead to an entirely new phase of search engine development in the quest to deliver the right information to the right user at the right time.

18.2 A Brief History of Web Search

Before considering some of the emergent search technologies that have the potential to disrupt the search industry, it is first worth briefly reviewing the history of web search over the past 15 years, to better understand the evolution of modern web search. The early web was not a place of search. Instead if you wanted to get to a particular web page then you either typed the URL directly into your browser, or you used a portal like Yahoo as a starting point to navigate to this page. As the web grew (and grew, and grew) it became clear that portal browsing would not scale, and web search began to emerge in the guise of early search engines such as Lycos, Excite, and Altavista.

These search engines all relied on so-called information retrieval (IR) technologies that had been around since the 1970's [104, 4]. A simplified schematic of a typical search engine architecture is preseneted in Fig. 18.1. Briefly, early search engines constructed their own index of the web, by *crawling* the web's network of pages and analysing the content of each page in turn, recording the words, and their frequencies, contained in each page. To respond to a search query, the search engine retrieves and ranks pages that contain query terms. During the early days of web search, the emphasis was very much on the size of the index, and search engines that had indexed more of the web had a clear coverage advantage over their rivals. Attention was also paid to the ranking of search results; for the most part, these search engines relied on the frequency of query terms in a web page (relative to the index as a whole) as the primary arbiter of relevance [96], preferring pages that contained frequent occurrences of distinctive query terms. While this approach worked reasonably well in the well-structured, closed-world of information retrieval systems, where information retrieval experts could be relied upon to submit detailed, well-formed queries, it did not translate well to the scale and heterogenous nature of web content or our vague search queries. The outcome was a poor search experience for most searchers, with relevant results hidden deep within result-lists dominated by results that were, at best, only superficially relevant to the query.

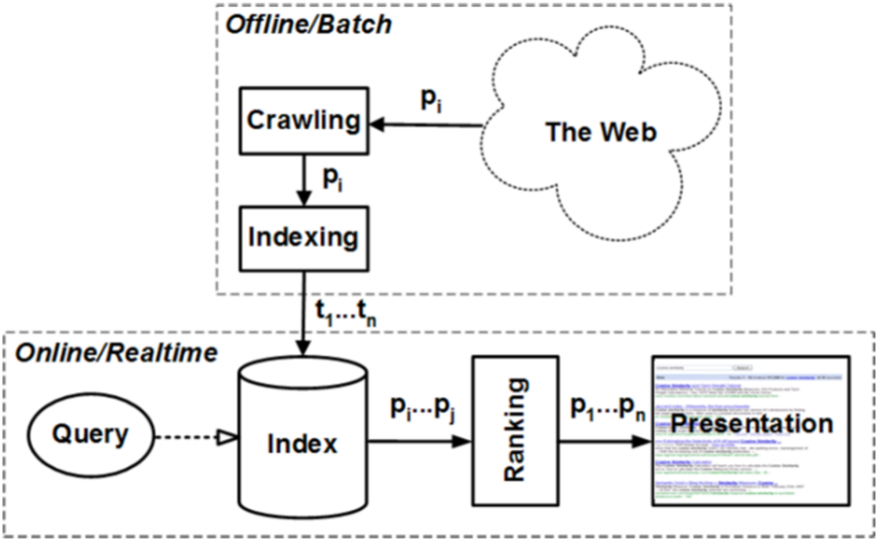

Fig. 18.1: Functional components of a typical web search engine. A page, p_i , is located on the web by the crawler and its content, the terms $t_1,...,t_n$, are retrieved and indexed as part of an offline process. In response to a search query, the engine probes the index to retrieve results which match the query terms, $p_i,..., p_j$, which are then ranked by their relevance according to the search engines particular ranking metrics, before being presented to the searcher as a result-list.

Improving the ranking of search results became the challenge for these early search engines and even the race for the largest search index took a back seat in the face of this more pressing need. It soon became clear, however, that relying solely on the terms in a page was not going to be sufficient, no matter how much time was invested in tweaking these early ranking algorithms. Simply put, there were lots of pages that scored equally well when it came to counting matching query and page terms, but few of these pages turned out to be truly relevant and authoritative. Although term matching information had a role to play in overall relevance, on its own it was insufficient, and it was clear that there was vital information missing from the ranking process.

The missing ingredient came about as a result of research undertaken by a number of groups during the mid 1990's. This included the work of John Kleinberg [40] and, most famously, the work of Google founders Larry Page and Sergey Brin [13].

These researchers were among the first to take advantage of the connectedness of web pages, and they used this information to evaluate the relative importance of individual pages. Kleinberg, Page, and Brin recognised the web as a type of *citation network* (see for example, [60]). Instead of one paper citing another through a bibliographic reference, on the web one page cited another page through a hyperlink connecting the two. Moreover, it seemed intuitive that the importance of a given page should be a function of the various pages that linked to it; the so-called *back-links* of the page. Thus a page could be considered important if lots of other important pages linked to it. This provided the starting point for a fundamentally new way to measure the importance of a page and, separately, the work of [40, 17] and [13] led to novel algorithms for identifying authoritative and relevant pages for even vague web search queries. By the late 1990's Page and Brin's so-called *PageRank* algorithm was implemented in the first version of Google, which combined traditional term-matching techniques with this new approach to link analysis, to provide search results that were objectively superior to the results of other search engines of the day. The rest, as they say, is history.

18.3 The Future of Web Search

There is no doubt that web search represents a very significant recommendation challenge. The size and growth characteristics of the web, and the sheer diversity of content types on offer represent formidable information retrieval challenges in their own right. At the same time, as the demographics of the web's user-base continues to expand, search engines must be able to accommodate a diverse range of user types and search skill levels. In particular, most of us fail to live up to the expectations of the document-centric, term-based information retrieval engines that lie at the heart of modern search technology. These engines, and the techniques they rely upon, largely assume well-formed, detailed search queries, but such queries are far from common in web search today [36, 37, 100, 45]. Instead most web search queries are vague or ambiguous, with respect to the searcher's true information needs, and many queries can contain terms that are not even reflected in the target document(s).

Given that many queries fail to deliver the results that the searcher is looking for there is considerable room for improvement in this most fundamental feature of the search experience. While the problem may reside, at least in part, with the nature of web search queries, as discussed above, it is unlikely that users will improve their query-skills any time soon. In response, researchers have begun to explore two complementary strands of research as a way to improve the overall searcher experience. One widely held view is that web search needs to become more personalized: additional information about users, their preferences and their current context, for example, should be used to deliver a more personalized form of web search by selecting and ranking search results that better match the preferences and context of the individual searcher (see for e.g. [86, 14, 31, 22, 2, 48]). Another view is that there is an opportunity for web search to become more collaborative, by allowing communities of users to co-operate (implicitly or overtly) as they search (see for e.g. [70, 71, 73, 72, 58, 59, 94, 1]).

In the following sections we will review this research landscape, describing a number of initiatives that are attempting to transform static (non-personalized), solitary (non-collaborative), mainstream search engines into more personalized (see Section 18.3.1) or more collaborative (see Section 18.3.2) search services. These initiatives borrow ideas from recommender systems, user profiling, and computer-supported collaborative working research; see for example [84, 41, 89, 35, 52]. We will also highlight recent research that seeks to bring both of these approaches together leading to a new generation of search services that are both collaborative and personalized. We will refer to these hybrid services as *social search* services and in the remainder of this chapter we will describe two detailed case-studies of two different approaches to social search.

18.3.1 Personalized Web Search

Many recommender systems are designed to make suggestions to users that are relevant to their particular circumstances or their personal preferences — for example, recommender systems help users to identify personally relevant information such as news articles [8, 9, 41], books [46], movies [54, 27, 42], and even products to buy [83, 74, 76, 51, 75, 20]— and the application of recommender technologies to web search allows for a departure from the conventional one-size-fits-all approach to mainstream web search. When it comes to delivering a more personalized search experience there are two key requirements: firstly, we must understand the needs of searchers (*profiling*); secondly, we must be able to use these profiles to influence the output of the search engine, for example by re-ranking results according to the profile, or, indeed, by influencing other components of the web search experience.

To put these research efforts into perspective it is useful to consider two important dimensions to personalizing web search. On the one hand we can consider the nature of the profiles that are learned: some approaches focus on *short-term* user profiles that capture features of the user's current search context (e.g. [86, 14, 31]), while others accommodate *long-term* profiles that capture the user's preferences over an extended period of time (e.g. [22, 2, 48]). On the other hand, when it comes to harnessing these profiles during search, we can usefully distinguish between those approaches that are guided by an *individual* target user's profile (e.g. [15, 89, 38, 43]) versus those that are *collaborative*, in the sense that they are guided by the profiles of a group of users (e.g. [46, 85, 41, 35, 90]).

Generally speaking, user profiles can be constructed in two ways. Explicit profiling interrogates users directly by requesting different forms of preference information, from categorical preferences [22, 48] to simple result ratings [2]. In contrast, *implicit profiling* techniques attempt to infer preference information by monitoring user behaviour, and without interfering with users as they go about their searches; e.g. [22, 47, 69].

With explicit profiling, the users themselves do the profiling work by either specifying search preferences up front, or by providing personal relevance feedback such as rating returned search results. Chirita et al [22] use individual user profiles which are defined by the searcher through ODP[1] web directory categories to re-rank results according to the distance between the profile and ODP categories for each result. They investigate a number of different distance metrics, and report the findings of a live user evaluation that shows that their personalized approach is capable of more relevant result rankings than standard Google search. One of the drawbacks of relying on ODP categories in this way however is that only a small proportion of the web is categorised in the ODP and so many of the returned search results have no category information to base the re-ranking on. Ma et al [48] propose a similar approach whereby user profiles are explicitly expressed through ODP categories, except they re-rank search results based on the cosine similarity between result page content and the ODP directory category profiles. In this way the search results themselves are not required to be categorised in the ODP.

In contrast, *ifWeb* [2] builds user profiles using a less structured approach through keywords, free-text descriptions, and web page examples provided by the user to express their specific information needs, which are stored as a weighted semantic network of concepts. *ifWeb* also takes advantage of explicit relevance feedback where the searcher provides result ratings that are used to refine and update their profile. A similar approach is used by the *Wifs* system [55] in which profiles initially built using terms selected from a list can be subsequently improved with feedback on viewed documents provided by the users. The major drawback with these types of explicit approaches to profiling is that the majority of users are reluctant to make the extra effort in providing feedback [16]. Furthermore, searchers may find it difficult to categorise their information needs and preferences accurately in the first place.

A potentially more successful approach to profiling is to infer user preferences implicitly (*implicit profiling*). As in the work of [22], Liu et al [47] also use hierarchical categories from the ODP to represent a searcher's profile, except in this work the categories are chosen automatically based on past search behaviour such as previously submitted queries and the content of selected result documents. A number of different learning algorithms are analysed for mapping this search behaviour onto the ODP categories, including those based on Linear Least Squares Fit (LLSF) [107], the Rocchio relevance feedback algorithm [78], and k-Nearest Neighbor (kNN) [28]. In a related approach, [103] use statistical language methods to mine contextual information from this type of long-term search history to build a language model based profile, and [69] also infer user preferences based on past behaviour, this time using the browser cache of visited pages to infer subject areas that the user is interested in. These subject areas, or categories, are combined into a hierarchical user profile where each category is also weighted according to the length of time the user spent viewing the pages corresponding to the category.

The above are all examples of long-term user profiles that seek to capture information about the user's preferences over an extended period of time, certainly

[1] The Open Directory Project, http://dmoz.org

beyond the bounds of a single search session. The alternative is to capture short-term profiling information, typically related to the particular context of the current information finding task. For example, the UCAIR system [86] concentrates on recently submitted queries and selected results to build a short-term profile that is used to personalize results for the current search task. When a new search session is initiated, a new profile for the user and their current information requirements is created. Similarly Watson [14] and IntelliZap [31] both generate short-term profiles from current context information. Watson identifies informative terms in local documents that the user is editing and web pages that are being browsed, and uses these to modify the user's search queries to personalize results. IntelliZap users initiate a search by selecting a textual query from within a document they are currently viewing, and the search is then guided by additional terms occurring in close proximity to the query terms in the document. In these examples, the profiles guiding the personalization of search results capture context which is pertinent to the users immediate, and possibly temporary, information needs.

The availability of profile and/or context information is the pre-requisite for personalization and there have been a wide range of techniques developed for utilizing profile information to influence different aspects of search experience. These techniques are not limited to influencing the retrieval and ranking of search results, for example, and in fact there has been research on how profiles can be used to influence many other stages in the web search pipeline including the spidering and indexing [32, 44, 34, 29] of raw page content, and query generation [3, 7, 56]. For example, one common way to personalize search results based on a user profile involves using the profile to re-write, elaborate, or expand the original search query so that it returns more specific results that better reflect search interests or context. For example, Koutrika and Ioannidis [43] propose an algorithm they call *QDP* (Query Disambiguation and Personalization) to expand a query submitted by the user according to a user profile represented by weighted relationships between terms. These relationships take the form of operators between terms, such as conjunction, disjunction, negation and substitution, and so in effect the user's profile provides a set of personalized query rewriting rules, which can be applied to the submitted query before it is dispatched to the search engine. Croft et al [26] describe how individualized language models can be used as user profiles with a view to supporting query expansion and relevance feedback. There is also much research in the area of query expansion and disambiguation from the perspective of short term, session-based user profiles from a relevance feedback standpoint which is also highly relevant to work in personalized search [82]. This perspective is not so much targeted at personalizing search per se, but rather at improving search at the level of independent search sessions and many of these approaches can be expanded to encompass longer-term personalized search profiles.

However, perhaps the most popular way to personalize search through user profiles is to directly influence the *ranking* of search results. For example, Jeh and Widom [38] do this by introducing a personalized version of PageRank [13] for setting the query-independent priors on web pages based on user profiles. These profiles consist of a collection of *preferred* pages with high PageRank values which are explicitly chosen by the user, and used to compute a personalized PageRank score for any arbitrary page based on how related it is to these highly scored preferred pages. Chirita et al [23] build on this idea by automatically choosing these profile pages by analysing the searcher's bookmarked pages and past surfing behaviour, along with a *HubFinder* algorithm that finds related pages with high PageRank scores which are suitable for driving the personalized PageRank algorithm. Both of these approaches are based on long-term user profiles drawn from an extended period of the user's browsing history.

Chang et al [19] propose a personalized version of Kleinberg's HITS [39] ranking algorithm. Their technique harnesses short-term feedback from the searcher, either explicitly or implicitly, to build a profile consisting of a personalized authority list which can then be used to influence the HITS algorithm to personalize the ranking of search results. Experimental results using a corpus of computer science research papers shows that personalized HITS is able to significantly improve result ranking in line with the searcher's preferences, even with only minimal searcher feedback.

Another popular ranking-based approach is the re-ranking of results returned from some underlying, generic web search engine according to searcher preferences without requiring access to the inner workings of the search engine. Speretta and Gauch [97] create individual user profiles by recording the queries and selected result snippets from results returned by Google which are classified into weighted concepts from a reference concept hierarchy. The results from future Google searches are then re-ranked according to the similarity between each result and the searcher's profile concept hierarchy. Rohini and Varma [79] also present a personalized search method where results from an underlying web search engine are re-ranked according to a collaborative filtering technique that harnesses implicitly generated user profiles.

All of the above techniques focus on harnessing single user profiles (the preferences of the target searcher) to personalize that user's search experience. In recommender systems research it is common to take advantage of groups of related profiles when it comes to generating recommendations for a target individual. For instance, the well known *collaborative filtering* approach to recommendation explicitly uses the preferences of a group of users who are similar to the target user when it comes to generating recommendations [77, 85, 46]; see also [35, 52] and Chapter 21. Similar ideas are beginning to influence web search and, indeed, in Section 18.4 we will describe one particular approach that harnesses the preferences of communities of users, albeit in the form of single community profiles rather than a collection of individual user profiles; see also [92, 90]. Sugiyama et al. [101] propose a method whereby long-term user profiles are constructed from similar searchers according to browsing history using a modified collaborative filtering algorithm. The idea is that searchers who issued similar queries and selected similar results in the past can benefit from sharing their search preferences. Sun et al. [102] propose a similar approach called CubeSVD which is also based on collaborative filtering to personalize web search results by analysing the correlation of users, queries and results in click-through data. Both these methods involve the identification of similar searchers to the current searcher in order to create a more comprehensive user profile for the individual. More recently, the work of [12] describes a peer-to-peer approach to personalizing web search that also leverages the profiles of similar users during result recommendation. Each searcher is profiled in terms of their prior queries and result selections (once again these are long-term profiles). In response to a new target query, recommendations are derived from the user's own personal profile, but in addition, the query is propagated through the peer-to-peer search network so that connected users can also suggest relevant results based on their prior search behaviours. The resulting recommendations are aggregated and ranked according to their relevance to the target query and also in terms of the strength of the *trust* relationship between the target user and the relevant peer; see also recent trust-based recommendation techniques by [63, 65, 64, 66, 67, 62] and Chapter 20.

18.3.2 Collaborative Information Retrieval

Recent studies in specialised information seeking tasks, such as military command and control tasks or medical tasks, have found clear evidence that search-type tasks can be collaborative as information is shared between team members [70, 71, 73, 72]. Moreover, recent work by [57] highlights the inherently collaborative nature of more general purpose web search. For example, during a survey of just over 200 respondents, clear evidence for collaborative search behaviour emerged. More than 90% of respondents indicated that they frequently engaged in collaboration at the level of the *search process*. For example, 87% of respondents exhibited "back-seat searching" behaviours, where they watched over the shoulder of the searcher to suggest alternative queries. A further 30% of respondents engaged in search coordination activities, by using instant messaging to coordinate searches. Furthermore, 96% of users exhibited collaboration at the level of *search products*, that is, the results of searches. For example, 86% of respondents shared the results they had found during searches with others by email. Thus, despite the absence of explicit collaboration features from mainstream search engines there is clear evidence that users implicitly engage in many different forms of collaboration as they search, although, as reported by [57], these collaboration "work-arounds" are often frustrating and inefficient. Naturally, this has motivated researchers to consider how different types of collaboration might be supported by future editions of search engines.

The resulting approaches to *collaborative information retrieval* can be usefully distinguished in terms of two important dimensions, *time* — that is, *synchronous* versus *asynchronous* search — and *place* — that is, *co-located* versus *remote* searchers. Co-located systems offer a collaborative search experience for multiple searchers at a single location, typically a single PC (e.g. [1, 87]) whereas remote approaches allow searchers to perform their searches at different locations across multiple devices; see e.g. [58, 59, 94]. The former enjoy the obvious benefit of an increased faculty for direct collaboration that is enabled by the face-to-face nature of co-located search, while the latter offer a greater opportunity for collaborative search. Alternatively, synchronous approaches are characterised by systems that broadcast a "call to search" in which specific participants are requested to engage in a well-defined search task for a well defined period of time; see e.g. [87]. In contrast, asynchronous approaches are characterised by less well-defined, ad-hoc search tasks and provide for a more open-ended approach to collaboration in which different searchers contribute to an evolving search session over an extended period of time; see e.g. [58, 92].

A good example of the co-located, synchronous approach to collaborative web search is given by the work of [1]. Their CoSearch system is designed to improve the search experience for co-located users where computing resources are limited; for example, a group of school children having access to a single PC. CoSearch is specifically designed to leverage peripheral devices that may be available (e.g. mobile phones, extra mice etc.) to facilitate distributed control and division of effort, while maintaining group awareness and communication. For example, in the scenario of a group of users collaborating though a single PC, but with access to multiple mice, CoSearch supports a *lead searcher* or *driver* (who has access to the keyboard) with other users playing the role of search *observers*. The former performs the basic search task but all users can then begin to explore the results returned by independently selecting links so that pages of interest are added to a page queue for further review. The CoSearch interface also provides various opportunities for users to associate notes with pages. Interesting pages can be saved and as users collaborate a *search summary* can be created from the URLs and notes of saved pages. In the case where observers have access to mobile phones, CoSearch supports a range of extended interface functionality to provide observers with a richer set of independent functionality via a bluetooth connection. In this way observers can download search content to their mobile phone, access the page queue, add pages to the page queue and share new pages with the group.

The purpose of CoSearch is to demonstrate the potential for productive collaborative web search in resource-limited environments. The focus is very much on dividing the search labour while maintaining communication between searchers, and live user studies speak to the success of CoSearch in this regard [1]. The work of [88] is related in spirit to CoSearch but focuses on image search tasks using a table-top computing environment, which is well suited to supporting collaboration between co-located users who are searching together. Once again, preliminary studies speak to the potential for such an approach to improve overall search productivity and collaboration, at least in specific types of information access tasks, such as image search, for example. A variation on these forms of synchronous search activities is presented in [87], where the use of mobile devices as the primary search device allows for a remote form of synchronous collaborative search. The iBingo system allows a group of users to collaborate on an image search task with each user using a ipod touch device as their primary search/feedback device (although conventional PCs appear to be just as applicable). Interestingly, where the focus of CoSearch is largely on the division of search labour and communication support, iBingo offers the potential to use relevance feedback from any individual searcher to the benefit of others. Specifically, the iBingo collaboration engine uses information about the activities of each user in order to encourage other users to explore different information trails and different facets of the information space. In this way, the ongoing activities of users can have an impact on future searches by the group and, in a sense, the search process is being "personalized" according to the group's search behaviour.

Remote search collaboration (whether asynchronous or synchronous) is the aim of SearchTogether, which allows groups of searchers to participate in extended shared search sessions as they search to locate information on particular topics; see also [58]. In brief, the SearchTogether system allows users to create shared search sessions and invite other users to join in these sessions. Each searcher can independently search for information on a particular topic, but the system provides features to allow individual searchers to share what they find with other session members by recommending and commenting on specific results. In turn, SearchTogether supports synchronous collaborative search by allowing searchers to invite others to join in specific search tasks, allowing cooperating searchers to synchronously view the results of each others' searches via a split-screen style results interface. As with CoSearch above, one of the key design goals in SearchTogether is to support a division of labour in complex, open-ended search tasks. In addition, a key feature of the work is the ability to create a shared awareness among group members by reducing the overhead of search collaboration at the interface level. SearchTogether does this by including various features, from integrated messaging, query histories, and recommendations arising out of recent searches.

In the main, the collaborative information retrieval systems we have so far examined have been largely focused on supporting collaboration from a division of labour and shared awareness standpoint, separate from the underlying search process. In short, these systems have assumed the availability of an underlying search engine and provided a collaboration interface that effectively *imports* search results directly, allowing users to share these results. As noted by [68], one of the major limitations of these approaches is that collaboration is restricted to the interface in the sense that while individual searchers are notified about the activities of collaborators, they must individually examine and interpret these activities in order to reconcile their own activities with their co-searchers. Consequently, the work of [68] describes an approach to collaborative search that is more tightly integrated with the underlying search engine resource so that the operation of the search engine is itself influenced by the activities of collaborating searchers in a number of ways. For example, mediation techniques are used to prioritise, as yet, unseen documents, while query recommendation techniques are used to suggest alternative avenues for further search exploration.

18.3.3 Towards Social Search

So far we have focused on two separate strands of complementary research in the field of web search and information finding motivated by questions that cut to the very core of conventional web search. The one-size-fits-all nature of mainstream web search is questioned by researchers developing more personalized web search techniques, and the assumption that search is largely a solitary experience is questioned by recent studies that highlight the inherently collaborative nature of many search scenarios.

To date, these different strands of research have been separated by different motivations and objectives. The world of personalized search, for example, has been largely guided by the need to produce result-lists that are better targeted to the needs of the individual searcher, whereas collaborative information retrieval has focused on supporting groups of searchers by facilitating the division of search labour and by promoting shared awareness among cooperating searchers. However both of these research communities are linked by a common thread of research from the recommender systems field and a recommender systems perspective has helped to identify opportunities to bring these two different strands of research together. In what follows we will describe two related case-studies that attempt to bring together these strands of research in the pursuit of web search techniques that are both collaborative and personalized. The result is an approach to web search that is both more collaborative — each case study assumes the involvement of groups of searchers — and more personalized, albeit at the level of the group rather than the individual searcher. Both of these case-studies will describe remote, asynchronous forms of collaborative web search and we will summarize the results of recent live-user studies to highlight their potential end-user benefits. In each case we will describe the central role that recommendations play in adding-value to a conventional search result-list. For example, we will describe how the preferences and activities of communities and groups of users can be harnessed to promote recommended search results in addition to conventional result-lists.

18.4 Case-Study 1 - Community-Based Web Search

In this first case-study we review recent work in the area of Community-based web search in which the search activities of communities of like-minded users are used to augment the results of a mainstream search engine to provide a more focused community-oriented result-list; see [91, 92]. This can include well-defined or ad-hoc communities, and our aim is to take advantage of the query repetition and selection regularity that naturally occurs within the search behaviour of such communities as a source of result recommendations. In this case-study we describe and evaluate one particular implementation of this approach to web search that has been designed to work with a mainstream search engine such as Google.

18.4.1 Repetition and Regularity in Search Communities

There are many scenarios in which search can be viewed as a community-oriented activity. For example, the employees of a company will act as a type of search community with overlapping information needs. Similarly, students in a class may serve as a search community as they search for information related to their class-work. Visitors to a themed website (e.g., a wildlife portal or a motoring portal) will tend to share certain niche interests and will often use the site's search facilities to look for related information. And of course, groups of friends on a social networking site may act as a community with shared interests.

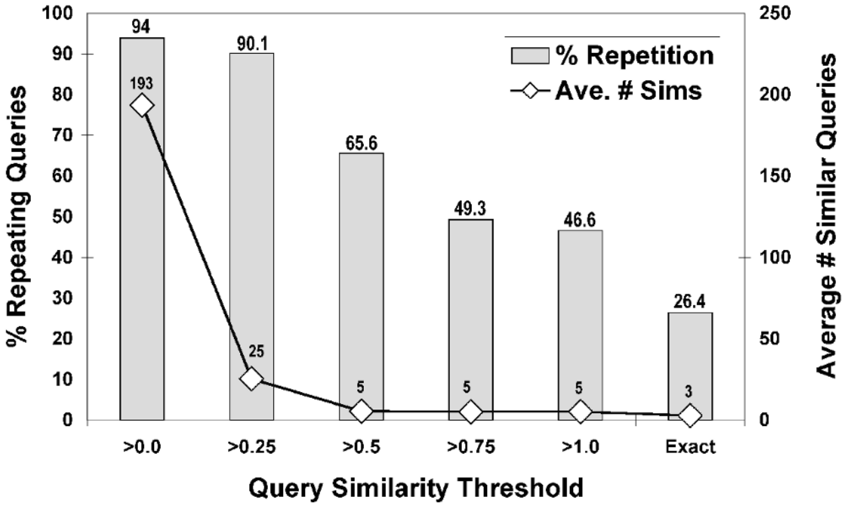

Fig. 18.2: Repetition and similarity amongst the search queries used by the employees of a software company.

We became interested in these emergent search communities because we believed that there was a high likelihood that similarities would exist between the search patterns of community members. For example, Figure 18.2 presents the results of a recent 17-week study of the search patterns for 70 employees of a local software company; this study preceded the trial discussed later in this paper. During the study we examined more than 20,000 individual search queries and almost 16,000 result selections. We see that, on average, just over 65% of queries submitted shared at least 50% (>0.5 similarity threshold) of their query terms with at least 5 other queries; and more than 90% of queries shared at least 25% of their terms with about 25 other queries. In other words, searchers within this ad hoc corporate search community do search for similar things in similar ways, much more so than in generic search scenarios where we typically find much lower repetition rates of about 10% at the 0.5 similarity threshold [92].

This is an important result which is supported by similar studies on other communities of searchers [92], and which motivates our collaborative web search approach. It tells us that, in the context of communities of like-minded searchers, the world of web search is a repetitive and regular place. A type of community search knowledge is generated from the search experiences of individuals as they search. This in turn suggests that it may be possible to harness this search knowledge by facilitating the sharing of search experiences among community members. So, as a simple example, when a visitor to the previously mentioned wildlife portal searches for "jaguar pictures" they can be recommended search results that have been previously selected by other community members for *similar* queries. These results will likely relate to the wildlife interests of the community and so, without any expensive processing of result content, we can personalize search results according to the learned preferences of the community. In this way, novice searchers can benefit from the shared knowledge of more experienced searchers.

18.4.2 The Collaborative Web Search System

Figure 18.3 presents the basic architecture for our collaborative web search system, which is designed to work alongside an underlying mainstream search engine — in this case, Google. Briefly, a proxy-based approach is adopted to intercept queries on their way to the underlying search engine, and to manipulate the results that are returned from this engine back to the searcher. In this way users get to use their favourite search engine in the normal way, but with collaborative web search (CWS) promotions incorporated into the result-lists directly via the proxy. For example, consider a user U_i submitting query q_T to Google. This request is redirected to the CWS system whereupon two things happen. First, the query is passed on to Google and the result-list R_S is returned in the normal way. Second, in parallel the query is also used to access a local store of the search activity for $U_i^{\prime}s$ community – the CWS *hit-matrix* – to generate a ranked set of promotion candidates, R_P, as outlined below. These promotion candidates are annotated by the *explanation engine* to present the searcher with a graphical representation of their community history. Result-lists R_P and R_S are merged and the resulting list R_{final} is returned to the user; typically this merge involves promoting the k (e.g., $k = 3$) most relevant promotions to the head of the result-list.

Thus for a target search query, CWS combines a default result-list, R_S, from a standard search engine, with a set of recommended (*promoted*) results, R_P, drawn from the community's past search history. To do this the search histories of a given community, C, of users ($C = \{U_1,...,U_n\}$) are stored in a *hit-matrix*, H^C, such that each row corresponds to some query q_i and each column to some selected result page p_j. The value stored in H_{ij}^C refers to the number of times that page p_j has been selected for query q_i by members of C. In this way, each hit-matrix acts as a repos-

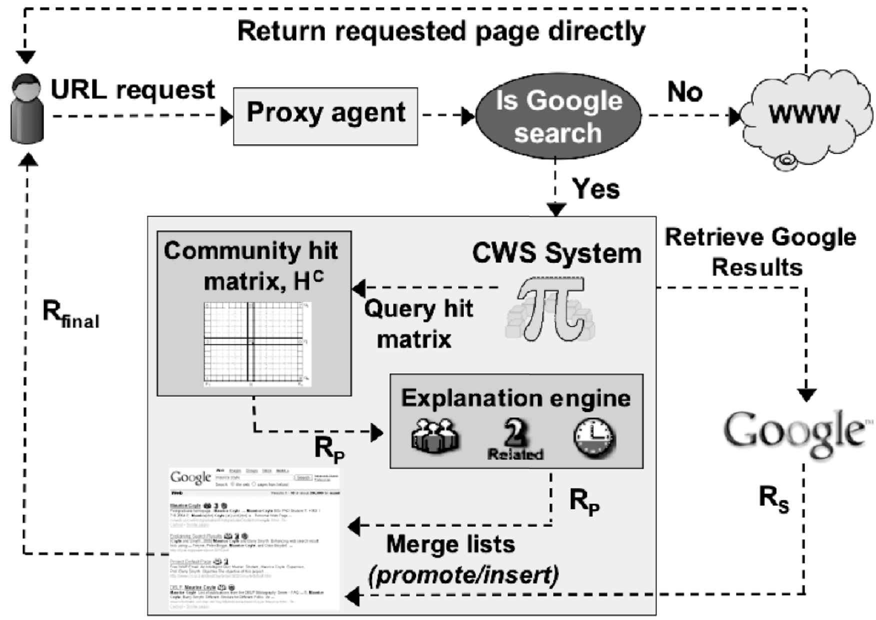

Fig. 18.3: Proxy architecture for a CWS system.

itory of community search experiences: the results that the community members have found to be relevant for their queries.

$$Relevance(p_j, q_i) = \frac{H_{ij}}{\sum_{\forall j} H_{ij}} \tag{18.1}$$

$$Sim(q, q') = \frac{|q \cap q'|}{|q \cup q'|} \tag{18.2}$$

$$WRel(p_j, q_T, q_1, ..., q_n) = \frac{\sum_{i=1...n} Relevance(p_j, q_i) \bullet Sim(q_T, q_i)}{\sum_{i=1...n} Exists(p_j, q_i) \bullet Sim(q_T, q_i)} \tag{18.3}$$

When responding to a new target query, q_T, H_C is used to identify and rank results that have been regularly selected in the past. The relevance of a result p_j in relation to a query q_i can be estimated by the relative frequency that p_j has been selected for q_i in the past, as shown in Equation 18.1. More generally, we can pool the results that have been selected for queries that are similar to q_T (see Equation 18.2) and rank each result according to the weighted model of relevance shown in Figure 18.3, which weights each individual result's relevance by the similarity of the associated query to q_T; note that the predicate *Exists* returns 1 if page p_j has been previously selected for query q_i in the target community, and 0 otherwise.

Fig. 18.4: The result-list returned by Google in response to the query ‘michael jordan’.

Figures 18.4 and 18.5 present example screen shots for the result-list returned by Google for the query *‘Michael Jordan’*. In the case of Figure 18.4 we see the default Google result-list, with results for the basketball star clearly dominating. In Figure 18.5, however, we see a result-list that has been modified by our proxy-based version of CWS, trained by (in this example) a community of computer science researchers. The results are presented through the standard Google interface, but we see that the top 3 results are promotions for the well-known Berkeley professor. In addition, promoted results are annotated with explanation icons designed to capture different aspects of the result’s community history. These include icons that capture the popularity of the result among community members, information about how recently it has been selected, and information about the other queries that have led to its selection.

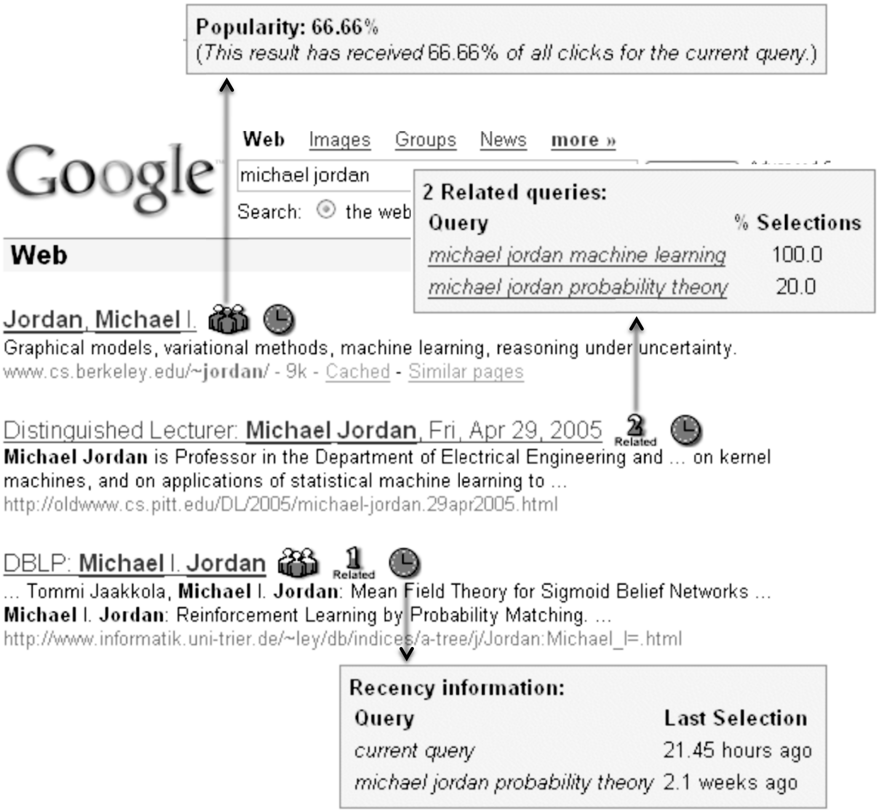

Fig. 18.5: The result-list returned by CWS in response to the query 'michael jordan' issued within a community with a shared interest in computer science. The extra explanation information available by mousing-over each promoted result icon type is also shown.

18.4.3 Evaluation

The current proxy-based architecture has been used as the basis of a long-term trial of the CWS approach in a corporate search scenario. In this section we will describe some recent results drawn from this trial, which speak to the value of the community-based promotions offered by CWS.

The trial participants included the 70+ employees of a local Dublin software company where the CWS architecture was configured to work with the standard Google search engine so that all Google requests were redirected through the CWS system. The search experience was based on the standard Google interface with a maximum of 3 results promoted (and annotated with explanations) in any session;

if more than 3 promotions were available then non-promoted results were annotated with explanation icons, but left in their default Google position. The results presented here are drawn from just over 10 weeks of usage and cover a total of 12,621 individual search sessions.

One of the challenges in evaluating new search technologies in a natural setting is how to evaluate the quality of individual search sessions. Ideally we would like to capture direct relevance feedback from users as they search. While it would be relatively straightforward to ask users to provide such feedback during each session, or as they selected specific results, this was not feasible in the current trial because participants were eager to ensure that their search experience did not deviate from the norm, and were unwilling to accept pop-ups, form-filling or any other type of additional feedback. As an alternative, in this evaluation, we used a less direct measure of relevance based on the concept of a *successful session* (see also [92, 91]). We define a successful session to be one where at least one search result has been selected, indicating that the searcher has found at least one (partially) relevant result. In contrast, search sessions where the user does not select any results are considered to be unsuccessful, in the sense that the searcher has found no relevant results. While this is a relatively crude measure of overall search performance, it at least allows us to compare search sessions in a systematic way.

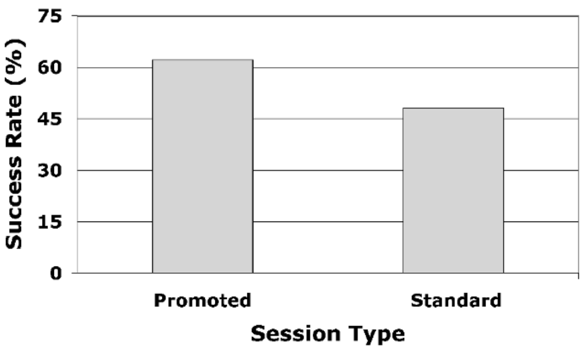

Fig. 18.6: The success rates for sessions containing promotions compared to those without promotions.

A comparison of success rates between sessions with promotions (*promoted sessions*) and search sessions without promotions (*standard sessions*) is presented as Figure 18.6. The results show that during the course of the 10 week trial, on average, sessions with promotions are more likely to be successful (62%) than standard sessions (48%) containing only Google results, a relative benefit of almost 30% due to the community-based promotion of results. In other words, during the course of the trial we found that for more than half of the standard Google search sessions users failed to find any results worth selecting. In contrast, during the same period, the same searchers experienced a significantly greater success rate for sessions that contained community promotions, with less than 40% of these sessions failing to attract user selections. Within an enterprise these results can have an important impact when it comes to overall search productivity because there are significant savings to be made by eliminating failed search sessions in many knowledge-intensive business scenarios. For example, a recent report [30] by the International Data Corporation (IDC) found that, on average, knowledge workers spend 25% of their time searching for information, and an enterprise employing 1,000 knowledge workers will waste nearly $2.5 million per year (at an opportunity cost of $15 million) due to an inability to locate and retrieve information. In this context any significant reduction in the percentage of failed search sessions can play an important role in improving enterprise productivity, especially in larger organisations.

18.4.4 Discussion

The model of collaborative web search presented here is one that seeks to take advantage of naturally occurring query repetition and result selection regularity among communities of like-minded searchers. In this case-study we have focused on one particular type of search community in the form of a group of employees. Obviously this is a reasonably straightforward community to identify and it is perhaps not surprising that we have found a high degree of repetition and regularity to take advantage of during collaborative web search. Nonetheless, this type of community, where groups of individuals come together to perform similar information finding tasks, is a common one, whether it is employees in a company or students in a class or researchers in a research group.

There are of course many other types of community. For example, we have already mentioned the scenario where a group of visitors to a themed web site can be considered to be an ad-hoc search community. More generally, it is interesting to consider the open question of community discovery and identification, and there is considerable research at the present time devoted to exploring various approaches to automatically identifying online communities; see for example [11, 5, 21, 106, 105]. And as we develop a better understanding of the nature of online communities in the new world of the social web it may be possible to offer a more flexible form of search collaboration, facilitated by a more flexible and dynamic definition of search community.

18.5 Case-Study 2 - Web Search. Shared.

The previous case-study looked at a community-oriented view of collaborative web search, where the search activities of like-minded communities of searchers were used to influence mainstream search engine results. In this section we describe an alternative model of collaborative web search, as implemented in a system called HeyStaks, that is different in two important ways. First of all, HeyStaks adopts more user-led approach to collaborative web search, one that is focused on helping users to better organise and share their search experiences. HeyStaks does this by allowing users to create and share repositories of search experiences as opposed to coordinating the participation of search communities. Secondly, we adopt a very different approach to search engine integration. Instead of the proxy-based approach described in the previous case-study, HeyStaks is integrated with a mainsream search engine, such as Google, through a browser toolbar, which provides the collaborative search engine with the ability to capture and guide search activities. Finally, we will also summarize the findings of a recent live-user study to investigate the nature of search collaboration that manifests within HeyStaks' user population.

18.5.1 The HeyStaks System

HeyStaks adds two basic features to a mainstream search engine. First, it allows users to create *search staks*, as a type of folder for their search experiences at search time. Staks can be shared with others so that their searches will also be added to the stak. Second, HeyStaks uses staks to generate recommendations that are added to the underlying search results that come from the mainstream search engine. These recommendations are results that stak members have previously found to be relevant for similar queries and help the searcher to discover results that friends or colleagues have found interesting, results that may otherwise be buried deep within Google's default result-list.

As per Fig. 18.7, HeyStaks takes the form of two basic components: a client-side *browser toolbar* and a back-end *server*. The toolbar allows users to create and share staks and provides a range of ancillary services, such as the ability to tag or vote for pages. The toolbar also captures search click-throughs and manages the integration of HeyStaks recommendations with the default result-list. The back-end server manages the individual stak indexes (indexing individual pages against query/tag terms and positive/negative votes), the stak database (stak titles, members, descriptions, status, etc.), the HeyStaks social networking service and, of course, the recommendation engine. In the following sections we will briefly outline the basic operation of HeyStaks and then focus on some of the detail behind the recommendation engine.

Consider the following motivating example. Steve, Bill and some friends are planning a European vacation and they know that during the course of their research they will use web search as their primary source of information about what to do and where to visit. Steve creates a (private) search stak called "European Vacation 2008" and shares this with Bill and friends, encouraging them to use this stak for their vacation-related searches.

Fig. 18.8 shows Steve selecting this stak as he embarks on a new search for "Dublin hotels", and Fig. 18.9 shows the results of this search. The usual Google results are shown, but in addition HeyStaks has made two promotions. These have

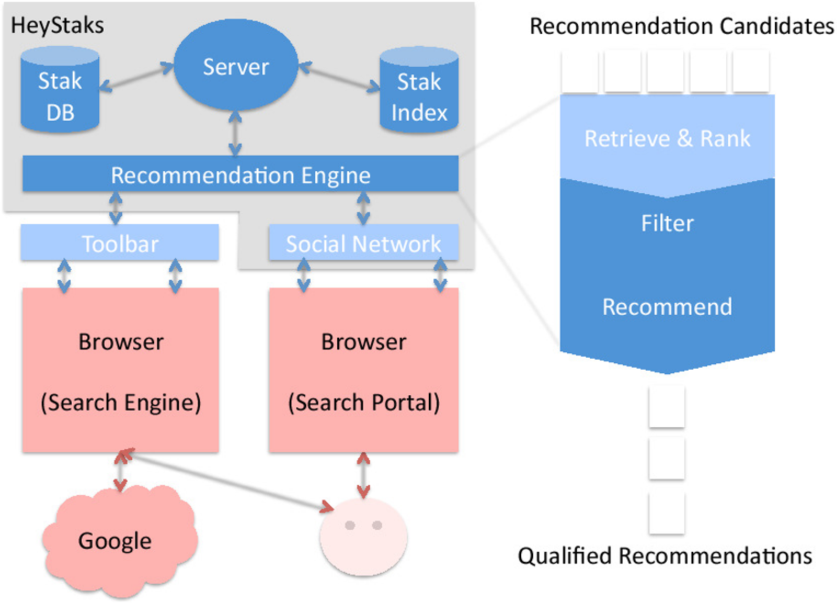

Fig. 18.7: The HeyStaks system architecture and outline recommendation model.

Fig. 18.8: Selecting a new active stak.

been promoted because other members of the “European Vacation 2008“ stak had recently found these results to be relevant; perhaps they selected them for *similar* queries, or voted for them, or tagged them with related terms. These recommendations may have been promoted from much deeper within the Google result-list, or they may not even be present in Google’s default results for the target query. Other relevant results may also be highlighted by HeyStaks, but left in their default Google position. In this way Steve and Bill benefit from promotions that are based on their previous similar searches. In addition, HeyStaks can recommend results from other related public staks as appropriate, helping searchers to benefit from the search knowledge that other groups and communities have created.

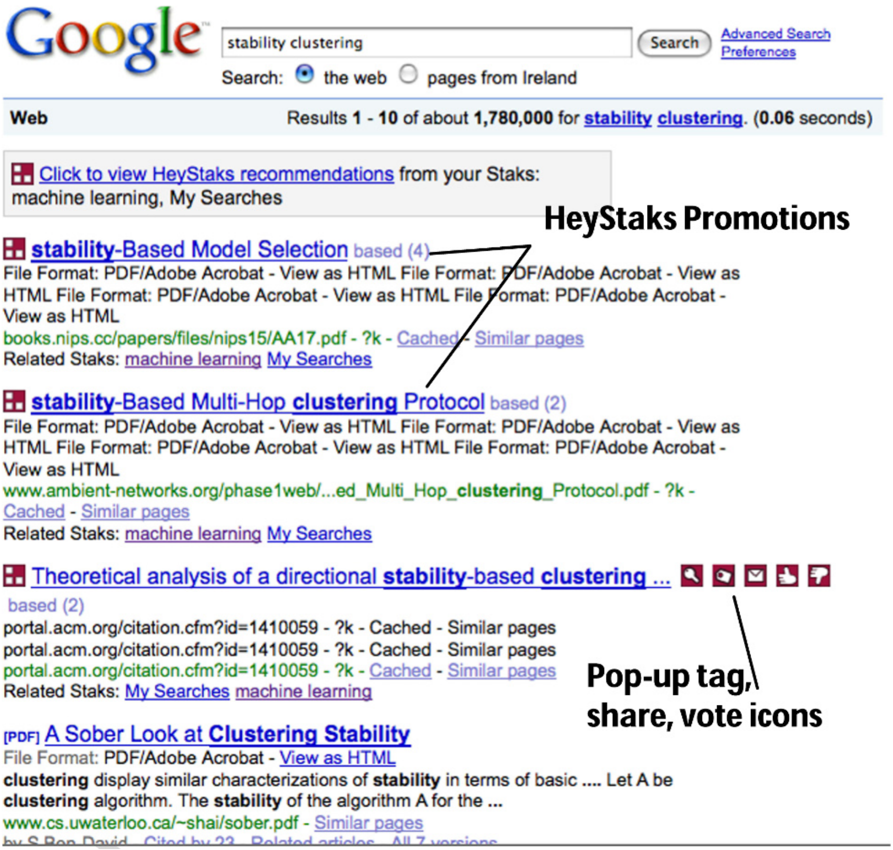

Fig. 18.9: Google search results with HeyStaks promotions.

Separately from the toolbar, HeyStaks users can also benefit from the HeyStaks *search portal*, which provides a social networking service built around people’s search histories. For example, Fig. 18.10 shows the portal page for the “European Vacation 2008“ stak, which is available to all stak members. It presents an activity feed of recent search history and a query cloud that makes it easy for the user to find out about what others have been searching for. The search portal also provides users with a wide range of features such as stak maintenance (e.g., editing, moving, copying results in staks and between staks), various search and filtering tools, and a variety of features to manage their own search profiles and find new search partners.

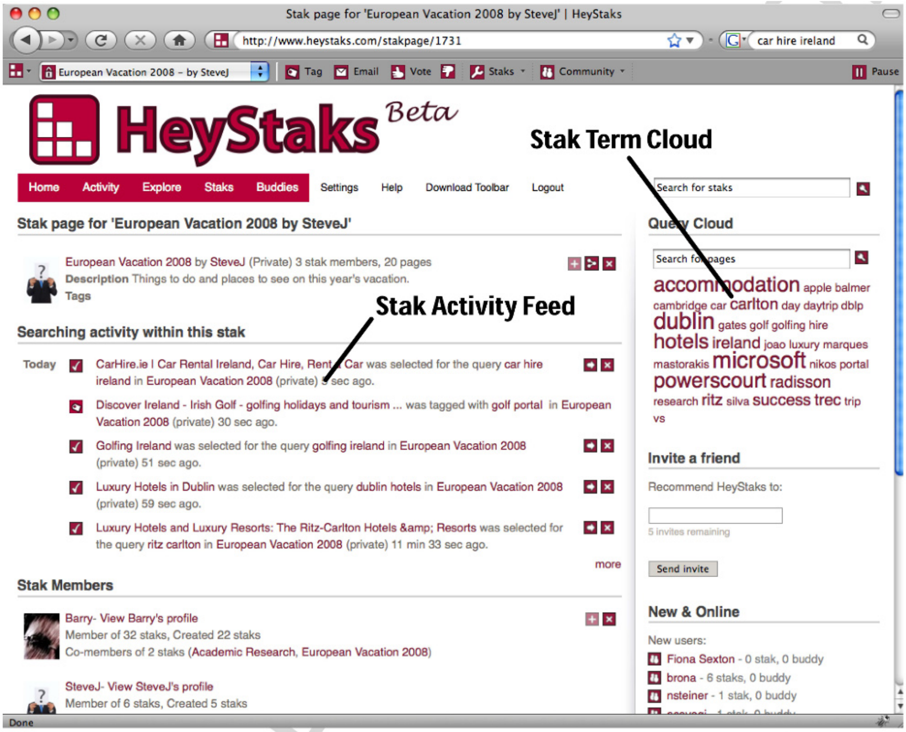

Fig. 18.10: The HeyStaks search portal provide direct access to staks and past searches.

18.5.2 The HeyStaks Recomendation Engine

In HeyStaks each search stak (S) serves as a profile of the search activities of the stak members and HeyStaks combines a number of implicit and explicit profiling techniques to capture a rich history of search experiences. Each stak is made up of a set of result pages ($S = \{p_1, \ldots, p_k\}$) and each page is anonymously associated with a number of implicit and explicit interest indicators, including the total number of times a result has been selected (sel), the query terms ($q_1, \ldots, q_n$) that led to its selection, the number of times a result has been tagged (*tag*), the terms used to tag it ($t_1,...,t_m$), the votes it has received (v^+, v^-), and the number of people it has been shared with (*share*) (all explicit indicators of interest) as indicated by Eq. 18.4.

$$p_i^S = \{q_1,...,q_n,t_1,...,t_m,v^+,v^-,sel,tag,share\} \tag{18.4}$$

In this way, each page is associated with a set of *term data* (query terms and/or tag terms) and a set of *usage data* (the selection, tag, share, and voting count). The term data is represented as a Lucene (*lucene.apache.org*) index table, with each page indexed under its associated query and tag terms, and provides the basis for retrieving and ranking *promotion candidates*. The usage data provides an additional source of evidence that can be used to filter results and to generate a final set of recommendations. At search time, a set of recommendations is produced in a number of stages: relevant results are retrieved and ranked from the Lucene stak index; these promotion candidates are filtered based on an *evidence model* to eliminate noisy recommendations; and the remaining results are added to the Google result-list according to a set of *recommendation rules*.

Briefly, there are two types of promotion candidates: *primary promotions* are results that come from the active stak S_t; whereas *secondary promotions* come from other staks in the searcher's stak-list. To generate these promotion candidates, the HeyStaks server uses the current query q_t as a probe into each stak index, S_i, to identify a set of relevant stak pages $P(S_i,q_t)$. Each candidate page, p, is scored using Lucene's *TF.IDF* retrieval function as per 18.5, which serves as the basis for an initial recommendation ranking.

$$score(q_t,p) = \sum_{t \in q_t} tf(t \in p) \bullet idf(t)^2 \tag{18.5}$$

Staks are inevitably noisy, in the sense that they will frequently contain pages that are not on topic. For example, searchers will often forget to set an appropriate stak at the start of a new search session and, although HeyStaks includes a number of automatic stak-selection techniques to ensure that the right stak is active for a given search, these techniques are not perfect, and misclassifications do inevitably occur; see also [18, 95]. As a result, the retrieval and ranking stage may select pages that are not strictly relevant to the current query context. To avoid making spurious recommendations HeyStaks employs an *evidence filter*, which uses a variety of threshold models to evaluate the relevance of a particular result, in terms of its usage evidence; tagging evidence is considered more important than voting, which in turn is more important than implicit selection evidence. For example, pages that have only been selected once, by a single stak member, are not automatically considered for recommendation and, all other things being equal, will be filtered out at this stage. In turn, pages that have received a high proportion of negative votes will also be eliminated. The precise details of this model are beyond the scope of this paper but suffice it to say that any results which do not meet the necessary evidence thresholds are eliminated from further consideration.

After evidence pruning we are left with revised primary and secondary promotions and the final task is to add these *qualified recommendations* to the Google result-list. HeyStaks uses a number of different recommendation rules to determine how and where a promotion should be added. Once again, space restrictions prevent a detailed account of this component but, for example, the top 3 primary promotions are always added to the top of the Google result-list and labelled using the HeyStaks promotion icon. If a remaining primary promotion is also in the default Google result-list then this is labeled in place. If there are still remaining primary promotions then these are added to the secondary promotion list, which is sorted according to TF.IDF scores. These recommendations are then added to the Google result-list as an optional, expandable list of recommendations; for further details see [93, 94]

18.5.3 Evaluation

In this section we examine a subset of 95 HeyStaks users who have remained active during the course of the early beta release of the toolbar and service. These users registered with HeyStaks during the period October-December 2008 and the results below represent a summary of their usage during the period October 2008 - January 2009. Our aim is to gain an understanding of both how users are using HeyStaks, and whether they seem to be benefiting from its search promotions. Because this is a study of live-users *in the wild* there are certain limitations about what we have been able to measure. There is no control group, for example, and it was not feasible, mainly for data privacy reasons, to analyse the relative click-through behaviour of users, by comparing their selections of default Google results to their selections of HeyStaks promotions. However, for the interested reader, our earlier work does report on this type of analysis in more conventional control-group laboratory studies [10, 25, 92].

Key to the HeyStaks proposition is that searchers need a better way to organise and share their search experiences. HeyStaks provides these features but do users actually take the time to create staks? Do they share them with others or join those created by others?

During the course of the initial deployment of HeyStaks users did engage in a reasonable degree of stak creation and sharing activity. For example, as per Fig. 18.11, on average, beta users created just over 3.2 new staks and joined a further 1.4. Perhaps this is not surprising: most users create a few staks and share them with a small network of colleagues or friends, at least initially.

In total there were over 300 staks created on a wide range of topics, from broad topics such as travel, research, music and movies, to more niche interests including archaeology, black and white photography, and mountain biking. A few users were prolific stak creators and joiners: one user created 13 staks and joined another 11, to create a search network of 47 other searchers (users who co-shared the same staks).

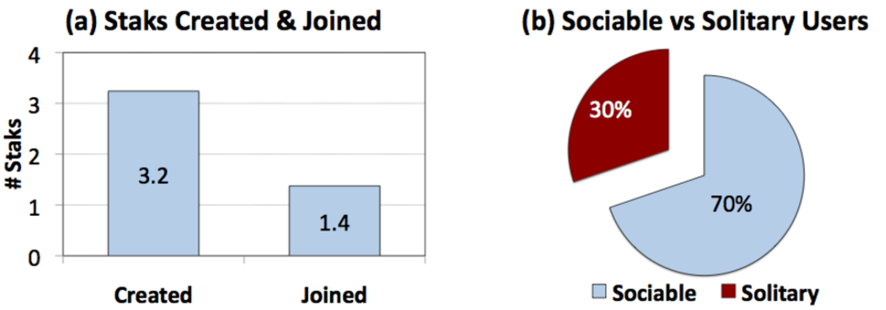

Fig. 18.11: (a) Average staks created and joined per user. (b) The percentage of *sociable* and *solitary* users.

In fact on average, each user was connected to a search network of just over 5 other searchers by the staks that they shared.

The vast majority of staks were created as public staks, although most (52%) remained the domain of a single member, the stak creator. Thus 48% of staks were shared with at least one other user and, on average, these staks attracted 3.6 members. Another way to look at this is as depicted in Fig. 18.11(b): 70% of users make the effort to share or join staks (*sociable* users); and only 30% of users created staks just for their own personal use and declined to join staks created by others (*solitary* users).

At its core HeyStaks is motivated by the idea that web search is an inherently social or collaborative activity. And even though mainstream search engines do not support this, searchers do find alternative collaboration channels (e.g., email, IM, etc.) with which to partially, albeit inefficiently, share their search experiences; see for example [57]. One of the most important early questions to ask about HeyStaks users concerns the extent to which their natural search activity serves to create a community of collaborating searchers. As users search, tag, and vote they are effectively producing and consuming community search knowledge. A user might be the first to select or tag a given result for a stak and, in this context, they have *produced* new search knowledge. Later, if this result is promoted to another user and then re-selected (or tagged or voted on), then this other user is said to have *consumed* that search knowledge; of course they have also produced search knowledge as their selection, tag, or vote is added to the stak.

We have found that 85% of users have engaged in search collaborations. The majority have consumed results that were produced by at least one other user, and on average these users have consumed results from 7.45 other users. In contrast 50% of users have produced knowledge that has been consumed by at least one other user, and in this case each of these producers has created search knowledge that is consumed by more than 12 other users on average.

One question we might ask is to what *degree* individual users tend to be producers or consumers of search knowledge. Are some searchers *net producers* of search knowledge, in the sense that they are more inclined to create search knowledge that

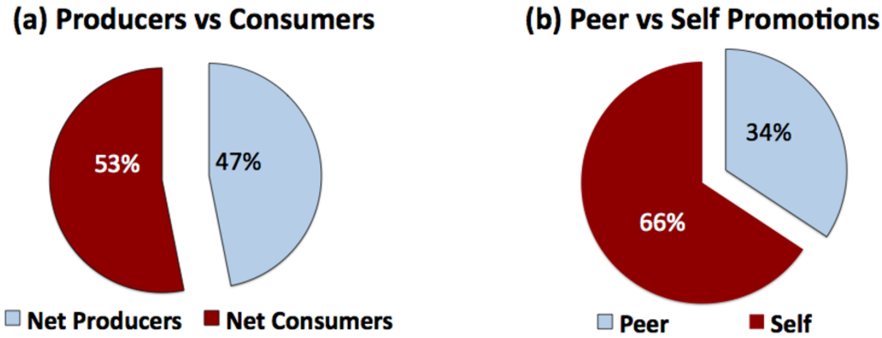

Fig. 18.12: (a) Net producers vs. consumers. (b) Promotion sources (self vs. peer).

is useful to others? Are other users *net consumers*, in the sense that they are more inclined to consume search knowledge that others have created? This data is presented in Fig. 18.12(a). To be clear, a net producer is defined as a user who has helped more other users than they themselves have been helped by, whereas a net consumer is defined as a user who has been helped by more users than they themselves have helped. The chart shows that 47% of users are net producers. Remember that, above, we noted how 50% of users have produced at least *some* search knowledge that has been consumed by some other user. It seems that the vast majority of *these* users, 94% of them in fact, are actually helping more people than they are helped by in return.

So, we have found that lots of users are helping other users, and lots of users are helped by other users. Perhaps this altruism is limited to a small number of searches? Perhaps, most of the time, at the level of individual searches, users are helping themselves? A variation on the above analysis can help shed light on this question by looking at the source of promotions that users judge to be relevant enough to select during their searches. Overall, the beta users selected more than 11,000 promotions during their searches. Some of these promotions will have been derived from the searcher's own past history; we call these *self* promotions. Others will have been derived from the search activities of other users who co-share staks with the searcher; we call these *peer* promotions. The intuition here is that the selection of self promotions corresponds to examples of HeyStaks helping users to *recover* results they have previously found, whereas the selection of promotions from peers corresponds to *discovery* tasks, where the user is benefiting from focused new content that might otherwise have been missed, or have been difficult to find; see [61, 50]. Thus Fig. 18.12(b) compares the percentage of peer and self promotions and shows that two-thirds of selected promotions are generated from the searcher's own past search activities; most of the time HeyStaks is helping searchers to recover previously found results. However, 33% of the time peer promotions are selected (and we already know that these come from many different users), helping the searcher to discover new information that others have found.

The bias towards self promotions is perhaps not surprising, especially given the habits of searchers, and especially during the early stages of stak development. The growth of most staks is initially led by a single user, usually the creator, and so inevitably most of the promotions are generated in response to the creator's own search queries. And most of these promotions will be self promotions, derived from the leader's own search activities. Many staks are not shared and so are only capable of making self promotions. As staks are shared, however, and more users join, the pool of searchers becomes more diverse. More results are added by the actions of peers and more peer promotions are generated and selected. It is an interesting task for future work to explore the evolution of a search stak and to investigate how stak content and promotions are effected as more and more users participate. Are there well-defined stages in stak evolution, for example, as self promotions give way to peer promotions? For now it is satisfying to see that even in the early stages of stak evolution, where the average stak has between 3 and 4 members, that 34% of the time members are benefiting from promotions that are derived from the activities of their peers.

18.5.4 Discussion

Compared to the first case-study, HeyStaks promotes a much more explicit form of search collaboration — search staks are explicitly created and shared by users — and the result is the formation of *micro* search communities in which small groups of searchers collaborate on particular search themes or topics. Of course this does not preclude the formation of larger groups of collaborating searchers, and it is entirely likely that certain types of search stak will evolve to become search communities in a manner that fits well with those contemplated by the previous case-study.

Once again, there are many questions left unanswered by this case-study as it provides a fertile ground for further research. For example, the potential proliferation of search staks leads to entirely new recommendation opportunities as users may benefit from suggestions about which staks to join, for example. Moreover, it may be interesting to consider the merging and/or splitting of staks in certain circumstances, allowing users to create staks by combining existing staks, for instance.

18.6 Conclusions

Web search engines are, and no doubt will continue to be, the primary tools that we will use to discover and explore online information. For all of the success of mainstream search engines like Google, the web search problem is far from being solved and research into a new generation of web search technologies is maturing. In the future it is likely that mainstream search engines will evolve to offer users greater support when it comes to finding the right information at the right time, and recommendation technologies are set to play an important part of this future.

Already, for example, researchers are exploring how to make search engines more responsive to our particular, individual needs and preferences by combining user profiling and recommendation technologies to deliver a more personalized user experience, whether through the generation of targeted result-lists or improved query recommendation, for example. Another strand of research seeks to take advantage of the inherently collaborative nature of many web search tasks by providing searchers with new tools to foster and promote search collaboration between small groups and even large communities of searchers.

In this chapter we have provided a snapshot of these interesting areas of independent research by surveying a number of representative systems and techniques. In turn we have highlighted how these complementary approaches to collaborative and personalized web search are beginning to come together to offer users improved personalization as a side-effect of collaboration, with recommender systems playing a central role in a new type of social search service. In this regard we have presented two separate case-studies of these social search systems to show how mainstream search engines like Google may be enhanced by such approaches in practice.

In the future it is likely that mainstream search engines will evolve to accommodate many elements of these approaches, as recommendation technologies play an increasing role in web search. Where today the burden of web search is very much on the individual searcher, we believe that the introduction of recommendation technologies will provide search engines with the opportunity to be a lot more proactive as they work to anticipate, rather than respond to, a user's information needs. This in turn will lead to many new research opportunities, especially at the level of the search interface, as we look for new ways to incorporate recommendation techniques into the very fabric of web search. Indeed, already we are seeing some early examples of this as, for instance, search engines like Google and Yahoo, incorporate query recommendation techniques in to their regular search boxes. But this is just the beginning and as researchers address the challenges of profiling, privacy, and recommendation head-on, search engines will provide a unique platform for the next generation of recommendation technologies. And just as the e-commerce sites have served as an early platform for recommender systems, search engines will help to introduce a new era of recommendation technologies to a much wider audience.

Acknowledgements

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

References

1. Saleema Amershi and Meredith Ringel Morris. Cosearch: a system for co-located collaborative web search. In *Proceedings of the annual SIGCHI conference on Human factors in computing systems (CHI)*, pages 1647–1656, 2008.
2. Fabio A. Asnicar and Carlo Tasso. Ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In *Proceedings of the Workshop on Adaptive Systems and User Modeling on the World Wide Web at the Sixth International Conference on User Modeling*, pages 3–11, 1997.
3. Ricardo A. Baeza-Yates, Carlos A. Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In *Current Trends in Database Technology - EDBT 2004 Workshops*, pages 588–596, 2004.
4. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
5. Lori Baker-Eveleth, Suprateek Sarker, and Daniel M. Eveleth. Formation of an online community of practice: An inductive study unearthing key elements. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 254–256, 2005.
6. M. Balabanovic and Y. Shoham. FAB: Content-Based Collaborative Recommender. *Communications of the ACM*, 40(3):66–72, 1997.
7. Evelyn Balfe and Barry Smyth. Improving web search through collaborative query recommendation. In *Proceedings of the European Conference on Artificial Intelligence (ECAI*, pages 268–272, 2004.
8. D. Billsus and M. Pazzani. A Hybrid User Model for News Story Classification. In *Proceedings of the Seventh International Conference on User Modeling, UM '99*, 1999. Banff, Canada.
9. Daniel Billsus, Michael J. Pazzani, and James Chen. A learning agent for wireless news access. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 33–36, New York, NY, USA, 2000. ACM Press.
10. Oisín Boydell and Barry Smyth. Enhancing case-based, collaborative web search. In *Proceedings of International Conference on Case-Based Reasoning (ICCBR)*, pages 329–343, 2007.
11. John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker. Towards semantically-interlinked online communities. In *European Semantic Web Conference (ESWC)*, pages 500–514, 2005.
12. Peter Briggs and Barry Smyth. Provenance, trust, and sharing in peer-to-peer case-based web search. In *Proceedings of European Conference on Case-Based Reasoning (ECCBR)*, pages 89–103, 2008.
13. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.
14. Jay Budzik and Kristian J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 44–51, New York, NY, USA, 2000. ACM.
15. R. Burke. The Wasabi Personal Shopper: A Case-Based Recommender System. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. AAAI Press, 1999.
16. John M. Carroll and Mary Beth Rosson. Paradox of the active user. In John M. Carroll, editor, *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, chapter 5, pages 80–111. Bradford Books/MIT Press, 1987.
17. Soumen Chakrabarti, Byron Dom, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon M. Kleinberg. Mining the web's link structure. *IEEE Computer*, 32(8):60–67, 1999.
18. Pierre-Antoine Champin, Peter Briggs, Maurice Coyle, and Barry Smyth. Coping with noisy search experiences. In *Twenty-ninth SGAI International Conference on Artificial Intelligence (AI-2009)*. Springer-Verlag, 2009.

19. Huan Chang, David Cohn, and Andrew McCallum. Learning to create customized authority lists. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 127–134, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
20. Li Chen and Pearl Pu. Evaluating critiquing-based recommender agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2006.
21. Alvin Chin and Mark H. Chignell. Identifying active subgroups in online communities. In *CASCON*, pages 280–283, 2007.
22. Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. 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, New York, NY, USA, 2005. ACM Press.
23. Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. Pros: A personalized ranking platform for web search. In Paul De Bra and Wolfgang Nejdl, editors, *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, volume 3137 of *Lecture Notes in Computer Science*, pages 34–43. Springer, 2004.
24. Maurice Coyle and Barry Smyth. Information recovery and discovery in collaborative web search. In *Proceedings of the European COnference on Information retrieval (ECIR)*, pages 356–367, 2007.
25. Maurice Coyle and Barry Smyth. Supporting intelligent web search. *ACM Trans. Internet Techn.*, 7(4), 2007.
26. W. Bruce Croft, Stephen Cronen-Townsend, and Victor Larvrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.
27. B.J. Dahlen, J.A. Konstan, J.L. Herlocker, N. Good, A. Borchers, and J. Riedl. Jump-starting movieLens: User benefits of starting a collaborative filtering system with "dead-data". In . University of Minnesota TR 98-017, 1998.
28. B. V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
29. R. Dolin, D. Agrawal, A. El Abbadi, and L. Dillon. Pharos: a scalable distributed architecture for locating heterogeneous information sources. In *CIKM '97: Proceedings of the sixth international conference on Information and knowledge management*, pages 348–355, New York, NY, USA, 1997. ACM.
30. Susan Feldman and Chris Sherman. The High Cost of Not Finding Information. In *(IDC White Paper)*. IDC Group, 2000.
31. L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: the concept revisited. In *WWW '01: Proceedings of the 10th International Conference on the World Wide Web*, pages 406–414. ACM Press, 2001.
32. C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: an automatic citation indexing system. In *DL '98: Proceedings of the third ACM conference on Digital libraries*, pages 89–98, New York, NY, USA, 1998. ACM.
33. Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 478–479, New York, NY, USA, 2004. ACM.
34. Luis Gravano, Héctor García-Molina, and Anthony Tomasic. Gloss: text-source discovery over the internet. *ACM Trans. Database Syst.*, 24(2):229–264, 1999.
35. Anthony Jameson and Barry Smyth. Recommendation to groups. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, pages 596–627. Springer-Verlag, 2007.
36. Bernard J. Jansen and Amanda Spink. An analysis of web searching by european alltheweb.com users. *Inf. Process. Manage.*, 41(2):361–381, 2005.
37. Bernard J. Jansen, Amanda Spink, Judy Bateman, and Tefko Saracevic. Real life information retrieval: a study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.

38. Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 271–279, New York, NY, USA, 2003. ACM.
39. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
40. Jon M. Kleinberg. Hubs, authorities, and communities. *ACM Comput. Surv.*, 31(4):5, 1999.
41. J.A. Konstan, B.N Miller, D. Maltz, J.L. Herlocker, L.R. Gorgan, and J. Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
42. Yehuda Koren. Tutorial on recent progress in collaborative filtering. In *Proceedings of the International Conference on Recommender Systems (RecSys)*, pages 333–334, 2008.
43. G. Koutrika and Y. Ioannidis. A unified user-profile framework for query disambiguation and personalization. In *Proc. of the Workshop on New Technologies for Personalized Information Access*, pages 44–53, 2005.
44. A. Kruger, C. L. Giles, F. M. Coetzee, E. Glover, G. W. Flake, S. Lawrence, and C. Omlin. Deadliner: building a new niche search engine. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pages 272–281, New York, NY, USA, 2000. ACM.
45. S. Lawrence and C. Lee Giles. Accessibility of Information on the Web. *Nature*, 400(6740):107–109, 1999.
46. Greg Linden, Brent Smith, and Jeremy York. Industry report: Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Distributed Systems Online*, 4(1), 2003.
47. Fang Liu, Clement Yu, and Weiyi Meng. 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, New York, NY, USA, 2002. ACM Press.
48. Zhongming Ma, Gautam Pant, and Olivia R. Liu Sheng. Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25(1):5, 2007.
49. Christos Makris, Yannis Panagis, Evangelos Sakkopoulos, and Athanasios Tsakalidis. Category ranking for personalized search. *Data Knowl. Eng.*, 60(1):109–125, 2007.
50. Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.
51. Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. Experiments in dynamic critiquing. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 175–182, 2005.
52. Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. Cats: A synchronous approach to collaborative group recommendation. In *Proceedings of the International FLAIRS Conference*, pages 86–91, 2006.
53. L. McGinty and B. Smyth. Comparison-Based Recommendation. In Susan Craw, editor, *Proceedings of the Sixth European Conference on Case-Based Reasoning (ECCBR 2002)*, pages 575–589. Springer, 2002. Aberdeen, Scotland.
54. Ryan J. Meuth, Paul Robinette, and Donald C. Wunsch. Computational intelligence meets the netflix prize. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 686–691, 2008.
55. Alessandro Micarelli and Filippo Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200, 2004.
56. M. Mitra, A. Singhal, and C. Buckley. Improving Automatic Query Expansion. In *Proceedings of ACM SIGIR*, pages 206–214. ACM Press, 1998.
57. Meredith Ringel Morris. A survey of collaborative web search practices. In *Proceedings of the annual SIGCHI conference on Human factors in computing systems (CHI)*, pages 1657–1660, 2008.
58. Meredith Ringel Morris and Eric Horvitz. S^3: Storable, shareable search. In *INTERACT (1)*, pages 120–123, 2007.

59. Meredith Ringel Morris and Eric Horvitz. Searchtogether: an interface for collaborative web search. In *UIST*, pages 3–12, 2007.
60. Sridhar P. Nerur, Riyaz Sikora, George Mangalaraj, and Venugopal Balijepally. Assessing the relative influence of journals in a citation network. *Commun. ACM*, 48(11):71–74, 2005.
61. Vicki L. O'Day and Robin Jeffries. Orienteering in an information landscape: how information seekers get from here to there. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 438–445, New York, NY, USA, 1993. ACM Press.
62. John O'Donovan, Vesile Evrim, and Barry Smyth. Personalizing trust in online auctions. In *Proceedings of the European Starting AI Researcher Symposium (STAIRS)*, Trento, Italy, 2006.
63. John O'Donovan and Barry Smyth. Eliciting trust values from recommendation errors. In *Proceedings of the International FLAIRS Conference*, pages 289–294, 2005.
64. John O'Donovan and Barry Smyth. Trust in recommender systems. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 167–174, 2005.
65. John O'Donovan and Barry Smyth. Trust no one: Evaluating trust-based filtering for recommenders. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1663–1665, 2005.
66. John O'Donovan and Barry Smyth. Is trust robust?: an analysis of trust-based recommendation. In *Intelligent User Interfaces*, pages 101–108, 2006.
67. John O'Donovan and Barry Smyth. Mining trust values from recommendation errors. *International Journal on Artificial Intelligence Tools*, 15(6):945–962, 2006.
68. Jeremy Pickens, Gene Golovchinsky, Chirag Shah, Pernilla Qvarfordt, and Maribeth Back. Algorithmic mediation for collaborative exploratory search. In *SIGIR*, pages 315–322, 2008.
69. Alexander Pretschner and Susan Gauch. Ontology based personalized search. In *ICTAI '99: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, page 391, Washington, DC, USA, 1999. IEEE Computer Society.
70. Madhu C. Reddy and Paul Dourish. A finger on the pulse: temporal rhythms and information seeking in medical work. In *CSCW*, pages 344–353, 2002.
71. Madhu C. Reddy, Paul Dourish, and Wanda Pratt. Coordinating heterogeneous work: Information and representation in medical care. In *ECSCW*, pages 239–258, 2001.
72. Madhu C. Reddy and Bernard J. Jansen. A model for understanding collaborative information behavior in context: A study of two healthcare teams. *Inf. Process. Manage.*, 44(1):256–273, 2008.
73. Madhu C. Reddy and Patricia Ruma Spence. Collaborative information seeking: A field study of a multidisciplinary patient care team. *Inf. Process. Manage.*, 44(1):242–255, 2008.
74. J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Dynamic Critiquing. In Peter Funk and Pedro A. Gonzalez Calero, editors, *Proceedings of the 7th European Conference on Case-Based Reasoning*, pages 763–777. Springer-Verlag, 2004.
75. James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Incremental critiquing. *Knowl.-Based Syst.*, 18(4-5):143–151, 2005.
76. James Reilly, Barry Smyth, Lorraine McGinty, and Kevin McCarthy. Critiquing with confidence. In *Proceedings of International Conference on Case-Based Reasoning (ICCBR)*, pages 436–450, 2005.
77. Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, 1997.
78. J. Rocchio. *Relevance Feedback in Information Retrieval*. G. Salton (editor), The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice–Hall, Inc., Englewood Cliffs, NJ, 1971.
79. U. Rohini and Vasudeva Varma. A novel approach for re-ranking of search results using collaborative filtering. In *Proceedings of the International Conference on Computing: Theory and Applications*, volume 00, pages 491–496, Los Alamitos, CA, USA, 2007. IEEE Computer Society.
80. Lior Rokach, Genetic algorithm-based feature set partitioning for classification problems,Pattern Recognition, 41(5):1676–1700, 2008.

81. Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the International World-Wide Web Conference*, pages 377–386, 2006.
82. Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. In *Readings in information retrieval*, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
83. J. Ben Schafer, Joseph Konstan, and John Riedi. Recommender systems in e-commerce. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, New York, NY, USA, 1999. ACM Press.
84. U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '95)*, pages 210–217. ACM Press, 1995. New York, USA.
85. U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 210–217, 1995.
86. Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit User Modeling for Personalized Search. In *Proceedings of the Fourteenth ACM Conference on Information and Knowledge Management (CIKM 05)*, 2005.
87. Alan F. Smeaton, Colum Foley, Daragh Byrne, and Gareth J. F. Jones. ibingo mobile collaborative search. In *CIVR*, pages 547–548, 2008.
88. Alan F. Smeaton, Hyowon Lee, Colum Foley, and Sinéad McGivney. Collaborative video searching on a tabletop. *Multimedia Syst.*, 12(4-5):375–391, 2007.
89. Barry Smyth. Case-based recommendation. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, pages 342–376. Springer-Verlag, 2007.
90. Barry Smyth. A community-based approach to personalizing web search. *IEEE Computer*, 40(8):42–50, 2007.
91. Barry Smyth, Evelyn Balfe, Oisín Boydell, Keith Bradley, Peter Briggs, Maurice Coyle, and Jill Freyne. A Live-user Evaluation of Collaborative Web Search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pages 1419–1424. Morgan Kaufmann, 2005. Ediburgh, Scotland.
92. Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle, and Oisín Boydell. Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Model. User-Adapt. Interact.*, 14(5):383–423, 2004.
93. Barry Smyth, Peter Briggs, Maurice Coyle, and Michael P O'Mahony. A case-based perspective on social web search. In *Proceedings of International Conference on Case-Based Reasoning (ICCBR)*, 2009.
94. Barry Smyth, Peter Briggs, Maurice Coyle, and Michael P. O'Mahony. Google. shared! a case-study in social search. In *Proceedings of the International Conference on User Modeling, Adaptation and Personalization (UMAP)*, pages 494–508. Springer-Verlag, 2009.
95. Barry Smyth and Pierre-Antoine Champin. The experience web: A case-based reasoning perspective. In *Workshop on Grand Challenges for Reasoning from Experiences (IJCAI 2009)*, 2009.
96. Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. In *Document retrieval systems*, pages 132–142. Taylor Graham Publishing, London, UK, UK, 1988.
97. Micro Speretta and Susan Gauch. Personalized search based on user search histories. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 622–628, Washington, DC, USA, 2005. IEEE Computer Society.
98. Amanda Spink, Judy Bateman, and Major Bernard Jansen. Searching heterogeneous collections on the web: behaviour of excite users. *Information Research: An Electronic Journal*, 4(2), 1998.
99. Amanda Spink and Bernard J. Jansen. A study of web search trends. *Webology*, 1(2):4, 2004.

100. Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. Searching the Web: the Public and their Queries. *Journal of the American Society for Information Science*, 52(3):226–234, 2001.
101. Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2004. ACM Press.
102. Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cubesvd: a novel approach to personalized web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 382–390, New York, NY, USA, 2005. ACM Press.
103. Bin Tan, Xuehua Shen, and ChengXiang Zhai. Mining long-term search history to improve search accuracy. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723, New York, NY, USA, 2006. ACM Press.
104. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.
105. Xin Wang and Ata Kabán. State aggregation in higher order markov chains for finding online communities. In *IDEAL*, pages 1023–1030, 2006.
106. Xin Wang and Ata Kabán. A dynamic bibliometric model for identifying online communities. *Data Min. Knowl. Discov.*, 16(1):67–107, 2008.
107. Yiming Yang and Christopher G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Trans. Inf. Syst.*, 12(3):252–277, 1994.
108. Baoyao Zhou, Siu Cheung Hui, and Alvis C. M. Fong. An effective approach for periodic web personalization. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 284–292, Washington, DC, USA, 2006. IEEE Computer Society.