Collaboration, Reputation and Recommender Systems in Social Web Search

Barry Smyth, Maurice Coyle, Peter Briggs, Kevin McNally, Michael P. O'Mahony

Abstract Modern web search engines have come to dominate how millions of people find the information that they are looking for online. While 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 two decades, mainstream search is not without its challenges. Mainstream search engines continue to provide a largely *one-size-fits-all* service to their user-base, ultimately limiting the relevance of their result-lists. And they have only very recently begun to consider how the rise of the social web may support novel approaches to search and discovery, or how such signals can be used to inform relevance. In this chapter we will explore recent research which aims to do just that: to make web search a more personal and collaborative experience and to leverage important information such as the reputation of searchers during result-ranking. In short we look towards a more social future for mainstream search.

1 Introduction

The web is probably one of the most important and wide-spread information tools in use today. Many of us interact with general purpose search engines such as Goggle and Bing many times a day, while some of us use more specialized search services to cater for niche needs from time to time. Indeed mainstream search has become so much a part of everyday life that one would be forgiven for assuming that all of the major web search challenges have been overcome The reality is very different, however, and by some measures the pace of innovation in web search has never been greater as leading services continue to look for new ways to cope with the many

Barry Smyth, Michael P. O'Mahony at the Insight Centre for Data Analytics, University College Dublin. Maurice Coyle and Peter Briggs at HeyStaks Technologies Ltd., NovaUCD. Kevin McNally at CLARITY: Centre for Sensor Web Technologies, University College Dublin, Ireland.

challenges that remain in order to satisfy their users' changing needs and evolving expectations.

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, a disappointing and somewhat surprising success rate by any standard. In large part this problem is as much due to the searcher as it is the search engine: our search queries tend to be vague and under-specified, and rarely provide a clear indication of our search needs [49,113,123–125]. Mostly we have adapted to these low success rates. We respond to poor result-lists with follow-up or alternative queries until we find what we are looking for. And while we usually do find what we are looking for it comes at a cost — wasted time and effort — and sometimes we may abandon our efforts altogether. 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, even though it exists somewhere.

Thus, there remains plenty of scope for improvement in mainstream search. This is particularly true as the web evolves to become a more social and collaborative world, creating new opportunities to learn about and harness user preferences and relationships. 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 approach to web search led to its relentless rise over the past 15 years, so too will new search technologies emerge to have a similarly disruptive effect on the market over the next 15 years.

It can be useful to view modern search engines as a type of recommender system: they respond to user queries with a set of result-page recommendations. But unlike many conventional recommender systems, search engines have focused on text and link analysis rather than the user interactions and the similarity relationships that drive recommender systems. There is now an opportunity to recommendation technologies 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 [102]. 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 [5,37,59,94,104,112]. They question the very core assumptions of mainstream web search engines and suggest important adaptations to conventional approaches to web search.

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, regardless of their preferences — and argues that web search needs to become more *personalized* by considering the implicit needs and preferences of searchers. We will review a number of different approaches to personalizing web search which harness different types of user preference and context information to influence the search experience; see for example [2, 15, 20, 22, 23, 31, 33, 35, 53, 54, 85, 108, 122, 131]. That being said many mainstream search engines are beginning to adapt the results that they return to users, based on factors such as location, time of day etc. but less so based on an understanding of user preferences or needs. A valid concern when it comes to adapting or personalizing result-lists is the extent to which it may blinker the searcher and limit the possible views and opinions that the searcher is exposed to in the long-run; see [81]. However, personalization does not necessarily oblige a narrowing of results. And one of the most interesting dimensions to modern recommender systems is the extent to which they seek to explore issues such as diversity and novelty as well as relevance when it comes to evaluating the quality of result-lists; see for example, [7, 10, 36, 56, 60, 111]. In this sense the solution to Pariser's Filter Bubble is the recommendation of more diverse results and/or results that express novel and divergent viewpoints.

The second assumption to be questioned concerns the *solitary nature* of web search. By and large modern web search takes the form of an isolated interaction between a lone searcher and search engine. However, recent research suggests that there are many circumstances where the search for information can (and should) have a more collaborative flavour. Often it makes sense for groups of searchers (e.g., friends, colleagues, classmates) to cooperate 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 this potential for collaboration during a variety of information seeking tasks; see for example, [1, 69, 70, 87–90, 117].

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. This provides a new source of information for search engines to use during retrieval, specifically collaboration and reputation 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 derived from the preferences and activities of searchers. In doing so we will bring together recommendation and search in a way that points to a new future for search engine development in the ongoing quest to deliver the right information to the right user at the right time.

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 [4, 129]. A simplified schematic of a typical search engine architecture is presented in Fig. 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 [121], 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.

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 [43] 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

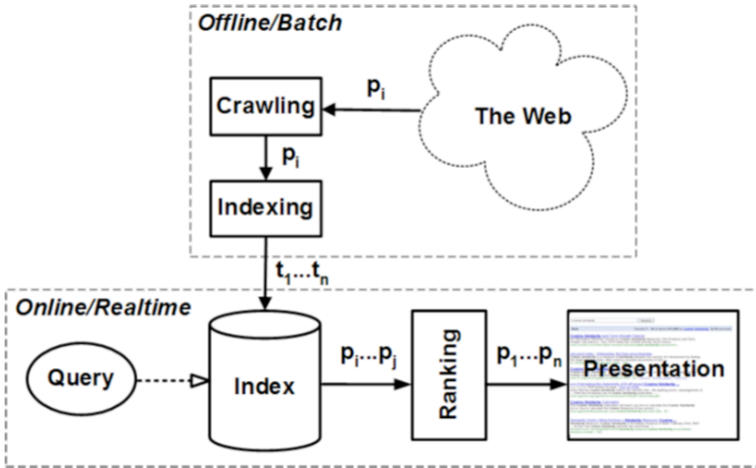


Fig. 1 Functional components of a typical web search engine. A page, p_i , is located on the web by the crawler and its content, the terms $t_1,...,t_n$, are retrieved and indexed as part of an offline process. In response to a search query, the engine probes the index to retrieve results which match 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.

individual pages. Kleinberg, Page, and Brin recognised the web as a type of *citation network* (see for example, [71]). 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 [18, 43] 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.

3 The Future of Web Search

There is no doubt that web search represents a very significant information discovery and 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 [38, 39, 49, 125]. 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 (for example, see [2, 15, 22, 31, 53, 108]). 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 (for example, see [1, 69, 70, 87–90, 117]).

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 3.1) or more collaborative (see Section 3.2) search services. These initiatives borrow ideas from recommender systems, user profiling, and computer-supported collaborative working research (for example, see [37, 44, 58, 106, 112]). 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 later in this chapter we will describe two detailed case-studies of two different approaches to social search.

3.1 Personalizing 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, 44, 82], books [51], movies [27, 45, 65], and even products to buy [21, 57, 91–93, 104] — 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 [15, 31, 108], while others accommodate *long-term* profiles that capture the user's preferences over an extended period of time [2, 22, 53]. 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 (for example, see [16, 40, 46, 112]) versus those that are *collaborative*, in the sense that they are guided by the profiles of a group of users (for example, see [37, 44, 51, 107, 113]).

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, 53] 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 [22, 52, 85].

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. [53] 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 [66] in which profiles initially built using terms selected from a list can be subsequently improved with feedback on

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

viewed documents provided by 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 [17]. 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. [52] 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) [130], the Rocchio relevance feedback algorithm [96], and k-Nearest Neighbor (kNN) [28]. In a related approach, [128] use statistical language methods to mine contextual information from this type of long-term search history to build a language model based profile, and [85] 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 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 [108] 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 [15] 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 [29, 32, 34, 47] of raw page content and query generation [3, 6, 67]. 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 [46] 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 [103]. 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 [40] do this by introducing a personalized version of PageRank [14] 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. [20] propose a personalized version of Kleinberg's HITS [42] 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 [122] 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 [97] 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 [51, 94, 107]; see also [37, 37, 58]. Similar ideas are beginning to influence web search such as approaches that harness the preferences of communities of users; see also [113, 114]. Sugiyama et al. [126] 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. [127] 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 [11] 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 users 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 in [73–75, 78–80].

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 [87–90]. Moreover, recent work by [68] 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 [68], 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 [1, 109] whereas remote approaches allow searchers to perform their searches at different locations across multiple devices [69,70,117]. 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 [109]. 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 [70, 114].

A good example of the co-located, synchronous approach to collaborative web search is given by the work in [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 (for example, 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 in [110] 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 [109], 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 an iPod touch device as their primary search/feedback device (although conventional PCs appear to be just as applicable). Interestingly, where the focus on 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 [70]. 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 in [83], 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 in [83] 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.

3.3 On Reputation and Recommendation

Collaborative information retrieval has highlighted the importance of the searcher in web search tasks and the potential for groups of searchers to collaborate (implicitly or explicitly) during complex search tasks. This perspective suggests that the *reputation* of a user may have an important role to play in guiding collaboration; it seems natural to give greater emphasis to the opinions and/or suggestions of more reputable users.

Recently there has been considerable interest in *reputation systems* to evaluate user reputation and inter-user trust across social web and e-commerce applications. For example, the reputation system used by eBay has been examined by Jøsang et al. [41] and Resnick et al. [95]. Briefly, eBay elicits feedback from buyers and sellers regarding their interactions with each other, and that information is aggregated in order to calculate user reputation scores. The aim is to reward good behaviour on the site and to improve robustness by leveraging reputation to predict whether a vendor will honour future transactions.

The work of O'Donovan and Smyth [76] considers the role of reputation in recommender systems. In this case, a standard collaborative filtering algorithm is modified to add a trust score to complement the normal profile or item-based similarity score, so that recommendation partners are chosen from those users that are not only similar to the target user, but who have also had a successful recommendation history with that user. It is posited that this trust information can be estimated by measuring the accuracy of a profile at making predictions over time, and using this approach the average prediction error is improved significantly in comparison with conventional collaborative filtering approaches.

Other research has examined reputation systems employed in social networking platforms. Lazzari [50] performed a case study of the professional social networking site Naymz. He warns that calculating reputation on a global level allows users who have interacted with only a small number of others to accrue a high degree of reputation, making the system vulnerable to malicious use. Similar to Jøsang et al. [41], Lazzari [50] suggests that vulnerability lies in the site itself, allowing malicious users to game the reputation system for their own ends. However, applying reputation globally affords malicious users influence over the entire system, which adds to its vulnerability.

In this chapter we consider the role of reputation models in a social search service in order to capture the *quality* of search knowledge that is contributed by users and how this reputation data can be leveraged to improve overall recommendation quality.

3.4 Towards Social Search

So far we have touched on separate strands of complementary research in the field of web search, recommender systems 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 undermined 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 this perspective has helped to identify opportunities to bring these two different strands of research together. In what follows we will describe two complementary case-studies that describe the evolution of one particular approach to making conventional web search more collaborative and personal.

To begin with we will describe the HeyStaks social search system [115, 120], which adds a layer of community-based collaboration atop conventional search services such as Google or Bing. HeyStaks is an example of a remote, asynchronous form of collaborative web search and we will summarize the results of recent live-user studies to highlight its potential end-user benefits. The second case-study will introduce the notion of *reputation* as a novel relevance signal that can further improve the quality of the HeyStaks suggestions, by weighting the influence of other searchers differently depending on their past search successes.

4 Case-Study 1: HeyStaks, A Social Search Utility

We describe a model of collaborative web search as implemented in a system called *HeyStaks*, which is novel in two important ways. First of all, HeyStaks adopts a 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, curate and share repositories of search experiences as opposed to coordinating the participation of search communities. Secondly, HeyStaks is tightly coupled to a mainstream search engine, such as Google, through a browser toolbar, which provides the collaborative search engine with the ability to capture and guide search activities. This means that users can enjoy the benefits of collaborative search while continuing to use their favourite search engine. 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.

4.1 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 shown in Fig. 2, 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, as a motivating example, the scenario of a group of friends planning a trip (to Canada, in this case). They know that during the course of their trip research they will use web search as their primary source of information about what to do and where to visit. So, one of the group creates a stak called "Canada Trip" and shares it with the other travellers, encouraging them to use this stak for all of their Canada-related searches regarding the trip.

Fig. 3 shows one of the group searching for information related to "visa Canada"; the "Canada Trip" stak has been automatically suggested as their search context in the HeyStaks Toolbar based on their query. In addition to the expected Google search results, they also see a number of pages that have been recommended from

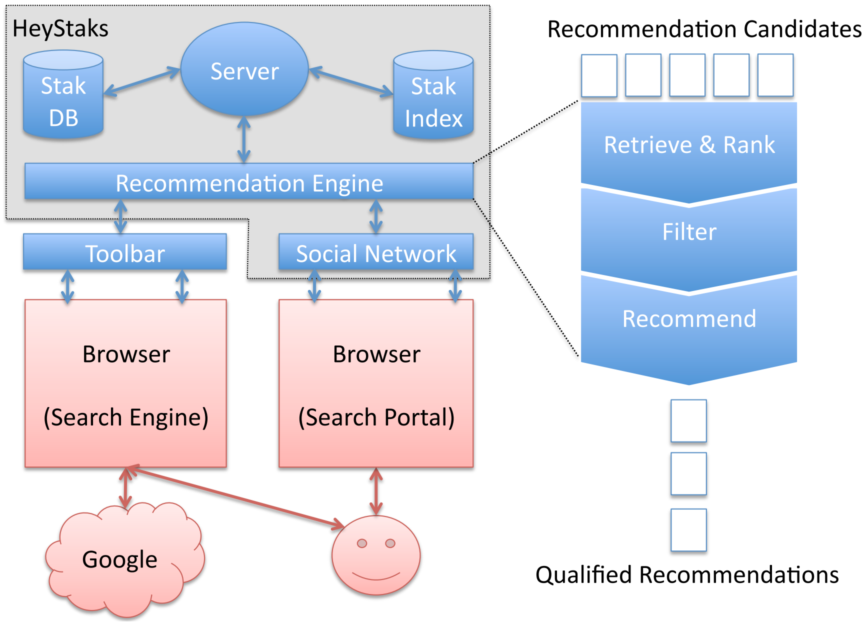


Fig. 2 The HeyStaks system architecture and outline recommendation model.

this suggested stak. These are results that other travellers have found to be useful on their searches for related queries. These recommendations may have been previously selected or tagged or otherwise shared during recent searchers by group members. 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.

4.2 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 ($S = \{p_1, ..., p_k\}$) 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 ($q_1, ..., q_n$) 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 ($share$) as indicated by Eq. 1.

$$p_i^S = \{q_1, ..., q_n, t_1, ..., t_m, v^+, v^-, sel, tag, share\} \tag{1}$$

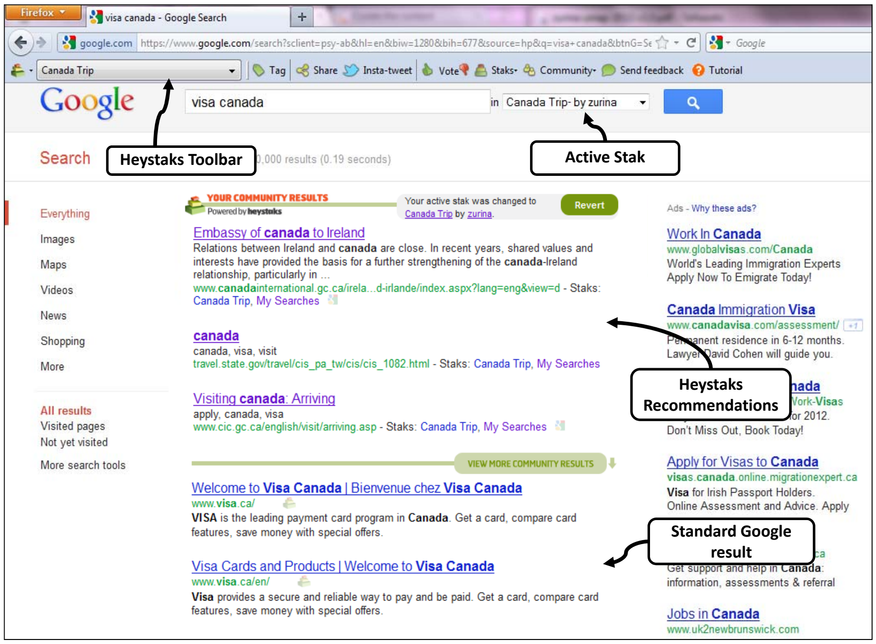


Fig. 3 Google search results with HeyStaks promotions.

In this way, each page is associated with *term data* (query terms and/or tag terms) and *usage data* (the selection, tag, share, and voting count). The term data is represented as a Lucene[2] 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 Eq. 2, which serves as the basis for an initial recommendation ranking.

[2] http://lucene.apache.org.

$$rel(q_t, p) = \sum_{t \in q_t} tf(t \in p) \times idf(t)^2 \tag{2}$$

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 [19, 98–101, 119]. 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 chapter 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 [116, 118]

4.3 Evaluation

To gain an understanding of both how users are using HeyStaks, and whether they seem to be benefiting from its search promotions, we consider a subset of 95 HeyStaks users who remained active during the course of the early beta release of the toolbar and service. These users registered with HeyStaks during the period October to December 2008 and the results below represent a summary of their usage during the period October 2008 to January 2009. Because this is a study of live-users *in the wild* there are certain limitations about what we can measure. There is no control group, for example, and it has not been 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, other studies do report on this type of analysis in more conventional control-group laboratory studies [9, 25, 114, 120].

Key to the HeyStaks proposition is that searchers need a better way to organise and share their search experiences, as opposed to the largely ad-hoc and very manually mechanisms (email, word of mouth, face-to-face collaboration) that are currently the norm. HeyStaks provides these features but do users actually take the time to create staks as a gateway to better search collaboration? Do they share them or join those created by others?

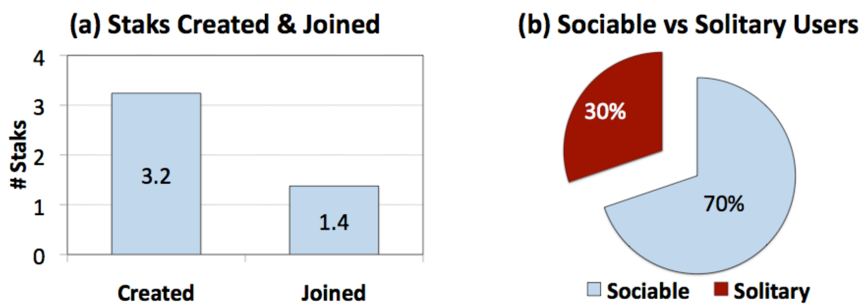


Fig. 4 (a) The average number of staks created and joined per user. (b) The percentage of *sociable* and *solitary* users.

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 shown in Fig. 4, 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). 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. 4(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 (for example, email, IM, etc.) with which to partially, albeit inefficiently, share their search experiences; see for example [68]. 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 7.5 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.

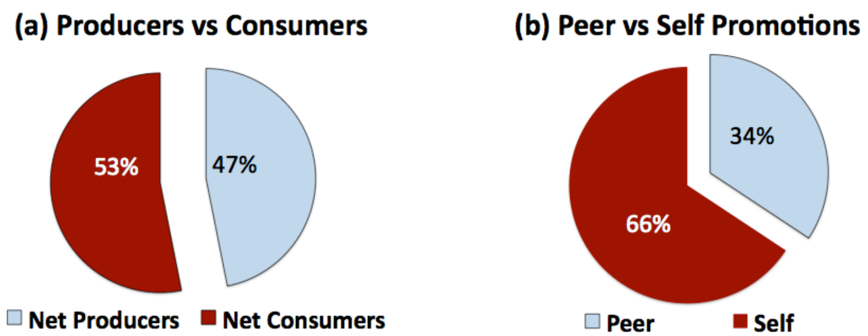


Fig. 5 (a) Net producers vs. consumers. (b) Promotion sources (self vs. peer).

Another matter we might consider 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 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. 5(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 [55,72]. Thus Fig. 5(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 affected 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.

5 Case-Study 2: A Reputation Model for Social Search

As described previously, the many and varied different types of activities that a HeyStaks user can perform (click-throughs, tagging, voting, sharing) on a web page are ultimately combined and leveraged by HeyStaks to make recommendations at search time. While the recommendation algorithm used differentially weights different activity types (so that tagging, for example, is considered a more reliable indicator of interest that a simple result click-through), the source of the activity (the user performing the activity) is not considered explicitly. Intuitively, we might expect that some users are more experienced searchers than others and, as such, perhaps their activities should be considered as more reliable at recommendation time. In other words promotion candidates that come from the activities of very experienced users might be considered ahead of candidates that come from the activity of less experienced users. This is particularly important given the potential for malicious users to disrupt stak quality by introducing dubious results to a stak; see also [78, 79] for related matters.

In this case-study we describe how user activities in HeyStaks can be harnessed to generate a computational model of searcher reputation, based on the collaboration events that naturally occur between HeyStaks users as they share their search experiences. We describe an algorithm for maintaining an up-to-date reputation model at search time and go on to propose a mechanism for incorporating reputation into the HeyStaks result recommendation subsystem.

5.1 From Activities to Reputation

It seems natural that the reputation of searchers should be linked to the search knowledge that they contribute. In simple terms this search knowledge is based on the creation and sharing of search staks and, ultimately, the web pages that are added to these staks according to a variety of different types of user activities (selections, voting, sharing, tagging). Each of these activities results in the creation of new search knowledge. If the target page is not present in a stak, then its selection, sharing, voting, or tagging will cause it to be added to the stak for the first time. If the page is already present, as a result of an earlier activity, then the page's stak record will be updated to reflect the additional activity.

What then is the relationship between search activity and searcher reputation? Under the heading of "more search knowledge is better than less search knowledge", it might make sense to model reputation as a direct function of the volume of activity that a given searcher has engaged in. This would be a mistake. For a start, just because a user is creating a lot of search knowledge, by adding many pages to many staks, it does not mean that this new knowledge is useful, especially to others. On the contrary, and as already mentioned, one of the major concerns in any social recommender is the potential for misuse through the actions of malicious users, a problem that would no doubt be exacerbated by valuing the contribution of very 'productive' malicious users.

Ultimately, in a social context, reputation is a form of *incentive*. It allows HeyStaks to capture and encode the value of user contributions [84, 86]. This is related to the concept of trust in recommender systems and social networks [48, 77] where, for example, the accumulation of trust scores can motivate users to enhance the quantity and quality of their contributions. But like any incentive, reputation can be *gamed* and so it is important that the incentive is correctly aligned with the sort of behaviour that benefits the system and its users as a whole. A reputation model that is the sum of all user activities does not meet this requirement since it is not necessarily to anyone's benefit to create a system that is measured simply by the volume of its search knowledge.

5.2 Reputation as Collaboration

Thus, our model of reputation must recognise the *quality* of *shared* search knowledge. To do this the key idea is that the quality of shared search knowledge can be estimated by looking at the frequency of *search collaborations* within HeyStaks. If HeyStaks recommends a result to a searcher, and the searcher chooses to act on (select, tag, vote on or share) this result, then we can view this as an instance of search collaboration (a *collaboration event*). The current searcher — the one who chooses to act on the recommendation — is known as the *consumer* and, in the simplest case, the original searcher — the one whose earlier activity caused this result to be added to the stak — is known as the *producer*. In other words, the producer creates search knowledge by adding the page to a stak, while the consumer consumes this knowledge by acting on the page when it is recommended.

The basic idea behind our reputation model is that this *implicit* collaboration between producer and consumer confers a *unit of reputation* on the producer (Fig. 6); incidentally, it is implicit because neither the producer nor the consumer are consciously or actively collaborating, rather the collaboration is a side-effect of recommendation, but a side-effect that creates a connection between the producer and consumer. If a given user is a regular producer of search knowledge (pages that are frequently recommended to, and acted on, by many other users) then this producer should enjoy a high reputation score. Moreover, if users create lots of staks and share these staks with many other users, or simply join staks that have been created by others, then they create an opportunity for more collaboration events to occur; and if users contribute good search knowledge to shared staks then their reputation score will benefit from the realisation of these frequent collaboration opportunities. In this way, this collaboration-based model of reputation is incentivizing users not just to create search knowledge but also to share it with others.

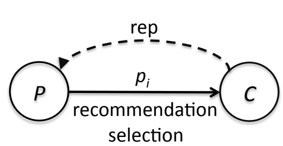


Fig. 6 Producer (P) and consumer (C) collaboration: C selects page p_i, which has been recommended to C based on P's previous activity. In turn, C confers reputation on P.

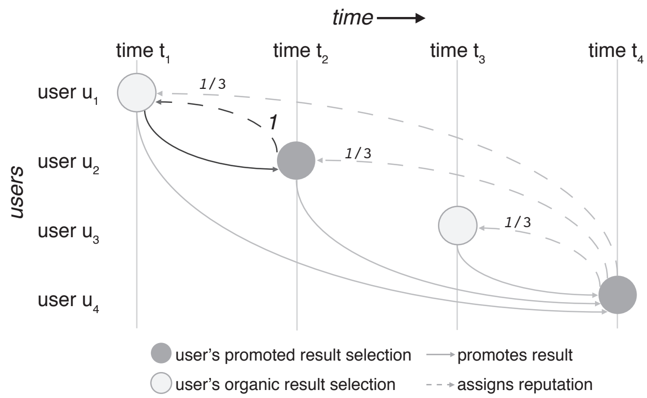


Fig. 7 Simple example of user reputation calculations in HeyStaks.

In reality the conferral of reputation by a consumer on a producer is more complicated than just described. In the general case, when a consumer acts on the promoted result, there may be many different relevant producers. One producer will have been the first to act on the result in question, causing it to be added to a stak, but subsequent users may have (re)selected it for similar or different queries, or they may have voted on it or tagged it or shared it with others independently of its other producers. These other users are also producers in the sense that their actions will be considered at recommendation time. In this case we should share the unit of reputation between these multiple producers. We propose to do this using a simple model so that if, at time t, a consumer acts on a promoted result, then the reputation score of each of its k producers is incremented by $1/k$; that is, a single unit of reputation is shared equally among the producers. It is worth noting that although this approach shares out reputation equally at any given collaboration event, because producers accumulate over time, it will naturally be the case that early producers will tend to enjoy greater reputation if the result in question features in multiple collaboration events over an extended time period.

5.3 An Example

To illustrate our user reputation model, consider the simple scenario as depicted in Fig. 7. Here, the activity of four users, $\{u_1,\ldots,u_4\}$, with respect to a single search result-page r, is shown at four points in time t_i, where $t_4 > t_3 > t_2 > t_1$. Further, assume that all four users are members of a particular stak S, which is currently the active stak for each of these users. The sequence of events at each time step t_i is as follows:

t_1: User u_1 organically selects result r for some search query q, causing result r to be added to stak S.

t_2: User u_2 selects result r, which has been promoted by HeyStaks, for a search query that is related to q. Since user u_1 is the only user to have previously selected result r in stak S, we say that user u_1 (the *producer*) has *promoted* result r to user u_2 (the *consumer*). Consequently, user u_2 assigns a *reputation* score of 1 to user u_1.

t_3: User u_3 organically selects result r for an unrelated search query q'. This time, result r is not promoted by HeyStaks and hence no reputation is assigned by user u_3 to any of the other users.

t_4: Finally, user u_4 selects result r, which has been promoted by HeyStaks, for a search query that is again related to q. Since users u_1, u_2 and u_3 have all previously selected (either organically or by promotion) result r, on this occasion reputation is assigned by user u_4 to each of these users. Thus, in Fig. 7, the reputation score is distributed equally among the three users, such that each user receives a score of $1/3$.

At the end of the time period, overall user reputation can be calculated, for example, by simply summing the individual reputation scores that each user has received. For example, in the above scenario, the overall reputation scores for users u_1, u_2, u_3 and u_4 are $4/3$, $1/3$, $1/3$ and 0, respectively.

5.4 Graph-Based Reputation Models

In fact we can treat the collaborations that occur among users as a type of graph, a *collaboration graph*. Each node represents a unique user and the edges represent collaborations events between pairs of users. These edges are directed to reflect the producer/consumer relationships and reputation flows along these edges, and is aggregated at the nodes. In the above example, user reputation was calculated as a simple weighted sum of collaboration events but we could so also consider other types of aggregation approaches. Below we formalise this model and also describe an alternative based on PageRank [12].

5.4.1 Reputation as a Weighted Sum of Collaboration Events

As previously described, according to this aggregation approach, producer reputation is calculated as a sum of the collaboration events in which they have participated. Consider the selection of result r by consumer c at time t. The producers responsible for this result recommendation are given by $producers(r,t)$ (Eq. 3) such that each p_i denotes a specific user u_i in a specific stak S_j.

$$producers(r,t) = \{p_1, ..., p_k\} \ . \tag{3}$$

Then, for each producer of r, p_i, we update its reputation according to Eq. 4. In this way reputation is shared equally among its k contributing producers.

$$rep(p_i,t) = rep(p_i,t-1) + 1/k \; . \tag{4}$$

As users participate in more and more collaboration events, their reputation grows over time. See [61] for further details on this approach.

5.4.2 Reputation as PageRank

PageRank [12] can also be applied to compute the reputation of HeyStaks users, which take the place of web pages in the collaboration graph. When a collaboration event occurs, directed links are inserted from the consumer to each producer. Once all collaboration events up to some point in time, t, have been captured, the reputation of each user p_i at time t is given by:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{|L(p_j|)} \; , \tag{5}$$

where d is a damping factor, N is the number of users, $M(p_i)$ is the set of inlinks (from consumers) to (producer) p_i and $L(p_j)$ is the the set of outlinks from p_j (i.e. the other users from whom p_j has consumed results).

5.5 From User Reputation to Result Promotion

In the previous case-study the standard HeyStaks recommendation engine scores each recommendation candidate based on how relevant it is to the target query ($rel(q_t,r)$ as per Eq. 2, but here with p replaced by r to avoid confusion between pages and producers). If reputation is to influence recommendation ranking, as well as relevance, then we need to transform the above user-based reputation measures into a page-based reputation measure, which can be incorporated into recommendation. Recommendation candidates can then be ranked according to a weighted score of *relevance* ($rel(q_t,r)$) and *reputation* ($rep(r,t)$) by Eq. 6, where w is used to adjust the relative influence of relevance and reputation.

$$score(r,q_t) = w \times rep(r,t) + (1-w) \times rel(q_t,r) \; . \tag{6}$$

Eq. 6 describes one simple approach to combining result reputation and relevance at recommendation time and we now consider two ways to transform user reputation into a page reputation score; as mentioned above, here we use r to refer to a result-page instead of p since the latter is more conveniently associated with a producer. In what follows then we describe two alternative approaches to transferring reputation from producers to pages for the purpose of recommendation.

5.5.1 Max Reputation

The first page reputation score calculates the reputation of a result-page r (at time t) as the maximum reputation of its associated producers, $\{p_1, \ldots, p_k\}$; see Eq. 7. Scoring results in this way provides the advantage that the reputation of a page will not be prematurely increased if, for example, many new, but not yet reputable, users have selected the page.

$$rep(r,t) = \max_{\forall p_i \in \{p_1,\ldots,p_k\}} \big(rep(p_i,t)\big) \, . \tag{7}$$

5.5.2 Hooper's Reputation

Hooper's *Rule for Concurrent Testimony* was proposed to calculate the credibility of human testimony [105]. Hooper gives to a report a credibility of $1-(1-c)^k$, assuming k reporters, each with a credibility of c ($0 \leq c \leq 1$). For HeyStaks, result reputation can be determined by performing a similar calculation across the reputation scores of its producers as in Eq. 8.

$$rep(r,t) = 1 - \prod_{i=1}^{k}(1 - rep(p_i,t)) \, . \tag{8}$$

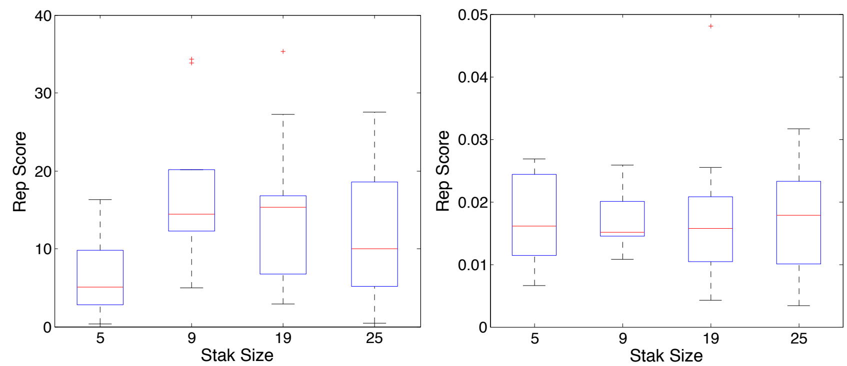
5.6 Evaluation

The key hypothesis in this case-study has been that by allowing reputation, as well as relevance, to influence the ranking of result recommendations, we can improve the overall quality of search results. In this section we evaluate our reputation models using data generated during a closed, live-user trial of HeyStaks, designed to evaluate the utility of the HeyStaks approach to collaborative web search in fact-finding information discovery tasks.

5.6.1 Dataset and Methodology

Our live-user trial involved 64 first-year undergraduate university students with varying degrees of search expertise; see [63]. Users were asked to participate in a general knowledge quiz, during a supervised laboratory session, answering as many questions as they could from a set of 20 questions in the space of 1 hour. Each student received the same set of questions which were randomly presented to avoid any ordering bias. See [63] for a list of questions used in the trial.

Each user was allocated a desktop computer with the Firefox web browser and the HeyStaks toolbar pre-installed; they were permitted to use Google, enhanced



(a) WeightedSum (b) PageRank

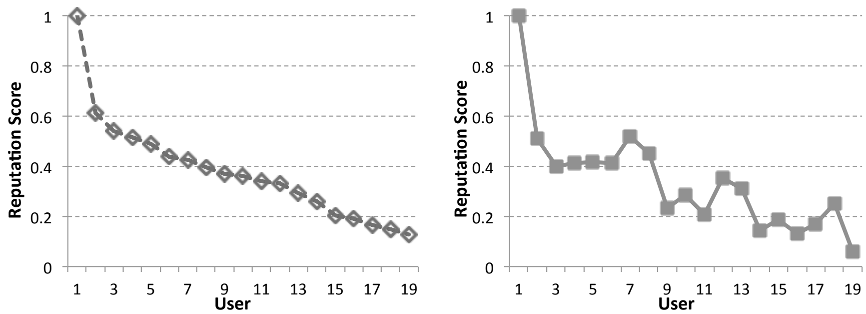
Fig. 8 User reputation scores: WeightedSum (a) and PageRank (b) reputation models.

by HeyStaks functionality, as an aid in the quiz. The 64 students were randomly divided into search groups. Each group was associated with a newly created search stak, which would act as a repository for the group's search knowledge. We created 6 *solitary* staks, each containing just a single user, and 4 *shared* staks containing 5, 9, 19, and 25 users. The solitary staks served as a benchmark to evaluate the search effectiveness of individual users in a non-collaborative search setting, whereas the different sizes of shared staks provided an opportunity to examine the effectiveness of collaborative search across a range of different group sizes. All activity on both Google search results and HeyStaks recommendations was logged, as well as all queries submitted during the experiment. During the 60 minute trial, some 3,124 queries and 1,998 result activities (selections, tagging, voting, popouts) were logged, and 724 unique results were selected.

While the reputation model was not used during this original live-user trial — recommendations were ranked based on relevance only — the data produced does make it possible for us to *replay* the trial to construct reputation models and use them to re-rank the recommendations made by HeyStaks. We can then retrospectively test the quality of re-ranked results versus the original ranking against a ground-truth relevance. As part of the post-trial analysis, each selected result was manually classified as *relevant* (the result contained the answer to a question), *partially relevant* (the result referred to an answer, but not explicity), or *not-relevant* (the result did not contain an explicit or implicit reference to an answer) by experts.

5.6.2 User Reputation

To get a sense of how users were scored by the two reputation models described in Section 5.4, we now examine the type of user reputation values that are generated from the trial data. In Fig. 8, box-plots are shown for the reputation scores across the 4 shared staks and for each reputation model. Here we see that for the Weight-



(a) Weighted Sum (b) PageRank

Fig. 9 User reputation scores for members of the 19-person stak: WeightedSum (a) and PageRank (b) reputation models.

edSum model there is a clear difference in the median reputation score for members of the 5 person stak when compared to members of the larger staks. This is not evident in results for the PageRank model, which shows very similar reputation scores, regardless of stak size.

Fig. 9 shows the reputation scores (normalised by the maximum user reputation score for each model) that members of the 19-person stak had accumulated at the end of the trial. Users are ranked according to their WeightedSum score in descending order, and this ordering is maintained in both graphs. The long-tail distribution of reputation scores is representative of that found in the other staks, where a small number of users had accumulated high reputation scores and the remainder relatively low scores. Users with high reputation can be considered as *search leaders*, who are among the first to locate and add relevant results to staks, and whose search contributions are deemed to be particularly useful by other stak members. Further, the graphs indicate a strong correlation exists between reputation scores according to the WeightedSum and PageRank models (Spearman rank correlation of 0.91).

5.6.3 From Reputation to Quality

Of course the true test of the reputation models is the extent to which they improve the quality of results recommended by HeyStaks. We have described how user reputation can be combined with term-based relevance to generate recommendations; see Eq. 6. Accordingly, as mentioned above, we re-rank each of the (relevance-based) recommendation lists produced during the trial using the item reputation models, based on the user reputation scores calculated at the appropriate point in time. In what follows, four combinations of user and result reputation models are considered to re-rank recommendation lists: WeightedSum or PageRank combined with the Max or Hooper models. Since we have ground-truth relevance information for all of the recommendations (relative to the quiz questions), we can determine the

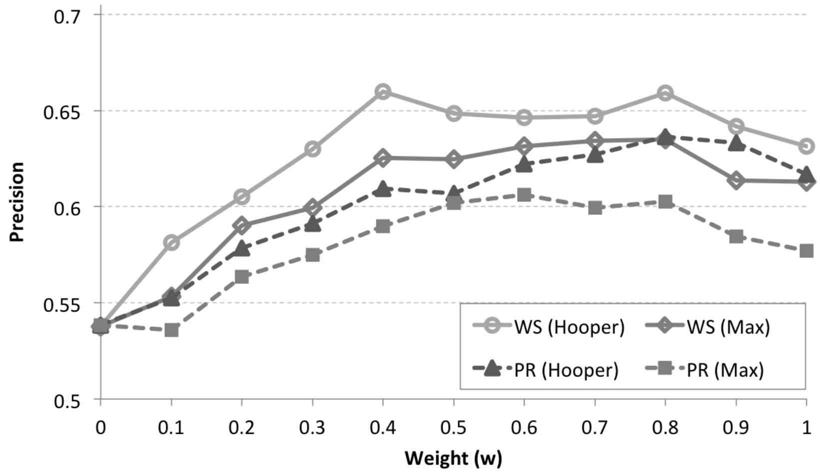


Fig. 10 Precision for different combinations of relevance and reputation.

quality of the new recommendations rankings. Specifically, for each combination of user and item reputation model, we count how often the top-ranked recommendation is relevant for each query and then compute a precision metric by dividing this count by the total number of queries considered. Thus, precision returns a value between 0 and 1 and a precision of 0.5, for example, means that 50% of top-ranked results over all queries are relevant for a given condition.

Fig. 10 shows precision versus the weighting (w) used in Eq. 6 to adjust the influence of term-based relevance versus reputation during recommendation. The results for each combination of user and result reputation model show an increase in precision when compared to recommendations based on relevance ranking only; at $w = 0$, reputation is not an influencing factor and in all cases the precision is 0.54. As the influence of reputation over relevance during recommendation is increased (by increasing w), an improvement in precision is seen up to values of w in the range 0.4–0.8. For example, at $w = 0.5$, the reputation models achieve a precision of 0.60–0.65 compared to 0.54 for the default HeyStaks relevance-only recommendations, a percentage improvement of 11–20%. In all cases, precision decreases as w approaches 1, indicating that the relevance information HeyStaks uses to rank recommendations is needed in order to optimally rank recommendations; i.e. reputation alone does not provide best performance.

The WeightedSum user reputation model, when paired with the Hooper result reputation model, is the best performing technique, peaking at $w = 0.4$ and $w = 0.8$, each time achieving a precision of approximately 0.66. Hooper's model also performed well when combined with PageRank, achieving a precision of 0.64 at $w = 0.8$. This leads us to believe that Hooper may be the most suitable option for result reputation. The score it produces for a result is a consensus based on the reputation of its producers. The model promotes the idea that a result will have a high score by way of reinforcement from its producers, assuming they are reputable; for further discussion and results see [61, 62, 64].

6 Conclusions

Mainstream search engines are evolving 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 role in their evolution going forward. For example, researchers are exploring how to make search engines more responsive to the particular needs and preferences of individuals, and how to introduce greater opportunity for collaboration into the standard search model.

In the original version of this chapter, published as part of the first edition of this volume, we predicted that mainstream search engines would likely evolve to accommodate many elements of the personalization and collaboration ideas surveyed and presented. This prediction has come to pass, at least in part. Mainstream search engines such as Google and Bing are increasingly personalizing their results based on the searcher and her context. In some cases, mainstream search engines have also begun to incorporate social signals into their ranking engines. For example, Bing has partnered with social media services such as Twitter and Facebook to incorporate content and signals from these networks during result selection and ranking, while Google now emphasises content from its own social network (Google+) in similar ways. Indeed Google prioritises content from verified Google+ users, lending a form of reputation to its rankings.

Where today the burden of web search is still very much on the individual searcher, we believe that the introduction of recommendation technologies will provide search engines with the opportunity to function more proactively as they work to anticipate, rather than respond to, a user's information needs. For example, the Google Now service goes some way to realising this by making suggestions to users based on various signals that are relevant to their needs and context. 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 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: through the CLARITY Centre for Sensor Web Technologies under grant number 07/CE/I1147; and through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289.

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. Florence, Italy.
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 Proceedings of the Annual SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 3–11, 1997. Chia Laguna, Sardinia.
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. Heraklion, Greece.
4. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
5. M. Balabanovic and Y. Shoham. FAB: Content-Based Collaborative Recommender. *Communications of the ACM*, 40(3):66–72, 1997.
6. 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.
7. Fabiano Belém, Eder Ferreira Martins, Jussara M. Almeida, and Marcos André Gonçalves. Exploiting novelty and diversity in tag recommendation. In *ECIR*, pages 380–391, 2013.
8. Daniel Billsus, Michael J. Pazzani, and James Chen. A learning agent for wireless news access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI)*, pages 33–36, 2000. New Orleans, Louisiana, United States.
9. 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.
10. Derek Bridge and John Paul Kelly. Diversity-enhanced conversational collaborative recommendations. In N. Creaney, editor, *Procs. of the Sixteenth Irish Conference on Artificial Intelligence and Cognitive Science*, pages 29–38, 2005.
11. 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.
12. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International Conference on World Wide Web (WWW '98)*, pages 107–117, Brisbane, Australia, 1998. ACM.
13. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.
14. 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.
15. Jay Budzik and Kristian J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI)*, pages 44–51, 2000. New Orleans, Louisiana, United States.
16. Robin Burke. The Wasabi Personal Shopper: A Case-Based Recommender System. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 1999. Orlando, Florida, USA.
17. 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.
18. 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.

19. Pierre-Antoine Champin, Peter Briggs, Maurice Coyle, and Barry Smyth. Coping with Noisy Search Experiences. In *29th SGAI International Conference on Artificial Intelligence (AI)*, pages 5–18, 2009. Cambridge, UK.
20. Huan Chang, David Cohn, and Andrew McCallum. Learning to Create Customized Authority Lists. In *ICML '00: Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 127–134, 2000.
21. Li Chen and Pearl Pu. Evaluating Critiquing-based Recommender Agents. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006. Boston, Massachusetts.
22. Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. Using ODP Metadata to Personalize Search. In *Proceedings of the 28th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 178–185, 2005. Salvador, Brazil.
23. Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. PROS: A Personalized Ranking Platform for Web Search. In *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH)*, pages 34–43, 2004. Eindhoven, The Netherlands.
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. Rome, Italy.
25. Maurice Coyle and Barry Smyth. Supporting Intelligent Web Search, volume = 7, year = 2007. *ACM Trans. Internet Techn.*, (4).
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 *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 348–355, 1997. Las Vegas, Nevada, United States.
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, url = citeseer.ist.psu.edu/finkelstein01placing.html, year = 2001, bdsk-url-1 = citeseer.ist.psu.edu/finkelstein01placing.html. In *Proceedings of the 10th International Conference on the World Wide Web (WWW)*, pages 406–414. Hong Kong.
32. C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the 3rd ACM Conference on Digital Libraries (DL)*, pages 89–98, 1998. Pittsburgh, Pennsylvania, United States.
33. Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-Tracking Analysis of User Behavior in WWW Search. In *Proceedings of the 27th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 478–479, 2004. Sheffield, United Kingdom.
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. Ahmed Hassan and Ryen W. White. Personalized models of search satisfaction. In *CIKM*, pages 2009–2018, 2013.
36. Neil Hurley and Mi Zhang. Novelty and diversity in top-n recommendation - analysis and evaluation. *ACM Trans. Internet Techn.*, 10(4):14, 2011.

37. 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.
38. Bernard J. Jansen and Amanda Spink. An Analysis of Web Searching by European AlltheWeb.com Users. *Inf. Process. Manage.*, 41(2):361–381, 2005.
39. 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.
40. Glen Jeh and Jennifer Widom. Scaling Personalized Web Search. In *Proceedings of the 12th International conference on World Wide Web (WWW)*, pages 271–279. ACM, 2003. Budapest, Hungary.
41. Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.
42. Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *J. ACM*, 46(5):604–632, 1999.
43. Jon M. Kleinberg. Hubs, Authorities, and Communities. *ACM Comput. Surv.*, 31(4):5, 1999.
44. 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.
45. Yehuda Koren. Tutorial on Recent Progress in Collaborative Filtering. In *Proceedings of the International Conference on Recommender Systems (RecSys)*, pages 333–334, 2008.
46. 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. Edinburgh, UK.
47. 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 *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)*, pages 272–281, New York, NY, USA, 2000. McLean, Virginia, United States.
48. Ugar Kuter and Jennifer Golbeck. SUNNY: A new algorithm for trust inference in social networks, using probabilistic confidence models. In *AAAI*, pages 1377–1382, 2007.
49. S. Lawrence and C. Lee Giles. Accessibility of Information on the Web. *Nature*, 400(6740):107–109, 1999.
50. Marco Lazzari. An experiment on the weakness of reputation algorithms used in professional social networks: The case of naymz. In *Proceedings of the IADIS International Conference e-Society 2010*, pages 519–522, Freiburg, Germany, 2010. IADIS.
51. Greg Linden, Brent Smith, and Jeremy York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Distributed Systems Online*, 4(1), 2003.
52. Fang Liu, Clement Yu, and Weiyi Meng. Personalized Web Search by Mapping User Queries to Categories, year = 2002, bdsk-url-1 = http://doi.acm.org/10.1145/584792.584884. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM)*, pages 558–565, New York, NY, USA. ACM Press. McLean, Virginia, USA.
53. Zhongming Ma, Gautam Pant, and Olivia R. Liu Sheng. Interest-Based Personalized Search. *ACM Trans. Inf. Syst.*, 25(1):5, 2007.
54. Christos Makris, Yannis Panagis, Evangelos Sakkopoulos, and Athanasios Tsakalidis. Category Ranking for Personalized Search. *Data Knowl. Eng.*, 60(1):109–125, 2007.
55. Gary Marchionini. Exploratory search: From Finding to Understanding. *Communications of the ACM*, 49(4):41–46, 2006.
56. Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. An analysis of critique diversity in case-based recommendation. In *Proceedings of the International FLAIRS Conference*, pages 123–128, 2005.
57. 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.
58. 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.

59. 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.
60. L. McGinty and B. Smyth. On The Role of Diversity in Conversational Recommender Systems. In Kevin D. Ashley and Derek G. Bridge, editors, *Proceedings of the 5th International Conference on Case-Based Reasoning*, pages 276–290. Springer-Verlag, 2003.
61. Kevin McNally, Michael P. O'Mahony, Maurice Coyle, Peter Briggs, and Barry Smyth. A case study of collaboration and reputation in social web search. *ACM TIST*, 3(1):4, 2011.
62. Kevin McNally, Michael P. O'Mahony, and Barry Smyth. A model of collaboration-based reputation for the social web. In *ICWSM*, 2013.
63. Kevin McNally, Michael P. O'Mahony, Barry Smyth, Maurice Coyle, and Peter Briggs. Social and collaborative web search: An evaluation study. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '11)*, pages 387–390, Palo Alto, California USA, 2011. ACM.
64. Kevin McNally, Michael P. O'Mahony, Barry Smyth, Maurice Coyle, and Peter Briggs. Social and collaborative web search: an evaluation study. In *IUI*, pages 387–390, 2011.
65. 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.
66. 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.
67. M. Mitra, A. Singhal, and C. Buckley. Improving Automatic Query Expansion. In *Proceedings of ACM SIGIR*, pages 206–214. ACM Press, 1998.
68. 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.
69. Meredith Ringel Morris and Eric Horvitz. SearchTogether: An Interface for Collaborative Web Search. In *UIST*, pages 3–12, 2007.
70. Meredith Ringel Morris and Eric Horvitz. S^3: Storable, Shareable Search. In *INTERACT (1)*, pages 120–123, 2007.
71. 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.
72. Vicki L. O'Day and Robin Jeffries. Orienteering in an Information Landscape: How Information Seekers Get from Here to There. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 438–445, New York, NY, USA, 1993. ACM Press. Amsterdam, The Netherlands.
73. 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.
74. John O'Donovan and Barry Smyth. Eliciting Trust Values from Recommendation Errors. In *Proceedings of the International FLAIRS Conference*, pages 289–294, 2005.
75. 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.
76. John O'Donovan and Barry Smyth. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*, pages 167–174, San Diego, California, USA, 2005. ACM.
77. John O'Donovan and Barry Smyth. Trust in recommender systems. In *IUI*, pages 167–174, 2005.
78. 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.

79. John O'Donovan and Barry Smyth. Is Trust Robust?: An Analysis of Trust-Based Recommendation. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 101–108, 2006.
80. John O'Donovan and Barry Smyth. Mining trust values from recommendation errors. *International Journal on Artificial Intelligence Tools*, 15(6):945–962, 2006.
81. Eli. Pariser. *The Filter Bubble : What the Internet is Hiding from You, year = 2011*. Penguin Press.
82. Daniel Billsus Michael Pazzani. A Hybrid User Model for News Story Classification. In *Proceedings of the 7th International Conference on User Modeling (UM)*, 1999. Banff, Canada.
83. Jeremy Pickens, Gene Golovchinsky, Chirag Shah, Pernilla Qvarfordt, and Maribeth Back. Algorithmic Mediation for Collaborative Exploratory Search. In *Proceedings of the International Conference on Information retrieval Research and Development (SIGIR)*, pages 315–322, 2008.
84. Jennifer Preece and Ben Shneiderman. The reader to leader framework: Motivating technology-mediated social participation. *AIS Trans. on Human-Computer Interaction*, 1(1):13–32, 2009.
85. Alexander Pretschner and Susan Gauch. Ontology Based Personalized Search. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, page 391, Washington, DC, USA, 1999. IEEE Computer Society.
86. Al Mamunur Rashid, Kimberly Ling, Regina D. Tassone, Paul Resnick, Robert Kraut, and John Riedl. Motivating participation by displaying the value of contribution. In *CHI*, pages 955–958, 2006.
87. 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.
88. Madhu C. Reddy, Paul Dourish, and Wanda Pratt. Coordinating Heterogeneous Work: Information and Representation in Medical Care. In *CSCW*, pages 239–258, 2001.
89. 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.
90. 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.
91. 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.
92. James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Incremental critiquing. *Knowl.-Based Syst.*, 18(4-5):143–151, 2005.
93. 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.
94. Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, 1997.
95. Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Advances in Applied Microeconomics*, 11:127–157, 2002.
96. 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.
97. 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.
98. Zurina Saaya, Rachael Rafter, Markus Schaal, and Barry Smyth. The curated web: a recommendation challenge. In *RecSys*, pages 101–104, 2013.

99. Zurina Saaya, Markus Schaal, Maurice Coyle, Peter Briggs, and Barry Smyth. A comparison of machine learning techniques for recommending search experiences in social search. In *SGAI Conf.*, pages 195–200, 2012.
100. Zurina Saaya, Markus Schaal, Maurice Coyle, Peter Briggs, and Barry Smyth. Exploiting extended search sessions for recommending search experiences in the social web. In *ICCBR*, pages 369–383, 2012.
101. Zurina Saaya, Markus Schaal, Rachael Rafter, and Barry Smyth. Recommending topics for web curation. In *UMAP*, pages 242–253, 2013.
102. 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.
103. 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.
104. 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. Denver, Colorado, United States.
105. G. Shafer. The combination of evidence. *International Journal of Intelligent Systems*, 1(3):155–179, 1986.
106. 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.
107. 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.
108. 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. Bremen, Germany.
109. Alan F. Smeaton, Colum Foley, Daragh Byrne, and Gareth J. F. Jones. ibingo mobile collaborative search. In *CIVR*, pages 547–548, 2008.
110. Alan F. Smeaton, Hyowon Lee, Colum Foley, and Sinéad McGivney. Collaborative video searching on a tabletop. *Multimedia Syst.*, 12(4-5):375–391, 2007.
111. B. Smyth and P. McClave. Similarity v's Diversity. In David Aha and Ian Watson, editors, *Proceedings of the 3rd International Conference on Case-Based Reasoning*, pages 347–361. Springer, 2001.
112. Barry Smyth. Case-Based Recommendation. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, pages 342–376. Springer-Verlag, 2007.
113. Barry Smyth. A community-based approach to personalizing web search. *IEEE Computer*, 40(8):42–50, 2007.
114. 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.
115. Barry Smyth, Peter Briggs, Maurice Coyle, and Michael P. O'Mahony. A case-based perspective on social web search. In *ICCBR '09: Proceedings of the 8th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, pages 494–508, Seattle, Washington, USA, 2009. Springer-Verlag.
116. 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.
117. Barry Smyth, Peter Briggs, Maurice Coyle, and Michael P. O'Mahony. Google shared. A case study in social search. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization (UMAP '09)*, pages 283–294, Trento, Italy, June 2009. Springer-Verlag.

118. 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.
119. Barry Smyth and Pierre-Antoine Champin. The Experience Web: A Case-Based Reasoning Perspective. In *Workshop on Grand Challenges for Reasoning from Experiences at the International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.
120. Barry Smyth, Maurice Coyle, and Peter Briggs. Heystaks: a real-world deployment of social search. In *RecSys*, pages 289–292, 2012.
121. 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.
122. 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.
123. 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.
124. Amanda Spink and Bernard J. Jansen. A Study of Web Search Trends. *Webology*, 1(2):4, 2004.
125. 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.
126. 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. New York, NY, USA.
127. 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. Chiba, Japan.
128. 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. Philadelphia, PA, USA.
129. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.
130. 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.
131. 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.