# Recommendation as Collaboration in Web Search

*Barry Smyth, Jill Freyne, Peter Briggs, and Maurice Coyle*

■ *Recommender systems now play an important role in online information discovery, complementing traditional approaches such as search and navigation with a more proactive approach to discovery that is informed by the users' interests and preferences. To date recommender systems have been deployed within a variety of e-commerce domains, covering a range of products such as books, music, and movies, and have proven to be a successful way to convert browsers into buyers. Recommendation technologies have a potentially much greater role to play in information discovery, however, and in this article we consider recent research that takes a fresh look at web search as a fertile platform for recommender systems research as users demand a new generation of search engines that are less susceptible to manipulation and more responsive to searcher needs and preferences.*

## The Problem with Web Search

The quantity and diversity of the information content that is now available online is without precedent, and its relentless growth is set to continue, if not accelerate, as the web continues to evolve to accommodate new forms of user-generated content (Gulli and Signorini 2005). The popularity of the social web further escalates this issue by facilitating easy contribution and sharing of information at a pace that places strain on search engines to locate and index such a fast-changing space. The sheer scale and increasing heterogeneity of the modern web make for further significant challenges when it comes to providing individual users and communities with access to the right information at the right time. Despite all of the recent developments in search engine technologies, modern search engines continue to struggle when it comes to providing users with fast and efficient access to information. For example, recent studies have highlighted how even today's leading search engines fail to satisfy 50 percent of user queries (Smyth et al. 2005). Part of the problem rests with the searchers themselves: with an average of only two to three terms per query (Lawrence and Giles 1998, Spink and Jansen 2004), the typical web search query is often vague with respect to the searcher's true intentions or information needs (Song et al. 2007). Moreover, searchers sometimes choose query terms that are not well represented in the page that they are seeking, and so simply increasing the length of queries will not necessarily improve search performance.

Problems with search queries aside, another challenge that faces all mainstream search engines is the ongoing battle with search engine spam through so-called black-hat search engine optimization (SEO) techniques and the rise of the content farms. The former involve the manipulation of search engine rankings in order to promote target pages. At the time of writ-

 ISSN 0738-4602

ing, the *New York Times* had reported how one big-brand retailer in the United States had apparently engaged in black-hat SEO techniques involving the paid procurement of inbound links in order to boost themselves in the search rankings for *head queries* (that is, popular queries) for ladies apparel. Content farms have taken SEO strategies to an entirely new level by funding the mass production of content in response to contemporary query trends and then boosting this content in mainstream search rankings through aggressive SEO techniques. The end result of these types of activities, for the searcher, is that we are increasingly faced with less-relevant, lower-quality research results that are boosted by services that seek to manipulate the ranking functions of mainstream search engines. And while search engines can, and do, frequently change their ranking metrics in response to aggressive SEO tactics, this is an arms race that cannot be easily won.

## Toward Social Search

Two important ideas in web search have emerged in recent times — *personalization* and *collaboration*. These approaches question the core assumptions of mainstream web search engines and suggest important adaptations to conventional web search techniques. The first assumption concerns the *one size fits all* nature of mainstream web search — two different users with the same query will, more or less, receive the very same result list, despite their different preferences — and argues that web search needs to become more *personalized* so that the implicit needs and preferences of searchers can be accommodated (Chang, Cohn, and McCallum 2000; Chirita, Olmedilla, and Nejdl 2004; Granka, Joachims, and Gay 2004; Speretta and Gauch 2005; Asnicar and Tasso 1997; Ma, Pant, and Sheng 2007; Makris et al. 2007; Chirita et al. 2005; Pretschner and Gauch 1999; Shen, Tan, and Zhai 2005; Finkelstein et al. 2001).

The second assumption, which we will primarily focus on in this article, concerns the *solitary nature* of web search. Traditionally most web-search activities involved isolated interactions between a web user and the online system. Recently, there has been considerable interest in the potential for web search to evolve to become a more *social* activity (Morris, Teevan, and Panovich 2010; Golovchinsky, Qvarfordt, and Pickens 2009; Evans, Kairam, and Pirolli 2009 and 2010), whereby the search efforts of a user might be influenced by the user's social graph or the searches of others, potentially leading to a more *collaborative* model of search. We have seen an emergence of previously nonsocial systems now exploiting social relationships for more effective and inclusive applications that harness explicit and implicit linkages between individuals. For example, Last.fm[1] allows users to listen to each other's music, and Flickr[2] allows friends to exchange digital photographs online and facilitates the collaborative indexing of images by encouraging users to submit index terms for images. Finally, Wikipedia[3] harnesses the wisdom of the crowd to provide a real-time, collaboratively built encyclopedia where individuals can directly learn through the knowledge of others.

This has led many commentators to look toward a new era of *social search* in an attempt to unify two distinctive information-discovery worlds: the traditional world of web search and the information-sharing world of social networks. Only a few years ago, by and large, the majority of people located information of interest through their favorite mainstream search engine. However, recently there has been a very noticeable change in how many web users satisfy their information needs. In addition to web search, many of us are finding relevant information online through our social networks. This is not only a matter of keeping up with the daily lives of our friends and colleagues but also an important way for individuals to locate highly targeted content that is relevant to their long- and short-term needs. For example, for many Twitter users, the service is as much about consumed Twitter content as it is about generating their own tweets, and many (up to 25 percent according to a recent survey)[4] of the tweets we read contain links to pages of interest. In this way, our Twitter network plays the role of a social filter that is capable of recommending highly relevant and targeted pages based on our needs and preferences, as reflected by the users we follow. Indeed recent statistics from Twitter claim that its users are explicitly searching tweet content 24 billion times per month[5] as compared to approximately 88 billion queries per month for Google, and less than 10 billion queries per month for Yahoo. Similarly, at the time of writing, Facebook's own statistics highlight how its users are sharing upwards of 30 billion items of content every month.[6] Many of these items of content would have previously been located through mainstream search engines. Instead, today, they are being accessed through our social networks (figure 1) and, in terms of raw volume of information-seeking activity, the social networks are now beginning to compete with mainstream search engines.

The concept of social search has become somewhat muddled in the world of the web. On the one hand there are those approaches that seek to extend search beyond the web of pages and into the world of social networks. In this case social search refers to the indexing and searching of our social content (for example, blog posts, tweets, Facebook status updates, Flickr photos, Quora questions, and others). This is largely a matter for

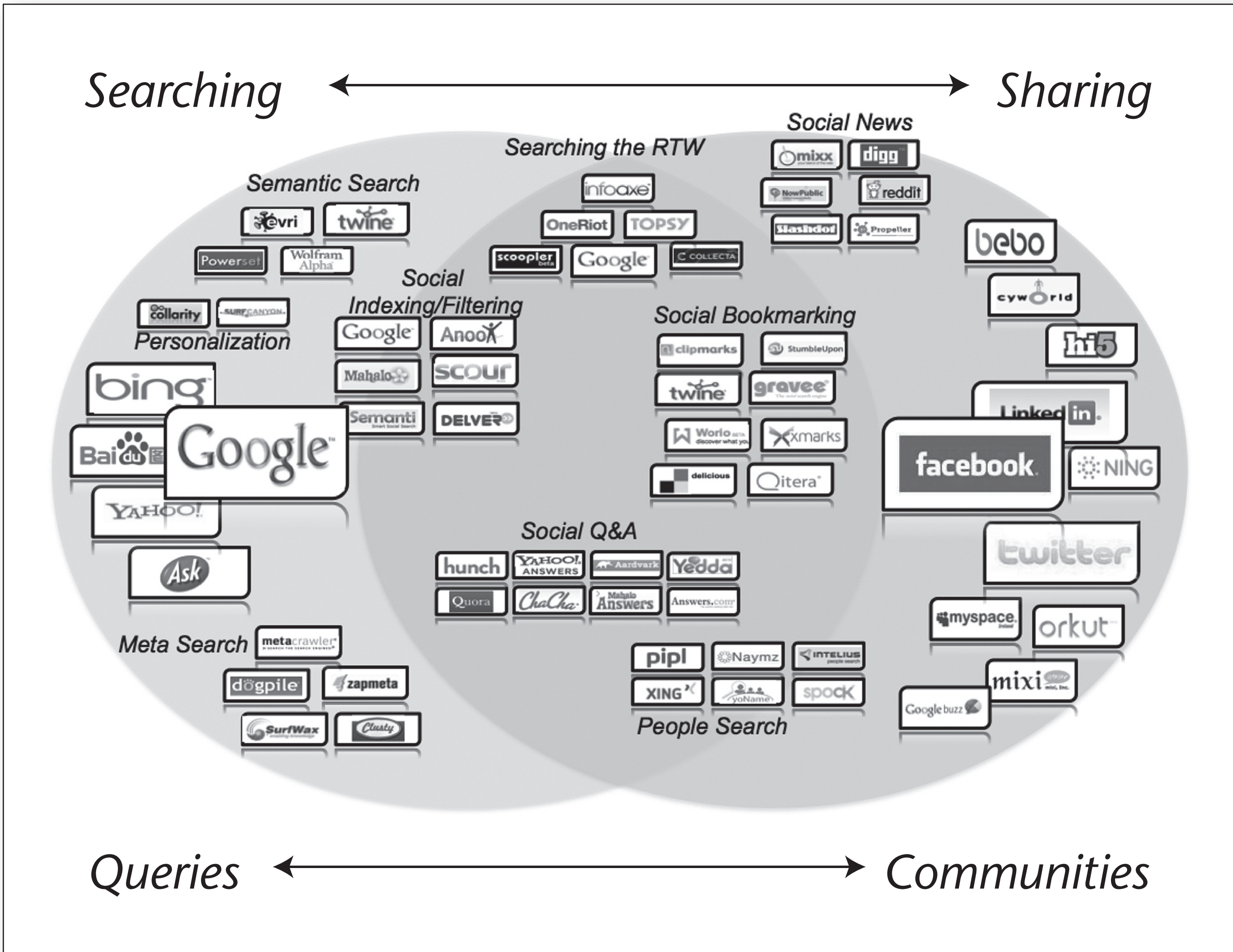


*Figure 1. Social Search Attempts to Bridge the Traditional, Query-Based World of Web Search with the Information-Sharing World of Social Networks.*

A variety of social search and sharing services have emerged to help users harness their social networks in pursuit of more effective information discovery across a variety of application contexts. This figure lists a number of well-know services, both startup and more mature, that have emerged to fill the gap between the mainstream search world such as Google, Yahoo, and Bing, and the major social networks such as Facebook, Twitter, and LinkedIn.

traditional information-retrieval technologies, adapted to the real-time nature of the social web. Recently mainstream search engines like Bing and Google have started to include user-generated content from our social graph within their mainstream search results.

In this article we focus on a different type of social search. Its aim is to help people during mainstream search tasks — that is, when they are using mainstream search engines — by harnessing the recent search experiences of their friends and colleagues through their social networks. The emphasis then is on making the solitary world of web search more collaborative. This relates to recent work in the area of *collaborative information retrieval*, which attempts to capitalize on the potential for collaboration during a variety of information-seeking tasks (Smyth et al. 2004; Morris and Horvitz 2007a and 2007b; Smyth et al. 2009b; Amershi and Morris 2008).

We will review the HeyStaks search service[7] as an example case-study in social search. HeyStaks is designed to add a layer of collaborative/social search on top of mainstream search engines: users continue to search as normal, using their search engine of choice, while benefiting from the past search experiences of their social networks. In particular, we will describe how HeyStaks generates result recommendations at search time and present a number of examples of the system in action.

## Collaborative Web Search

Recent studies in specialized 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 (Reddy and Dourish 2002; Reddy, Dourish, and Pratt 2001; Reddy and Spence 2008; Reddy and Jansen 2008). Moreover, recent work by Morris (2008) 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 behavior emerged. More than 90 percent of respondents indicated that they frequently engaged in collaboration at the level of the *search process*. For example, 87 percent of respondents exhibited "back-seat searching" behaviors, where they watched over the shoulder of the searcher to suggest alternative queries. A further 30 percent of respondents engaged in search coordination activities by using instant messaging to coordinate searches. Furthermore, 96 percent of users exhibited collaboration at the level of *search products*, that is, the results of searches. For example, 86 percent 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 Morris (2008), 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, tim*e* — that is, synchronou*s* versus asynchronou*s* search — and place — that is, colocated versus remote searchers. Colocated systems offer a collaborative search experience for multiple searchers at a single location, typically a single PC (for example, Amershi and Morris [2008], Smeaton et al. [2008]) whereas remote approaches allow searchers to perform their searches at different locations across multiple devices; see for example, Morris and Horvitz (2007a) and Smyth et al. (2009b). The former enjoy the obvious benefit of an increased facility for direct collaboration that is enabled by the face-to-face nature of colocated search, while the latter offer a greater opportunity for collaborative search. Alternatively, synchronous approaches are characterized by systems that broadcast a "call to search" in which specific participants are requested to engage in a well-defined search task for a well defined period of time; see for example, Smeaton et al. (2008). In contrast, asynchronous approaches are characterized by less well-defined, ad hoc search tasks and provide for a more open-ended approach to collaboration in which different searchers contribute to an evolving search session over an extended period of time; see for example, Morris and Horvitz (2007a).

A good example of the colocated, synchronous approach to collaborative web search is given by the work of Amershi and Morris (2008). Their CoSearch system is designed to improve the search experience for colocated 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, and others) 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 through 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 labor while maintaining communication between searchers, and live user studies speak to the success of CoSearch in this regard (Amershi and Morris 2008). The work of Smeaton, Lee, Foley, and McGivney (2007) is related in spirit to CoSearch but focuses on image search tasks using a tabletop computing environment, which is well suited to supporting collaboration between colocated 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 Smeaton et al. (2008), where the use of mobile

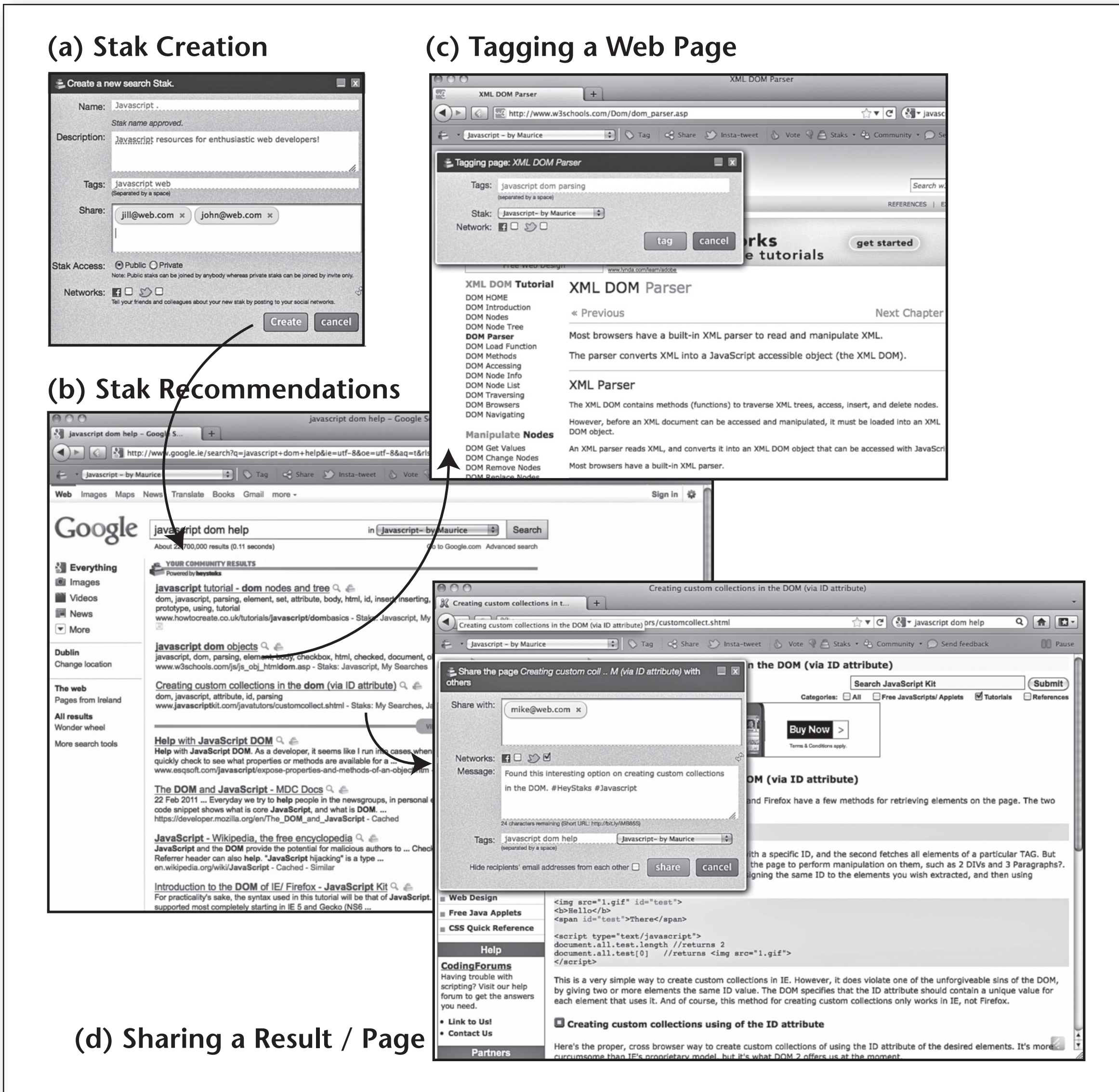


*Figure 2. HeyStaks in Action.*

(a) Stak creation: A user creates a JavaScript stak and shares it with developer colleagues. (b) Result recommendations: At Search time stak results are recommended based on the search experiences of stak members. (c) Tagging a web page: The user explicitly tags the page for the Javascript stak as relevant to "DOM parsing." (d) Sharing a page directly with others: Users can share pages with stak members and friends directly from the browser.

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 his or her 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 labor 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 behavior.

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 Morris and Horvitz (2007a). 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 cooperating searchers to synchronously view the results of each other's searches through a split-screen-style results interface. As with CoSearch above, one of the key design goals in SearchTogether is to support a division of labor 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 labor and shared awareness standpoint, separate from the underlying search process. In short, these systems have assumed the availability of an underlying search engine and provided a collaboration interface that effectively *imports* search results directly, allowing users to share these results. As noted by Pickens et al. (2008), 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 cosearchers. Consequently, the work of Pickens et al. (2008) 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 prioritize as yet unseen documents, while query recommendation techniques are used to suggest alternative avenues for further search exploration.

## Web Search Shared: A Case Study

In this section we describe a case study of a novel approach that brings collaboration to mainstream search engines. We describe the HeyStaks system, which has been designed to support collaborative web search tasks that are asynchronous and remote. A key objective is to tightly integrate this form of collaborative web search with existing mainstream search engines, which is a point of differentiation with respect to the previous collaborative search approaches outlined above. HeyStaks is designed to operate in parallel with search engines such as Google, Yahoo, and Bing through a browser plugin/toolbar. The benefit of this approach is that searchers can search as normal, using their preferred search engine, while still benefiting from the inherent collaboration potential of their friends and colleagues.

HeyStaks adds two basic features to any mainstream search engine. First, it allows users to create *search staks*, a type of folder for search experiences, at search time and invite others to join the staks by providing their email addresses. Staks can be configured to be *public* (anyone can join) or *private* (invitation only). 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.

In the following sections we showcase the functionality of HeyStaks in a worked example before reviewing the HeyStaks architecture and detailing how HeyStaks captures search activities within search staks and how this search knowledge is used to generate and filter result recommendations at search time; more detailed technical details can be found in Smyth et al. (2009a and 2009b).

### A Worked Example

To illustrate how HeyStaks operates consider a common use case of a small group of web developer colleagues working on a JavaScript project together. They know that up until now, they have been wasting a lot of search time refinding pages that they have found in the past or searching from scratch for solutions, tools, or hacks that one of their colleagues has already found recently. Recognizing the potential for HeyStaks to help with this type of wasted search effort, one of the group creates a new JavaScript stak as shown in figure 2a.

The user provides a stak title, a brief description, and the email addresses of fellow JavaScript developers to invite them to use this new stak.

Once the stak has been created and shared it will appear in the stak list of the HeyStaks toolbar for each of these users. As they each search for JavaScript-related information, their search actions will add new content to the stak. For example, in figure 2b a search for *javascript DOM help* by one stak member results in a set of recommendations from HeyStaks based on similar

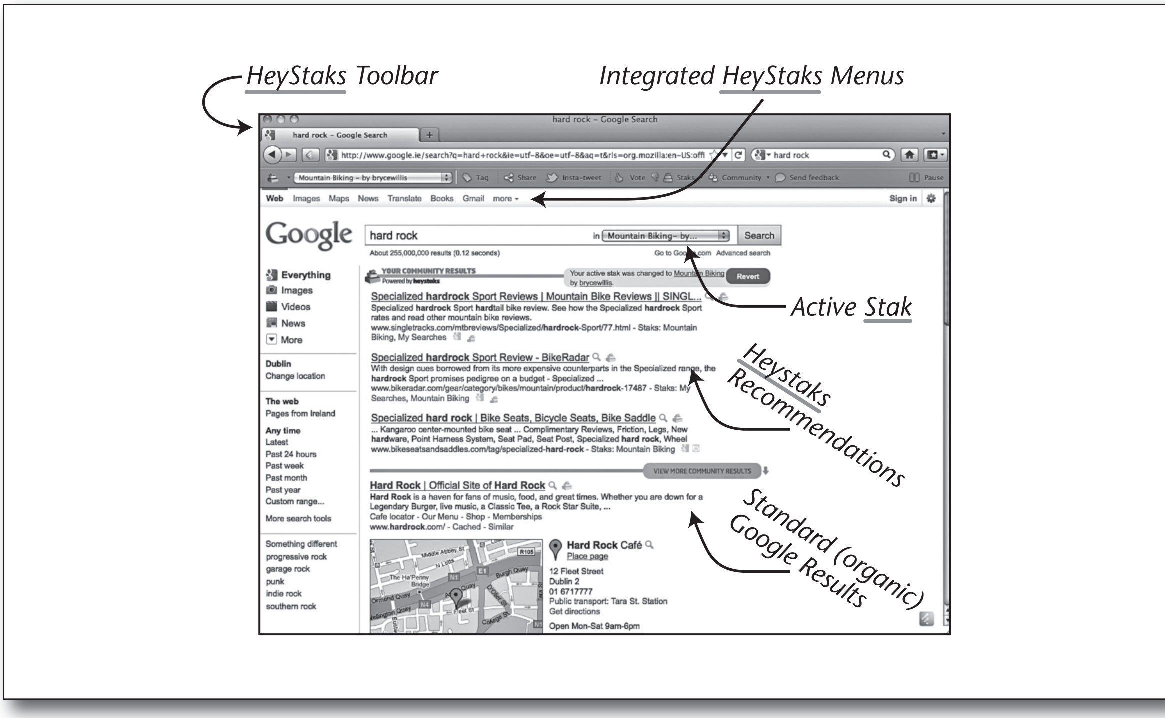


*Figure 3. HeyStaks Promotions Are Seamlessly Integrated with Mainstream Search Engine Content.*

searches by other members. Some of these recommendations originate from Google but others come from Yahoo and Bing, meaning that other stak members found these results on Yahoo or Bing; note the search engine icons after each recommendation to indicate the origin of the result. In this way HeyStaks also provides a form of collaborative metasearch.

In figure 2c our searcher chooses explicitly to tag a given result with the tags *javascript parsing DOM,* thereby adding to the stak index for this particular page and helping to reinforce the relevance of this page across these tag/query terms. In turn, in figure 2d we can see our searcher opting to share a different recommendation directly with other users, right from the browser; they can share the page by email or through social networks such as Twitter or Facebook. Just like tagging, this explicit sharing of pages provides another strong indicator to HeyStaks regarding the relevance of this page for future similar queries. Over time, these JavaScript developers will find that their stak becomes an important repository of JavaScript knowledge that is integrated directly with their everyday search tools (in this case, Google, Bing, and Yahoo). Indeed by making the stak public it can be recommended to other HeyStaks users who are also looking for JavaScript information and so promote the rapid growth of stak membership and content.

A further example is presented in figure 3, which focuses on the result recommendations that HeyStaks provides to users at search time. In this example the searcher, a mountain biker, is looking for information from the specialist mountain biking brand, Hard Rock. The query submitted is clearly ambiguous and Google responds with results related to the restaurant/hotel chain. However, HeyStaks recognizes the query as relevant to the *Mountain Biking* search stak that the searcher has previously joined and presents a set of more relevant results drawn from this stak. Thus HeyStaks automatically identifies the stak that is relevant for this user, given the user's query, and responds with more relevant, community-validated results from this particular stak.

## System Architecture

Figure 4 presents the overall HeyStaks architecture, which takes the form of two key components: a client-side *browser toolbar/plugin* and a back-end *server*. The toolbar has a dual purpose. On the one hand it provides users with direct access to the HeyStaks functions allowing them to create and share staks, tag or vote for pages, and perform other operations. Importantly the toolbar also provides for the type of deep integration that HeyStaks requires with an underlying search engine. For example, the toolbar captures the routine search activities of the user (query submissions and result clickthroughs). Moreover, the toolbar also makes it possible for HeyStaks to augment the mainstream search engine interface so that, for example, HeyStaks' recommendations can be integrated directly into a search engine's results page. The

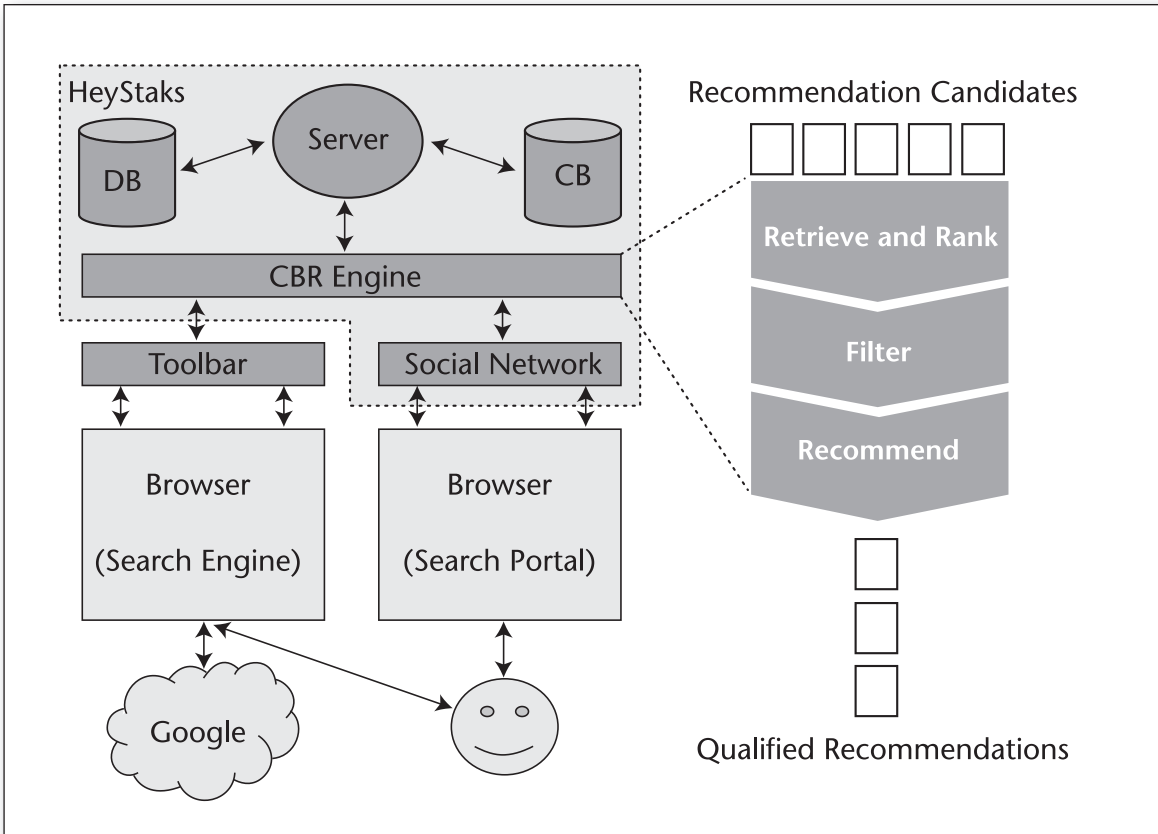


*Figure 4. The HeyStaks System Architecture and Outline of the Recommendation Model.*

toolbar also manages the communication with the back-end HeyStaks server. Search activities (queries, clickthroughs, tags, votes, shares, and so on) are used by the server to update the HeyStaks stak indexes. These stak indexes provide the primary source of recommendations so that when a user submits a query to a mainstream search engine, in a given stak context, this query is fed to the HeyStaks server in order to generate a set of recommendations based on the target stak and, possibly, other staks that the user has joined.

## Profiling Stak Pages

In HeyStaks each search stak ($S$) serves as a profile of the search activities of the stak members. Each stak is made up of a set of result pages ($S = \{r_1, \ldots, r_k\}$), and each result is anonymously associated with a number of implicit and explicit interest indicators, based on the type of actions that users can perform on these pages. A number of primary actions are facilitated, for example:

*Selections (or clickthroughs)* — that is, a user selects a search result (whether *organic* or *recommended*). Similarly, HeyStaks allows a user to *preview* a page by opening it in a frame (rather than a window), and *pop out* a page from a preview frame into a browser window.

*Voting* — that is, a user positively votes on a given search result or the current web page.

*Sharing* — that is, a user chooses to share a specific search result or web page with another user (by email or through posting to their Facebook Wall and so on).

*Tagging/Commenting* — that is, the user chooses to tag and/or comment on a particular result or web page.

Result selections are an example of an *implicit* action in the sense that this type of action is part and parcel of normal routine search activity. It is also a weak indicator of relevance because users will frequently select pages that turn out to be irrelevant to their current needs. Nevertheless, the frequent selection of a specific page in a specific stak, in response to a particular type of query, suggests relevance. The three other forms of actions (voting, sharing, tagging) we refer to as *explicit* actions in the sense that they are not part of the normal search process, but rather they are HeyStaks-specific actions that the user must choose to carry out. This type of deliberation suggests a stronger indicator of relevance and as such these actions are considered to be more reliable than simple result selections when it comes to evaluating the relevance of a page at recommendation time. Each result page $r_i^S$ from stak *S*, then, is associated with these indicators of relevance, 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 equation 1. This idea is related to earlier work by Amitay et al. (2005) and Smyth et al. (2004), which involves storing pages indexed by query terms. However, the present technology extends this to include other indicators such as snippets, tags, and votes.

$$r_i^S = \{q_1,\ldots q_n, t_1,\ldots,t_m, v^+, v^-, sel,\ tag,\ share\}. \quad (1)$$

In this way, each result page is associated with a set of *term data* (query terms and/or tag terms) and a set of *usage data* (the selection, tag, share, and voting count). The term data is represented as a Lucene (lucene.apache.org) index, with each result indexed under its associated query and tag terms.

## Retrieval and Ranking

At search time, the searcher's query $q_T$, current stak $S_T$, and other staks in the searcher's stak list are used to generate a list of recommendations to be returned to the searcher.

There are two types of recommendation candidates: *primary recommendations* are results that come from the active stak $S_t$; whereas *secondary recommendations* come from other staks in the stak list. There are two key steps when it comes to generating recommendations. First, a set of *recommendation candidates* is retrieved from each stak index, $S_i$, by querying the corresponding Lucene index with $q_T$. This effectively produces a list of recommendations based on the overlap between the query terms and the terms used to index each recommendation (query, snippet, and tag terms).

Second, these recommendations are filtered and ranked. Staks are inevitably noisy, in the sense that they will frequently contain results that are not on topic. Thus, the recommendation candidate selection stage may select results 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 article, but suffice it to say that any results that do not meet the necessary evidence thresholds are eliminated from further consideration; see also Smyth et al. (2009a and 2009b). The remaining recommendation candidates are then ranked according to two key factors: relevance and reputation. The relevance of a result *r* with respect to a query $q_T$ is computed based on Lucene's standard *TF*IDF* metric as per equation 2. The reputation of a result is a function of the reputation of the stak members who have added the result to the stak. And their reputation in turn is based on the degree to which results that they have added to staks have been subsequently recommended to, and selected by, other users; see McNally et al. (2010) for additional information. Essentially each result is evaluated using a weighted score of its relevance and reputation score as per equation 2, where *w* is used to adjust the relative influence of relevance and reputation and is usually set to 0.5.

$$rel(q_T, r) = \sum_{t \varepsilon q_t} tf(t \varepsilon r) \times idf(t)^2 . \quad (2)$$

$$score(r, q_T) = w \times rep(r) + (1-w) \times rel(q_T, r) \quad (3)$$

## Discussion

Currently HeyStaks has been deployed online as a search service offering. Interested users can download its browser toolbars or mobile apps at www.heystaks.com. During the course of its deployment there have been a number of opportunities to conduct live user trials and studies to explore how people engage with this new approach to search; see for example Smyth et al. (2009a and 2009b). In summary these studies highlight a number of interesting points. First and foremost, early users demonstrated a willingness to engage in a more collaborative approach to search: they frequently created and shared search staks and they often joined the staks created by others. Moreover, search collaboration was an inevitable and frequent result of this sharing. Those users who shared staks frequently received stak recommendations and often benefited directly from the searches of others.

# Conclusion

The purpose of this article has been to highlight a coming change in the world of mainstream web search and the important role that recommender systems and associated technologies have to play in this shift. Today mainstream search engines are largely solitary affairs, engaging the user in an isolated search for information, regardless of the fact that the user's social graph may have valuable contributions to make. And while recent experiments by Google and Bing have started to look at the value of added social content such as tweets to search results, this merely scratches the surface of what can be a much more fundamental change to the way in which we search online. Indeed recent trends in web search research point to a more social and collaborative approach to information discovery, one in which recommendation technologies have a key role to play. Web search needs to evolve if mainstream search engines are better to serve the needs of searchers, and social search technologies have the potential to harness the inherently collaborative nature of many search tasks in a way that is all but ignored by mainstream search engines today.

During the course of this article we have reviewed a variety of recent research that aims to make web search more collaborative and more social. This includes research that has emerged from more traditional information-retrieval groups and from the recommender systems community. In turn we have described in detail the HeyStaks system as a concrete case study in social search. HeyStaks is unique in the level of integration that it provides with mainstream search engines such as Google and Bing, allowing people to search as normal, while benefiting from recommendations that are derived from the searches of people they trust and on topics that matter to them.

## Acknowledgements

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361, by HeyStaks Technologies Ltd., and by the Australian government through the Intelligent Island Program and CSIRO Food and Nutritional Sciences.

## Notes

1. See www.last.fm.
2. See www.Flickr.com.
3. See www.wikipedia.org.
4. See techcrunch.com/2010/09/14/twitter-seeing-90-million-tweets-per-day.
5. Seewww.boygeniusreport.com/2010/07/07/twitter-handling-24-billion-search-queries-per-month.
6. See www.facebook.com/press/info.php?statistics.
7. See www.heystaks.com.

## References

Amershi, S., and Morris, M. R. 2008. CoSearch: A System for Co-Located Collaborative Web Search. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems*, 1647–1656. New York: Association for Computing Machinery.

Amitay, E.; Darlow, A.; Konopnicki, D.; and Weiss, U. 2005. Queries as Anchors: Selection by Association. In *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*, 193–201. New York: Association for Computing Machinery.

Asnicar, F., and Tasso, C. 1997. ifWeb: A Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. In *User Modeling: Proceedings of the Sixth International Conference*. New York: Springer Wien.

Chang, H.; Cohn, D.; and McCallum, A. K. 2000. Learning to Create Customized Authority Lists. In *Proceedings of the 7th International Conference on Machine Learning*, 127–134. San Francisco: Morgan Kaufmann.

Chirita, P. A.; Nejdl, W.; Paiu, R.; and Kohlschütter, C. 2005. Using ODP Metadata to Personalize Search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 178–185. New York: Association for Computing Machinery.

Chirita, P. A.; Olmedilla, D.; and Nejdl, W. 2004. Pros: A Personalized Ranking Platform for Web Search. In *Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Lecture Notes in Computer Science 3137, 34–43. Berlin: Springer.

Evans, B. M.; Kairam, S.; and Pirolli, P. 2009. Exploring the Cognitive Consequences of Social Search. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, Extended Abstracts Volume*, 3377–3382. New York: Association for Computing Machinery.

Evans, B. M.; Kairam, S.; and Pirolli, P. 2010. Do Your Friends Make You Smarter?: An Analysis of Social Strategies in Online Information Seeking. *Information Processing and Management* 46(6): 679–692.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2001. Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th International Conference on the World Wide Web*, 406–414. New York: Association for Computing Machinery.

Golovchinsky, G.; Qvarfordt, P.; and Pickens, J. 2009. Collaborative Information Seeking. *IEEE Computer* 42(3): 47–51.

Granka, L. A.; Joachims, T.; and Gay, G. 2004. Eye-Tracking Analysis of User Behavior in WWW Search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 478–479. New York: Association for Computing Machinery.

Gulli, A., and Signorini, A. 2005. The Indexable Web Is More Than 11.5 Billion Pages. In *Proceedings of the 14th International Conference on World Wide Web: Special Interest Tracks and Posters*, 902–903. New York: Association for Computing Machinery.

Lawrence, S., and Giles, C. L. 1998. Context and Page Analysis for Improved Web Search. *IEEE Internet Computing* 2(4): 38–46.

Ma, Z.; Pant, G.; and Sheng, O. R. L. 2007. Interest-Based Personalized Search. *ACM Transactions on Information Systems* 25(1).

Makris, C.; Panagis, Y.; Sakkopoulos, E.; and Tsakalidis, A. 2007. Category Ranking for Personalized Search. *Data & Knowledge Engineering* 60(1): 109–125.

McNally, K.; O'Mahony, M. P.; Smyth, B.; Coyle, M.; and Briggs, P. 2010. Towards a Reputation-Based Model of Social Web Search. In *Proceedings of the 2010 International Conference on Intelligent User Interfaces*, 179–188. New York: Association for Computing Machinery.

Morris, M. R. 2008. A Survey of Collaborative Web Search Practices. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems*, 1657–1660. New York: Association for Computing Machinery.

Morris, M., and Horvitz, E. 2007a. S3: Storable, Shareable Search. *Human-Computer Interaction INTERACT 2007,* Lecture Notes in Computer Science 4663, 120–123. Berlin: Springer.

Morris, M. R., and Horvitz, E. 2007b. SearchTogether: An Interface for Collaborative Web Search. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, 3–12. New York: Association for Computing Machinery.

Morris, M. R.; Teevan, J.; and Panovich, K. 2010. What Do People Ask Their Social Networks, and Why?: A Survey Study of Status Message Q&A Behavior. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, 1739–1748. New York: Association for Computing Machinery.

Pickens, J.; Golovchinsky, G.; Shah, C.; Qvarfordt, P.; and Back, M. 2008. Algorithmic Mediation for Collaborative Exploratory Search. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 315–322. New York: Association for Computing Machinery.

Pretschner, A., and Gauch, S. 1999. Ontology Based Personalized Search. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, 391–398. Los Alamitos, CA: IEEE Computer Society.

Reddy, M., and Dourish, P. 2002. A Finger on the Pulse: Temporal Rhythms and Information Seeking in Medical Work. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, 344–353. New York: Association for Computing Machinery.

Reddy, M., and Jansen, B. 2008. A Model for Understanding Collaborative Information Behavior in Context: A Study of Two Healthcare Teams. *Information Processing and Management* 44(1): 256–273.

Reddy, M. C., and Spence, P. R. 2008. Collaborative Information Seeking: A Field Study of a Multidisciplinary Patient Care Team. *Information Processing and Management* 44(1): 242–255.

Reddy, M.; Dourish, P.; and Pratt, W. 2001. Coordinating Heterogeneous Work: Information and Representation in Medical Care. In *Proceedings of the European Conference on the Semantic Web*, Lecture Notes in Computer Science, 239–258. Berlin: Springer.

Shen, X.; Tan, B.; and Zhai, C. 2005. Implicit User Modeling for Personalized Search. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 824–831. New York: Association for Computing Machinery.

Smeaton, A. F.; Foley, C.; Byrne, D.; and Jones, G. J. F. 2008. iBingo Mobile Collaborative Search. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval*, 547–548. New York: Association for Computing Machinery.

Smeaton, A. F.; Lee, H.; Foley, C.; and McGivney, S. 2007. Collaborative Video Searching on a Tabletop. *Multimedia Systems* 12(4-5): 375–391.

Smyth, B.; Balfe, E.; Boydell, O.; Bradley, K.; Briggs, P.; Coyle, M.; and Freyne, J. 2005. A Live-User Evaluation of Collaborative Web Search. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 1419–1424. Denver, CO: Professional Book Center.

Smyth, B.; Balfe, E.; Freyne, J.; Briggs, P.; Coyle, M.; and Boydell, O. 2004. Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web Search Engine. *User Modeling and User-Adaptated Interaction* 14(5):383–423.

Smyth, B.; Briggs, P.; Coyle, M.; and O'Mahony, M. P. 2009a. A Case-Based Perspective on Social Web Search. In *ICCBR '09: Proceedings of the 8th International Conference on Case-Based Reasoning*, 494–508. Berlin: Springer-Verlag.

Smyth, B.; Briggs, P.; Coyle, M.; and O'Mahony, M. P. 2009b. Google Shared. A Case-Study in Social Search. In *Proceedings of the 17th International User Modeling, Adaptation, and Personalization Conference*, Lecture Notes in Computer Science 5535, 283–294. Berlin: Springer.

Song, R.; Luo, Z.; Wen, J.-R.; Yu, Y.; and Hon, H.-W. 2007. Identifying Ambiguous Queries in Web Search. In *Proceedings of the 16th International Conference on the World Wide Web*, 1169–1170. New York: Association for Computing Machinery.

Speretta, M., and Gauch, S. 2005. Personalized Search Based on User Search Histories. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*. Los Alamitos, CA: IEEE Computer Society.

Spink, A., and Jansen, B. J. 2004. A Study of Web Search Trends. *Webology* 1(2).

**Barry Smyth** holds the Digital Chair of Computer Science at University College Dublin, Ireland. He is the director of the CLARITY Centre for Sensor Web Technologies and a fellow of the European Coordinating Committee on Artificial Intelligence. His research interests include artificial intelligence, recommender systems, and information retrieval. Smyth serves on the editorial boards of a number of journals including *AI Magazine* and the *Journal of Artificial Intelligence*.

**Jill Freyne** is a research scientist and project leader at the CSIRO Tasmanian ICT Centre. Freyne received her PhD from University College Dublin, where her research focused on collaborative, community-based web search. She continued her research into personalization and recommender technology at the CLARITY Centre for Sensor Web Technologies and IBM Research in Cambridge, MA, where she worked primarily on recommender technologies for social networking. At CSIRO, Freyne's research has focused on recommender and persuasive technologies to affect attitude and behavior in the health domain.

**Maurice Coyle** is a cofounder of HeyStaks Technologies Ltd., a venture-capital-funded startup in the social search space. Coyle received his PhD from University College Dublin, and his research interests include recommender systems and social search.

**Peter Briggs** is a cofounder of HeyStaks Technologies Ltd., a newly funded social search startup. Briggs graduated with a PhD from University College Dublin, and his research interests include social search and reputation in recommendation systems.