Title	A case-based perspective on social web search
Authors(s)	Smyth, Barry, Briggs, Peter, Coyle, Maurice, O'Mahony, Michael P.
Publication date	2009-07
Publication information	Smyth, Barry, Peter Briggs, Maurice Coyle, and Michael P. O'Mahony. "A Case-Based Perspective on Social Web Search." Springer, July 2009. https://doi.org/10.1007/978-3-642-02998-1_35.
Conference details	Paper presented at the 8th International Conference on Case-Based Reasoning (ICCBR 2009), Seattle, USA, July 20-23
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/1628
Publisher's version (DOI)	10.1007/978-3-642-02998-1_35

A Case-Based Perspective on Social Web Search*

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

CLARITY: Centre for Sensor Web Technologies
HeyStaks Ltd., NovaUCD, University College Dublin, Ireland.
http://www.clarity-centre.org,http://www.heystaks.com

Abstract. Web search is the main way for millions of users to access information every day, but we continue to struggle when it comes to finding the right information at the right time. In this paper we build on recent work to describe and evaluate a new application of case-based Web search, one that focuses on how experience reuse can support collaboration among searchers. Special emphasis is placed on the development of a case-based system that is compatible with existing search engines. We also describe the results of a live-user deployment.

1 Introduction

Mainstream search engines like Google are the primary way that people access information online, but their success is muted by the challenges that remain. Even Google fails to deliver relevant results as much as 50% of time [1]. This leads to poor search productivity but is as much a result of the vague queries that are commonplace in Web search (e.g. [2]) as it is due to failings in core search engine technology. Unfortunately vague queries are the reality of Web search and so in response there has been considerable research on different ways to improve result selection and ranking. For example, researchers have looked at ways to add context to bias search in the direction of special types of information (e.g., people, papers, etc.); see for e.g. [3]. Others have attempted to profile the preferences of searchers to deliver more personalized result-rankings [4–6].

Case-based reasoning (CBR) researchers have also recognised the opportunity for case-based techniques to improve Web search and information retrieval. For example, the work of Rissland [7] looks at the application of CBR to legal information retrieval, and [8] describe a case-based approach to question-answering tasks. Similarly, in recent years there has been considerable research looking at how CBR techniques can deal with less structured textual cases. This has led to a range of so-called *textual CBR* techniques [9]. In the context of Web search, one particularly relevant piece of work concerns the *Broadway* recommender system [10], and specifically the Broadway-QR query refinement technique that uses case-based techniques to reuse past query refinements in order to recommend new refinements to searchers. The work of [11] apply CBR techniques to Web search in a different way, by combining user profiling and textual case-based reasoning to dynamically filter Web documents according to a user's learned preferences.

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

This paper also focuses on how CBR techniques can be applied to Web search. It builds on previous work [12–14] which has already demonstrated the benefits of reusing search experiences within community-based search case bases; each case base representing the prior search experiences of a community of like-minded searchers. The main contribution of this paper is not a new case-based Web search technique per se – in fact we base our core CBR engine on the work of [13] – but rather a novel application of CBR techniques, which has the potential to contribute to mainstream Web search in a practical way. In the next section we describe the HeyStaks (www.heystaks.com) Web search utility, which is designed to work alongside search engines such as Google. HeyStaks allows searchers to create case bases to better organise their search experiences. These can be shared with friends and colleagues and used to augment Google's own search results with recommendations based on the search experiences of others. In Section 3 we describe a detailed evaluation of a recent HeyStaks deployment to demonstrate the value that HeyStaks brings to Web search; we show that users create and share search experiences at will, and benefit frequently from the experiences of others in a manner that mainstream search engines simply cannot support. In short we argue that this implementation of case-based Web search adds a new layer of collaboration to Web search by mediating the reuse of search experiences.

2 HeyStaks

HeyStaks is a new approach to case-based Web search, designed to work in cooperation *with* rather than competing *against* mainstream search engines. We describe HeyStaks as a *search utility* that seamlessly integrates with leading search engines such as Google, via a browser toolbar/plugin, to offer a number of practical features that are missing from today's Web search engines.

2.1 A Motivating Example

To understand the motivation for HeyStaks, consider a common use-case: Stella is a new machine learning PhD student. Her early work is dominated by background research and she spends a lot of time searching for relevant material and papers on a range of research related topics. As a newcomer Stella often struggles to find the right queries and successful searches can be elusive. She is not a dedicated bookmarker and so often wastes time re-searching for information she has found previously; recent research suggests that up to 30% of our searches are about re-finding information that we have previously found; see [15].

Fortunately for Stella, she has joined a mature, cooperative research group and she benefits from the generous wisdom of her colleagues when it comes to better understanding the things she should be searching for. In the spirit of collaboration colleagues will often email Stella links to Web sites and papers that they have found. Once again, research supports the value of this type of 'search' collaboration: up to 70% of the time we find that colleagues and friends have already found the type of things that we are looking for online; see [15].

Stella's experience is common: whether it is a group of colleagues at work, family members planning a vacation, or a set of friends researching an event, collaboration is frequently based on shared search experiences.Yet mainstream search engines offer no support for this type of collaboration, nor do they help individuals to organise their own search experiences. We believe that CBR techniques can play a key role in harnessing this type of collaboration in Web search.

Fig. 1. Selecting an active search stak at the start of a new search.

HeyStaks adds these missing features (the ability to *organise* and *share* search experiences) to Web search, via the HeyStaks browser toolbar. This allows Stella to group her searches into *staks* to reflect her different search interests (specific projects or tasks at work, travel, events etc.); these staks are case bases of search experiences. In the future, when Stella searches — her query defining a new information need — HeyStaks recommends results that she may have selected for similar queries. These reminders appear as promotions within the standard Google result-list. Moreover, Stella can chose to *share* her search staks with others, or she can join the search staks that others have created. Shared staks combine the search experiences of many, so that members can benefit from promotions that are derived from the experiences of others.

For example, in Fig 1 we see that Stella has joined staks created by other group members. She selects the "Machine Learning" stak as she embarks on a search for "stability clustering" and the result-list is shown in Fig. 2. The top 3 results are HeyStaks promotions, recommended from stak activity for similar queries; these promotions may have been promoted from much lower down the Google result-list, depending on stak activity. Google's normal results remain intact. At the top of the result-list is a reminder that HeyStaks has found additional recommendations. These come from Stella's own "My Searches" stak, as well as the "Machine Learning" stak, but are not quite relevant enough to merit an automatic promotion. By expanding this list Stella can see these additional promotions. In this way HeyStaks provides a very powerful collaboration platform for Web search: users search with their favourite search engine as normal; they organise their searches in to meaningful staks; and they benefit from the search experiences of their friends and colleagues.

Fig. 2. Google search results with HeyStaks promotions.

2.2 System Overview

Fig. 3 outlines the basic HeyStaks architecture. The system is made up of a number of key components including:

1. The HeyStaks *Server* is responsible for managing the core HeyStaks data including the user and stak databases (DB), which contain all of the basic

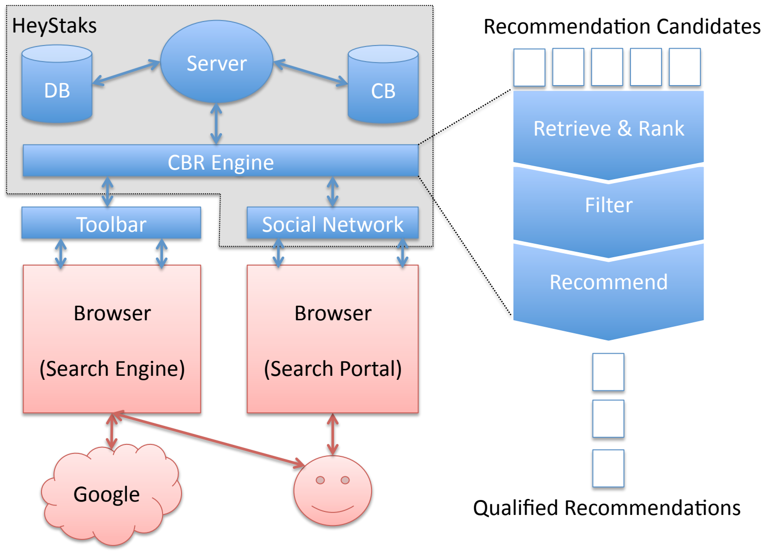

Fig. 3. The HeyStaks system architecture and outline recommendation model.

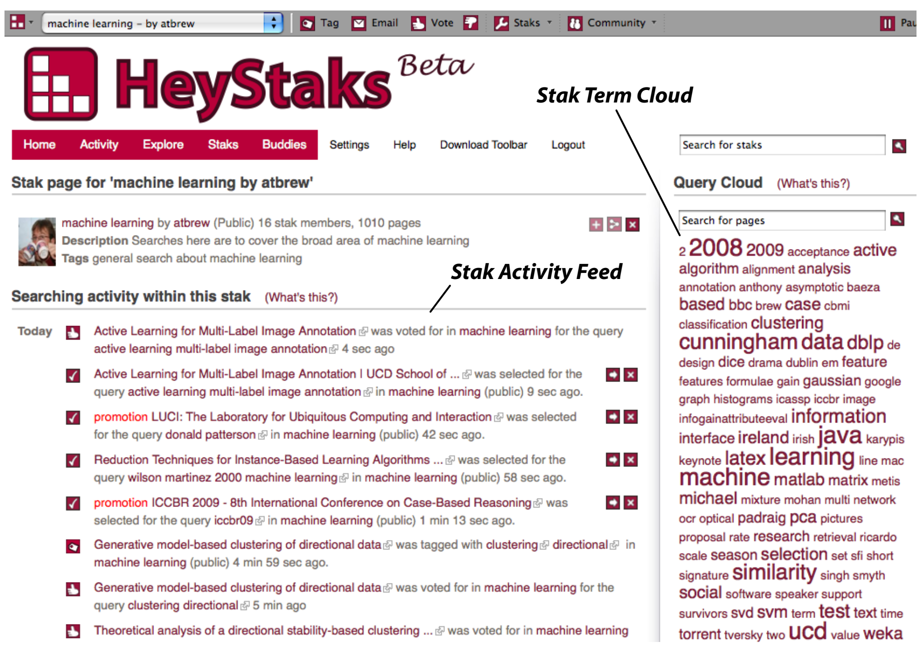

Fig. 4. The HeyStaks search portal provides direct access to staks and past searches.

user and stak information, and the search staks or case bases (CB), which store the individual stak-based search experiences.
2. The *Recommendation Engine* performs a number of recommendation tasks, the primary one being the selection of suitable results for promotion to the user, given the current target query and active stak. In addition the recommendation engine is also used to proactively suggest appropriate staks to searchers, according to their search context, but we will not focus on this feature in this paper.
3. The HeyStaks *Social Network* provides users with access to their individual profiles, the staks that they create and share, plus stak-related activity feeds as a way to keep up to date with what friends and colleagues have been searching for. It also provides a range of features to help users to maintain their staks and to help them discover interesting results and staks; for example Fig. 4 shows a stak page from the HeyStaks search portal for the "Machine Learning" stak discussed previously.
4. The HeyStaks *Toolbar* is the key client-side component. It provides users with various features to create and share staks and pages, and also acts as a link between the underlying search engine (e.g. Google) and the HeyStaks server, recording search experiences and inserting relevant promotions.

2.3 Case Bases of Search Experiences

As a CBR system, HeyStaks is distinguished in a number of interesting ways. In conventional CBR systems there is often a single case base or, at best, a small number of case bases, which have been created by trusted experts. HeyStaks is very different. Each stak is a case base and each user can create many staks. Moreover, there are few guarantees about the reliability of the information that makes its way in to these case bases. Users might forget to select an appropriate stak for their search, and although there is an automatic stak selection feature (beyond the scope of this work) it is not 100% reliable. These problems are exacerbated by the fact that a typical stak may be subscribed to by many different stak members, each with his/her own individual agenda for participating in the stak. All of this makes for a much more fluid and dynamic (and challenging) CBR environment, where experiences are distributed across a multitude of evolving case bases. It is, incidentally, also an environment in which understanding the source of case knowledge is likely to be particularly important, because understanding the provenance of a case, and the reputation of the case and/or stak creator, can play a vital role when it comes to evaluating the case's reliability; see [14, 16].

These ideas and challenges make fertile ground for future work but for now we will focus on describing the basic case representation and recommendation techniques that are used in HeyStaks. Each search stak (S) is a case base recording the search experiences of the stak members. Each stak is made up of a set of cases (($S = \{c_1, ..., c_k\}$) and each case corresponds to a single result page (p_i) that has been 'selected' for this stak. Each case is anonymously associated with a number of implicit and explicit interest indicators, including: the total number of times the result has been selected (sel) during a search, the query terms ($q_1, ..., q_n$) that led to its selection, the snippet terms associated with the result when it wasn't selected ($s_1, ..., s_m$), the total number of times a result has been tagged (tag) and the terms used to tag it ($t_1, ..., t_w$), the total votes it has received (v^+, v^-), and the number of people it has been shared with ($share$); see Eq. 1. In addition each term (query, tag, snippet) is linked with a hit-count that reflects the number of times that this term has been associated with the page in question; for example, h_{q_1} refers to the number of times that the query term q_1 has been found in queries that have led to the selection of the page p_i.

$$c_i^S = \{p_i, (q_1, h_{q_1})...(s_1, h_{s_1})...(t_1, h_{t_1})...(t_w, h_{t_w}), v^+, v^-, sel, tag, share\} \quad (1)$$

In this way, each case is associated with *term data* (query, snippet, tag terms) and *usage data* (the selection, tag, share, and voting counts). The former provides the basis for retrieving and ranking *promotion candidates*, while the latter provides a source of evidence that can be used to filter results and to generate a final set of recommendations. Thus, at search time, a set of recommendations is produced in a number of stages: relevant results are retrieved and ranked from a set of suitable staks; these promotion candidates are filtered, based on an *evidence model*, to eliminate noisy recommendations; and finally they are added to the Google result-list according to a set of *recommendation rules*.

Retrieval & Ranking. For a given target query, q_t, HeyStaks generates a set of promotion candidates. Briefly, there are two types of promotion candidates: *primary promotions* are results that come from the active stak[1] S_t; whereas *secondary promotions* come from other staks in the searcher's stak-list.

$$score(q_t, c_j) = \sum_{t \epsilon q_t} tf(t \epsilon c_j) \bullet idf(t)^2 \quad (2)$$

To generate these promotion candidates, the HeyStaks server uses q_t as a probe into each stak case base, S_i, to identify a set of relevant stak cases $C(S_i, q_t)$. Each candidate case, c_j is scored using a similar technique to that described by [13] by using a TFIDF (*term frequency* $\bullet$ *inverse document frequency*) function as the basis for an initial recommendation ranking; this approach prefers cases that match terms in the query which have occurred frequently in the case, but infrequently across the case base as a whole (see Eq. 2).

Evidence-Based Filtering. We have already mentioned that search staks are inevitably noisy because searchers will often forget to set an appropriate stak at the start of a new search session. As a result the retrieval and ranking stage may

[1] That is, the stak that is currently active in the HeyStaks toolbar. This stak will either have been selected by the user during a recent search or may have been automatically selected by HeyStaks based on the current query/search context.

select misclassified pages that are not relevant to the current query context. To avoid making spurious recommendations HeyStaks uses an *evidence filter*, which uses a variety of threshold models to further evaluate the relevance of a particular result in terms of its usage evidence; for instance, tagging evidence is considered more important than voting, which in turn is more important than implicit selection evidence. The precise details of this model are beyond the scope of this paper but, 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.

Recommendation Rules. The final task is to add the remaining *qualified recommendations* to the Google result-list. HeyStaks uses a number of different recommendation rules to determine when 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 icons. If a remaining primary promotion is also in the default Google result-list then this is labeled in its default Google position. If there are still remaining primary promotions then these are added to the secondary promotion list, which is sorted according to page TFIDF scores. These recommendations are then added to the Google result-list as an optional, expandable list of recommendations, such as that shown at the top of Fig. 2

3 Evaluation

As an application-paper our main aim in this work is to describe the HeyStaks application and, in particular, the results of a recent live-deployment obtained from a usage analysis of 95 HeyStaks beta users (excluding those users directly connected with the HeyStaks project) during the period October 2008 - January 2009. In this evaluation we will focus on two particular evaluation themes:

1. As a case-based reasoning system HeyStaks is interesting because it facilitates the casual creation and sharing of case knowledge, across multiple case bases, in the form of search experiences. But do users actually take the time to create these search case bases and do they share them with others?
2. As a search utility, HeyStaks is interesting because it promises to improve Web search by facilitating collaboration among searchers. But do users benefit from this collaboration? Do they respond positively to HeyStaks' promotions? Do they benefit from their own search experiences or those of others or a mixture of the two?

Since this is a study of live-users *in the wild* there are certain limitations on what we have been able to measure. There is no control group and it was not feasible, mainly for data privacy reasons, to analyse the relative click-through behaviour of users, by comparing their selections of Google results to their selections of promotions. However earlier work does report on these type of results in more conventional control-group laboratory studies (see for e.g. [1, 15]).

3.1 On the Creation and Sharing of Search Case Bases

A key element of the HeyStaks value proposition is that searchers need a better way to organise and share their search experiences and, specifically, that the ability to create and share search staks will provide them with these features. But do users actually take the time to create staks? Do they share them or join those created by others? As a user shares and joins staks they effectively create a *search network*, the members of which are the other users who co-share these staks. How do these search networks evolve as a result of stak sharing?

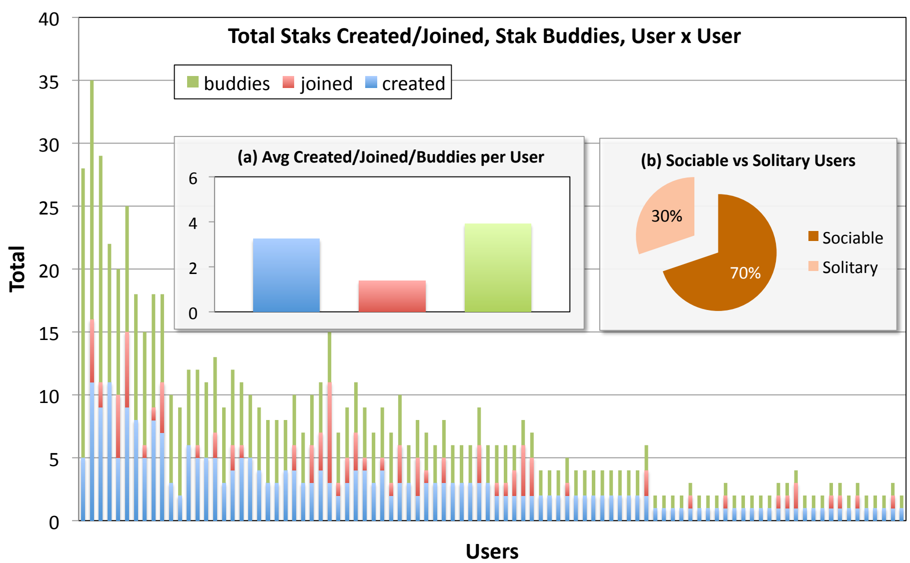

Fig. 5. The number of staks created and shared by users and the size of their respective search networks (buddy lists).

These questions are addressed by Figure 5 which plots the number of staks that are created and joined by the beta users; also shown are the number of stak members (buddies) per stak. The results demonstrate that users do actively engage in considerable stak creation activity and they do regularly join the staks created by others. For instance, the plot shows that the most prolific stak creator has created 11 new staks during the course of the beta trial, while other users join up to 8 staks that others have created. And this sharing of search experiences helps create search networks that extend to as many as 19 other users.

In fact, as per the Figure 5(a) insert we see that, on average, users create 3.2 staks and join another 1.4, to develop search networks that link them directly to about 4 other users. 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. Importantly though, the vast majority of users (70%) do make the effort to share/join staks — these are the *sociable* users of insert Figure 5(b) — and for staks that are shared, they benefit from an average of 3.6 members each.

In total more than 300 staks were created during the beta, on a wide range of interests, from broad topics such as travel, research, entertainment, to more niche interests including archaeology, black and white photography, and biking. These data speak to the inherent willingness of users to take advantage of the organisation and sharing features that HeyStaks makes available, resulting in the creation of a large number of casual search case-bases that have the potential to evolve over time to become significant repositories of topical search knowledge.

Of course the creation and sharing of search staks is really just a means to an end: it is a way to help users partition their search interests to facilitate recommendation and recovery. Ultimately the success of HeyStaks will depend on whether users find these recommendations to be useful, which we will explore further in the following sections.

3.2 Search Collaboration

One of the most important observations about users is the extent to which their natural search activity creates a community of collaborating searchers. HeyStaks is motivated by the idea that Web search is an inherently social or collaborative activity, despite the absence of such collaboration features from mainstream search engines. In this section we will examine the nature of this collaboration effect, and the extent to which actual collaboration occurs in practice. As users search, tag, and vote they effectively produce and consume search knowledge. A user who is the first to select or tag a given result for a stak *produces* new search knowledge. Later, if this result is promoted to another user and re-selected, then this other user is said to have *consumed* that search knowledge.

These relationships between the producers and consumers of search knowledge within staks effectively creates an implicit social network of search collaboration. Fig. 6 presents a visualization of this network for the beta users. Each node is a unique user and edges between nodes correspond to evidence for search collaboration. These edges are directed: an edge from *user A* (the producer) to *user B* (the consumer) signifies that *user B* has selected at least one of the search results that *user A* has been responsible for adding (through his/her own selections, tagging or voting activity) to a search stak, which is shared between both users. Of course a single edge can (and typically does) reflect many collaboration instances between two users. In this example the diameter of the nodes reflects the *reputation* of the user in terms of their relative ability to help other users to search; however a detailed discussion of this reputation mechanism is beyond the scope of this paper.

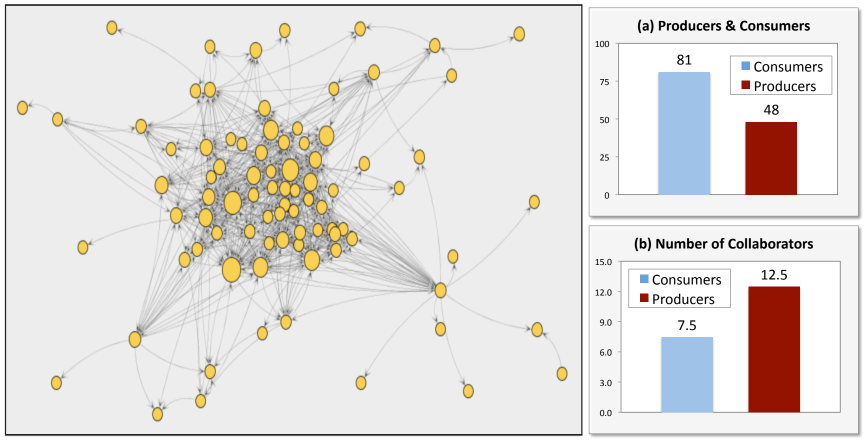

Fig. 6. A representation of the collaboration network among HeyStaks searchers: (a) The percentage of searchers adopting consumer and producer roles; (b) The average number of collaborating searchers for producers and cusumers.

Perhaps the first thing to notice is the extent of the collaboration that is evident among these users. From Fig. 6 we can see that the sharing of search knowledge is not limited to a small clique of especially social searchers, far from it: the graph includes 85% of beta users meaning that 85% of users have engaged in actual search collaborations. Most users (81 out of 95) have acted as consumers, benefiting from results that others have produced and 51% (48 out of 95) have acted as producers, creating at least some new search knowledge that others have consumed[2]; as shown in the Fig. 6(a) insert. Indeed when acting as a consumer, the average searcher benefits from results that have been produced by more than 7 other searchers. And when acting as a producer, searchers create search knowledge that is consumed by more than 12 other users on average; as shown in the Fig. 6(b) insert.

3.3 Search Leaders & Followers

These results tell us that searchers do appear to help others and that they are helped by others in return. In this section we take a more detailed look at the nature of search collaboration by examining the sources of promotions.

First let us start with an obvious next question by asking whether some users are especially altruistic in their searches, helping others more often that they are helped themselves? Figure 7 plots each user according to the number of other users they have helped (y-axis) compared to the number of other users they have been helped by (x-axis). As expected, given the data above, about half the users (those that are never producers) are situated on the x-axis [3]. These are the search

[2] Of course many users are both consumers and producers of search knowledge.

[3] Due to overlapping data points it is impossible to resolve each user along this axis.

followers within HeyStaks, the *novice* searchers who benefit disproportionately from the search experiences of others but are not seen to significantly contribute to search knowledge themselves. The other half of the user-base presents with a greater balance between the production and consumption of search knowledge, as evidenced by the clustering of data points along the main diagonal.

Users who have helped *more* people than they themselves have been helped by are *net producers*) and those who have been helped by more users than they have helped themselves, are *net consumers*). As the insert in Figure 7 shows, approximately 47% of users are net producers. This is especially important when we remember that above we noted how 51% of users have produced at least *some* search knowledge that has been consumed by some other user. The vast majority of *these* users, 94% of them in fact, are actually producing *more* search knowledge than they consume. These are the search *leaders* within HeyStaks.

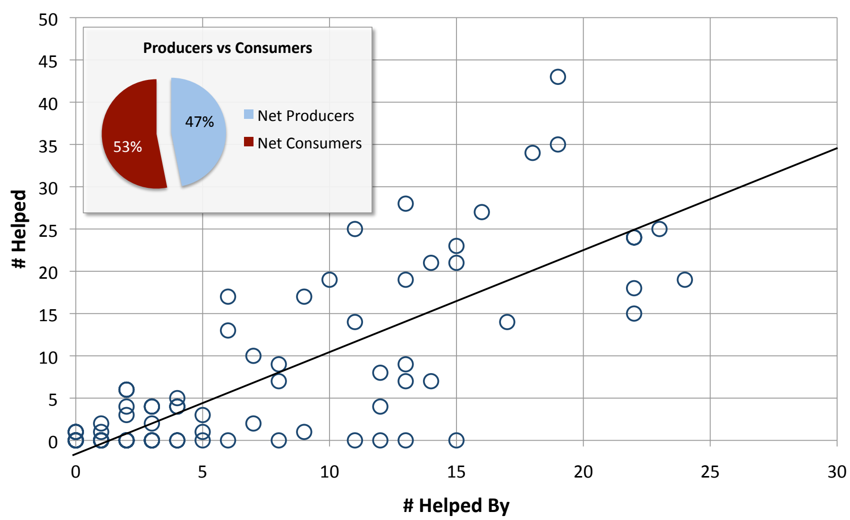

Fig. 7. Each HeyStaks user is plotted according to the number of other users that they have helped and been helped by. The points above the main diagonal indicate the net producers and those below are the net consumers.

3.4 Promotion Sources

So far we have looked at user-level collaboration and found that users do often collaborate through the implicit production and consumption of search knowledge. Now we will explore similar ideas but at the level of individual searches. Overall, the beta users selected more than 11,000 promotions as they searched. When a user performs a search and selects a promotion, how often is this promotion derived from their own past search activities (*self promotions*) and how often does it come from the shared search activities of other stak members (*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 (see [17]), where the user is benefiting from new content that might otherwise have been missed, or have been difficult to find.

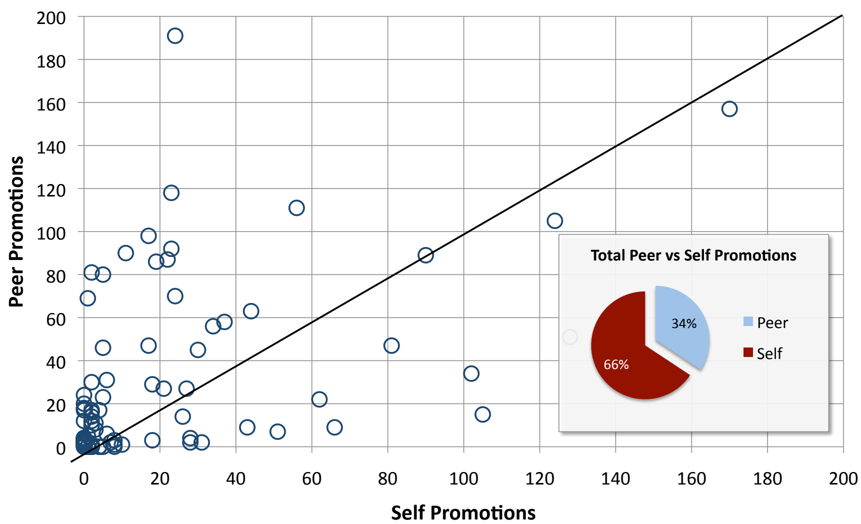

Fig. 8. Each HeyStaks user is plotted according to the number of peer and self promotions they have selected.

Figure 8 presents this information as a plot of users located according to the actual number of peer and self promotions selected. For example, the somewhat isolated user represented by the data point in the top-left quadrant of the graph has selected about 25 self promotions but more than 190 peer promotions, clearly demonstrating the 'discovery' benefits of shared staks for this particular user. At the other extreme there are users (towards the bottom right quadrant) who benefit dispropotionately from their own self promotions.

As per Figure 8, 66% of all selected promotions are self promotions; users are usually benefiting from their own past searches. This is not surprising, especially during the early stages of stak development. Stak growth is initially led by the stak creator, and so inevitably most of the promotions that are generated are in response to the stak creator's search queries. Most of these promotions will be self promotions, derived from the creator's own search activities. As staks are shared, 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 inevitably generated and selected. It is an interesting task for future work to explore the evolution of a search stak and to further 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.

4 Conclusions

The social Web places emphasis on the sharing of experiences (e.g., comments, opinions, ratings) and CBR has a opportunity to play an important role; see [18]. This paper adds to a body of research on case-based Web search. The main contribution is a description of the HeyStaks application and a detailed evaluation of its beta launch. The results of this evaluation highlight the potential for CBR to add value in the social Web. We have highlighted how the HeyStaks approach helps to support a very beneficial form of search collaboration that has been missing from mainstream search. In concluding, it is worth highlighting the CBR questions that have emerged during the course of this work:

1. We have focused on recommending cases from an individual case base (stak) but in HeyStaks there are many different case bases and the ability to suggest a case base is a useful feature in highlighting a new source of search knowledge to users. How might case bases be recommended?
2. Given that our search case bases are much noisier that expert-created case bases, how should this influence retrieval, reuse, and maintenance?
3. As cases are reused they are *enriched* by the activities of others: each case evolves to contain a *trace* of its reuse, as users re-select, tag, vote, and share the page associated with the case; see also [19].
4. As case bases evolve there may be opportunities to merge related case bases, or case bases may start to diverge as different contributors use them in different ways. How might these opportunities to merge or split case bases be recognised and handled?

These questions highlight new opportunities for research that will help to further strengthen the role of CBR in the social Web. In turn, HeyStaks presents many interesting challenges when it comes to addressing the type of scale that will be faced in a broader deployment context. For example, additional recommendation challenges exist when it comes to recommending pre-existing staks for users to join, based on a user's current search query or proactively, based on their previous queries. Moreover, the issue of malicious users, and their ability to influence stak promotions needs also to be addressed, perhaps by evaluating the reputation of users and allowing this to influence promotions.

References

1. Coyle, M., Smyth, B.: Information recovery and discovery in collaborative web search. In: ECIR. (2007) 356–367

2. Spink, A., Wolfram, D., Jansen, M.B.J., Saracevic, T.: Searching the web: the public and their queries. Journal of the American Society for Information Science and Technology **52**(3) (2001) 226–234
3. Lawrence, S.: Context in Web Search. IEEE Data Engineering Bulletin **23(3)** (2000) 25–32
4. Liu, F., Yu, C., Meng, W.: Personalized Web Search for Improving Retrieval Effectiveness. IEEE Transactions on Knowledge and Data Engineering **16(1)** (2004) 28–40
5. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized search on the world wide web. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web. Springer-Verlag (2007) 195–230
6. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing search via automated analysis of interests and activities. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2005) 449–456
7. Rissland, E.L., Daniels, J.J.: A hybrid CBR-IR Approach to Legal Information Retrieval. In: Proceedings of the 5th international conference on Artificial intelligence and law, ACM Press (1995) 52–61
8. Burke, R., Hammond, K., Kulyukin, V., Tomuro, S.: Question Answering from Frequently Asked Question Files. AI Magazine **18(2)** (1997) 57–66
9. Lenz, M., Ashley, K.: AAAI Workshop on Textual Case-Based Reasoning (1999) AAAI Technical Report WS-98-12.
10. Kanawati, R., Jaczynski, M., Trousse, B., J-M, A.: Applying the Broadway Recommendation Computation Approach for Implementing a Query Refinement Service in the CBKB Meta-search Engine. In: Conférence Française sur le Raisonnement á Partir de Cas (RáPC'99). (1999)
11. Godoy, D., Amandi, A.: PersonalSearcher: An Intelligent Agent for Searching Web Pages. In Monard, M.C., Sichman, J.S., eds.: IBERAMIA-SBIA. Volume 1952., Springer (2000) 43–52
12. Balfe, E., Smyth, B.: Case-based collaborative web search. In: ECCBR. (2004) 489–503
13. Boydell, O., Smyth, B.: Enhancing case-based, collaborative web search. In: ICCBR. (2007) 329–343
14. Briggs, P., Smyth, B.: Provenance, trust, and sharing in peer-to-peer case-based web search. In: ECCBR. (2008) 89–103
15. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community-based web search engine. User Model. User-Adapt. Interact. **14**(5) (2004) 383–423
16. Leake, D.B., Whitehead, M.: Case provenance: The value of remembering case sources. In: ICCBR. (2007) 194–208
17. O'Day, V.L., Jeffries, R.: 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 '93), New York, NY, USA, ACM Press (1993) 438–445
18. Plaza, E.: Semantics and experience in the future web. In: ECCBR. (2008) 44–58
19. Laflaquière, J., Settouti, L.S., Prié, Y., Mille, A.: Trace-based framework for experience management and engineering. In: KES. (2006) 1171–1178