HeyStaks: A Real-World Deployment of Social Search

Barry Smyth
CLARITY: Centre for Sensor Web Technologies
University College Dublin
Dublin, Ireland
barry.smyth@ucd.ie

Maurice Coyle
HeyStaks
NovaUCD
University College Dublin
Dublin, Ireland
maurice@heystaks.com

Peter Briggs
HeyStaks
NovaUCD
University College Dublin
Dublin, Ireland
maurice@heystaks.com

ABSTRACT

The purpose of this paper is to provide a deployment update for the HeyStaks social search system which uses recommendation techniques to add collaboration to mainstream search engines such as Google, Bing, and Yahoo. We describe our the results of initial deployments, including an assessment of the quality of HeyStaks' recommendations, and highlight some lessons learned in the marketplace.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.3.5 [**Online Information Services**]: Web-Based Services

Keywords

social search, recommender systems, deployed application

1. INTRODUCTION

Recently researchers have begun to recognise the potential of web search as a platform for a more social and collaborative approach to information discovery. For example, researchers in the area of *collaborative information retrieval* have sought to make web search more collaborative and more social in a variety of ways; see for example [1–3]. The driving insight is that search is an inherently collaborative affair as people frequently search for similar things in similar ways [6, 8, 10, 12]. By embracing collaboration mainstream search deliver improved result relevance especially in the face of ongoing challenges by co called *content farms* and increasingly aggress SEO (search engine optimization) strategies[1]. For instance, novice searchers have much to gain from the search experiences of more expert searchers. But such experiences and recommendations need to be integrated into mainstream search contexts.

[1]http://searchengineland.com/google-forecloses-on-content-farms-with-farmer-algorithm-update-66071

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
RecSys'12, September 9–13, 2012, Dublin, Ireland.
Copyright 2012 ACM 978-1-4503-1270-7/12/09 ...$15.00.

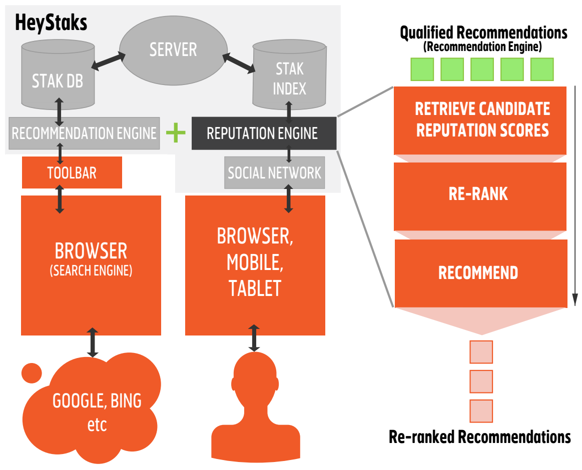
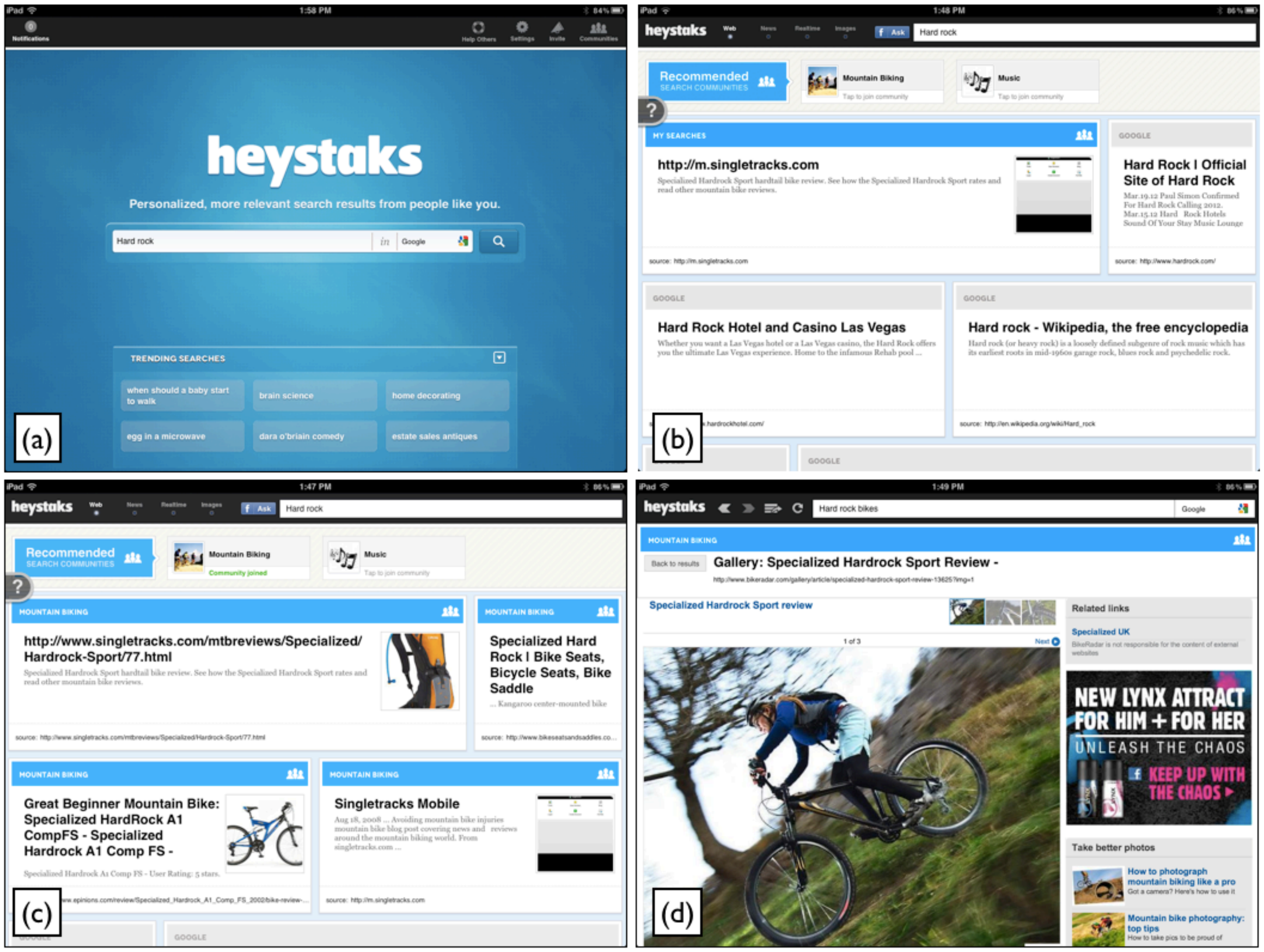


Figure 2: The HeyStaks system architecture and outline recommendation model.

This is a key objective for HeyStaks: an approach to social search that adds a layer of collaboration and recommendation to mainstream search engines like Google and Bing. The recommendation technology that drives HeyStaks has been described in detail elsewhere [5,9–11]. For this purpose of this short paper we review our deployment experiences as HeyStaks has graduated from the research lab to the marketplace. This includes a summary of the lessons learned and key evaluation results based on the first months of usage.

2. A REVIEW OF HEYSTAKS

HeyStaks combines ideas from social networking, content curation, and web search to provide a platform for search collaboration. Users can create named (public or private) staks on topics of interest and they can share these staks with friends and colleagues. As users search, their search experiences are used to populate staks; search histories are stored anonymously in staks. And when a stak member performs a new search, she may benefit from recommendations from her staks, perhaps based on the recent searches of friends. For example a group of vacationing friends might create a stak called *Ski Whistler 2012* to share their searches for their upcoming ski-trip. As they each search the results they find are added to the stak and recommendations are made based on the searches of the group. So, for example, one group member, eager to ensure they stay some-



http://m.singletracks.com

Specialized Hardrock Sport hardtail bike review. See how the Specialized Hardrock Sport rates and read other mountain bike reviews.

source: http://m.singletracks.com

Hard Rock | Official Site of Hard Rock

Mar.19.12 Paul Simon Confirmed For Hard Rock Calling 2012. Mar.15.12 Hard Rock Hotels Sound Of Your Stay Music Lounge

source: http://www.hardrock.com/

Hard Rock Hotel and Casino Las Vegas

Whether you want a Las Vegas hotel or a Las Vegas casino, the Hard Rock offers you the ultimate Las Vegas experience. Home to the infamous Rehab pool ...

Hard rock - Wikipedia, the free encyclopedia

Hard rock (or heavy rock) is a loosely defined subgenre of rock music which has its earliest roots in mid-1960s garage rock, blues rock and psychedelic rock.

source: http://en.wikipedia.org/wiki/Hard_rock

http://www.singletracks.com/mtbreviews/Specialized/Hardrock-Sport/77.html

Specialized Hardrock Sport hardtail bike review. See how the Specialized Hardrock Sport rates and read other mountain bike reviews.

source: http://www.singletracks.com/mtbreviews/Specialized/Hardrock-Sport/77.html

Specialized Hard Rock | Bike Seats, Bicycle Seats, Bike Saddle

... Kangaroo center-mounted bike

source: http://www.bikeseatsandsaddles.co...

Great Beginner Mountain Bike: Specialized HardRock A1 CompFS - Specialized Hardrock A1 Comp FS -

Specialized Hardrock A1 Comp FS - User Rating: 5 stars.

Singletracks Mobile

Aug 18, 2008 ... Avoiding mountain bike injuries mountain bike blog post covering news and reviews around the mountain biking world. From singletracks.com ...

source: http://m.singletracks.com

Gallery: Specialized Hardrock Sport Review -

http://www.bikeradar.com/gallery/article/specialized-hardrock-sport-review-13625?img=1

Specialized Hardrock Sport review

Specialized UK

How to photograph mountain biking like a pro

Mountain bike photography: top tips

Figure 1: The HeyStaks iPad app: (a) the search homescreen; (b) a search results display with recommended communities/staks; (c) A search results page with recommendations from a joined stak; (d) an individual results page.

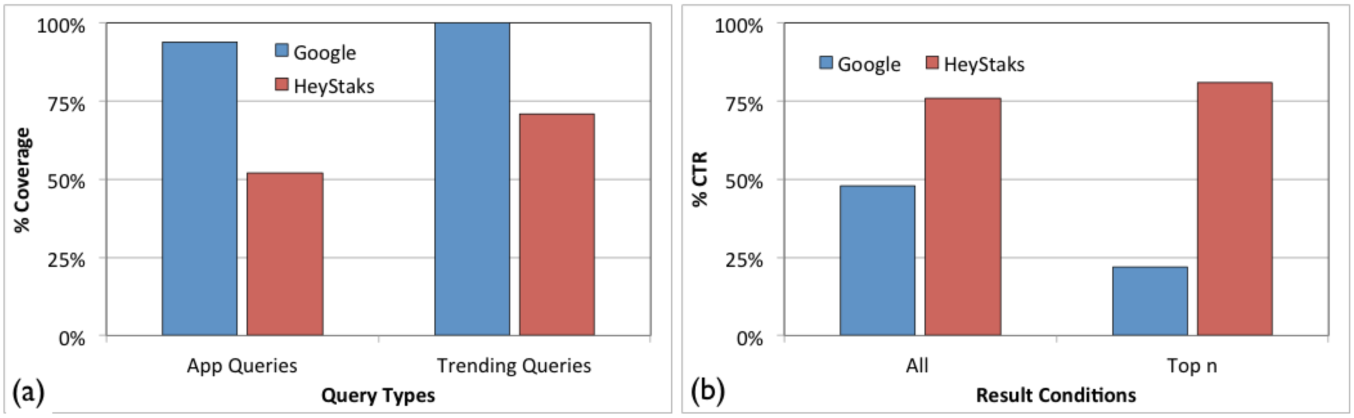


Figure 3: (a) Query coverage for Google and HeyStaks based on in-app and trending queries; (b) CTR for Google and HeyStaks for *All* and *Top n* conditions.

where with free wifi, might search for *"whistler accommodation free wifi"* and benefit from relevant specific results that were found by her friends.

Early versions of HeyStaks delivered social search via a browser plugin, to integrate directly with mainstream search engines at the interface level; examples of this can be seen in [9]. More recently HeyStaks has been deployed through a range of mobile and tablet apps for iOS and Android requiring a different approach to search engine integration via search engine APIs. In the remainder of this section we will provide some examples from the iPad HeyStaks app and summarise the system architecture and basic recommendation components.

2.1 HeyStaks for iPad

Figures 1(a-d) show screenshots from the iPad app. Our user starts with a query for *"hard rock"*; see Figure 1(a). The default results from Google tend towards music and casinos; see Figure 1(b). In this instance our searcher is interested in mountain biking; there is one recent example of this type of result recommended from their personal "My Searches" stak but in addition HeyStaks recommends two staks as possible sources of additional results, one is a "Mountain Biking" stak and the other a "Music" stak; these are presented above the main search results in Figure 1(b). By taping on either of these stak communities the user can join them and benefit from more topically focused recommendations. In this case, the user opts to join the "Mountain Biking" stak and instantly receives a new set of recommendations, this time from the relevant search histories of the most reputable members of this particular stak. The recommendations are inserted at the top of the results list as in Figure 1(c). Finally, the user selects one of the results and views it within the app from where they can chose to tag or share it with their contacts or social networks; see Figure 1(d).

2.2 System Architecture

The HeyStaks system architecture is shown in Figure 2, highlighting two important elements: the client-side apps (browser, mobile, tablet) and the backend HeyStaks server. The client apps provides direct access to key HeyStaks functionality and recommendations and integrate directly with mainstream search engines. The HeyStaks server manages the individual staks (indexing individual pages against query or tag terms and positive or negative votes), the stak database (stak titles, members, descriptions, status, etc.) and the core recommendation and reputation engines.

The recommendation engine at the heart of HeyStaks uses a combination of content-based and social recommendation strategies to identify and rank pages for recommendation to the searcher, based on their target query and stak membership. Its ranking component uses a relevance score to prioritise pages that are similar to the target query and that were frequently selected by stak members for similar search contexts. In addition, this relevance score is combined with the reputation of the stak members who originally sourced the page in question. This reputation score is based on a collaboration model in which reputation propagates between users when results which originated from one user are recommended and selected by another. For the interested reader, these relevance [9] and reputation [4] components have been described and evaluated elsewhere.

3. LIVE-USER EVALUATION

In October 2010 HeyStaks received seed funding to productise its social search technology and bring it to market as a consumer enhancement for traditional search engines. This culminated in key product releases in the second half of 2011. In this section we will summarise recent usage results based on an analysis of approximately 50,000 live users during the first 3 months of 2012, generating an average of about 4 unique searches per user per day. We will focus in particular on the coverage and relevance of HeyStaks' recommendations versus the organic results returned by Google.

3.1 Coverage

By coverage we mean the percentage of search queries for which Google/HeyStaks can return results. Early on coverage was one of the key challenges faced by HeyStaks: in the absence of a critical mass of staks and stak content recommendations were rare and so the potential benefits to new users were limited. This is a classic example of the cold-start problem in conventional recommender systems [7], which we shall return to later.

To evaluate coverage we use two sources of test queries. First there are the queries submitted by the HeyStaks users themselves – we refer to these as *app queries* – and they are important because their coverage tells us how often our current users benefit from HeyStaks recommendations. However, there is the concern that these early-adopter users are probably less representative of future users and so it is also important to understand the likely coverage of HeyStaks as it attracts more mainstream users in the future. To evaluate this coverage we use an alternative set of test queries based on the trending query lists available through Google; we refer to these as *trending queries*.

The results are presented in Figure 3(a). As expected Google offers near perfect coverage for both app and trending queries,; interestingly it fails to hit 100% for the app queries which highlights the somewhat specialised nature of these early-adopter queries. In contrast, HeyStaks does not provide such complete coverage. But it can generate recommendations for more than 50% of app queries and almost 75% of trending queries. In other words, the current users of HeyStaks enjoy at least some recommendations for half of their queries and so the majority of new users will see recommendations during their daily searches. The higher coverage for the trending queries bodes well for future users and reflects well on some of the bootstrapping techniques that have been used to drive coverage during the early stages of deployment (see Section 4).

3.2 Relevance

Query coverage is important but only if the recommendations prove to be relevant. The standard bearer for relevance in web search is the *click-through rate* (CTR) and, all other things being equal, results that attract a higher CTR are likely to be more relevant than those that do not. In this section we calculate two different variations on the CTR measure. First we calculate the CTR across all Google result lists and separately across all HeyStaks recommendation sets. We refer to this as the *All* condition. However, this is imperfect because it is clearly biased towards Google's longer result-lists; for a typical session Google includes 10 results where as HeyStaks includes at most 4 recommendations and usually fewer. As an alternative, we also calculate the CTR for the n HeyStaks recommendations in a session and compare this to the CTR for the top n Google results; we call this the *Top n* condition.

The results shown in Figure 3(b) are clearly very positive. In each condition we can see that HeyStaks enjoys a significantly higher CTR than Google. The difference is particularly striking for the *Top n Results* condition. In this case, HeyStaks enjoys a CTR of more than 80% compared to Google's 22% CTR on a like-for-like basis. This means that HeyStaks results attract selections almost 4 times as frequently as Google, which is surely s strong indicator that HeyStaks' more targeted, social recommendations are proving to be more relevant that the one-size-fits-all organic results from leading mainstream search engines.

4. DISCUSSION

Finally, we would like to briefly outline some of the lessons that have been learned as a result of our deployment experiences over the past 12 months. Many are not unique to the deployment of recommendation apps but we hope that they will nonetheless help others to learn from some of the challenges that we have faced.

Most recommender systems are subject to the cold-start problem and the challenge of making recommendations to users from the start. For HeyStaks this was a considerable challenge since the absence of a critical mass of users and stak content limited recommendations. To partially address this issue we added a number of high-level recommendation triggers or *filters* to staks, basically URL patterns that would trigger the promotion of certain types of organic Google or Bing results. For example, a *Gadgets & Technology* might include filters for *techcrunch.com, theverge.com, or mashable.com* to ensure these results would be promoted when returned by Google or Bing. This provided an opportunity to kick-start recommendations for new users by including reliable filters in newly created staks.

Another important lesson concerned the need to reduce app complexity. The HeyStaks browser apps contain a wide variety of functions, including allowing users to create and maintain their own staks. The motivation here was obviously to use this as an opportunity to crowd-source the early staks and their content. However, this functionality was rarely used and so increased app complexity. The mobile and tablet apps provided a more streamlined service, removing stak creation features, and instead prioritising the recommendation of pre-existing staks for users to join at search time. As well as simplifying the user experience this also helped ensure that new users were recommended high-quality staks on topics that mattered to them.

This simplification limits the opportunity to crowd-source new staks which in turn motivates the development of a semi-automatic stak creation workflow for use in-house. The precise details of this are beyond the scope of this short paper but briefly, by harvesting and clustering trending terms from public search and social network sources it is possible to produce a comprehensive list of stak topics. These topics and their associated seed terms are expanded to produce a comprehensive set of popular queries. And by submitting these queries to a range of search resources (Google, Bing, YouTube, Delicious, etc.) we obtain a large collection of candidate pages. We developed a rating tool to evaluate the quality of these pages, seeding new staks with those pages that passed the ratings test, and extracting recurring filter urls from these pages. Thus it was possible to create a large volume of staks to cover trending topics and ensure that they were bootstrapped with high quality content and thus capable of generating reliable recommendations.

We have provided a brief update on our experiences developing and deploying a recommendation system for social search. Early indications speak to the value of the approach as users benefit from superior search results when compared directly to mainstream search engines like Google. We have also outlined some important deployment lessons that we have learned that may be of interest for other recommender system deployments.

5. ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland under grant 07/CE/I1147 and HeyStaks Technologies Ltd

6. REFERENCES

[1] Brynn M. Evans and Ed H. Chi. An elaborated model of social search. *Inf. Process. Manage.*, 46(6):656–678, 2010.

[2] Brynn M. Evans, Sanjay Kairam, and Peter Pirolli. Do your friends make you smarter?: An analysis of social strategies in online information seeking. *Inf. Process. Manage.*, 46(6):679–692, 2010.

[3] Gene Golovchinsky, Pernilla Qvarfordt, and Jeremy Pickens. Collaborative information seeking. *IEEE Computer*, 42(3):47–51, 2009.

[4] 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.

[5] 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.

[6] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. A comparison of information seeking using search engines and social networks. In *ICWSM*, 2010.

[7] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260, 2002.

[8] 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.

[9] Barry Smyth, Peter Briggs, Maurice Coyle, and Michael P. O'Mahony. Google shared. a case-study in social search. In *UMAP*, pages 283–294, 2009.

[10] Barry Smyth, Maurice Coyle, and Peter Briggs. The altruistic searcher. In *CSE (4)*, pages 360–367, 2009.

[11] Barry Smyth, Maurice Coyle, and Peter Briggs. Communities, collaboration, and recommender systems in personalized web search. In *Recommender Systems Handbook*, pages 579–614. 2011.

[12] Barry Smyth, Jill Freyne, Maurice Coyle, and Peter Briggs. Recommendation as collaboration in web search. *AI Magazine*, 32(3):35–45, 2011.