<table>
<thead>
<tr>
<th>Title</th>
<th>Modeling user and result reputation in collaborative web search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors(s)</td>
<td>McNally, Kevin; O'Mahony, Michael P.; Smyth, Barry</td>
</tr>
<tr>
<td>Publication date</td>
<td>2011-08-31</td>
</tr>
<tr>
<td>Conference details</td>
<td>Paper presented at the 22nd Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2011), University of Ulster, Northern Ireland, 31 August - 2 September, 2011</td>
</tr>
<tr>
<td>Publisher</td>
<td>Intelligent Systems Research Centre</td>
</tr>
<tr>
<td>Item record/more information</td>
<td><a href="http://hdl.handle.net/10197/3449">http://hdl.handle.net/10197/3449</a></td>
</tr>
</tbody>
</table>
Modeling User and Result Reputation in Collaborative Web Search

Kevin McNally, Michael P. O’Mahony, and Barry Smyth

CLARITY: Centre for Sensor Web Technologies,
School of Computer Science and Informatics,
University College Dublin, Belfield, Dublin 4, Ireland
{kevin.mcnally,michael.omahony,barry.smyth}@ucd.ie

Abstract. Employing collaborative recommendation techniques allows for personalization of search results in social web search. If recommendations arise from past user activity, the expertise of those users driving the recommendation process can play an important role when it comes to ensuring recommendation quality. Hence the reputation of users is important, in addition to result relevance as traditionally considered in web search. In this paper we explore this concept of reputation; specifically, investigating how reputation can enhance the recommendation engine at the core of the HeyStaks social search utility. We evaluate a number of different reputation models in the context of the HeyStaks system, and demonstrate how incorporating reputation into the recommendation process can enhance the relevance of results recommended by HeyStaks.

Keywords: Social Search, Relevance, Reputation, HeyStaks

1 Introduction

The early years of web search (1995-1998) were characterised by considerable innovation as researchers came to discover some of the shortcomings of traditional term-based information retrieval techniques in the face of large-scale, heterogeneous web content, and in the face of queries from users who were far from search experts. While traditional term-based matching techniques played an important role in result selection, they were not sufficiently robust when it came to delivering a reliable and relevant ranking of search results. The significant breakthrough that led to modern web search engines came about through the work of Brin and Page [1], and Kleinberg [3], highlighting the importance of link connectivity when it came to understanding the importance of web pages.

By and large the world of web search has remained relatively stable over the past decade or more. Mainstream search engines have innovated around the edges of search but their core approaches have remained intact. However there are signs that this is now changing, as all of the mainstream players look to the world of social networks to provide new types of search content and new sources of ranking signals. There is now considerable interest in the concept of social search, based on the idea that information in our social graphs can be
used to improve mainstream search. For example, the HeyStaks system [11] adds a layer of social search onto mainstream search engines, using recommendation techniques to automatically suggest results to users based on pages that members of their social graphs have found to be interesting for similar queries in the past. HeyStaks adds collaboration to conventional web search and allows us to benefit from the past search histories of people we trust on topics that matter to us.

In this paper we examine the role of reputation in HeyStaks’ recommendation engine. Previously we have described how to estimate the reputation of a searcher by analysing how frequently their past search efforts have translated into useful recommendations for other users [5, 6]. For example, if my previous searches (and the pages that I find) lead to result recommendations to others that are regularly acted on (selected, tagged, shared etc.), then my reputation should increase, whereas if my past search efforts rarely translate into useful recommendations then my reputation should decline. In this paper we expand on previous work by considering a number of user reputation models, showing how these models can be used to estimate result reputation, and comparing the ability of these models to influence recommendation quality based on recent live-user data.

2 Related Work

Recently there has been considerable interest in reputation systems to evaluate user reputation and inter-user trust across social web and e-commerce applications. For example, the reputation system used by eBay has been examined by Jøsang et al. [2] and Resnick et al. [8]. Briefly, eBay elicits feedback from buyers and sellers regarding their interactions with each other, and that information is aggregated in order to calculate user reputation scores. The aim is to reward good behaviour on the site and to improve robustness by leveraging reputation to predict whether a vendor will honour future transactions.

The work of O’Donovan and Smyth [7] addresses reputation in recommender systems. In this case, a standard collaborative filtering algorithm is modified to add a user-user trust score to complement the normal profile or item-based similarity score, so that recommendation partners are chosen from those users that are not only similar to the target user, but who have also had a positive recommendation history with that user. It is posited that reputation can be estimated by measuring the accuracy of a profile at making predictions over time. Using this metric average prediction error is improved by 22%.

Other recent research has examined reputation systems employed in social networking platforms. Lazzari [4] performed a case study of the professional social networking site Naymz. He warns that calculating reputation on a global level allows users who have interacted with only a small number of others to accrue a high degree of reputation, making the system vulnerable to malicious use. Similar to Jøsang et al. [2], Lazzari [4] suggests that vulnerability lies in the site itself, allowing malicious users to game the reputation system for their own ends. However, applying reputation globally affords malicious users influence over the entire system, which adds to its vulnerability. In this paper we consider
reputation models in the context of the HeyStaks social search service which seek to capture the quality of search knowledge that is contributed by users and how this data can be leveraged to improve overall recommendation quality.

3 The HeyStaks Recommendation Engine

In this section we briefly review the HeyStaks recommendation engine to provide sufficient context for this work. Further details can be found in [11] (which focuses on the relevance model) and in [5] (which focuses on the reputation model).

Profiling Stak Pages. Each stak in HeyStaks captures the search activities of its stak members. The basic unit of stak information is a result (URL) and each stak \( S \) is associated with a set of results, \( S = \{r_1, \ldots, r_k\} \). Each result is also anonymously associated with a number of implicit and explicit interest indicators, based on the type of actions (for example, selecting, voting, tagging and sharing) that users can perform on these pages.

These actions can be associated with a degree of confidence that the user finds the page to be relevant. Each result page \( r_i^S \) from stak \( S \), is associated with relevance indicators: the number of times it has been selected \( (Sl) \), the query terms \( (q_1, \ldots, q_n) \) that led to its selection, the terms contained in the snippet of the selected result \( (s_1, \ldots, s_k) \), the number of times it has been tagged \( (Tg) \), the terms used to tag it \( (t_1, \ldots, t_m) \), the votes it has received \( (v^+, v^-) \), and the number of people it has been shared with \( (Sh) \) as per Equation 1.

\[
r_i^S = \{q_1\ldots q_n, s_1\ldots s_k, t_1\ldots t_m, v^+, v^-, Sl, Tg, Sh\}.
\]

Importantly, this means each result page is associated with a set of term data (query and/or tag terms) and a set of usage data (the selection, tag, share, and voting count). The term data provides the basis for retrieving and ranking recommendation candidates. The usage data provides an additional source of evidence that can be used to filter results and to generate a final set of recommendations.

Recommending Search Results. At search time, the query \( q \) and current stak \( S_T \) are used to generate a list of recommendations. Here we discuss recommendation generation from the current stak \( S_T \) only, although recommendations may also come from other staks that the user has joined or created. There are two key steps when it comes to generating recommendations. First, a set of recommendation candidates are retrieved from \( S_T \) based on the overlap between the query terms and the terms used to index each recommendation candidate (query, snippet, and tag terms). These recommendations are then filtered and ranked. Results that do not exceed certain activity thresholds are eliminated; such as, for example, results with only a single selection or results with more negative votes than positive votes (see [11]). Remaining recommendation candidates are then ranked according to a weighted score of its relevance and reputation (Equation 2), where \( w \) is used to adjust the relative influence of relevance and reputation.

\[
score(r, q) = w \times rep(r, t) + (1 - w) \times rel(q, r).
\]
The relevance of a result $r$ with respect to a query $q$ is computed using TF-IDF [9], which gives high weights to terms that are popular for a result $r$ but rare across other stak results, thereby serving to prioritise results that match distinguishing index terms.

The reputation of a result $r$ at time $t$ ($\text{rep}(r, t)$) is an orthogonal measure of recommendation quality. The intuition is that we should prefer results that originate from more reputable stak members. We explore user reputation and how it can be computed in the next section.

4 Reputation Models for Social Search

HeyStaks searchers themselves play a crucial role in determining what gets recommended and to whom, and so the expertise of these searchers is an important factor to consider. The key idea is that the quality of shared search knowledge can be estimated by looking at the search collaborations that naturally occur within HeyStaks. If HeyStaks recommends a result to a searcher, and the searcher chooses to act on this result (i.e. select, tag, vote or share), then we can view this as an instance of search collaboration. The current searcher who acts on the recommendation is known as the consumer and the original searcher(s), whose earlier action on this result caused it to be added to the search stak, and ultimately recommended, is known as the producer. In other words, the producer created search knowledge that was deemed to be relevant enough to be recommended and useful enough for the consumer to act upon it. The basic idea behind our reputation models is that this act of implicit collaboration between producer and consumer confers some unit of reputation on the producer.

4.1 Graph-Based Reputation Models

We can treat the collaborations that occur among users as a type of graph. Each node represents a unique user and the edges represent collaborations between pairs of users. These edges are directed to reflect the producer/consumer relationships and reputation flows along these edges, and is aggregated at the nodes. We now present two graph-based reputation model alternatives.

**Reputation as a Weighted Sum of Collaboration Events.** Producer reputation is calculated as a weighted sum of the collaboration events in which they have participated. Consider the selection of result $r$ by consumer $c$ at time $t$. The producers responsible for this result recommendation are given by $\text{producers}(r, t)$ (Equation 3) such that each $p_i$ denotes a specific user $u_i$ in a specific stak $S_j$.

$$\text{producers}(r, t) = \{p_1, \ldots, p_k\}. \quad (3)$$

Then, for each producer of $r$, $p_i$, we update its reputation as in Equation 4. In this way reputation is shared equally among its $k$ contributing producers.

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

As users participate in more and more collaboration events, their reputation grows over time. See [5] for further details on this approach.
Reputation as PageRank. PageRank [1] can be applied to compute the reputation of HeyStaks users, which take the place of web pages in the graph. When a collaboration event occurs, directed links are inserted from the consumer to each producer. Once all collaboration events up to some point in time, $t$, have been captured, the reputation of each user $p_i$ at time $t$ is given by:

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

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

4.2 Towards Result Reputation

The recommendation engine described in Section 3 scores each recommendation candidate based on how relevant it is to the target query. If reputation is to influence recommendation ranking, as well as relevance, then we need to transform the above user-based reputation measures into a result-based reputation measure. We now consider two approaches to compute result reputation.

Max Reputation. The reputation of a result $r$ (at time $t$) is simply the maximum reputation of its associated producers; see Equation 6. Scoring results in this way is advantageous as the reputation of a page will not be harmed if, for example, many new, not yet reputable users have selected the page.

$$\text{rep}(r, t) = \max_{p_i \in \{p_1, ..., p_k\}} (\text{rep}(p_i, t)).$$

Hooper’s Reputation. Hooper’s Rule for Concurrent Testimony was proposed to calculate the credibility of human testimony [10]. Hooper gives to a report a credibility of $1 - (1 - c)^k$, assuming $k$ reporters, each with a credibility of $c$ $(0 \leq c \leq 1)$. For HeyStaks, result reputation can be determined by performing a similar calculation across the reputation scores if its producers.

We can now evaluate the quality of a recommendation candidate by its relevance and reputation scores, which we combine using Equation 2 to calculate the rank score of a result $r$ and its producers $p_1, ..., p_k$ at time $t$, for query $q$.

5 Evaluation

The purpose of this paper has been to propose a number of alternatives to calculating the reputation of content based on that of its producers who are helping other users (consumers) to search within the HeyStaks social search service. The hypothesis is that by allowing reputation, as well as relevance, to influence the ranking of result recommendations, we can improve the overall
quality of search results. In this section we evaluate our page reputation models using data generated during a recent closed, live-user trial of HeyStaks, designed to evaluate the utility of the HeyStaks brand of collaborative search in fact-finding information discovery tasks.

5.1 Dataset and Methodology

Our live-user trial involved 64 first-year undergraduate university students with varying degrees of search expertise. Users were asked to participate in a general knowledge quiz, during a supervised laboratory session, answering as many questions as they could from a set of 20 questions in the space of 1 hour. Each student received the same set of questions which were randomly presented to avoid any ordering bias. See [6] for a list of questions used in the trial.

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

In recent work we described the performance results of this trial showing how larger groups tended to benefit from the increased collaboration effects of HeyStaks [6]. For example, members of shared staks answered significantly more
questions correctly, and with fewer queries, than the members of solitary staks who did not benefit from collaboration. In this paper we are interested in exploring reputation. No reputation model was used during the live-user trial and so recommendations were ranked based on relevance only. However the data produced makes it possible for us to effectively replay the user trial so that we can construct our reputation models and use them to re-rank HeyStaks recommendations. We can retrospectively test the quality of re-ranked results versus the original ranking against a ground-truth relevance; since as part of the post-trial analysis, each selected result was manually classified as relevant (the result contained the answer to a question), partially relevant (the result referred to an answer, but not explicitly), or not-relevant (the result did not contain an explicit or implicit reference to an answer) by experts.

5.2 User Reputation

To get a sense of how users were scored by the two reputation models described in Section 4.1, we now examine the type of user reputation values that are generated from the trial data. In Figure 1, box-plots are shown for the reputation scores across the 4 shared staks and for each reputation model. Here we see that for the WeightedSum model there is a clear difference in the median reputation score for members of the 5 person stak when compared to members of the larger staks. This is not evident in results for the PageRank model, which shows very similar reputation scores, regardless of stak size.

Further, a desirable property of a reputation model is that it should capture consumption diversity, meaning that in order for producers to gain high reputation, many consumers should benefit from the content that producers contribute to staks. Table 1 shows the Pearson correlation between the number of distinct consumers per producer (per stak) and producer reputation according to the two user reputation models tested. Across all staks, PageRank better reflects consumption diversity, indicating that this model is particularly effective in this regard. This is to be expected, given that user PageRank scores are directly influenced by the number of consumers interacting with them. However, WeightedSum does capture consumption diversity more effectively as stak size increases. In future work, we plan on refining our WeightedSum model in order to better reflect consumption diversity for such small-sized staks.

5.3 From Reputation to Quality

The true test of the user and result reputation models is the extent to which they improve in the quality of results recommended by HeyStaks. We have de-

<table>
<thead>
<tr>
<th>Stak Size</th>
<th>5</th>
<th>9</th>
<th>19</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>WeightedSum</td>
<td>0.41</td>
<td>0.52</td>
<td>0.78</td>
<td>0.85</td>
</tr>
<tr>
<td>PageRank</td>
<td>0.75</td>
<td>0.58</td>
<td>0.85</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table 1. Correlations between the number of distinct consumers per producer per stak and producer reputation.
scribed how HeyStaks combines term-based relevance and user reputation to generate its recommendation rankings; see Equation 2. For the purpose of this evaluation we regenerate each of the recommendation lists produced during the trial using the four possible combinations of user and result reputation models: WeightedSum and PageRank, each complemented with the Max and Hooper result reputation models. Since we have ground-truth relevance information for all of the recommendations (relative to the quiz questions), we can then determine the quality of the resulting recommendations. Specifically, we focus on the top recommended result and note whether it is relevant (that is, contains the answer to the question) or not relevant (does not contain the answer to the question). For each reputation model we compute an overall relevance rate, as the ratio of the percentage of recommendation sessions where the top result was deemed to be relevant, to the percentage of those where the top result was not-relevant. Moreover, we can compare this to the relevance rate of the recommendations made by the standard HeyStaks ranking (i.e. when $w = 0$ in Equation 2) in the trial to compute an overall relevance benefit; such that a relevance benefit of 40%, for a given reputation model, means that this model generated 40% more relevant recommendations than the standard HeyStaks ranking scheme.

Figure 2 presents a graph of relevance benefit versus the weighting ($w$) used in Equation 2 to adjust the influence of term-based relevance versus reputation during recommendation. The results for each combination of user and result reputation model indicate a significant benefit in recommendation quality when compared to the standard HeyStaks recommendations. As we increase the influence of reputation over relevance during recommendation (by increasing $w$) we see a consistent increase in the relevance benefit, up to values of $w$ in the range 0.4–0.8. For example, we can see that for $w = 0.5$, the reputation models are driving a relative improvement in recommendation relevance of about 30–50% compared to default HeyStaks’ relevance-only based recommendations. The Weighted Sum model, when paired with the Hooper result reputation model, was the best performing technique, peaking twice at $w = 0.4$ and $w = 0.8$, each time achieving around 55% relevance benefit. Hooper’s model also fared well when combined with PageRank, achieving a relevance benefit of 50% at $w = 0.8$. This leads us to believe that Hooper may be the most suitable option for result reputation. The score it produces for a result is a consensus based on the reputation of its producers. The model promotes the idea that a result will have a high score by way of reinforcement from its producers, assuming they are reputable.

5.4 Limitations

In this evaluation we have compared a number of reputation models based on live-user search data. One limitation of this approach is that although the evaluation uses live-user search data, the final recommendations are not themselves evaluated using live-users. Instead we replay users’ searches to generate reputation-enhanced recommendations. The main reason for this is the difficulty in securing sufficiently many live-users for a trial of this nature, which combines
a number of reputation models and therefore a number of experimental conditions. That being said, our evaluation methodology is sound since we evaluate the final recommendations with respect to their ground-truth relevance; we have an objective measure of page relevance based on the Q&A nature of the trial and we use this to evaluate the genuine relevance of the final recommendations.

Another point worth noting is that the live-user trial is limited to a specific type of search task, in this case a Q&A search task. As such it would be unsafe to draw general conclusions in relation to other more open-ended search tasks. However, this type of focused search task is not uncommon among web searchers and as such we feel it represents an important and suitable use-case that is worthy of evaluation. Moreover, previous work [11] has looked at the role of HeyStaks in more open-ended search tasks to note related benefits to end-users from its default relevance-based recommendations. In future work, we will deploy and evaluate our reputation model across similar general-purpose search tasks.

6 Conclusions

In this paper we have described a number of user and result reputation models designed to mediate result recommendation in collaborative search systems. We have described the results of a comparative evaluation in the context of real-user data which highlights the ability of these models to improve overall recommendation quality, when combined with conventional recommendation ranking metrics. We believe that this work lays the groundwork for future research in this area which will focus on scaling-up the role of reputation in HeyStaks and refining the combination of relevance and reputation during recommendation.

Currently, in HeyStaks the identity of users (producers and consumers) is not revealed and so users do not see where their recommendations come from. In the future it may be appropriate to relax this anonymity condition in certain circumstances (under user control). By doing so it will then be possible for individual users to better understand the source of their recommendations and
the reputation of their collaborating users. As such this model can ultimately lead to the formation of trust-based relationships via search collaboration.

7 Acknowledgments

This work is supported by Science Foundation Ireland under Grant 07/CE/I1147. Our thanks goes to Prof. Pádraig Cunningham of University College Dublin for his advice and comments during the preparation of the work.

References