Noname manuscript No.
(will be inserted by the editor)

A Comparative Study of Collaboration-based Reputation Models for Social Recommender Systems

Kevin McNally · Michael P. O'Mahony · Barry Smyth

Received: date / Accepted: date

Abstract Today, people increasingly leverage their online social networks to discover meaningful and relevant information, products and services. Thus, the ability to identify reputable online contacts with whom to interact has become ever more important. In this work we describe a generic approach to modeling user and item reputation in social recommender systems. In particular, we show how the various interactions between producers and consumers of content can be used to create so-called *collaboration graphs*, from which the reputation of users and items can be derived. We analyze the performance of our reputation models in the context of the HeyStaks social search platform, which is designed to complement mainstream search engines by recommending relevant pages to users based on the past experiences of search communities. By incorporating reputation into the existing HeyStaks recommendation framework, we demonstrate that the relevance of HeyStaks recommendations can be significantly improved based on data recorded during a live-user trial of the system.

Keywords Reputation · Social Recommender Systems · Collaboration Graphs

1 Introduction

As the online world has matured, the nature of the world-wide web has evolved. During the 1990s the web was about pages and links. Today it is as much about communities and

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

Kevin McNally
CLARITY Centre for Sensor Web Technologies
School of Computer Science and Informatics, University College Dublin, Ireland
E-mail: kevin.mcnally@ucd.ie

Michael P. O'Mahony
CLARITY Centre for Sensor Web Technologies
School of Computer Science and Informatics, University College Dublin, Ireland
E-mail: michael.omahony@ucd.ie

Barry Smyth
CLARITY Centre for Sensor Web Technologies
School of Computer Science and Informatics, University College Dublin, Ireland
E-mail: barry.smyth@ucd.ie

social collaboration. Indeed over the past decade the power of the web as a collaboration platform has been dramatically demonstrated time and time again and its collective wisdom has been applied to a diverse set of challenging scenarios, from the indexing of images to the creation of shared repositories of authoritative content (Lih, 2004; von Ahn, 2006; Siorpaes and Hepp, 2008; Hacker and von Ahn, 2009; Law and von Ahn, 2009; Walsh and Golbeck, 2010; De Alfaro et al, 2011; Goh et al, 2011; Guy et al, 2011). Importantly, the web, which heretofore created a network of content, has now matured to accommodate a network of relationships. Today most online users are connected to a variety of other people, near and far, by a complex and mature social graph, whether to our friends and classmates via Facebook, colleagues via LinkedIn, or our peers through services like Twitter.

In the early days of the *web of content*, the ability to evaluate content quality rapidly became a pressing need in order to help people locate the right information at the right time. Perhaps the single most important innovation of the web of content was the development of techniques for rating the authority of content in order to filter and rank pages during web search (Brin and Page, 1998; Kleinberg, 1999). This led directly to the age of web search and the success of search engines like Google. As the social web matures, and as our social graphs explode, we are faced with a similar need: the need to filter and rank users and relationships, which forms the basis of our social graph, which is increasingly playing the role of the key information filter in the social web of shared knowledge. As a result researchers have begun to explore the idea of trust and reputation on the social web, as an analogue for authority on the web of content (Kempe et al, 2003; Golbeck and Hendler, 2004; Zeng et al, 2006; Jøsang et al, 2007; Kuter and Golbeck, 2007; Wu and Tsang, 2008; O'Donovan, 2009; Duan et al, 2010; Lazzari, 2010; McNally et al, 2011; Kuter and Golbeck, 2010; Weng et al, 2010; Bakshy et al, 2011; Canini et al, 2011; Recuero et al, 2011; Pal and Counts, 2011; Cai et al, 2011). In this work we are also interested in reputation in the social web and the main contribution of this work is a description and evaluation of a set of reputation models that can be used to estimate the reputation of collaborating users and to use this information to infer the quality of the content or items that they interact with.

The perspective that drives this work is that the social web, at its heart, is about *collaboration* in the broadest sense of the word. For example, when a group of users edit a Wikipedia page they engage in a form of collaboration. Collaboration also occurs when an Amazon shopper acts on a recommendation derived from the ratings of other users. And when a user re-tweets another user we again see a form of collaboration in action. These are all examples of a *collaboration event*. They can occur anonymously as with Amazon and often Wikipedia, or we might know our collaborators, as with Twitter. Consequently, the key idea behind our reputation framework is that that the reputation of a user is based on the collaboration events that they participate in, and these collaboration events are the *fundamental unit of reputation*.

Very briefly, each collaboration event involves at least two users, a so-called *producer* and a *consumer* and the event represents some action by the consumer on an item/piece of content/asset contributed by the producer. For example, when Bob edits a Wikipedia article created by Mary, Bob is the consumer, Mary is the producer and Bob is effectively collaborating with Mary, albeit implicitly and without the need for explicit coordination beyond the collaboration platform provided by Wikipedia. Likewise, when Tom re-tweets a message posted by Suzie, a form of collaboration is occuring, with Tom the consumer and Suzie the producer. Of course, consumers may in turn become producers; if Tom's re-tweet is again re-tweeted by Alan then Tom is now a producer and Alan a consumer.

As we shall see this collaboration perspective means that we can represent the activities of users as a type of collaboration graph, the nodes of which represent users, and with

directed edges connecting producers to consumers. As new collaboration events occur, additional nodes and edges are created. Reputation can be inferred by examining the structure and dynamics of this collaboration graph. As collaboration events occur weighted edges from consumers to producers are inserted, with reputation accumulating at the graph nodes. In previous work we have presented a proof-of-concept of this approach to reputation modeling, focusing on one particular approach to accumulating reputation (McNally et al, 2010, 2011). In this paper we broaden this perspective in a number of important respects. Firstly we generalise our approach and describe a number of *user reputation* models based on the collaboration graph. In turn, we describe how these models can influence the recommendation of information to users as part of a recommender system, and evaluate a variety of different *item reputation* models that form the basis of this approach to recommendation.

While in this paper we are primarily concerned with computational models of reputation applied in an online setting, there is a long history of analogous work, particularly in the sociology literature, when considering the nature and role of trust and reputation in communities and societies; (Raub and Weesie, 1990; Mayer et al, 1995; McKnight and Chervany, 1996; Rousseau et al, 1998; Gambetta et al, 2000). We draw on this work, as well as related work in the online space, to evaluate this reputation framework in the context of a collaborative approach to Web Search (Speretta and Gauch, 2005; Morris and Horvitz, 2007a; Smyth, 2007; Morris and Horvitz, 2007b; Amershi and Morris, 2008). Our target system, HeyStaks, provides a novel approach to web search by adding a collaboration layer to mainstream search engines via a browser plugin. Briefly, HeyStaks facilitates implicit collaboration between searchers (Smyth et al, 2009). For example, when Nicola searches, in addition to Google's mainstream search results, she may receive recommendations from her social graph; these recommendations are results that other searchers have found to be relevant for similar queries. If Nicola selects one of these recommendations she is effectively engaged in a collaboration with the original searcher. In this paper we describe how our model for calculating reputation can be incorporated into HeyStaks' default recommendation engine and show that it has the potential to significantly enhance the quality of recommendations by evaluating it on live-user search data.

The remainder of this paper is organised as follows. In Section 2 we review recent related work on trust and reputation in the social web, highlighting a number of different approaches to measuring, analysing, and utilising this type of information across a variety of online services. Following this, in Section 3, we describe our reputation framework in detail, formalising the notion of collaboration, producers and consumers, and describing how the collaboration graph can be constructed based on the interactions between users in online social platforms. Section 4 describes a number of user reputation models that are derived from the collaboration graph, and how user reputation can be used to create models of item reputation. We go on to explain how these item reputation models can be applied to existing recommendation systems, effectively providing a complementary dimension to user/item similarity during recommendation. In Section 5 we describe the HeyStaks social search system as a test-bed for our approach to modeling user and item reputation. Finally, we give a detailed account of a live-user trial in Section 6, the data from which serves as the raw material for our reputation analysis and evaluation. We demonstrate, for example, that reputation can be captured effectively by the models described herein, and we show how different approaches to harnessing this reputation can positively influence recommendation quality. Finally, having explored this reputation design space, we go on to highlight how a specific reputation model instance stands out in terms of model performance and recommendation efficacy.

2 Background

In the early days of the web, the proliferation of online content meant it became increasingly difficult for people to distinguish between what content was useful and what was useless or even harmful. The work of Brin and Page (1998) and Kleinberg (1999) was significant in understanding that link connectivity can provide a very useful signal when it comes to the relative importance of web pages. Nowadays mainstream search engine technologies are largely based on this idea of connectivity. Around the same time, the popularity of online marketplaces exploded: Online platforms that allowed business-to-consumer or consumer-to-consumer transactions to take place became a key service provided by the web. Whereas traditionally interaction in business took place face-to-face, sites such as eBay and Amazon began to allow people to interact with each other remotely and even without interacting users knowing each other in the offline world. This presented a problem: How do buyers know which sellers are reliable and which are not? Where Google was effectively differentiating between reliable and unreliable pages, these online marketplaces needed a mechanism that distinguished reliable from unreliable users (Dellarocas, 2003).

2.1 Trust and Reputation in Societies

As mentioned previously, reputation, influence and trust have long been a topic of research within the sociology community as researchers have considered the origins of reputation, and the nature of trust between individual groups within societies. For example, early work by Raub and Weesie (1990) examined the origin of reputation within communities, highlighting the emergence of reputation when community members are knowledgeable of past actions. Moreover this work studied the effect of reputation on the efficiency of community interactions, concluding that knowledge of past performances drive reputation and facilitate more efficient interactions. Conversely they found that delays in the dissemination of this knowledge negatively impacts interaction efficiency. Considerable attention has been devoted to understanding the related matter of the trust between users in social contexts. For example, Gambetta et al (2000) explored the nature of trust in the context of social cooperation. Gambetta argues that when we say someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial, or at least not harmful to us, is high enough for us to consider engaging in some form of cooperation with him. As a result our beliefs about an individual can have greater importance than our motives for cooperation.

Mayer et al (1995) further unpacks the nature of trust relationships between individuals in cooperative contexts, by examining related factors such as confidence, predictability, and risk. Moreover, this work highlights the important issue of context in a trust-based relationship and sensitivity of trust to changes in context. An individual might trust another in a certain context but not in a different context, for example. Similar ideas are explored by Rousseau et al (1998), who also propose that there is no single definition of trust, that it is a concept that is always in transition, and that it evolves as our interactions with others evolve; see also the work of McKnight and Chervany (1996).

2.2 Defining Online Trust and Reputation

The main focus of this work, however, is on reputation in the online world. Much early research articulated the difficulty people have in determining the reliability of others in online scenarios (Jones et al, 2000; Olson and Olson, 2000; Resnick et al, 2000; Shneiderman, 2000). It focused on the idea that the system itself could not make the necessary decisions. Reliability of users was subjective and often relied on external factors (such as whether both participants in an online transaction behaved honorably). The online community itself should be allowed to determine who would or would not cause harm to others in the future. The research instead placed emphasis on the importance of facilitating the development of trust-based relationships between users. One user trusts another if they believe that any future transaction will be rewarding rather than detrimental (Golbeck and Hendler, 2004). However, how can two people make this determination if they have never interacted with each other before? There is a need for trust information to be publicly accessible so users can draw on the experiences of others. In order for a user to trust a stranger, they need to know how reputable they are (Resnick et al, 2000).

A key idea in this work is that the reputation of a single user can be deduced from looking at the outcome of transactions they have taken part in. If a user has made good on transactions with others in the past, it is likely they will continue to do so, and thus they are reputable in their community. The discussion on reputation put forth by Jøsang et al (2007) speaks to this definition: "Reputation can be considered as a collective measure of trustworthiness (in the sense of reliability) based on the referrals or ratings from members in a community." O'Donovan and Smyth (2005) come to a similar conclusion about how reputation can be couched in terms of trust. They believe that reputation is a function of trust, and both can be computed over time. From looking at this early work a view of the relationship between the concepts of online trust and reputation becomes clear: Trust occurs between a pair of users and is brought about by one or both users performing well in the context of some social or transactional contract. For example buyers and sellers will come to trust each other on eBay if a given transaction goes to plan. But if one or more parties break this implicit contract then trust will break down. It improves as more positive interactions take place between the pair. The reputation of a user can be seen as an aggregation of the trust that they have established with the set of users they have interacted with in the past. This research strongly indicates that although online trust and reputation can be distilled as concepts in their own right, they are closely interlinked, as explained by Mui et al (2002).

2.3 Transactional Feedback

Measuring and communicating this kind of reputation allows new users to base their decisions on the experiences of others. For example a new eBay buyer will trust a seller if there is information available that indicates this seller has performed well in a similar role in the past, even though the new buyer has never dealt with this seller themselves. However, in order to measure trust and reputation, they must be established as quantifiable entities. Early online reputation systems aggregated information users gave about seller performance and explicitly displayed it as an overall reputation score (Resnick et al, 2000). These early systems, largely employed by e-business websites, were based on aggregating buyer feedback data and displaying it as part of a seller's profile. The reputation system employed by eBay was one of the earliest widely popular systems of its kind.

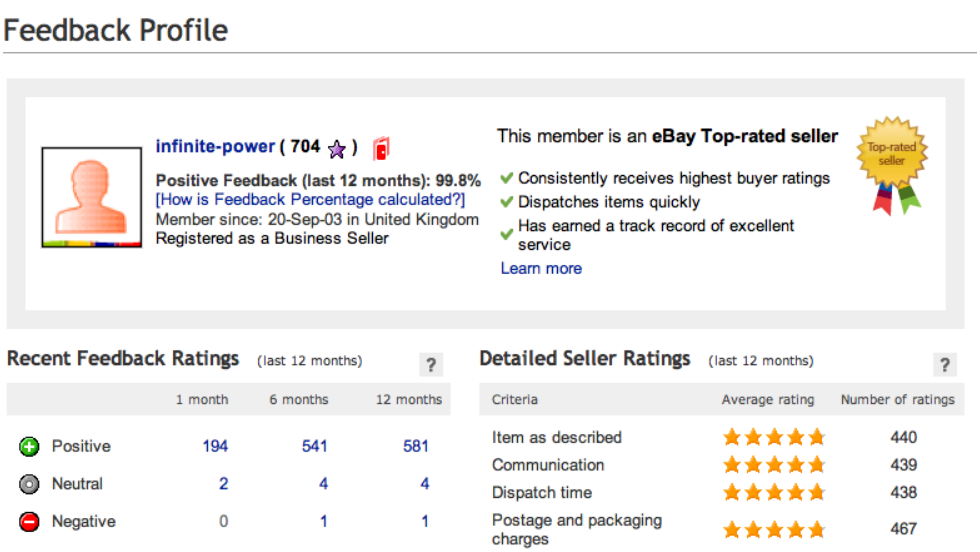


Fig. 1 An example profile page on eBay showing a seller's reputation score and feedback history.

This feedback-based mechanism has been examined by Resnick and Zeckhauser (2002) and later by Jøsang et al (2007). The system 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. An example profile page showing seller reputation can be seen in Figure 1. However, Resnick and Zeckhauser (2002) found that using information received directly from users to calculate reputation is not without its problems. Feedback is generally reciprocal; users almost always give positive feedback if they themselves had received positive feedback from the person they performed a transaction with. In many of these cases the information given is false, therefore reputation is not always a reliable indicator of future vendor performance. Jøsang et al (2007) confirms this, stating such systems require manual curation and protection from malicious users. For further discussion on how trust and reputation information can enhance system robustness, see Jøsang and Golbeck (2009); Lazzari (2010).

Since the development of these simple feedback-based mechanisms, work has been conducted to investigate how trust information can be utilized to improve specific platforms in different ways, for example by improving the quality of recommendations made by collaborative filtering systems (Massa and Bhattacharjee, 2004; O'Donovan and Smyth, 2005; Massa and Avesani, 2007; Kuter and Golbeck, 2010), enhancing the robustness of collaborative content websites like Wikipedia (Chatterjee et al, 2008; De Alfaro et al, 2011) and, more recently, incentivizing participation in online social networks (Yang et al, 2008; Lazzari, 2010; Li et al, 2011). In each case the interaction between users occurs in the context of some item or service or piece of content, and the quality of this interaction is measured as the basis of trust. Some of these systems focus only on calculating trust scores between pairs of users (O'Donovan and Smyth, 2005), some propagate these trust scores across a network (Guha et al, 2004; Kuter and Golbeck, 2010), and some aggregate these scores to achieve an overall reputation score for each individual (De Alfaro et al, 2011; Li et al, 2011). Each

method for gathering trust or reputation information is platform-specific, but all gather information by examining user-user interaction in the system being augmented. Specifically, one user provides some piece of information or service, and another user takes action on either the producing user or the produced item, from which trust information can be inferred.

2.4 Trust in Recommender Systems

In some cases it is not possible to gain trust information from the user directly, and many systems that rely on user interaction do not provide feedback functionality, so statements of trusts must be derived from implicit, indirect actions. An example of a mechanism for measuring trust based on implicit, indirect user interaction is given in the work of O'Donovan and Smyth (2005). The authors address the idea of interpersonal, context sensitive trust in recommender systems. Trust between users is not calculated by examining feedback received directly from users. Instead it is inferred that one user trusts another if their ratings can be used to reliably predict the other's rating for an item. The 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. O'Donovan and Smyth posit that trust can be estimated by measuring the accuracy of a profile at making predictions over time. Using this approach average prediction error is improved by 22% on standardized test sets.

Using socially-generated information such as the content of reviews and genre information to complement collaborative filtering systems was introduced by Basu et al (1998). This idea has been extended to infer trust and reputation information by augmenting the interface to allow users to directly and explicitly give feedback to other users, then aggregating this feedback in some way. Golbeck (2006) developed a trust-based recommender in which neighbours are not selected based on ratings similarity but rather based on explicitly provided trust data. Similar to Golbeck (2006), Avesani et al (2005) propose a trust algorithm called *MoleTrust* that can be used to augment an existing collaborative filtering system. The mechanism calculates a "trust metric" based on explicit, direct feedback given by users, which propagates across a network of content producers. This algorithm can be tuned to propagate over a specific depth across a social graph, controlling the prediction that one user trusts another even though they may have not explicitly interacted. They find that *MoleTrust* can improve the accuracy of predictions made by a recommender system, even in cases where users have provided few ratings. The authors provided similar work using data gained from the Epinions[1] consumer review website, a platform that again allows users to give explicit trust information regarding other users in the system (see Massa and Bhattacharjee, 2004; Massa and Avesani, 2007). The ideas put forward by the creators of MoleTrust have since been extended by Kuter and Golbeck (2010) to introduce a locally determined trust model known as SUNNY which uses a Bayesian approach to assign trust scores to a network of users. The algorithm utilizes explicit trust information provided by users of the FilmTrust network[2].

[1] http://www.epinions.com.

[2] http://trust.mindswap.org/FilmTrust.

2.5 Reputation and Collaborative Content Creation

Although collaborative filtering systems involve collaboration between users, the content with which the users interact is often not created by the users themselves (e.g. books, movies, music etc.). In scenarios where users do create content for public consumption, often finding the consensus is vitally important to ensure items of content are of the highest possible quality. The online encyclopedia Wikipedia[3] allows for this kind of collaborative creation of content. Due to the popularity of the site – the website's steady growth has resulted in the creation of nearly 4 million articles as of 2012[4] – manual maintenance of each article is becoming increasingly unfeasible. As such, research has been conducted that explores the idea of applying a reputation model for both contributors and articles. However, Wikipedia does not provide content authors with the facility to directly communicate their level of trust in others contributing to the system.

Work by Zeng et al (2006), Chatterjee et al (2008) and De Alfaro et al (2011) propose a novel way to infer overall reputation scores by examining the revision history of articles. It is suggested the extent to which one user edits another's content is in inverse proportion to the level of trust they have in that user's content, and thus the user himself. The earlier work of Zeng et al (2006) focuses on calculating article trustworthiness, but De Alfaro et al (2011) extend the idea to model contributor reputation. Each contributor's score, and thus the reputation of the content they provide, depends on how long their revisions survive before editing: If an article is fundamentally changed and that revision is accepted for a long amount of time, the reputation of the original author suffers while that of the reviser is boosted. This temporal, asymmetric aspect of the model promotes robustness as a user cannot instantly obtain high reputation, rather they have to earn it over time.

2.6 Trust and Reputation on the Social Web

Online interaction between users is nowhere more free than on today's most popular Social Web platforms. Trust between users, and indeed user reputation, can be inferred in many different ways using any number of signals. Work has been carried out to propagate scores across a network of users where trust information is given explicitly and directly by users on websites such as Epinions (Guha et al, 2004) and more recently the professional social network Naymz[5] (Lazzari, 2010). The author cautions that calculating reputation on a global level allows users who have interacted with only a small number of others to potentially accrue a high degree of reputation, making the system vulnerable to malicious use. Similar conclusions are made by Jøsang et al (2007), who suggest that vulnerability lies in the site itself, allowing malicious users to game the reputation system for their own ends.

Work conducted by Pal and Counts (2011) focuses on identifying topics on Twitter and ranking people according to their authority on that topic. They measure a number of different features related to a user's account such as number of original tweets, number of links shared and number of topically active followers. They examine "re-tweet impact" – a term described as the ratio of tweets a user has re-tweeted to the number of times that user's content has been re-tweeted by others. The authors' algorithm clusters users by topic, then ranks users according to a set of metrics that might be a good indication of authority in

[3] http://www.wikipedia.org.
[4] http://en.wikipedia.org/wiki/Wikipedia:Size_comparisons.
[5] http://www.naymz.com.

that topic; for example, the number of times a user's topically relevant tweets have been retweeted by others. Similar work was carried out by Weng et al (2010), where twitter users are assigned a score based partly on the idea of homophily, specifically how many people they follow who also follow them. Topic-sensitive graphs are constructed based on the content of users' tweets, and users are ranked according to the relative influence of others in their topic-based subgraph. Similar work has focused on homophily in large-scale instant messaging networks (Aral et al, 2009; Schaal et al, 2010). The latter work focuses on processing the implicit feedback between blogs and distinguishing between two types of signals per user with respect to their roles as authors and sources of feedback. More recently, Canini et al (2011) have examined ranking users of Twitter based on users' credibility per topic. This approach would allow users to input a search query, returning users who frequently tweet using those query terms, ranked by their credibility to the related topic. For similar work on influence on the Social Web, see Duan et al (2010); Bakshy et al (2011); Cai et al (2011).

2.7 Summary Discussion

In summary, the motivation behind applying reputation systems, as evidenced by this recent work, is to incentivize users to engage online with others and thus with the platforms that employ such systems. This can be aided by users trusting the system (Joinson, 2008), but also by rewarding users with positive feedback (Cheng and Vassileva, 2005) or with improved social standing (Recuero et al, 2011). It stands to reason that, if successful, a reputation system can not only distinguish trustworthy users from untrustworthy ones, but also determine the quality of the resources users provide. At their core, many of these systems calculate trust or reputation by building a network based on some explicitly or implicitly gained information in a specific context. Each incident edge in the network tells us something about how one user trusts another. To date all of these related works have developed specialised trust/reputation models in a specific setting or context. A key idea that underpins many of these approaches is the idea of collaboration, either implicit or explicit. Even in occasions where trust scores are propagated throughout a network, some form of user collaboration is required to infer trust. In this paper, we propose a generalized model of reputation by building a graph based not on trust, but on user collaboration: If one user produces a piece of content that is subsequently consumed by another user, their reputation increases. In the following section we describe how the interactions of users in online social platforms can be captured using a collaboration graph and then discuss how reputation can be calculated by examining this graph. By using the social search utility HeyStaks as a case study, we show how the model can be leveraged to positively influence the relevance of recommendations made by HeyStaks to users.

3 Collaboration in the Social Web

The idea of online collaboration has led to the democratization of content creation and consumption. Whereas once people spent large sums of money to buy a large encyclopedia written by a few choice experts, now authors collaborate online to create articles on Wikipedia. Similarly, users are regularly sharing content of different types with each other; from images

on sites like Imgur[6] and Flickr[7], news articles on Digg[8] and Reddit[9], or their knowledge of specific concepts on the StackExchange Network[10]. Such behaviour is becoming increasingly commonplace due to the proliferation of social media platforms. For example, the introduction of Facebook's "like" functionality has resulted in users endorsing content from their community 2.7 billion times daily, as of 2012[11]. Users are also producing content at staggering rates: over 200 million tweets are broadcast to the Twitter community on a daily basis[12], and a recent study by Phelan et al (2011) has shown that around 22% of these contain a URL to offsite content. In a single month in 2010, the social news website Reddit saw users post over 350,000 pieces of content. These posts are ranked according to the number of positive and negative votes they receive from users – an explicit indication of their quality as perceived by the community. In that month users voted on content over 3.5 million times[13]. Users can collaborate in groups on a single piece of information that is then offered for public consumption. On Wikipedia[14] users have created over 3.7 million articles in English alone, many of which have been revised and edited by multiple authors.

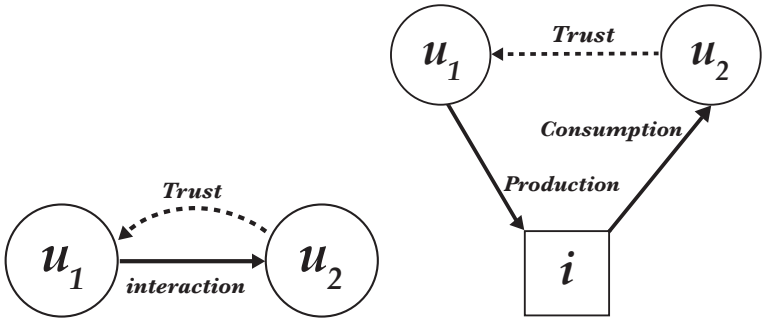
In the previous section we discussed various trust and reputation models in terms of how each system provides users with the means to communicate trust information. Pairs of users interact, mediated by some piece of information or content, and the consumer of that content gives feedback in some way, on the user or piece of information, and either explicitly or implicitly. The level at which these users interact determines the trust one user has in another, and the level to which a person performs in their community determines their reputation. We introduce a model that gathers reputation information which is applicable to any collaborative online environment, regardless of the nature of the feedback. Other systems may measure reputation by aggregating the level of trust between users (for example, Avesani et al (2005); Golbeck (2006)). Our model does not require explicit trust statements to be made between users to calculate overall user reputation. Instead, reputation is determined by measuring the level of collaboration that occurs between users.

In Section 4 we describe how we can use this collaboration model as the basis for calculating user and item reputation. In fact we will describe a number of different techniques for calculating reputation at the user and item level.

3.1 A Model of Online Collaboration

In what follows we will distinguish between trust (as a function of some user-to-user interaction) and reputation (as a function of a particular user) as discussed previously. In order to frame this research it is useful to consider two different dimensions by which we can understand the source and type of trust information used in many online social systems. Trust signals can be either *implicit* or *explicit* and they can be *direct* or *indirect* (see Figure 2) and different systems and approaches can be usefully compared along these dimensions. Examples of research previously discussed can be viewed in these terms; see Figure 3. In the case

[6] http://www.imgur.com.
[7] http://www.flickr.com.
[8] http://www.digg.com.
[9] http://www.reddit.com.
[10] http://www.stackexchange.com.
[11] http://ansonalex.com/infographics/facebook-user-statistics-2012-infographic/
[12] http://www.techcrunch.com/2011/06/30/twitter-3200-million-tweets/.
[13] http://www.reddit.com/r/Redditresearch/comments/de4re/basic_frequency_plots_for_a_months_worth_of.
[14] http://www.wikipedia.org.



(a) Direct Feedback (b) Indirect Feedback

Fig. 2 Two different types of feedback from which trust can be inferred. Figure (a) shows a simple, direct model where user u_2 gives feedback directly to user u_1. Figure (b) depicts the inference of trust based on the feedback that user u_2 gives about item i, rather than directly to user u_1 who produced the item.

of the former reputation information can be based on a direct user to user interaction, such as one user rating another user. Conversely reputation information may be derived indirectly if, for example, users interact by virtue of some item as is the case when users collaborate by editing a Wikipedia article.

Reputation information can also be based on implicit or explicit statements of trust. In eBay users provide explicit feedback on sellers or buyers; an example of a direct, explicit reputation model. In other scenarios reputation information is derived from implicit factors. In a recommendation scenario a user who receives and acts on a recommendation is implicitly acknowledging those similar users who served as the source of said recommendation; an example of an indirect, implicit reputation scenario because the users interact with respect to a separate recommended item. Likewise, when a user follows another user in Twitter we can view the user as displaying some level of trust on the followed user; this is an example of a direct, implicit reputation scenario.

Thus our approach to reputation acknowledges different sources and types of reputation information. In each case, however, the common denominator is an instance of collaboration between two users, which we view as a collaboration event. Such an event can occur directly or indirectly, and indeed explicitly or implicitly, as determined in the previous section. In fact, such collaboration can happen asynchronously, sometimes with users having interlinked but distinct goals. For example a user of the Stack Exchange network may answer another user's question some time after being first posed. This answerer has a different goal to the questioner, but these two people have indeed collaborated in their sharing of information. And the final piece of our approach proposes two possible ways in which users can participate in these collaboration events (see Figures 4 and 5). Specifically in each collaboration event there is a *producer* and a *consumer*. The producer p is the originator of some piece of information that is acted on, interacted with or consumed by the consumer c.

More generally a collaboration event can refer to a single consumer and multiple producers. For example, in a recommender system a consumer's selected recommendation may have come from many similar users. Similarly a user may read a tweet that has appeared on their Twitter feed as the result of a string of re-tweets by different users. Although these are cases where multiple users have produced a piece of information, this should still be viewed as a single instance of collaboration. Finally, the key idea in our approach is that these collaboration events can be scored in effect transferring a unit of trust from consumer to producer;

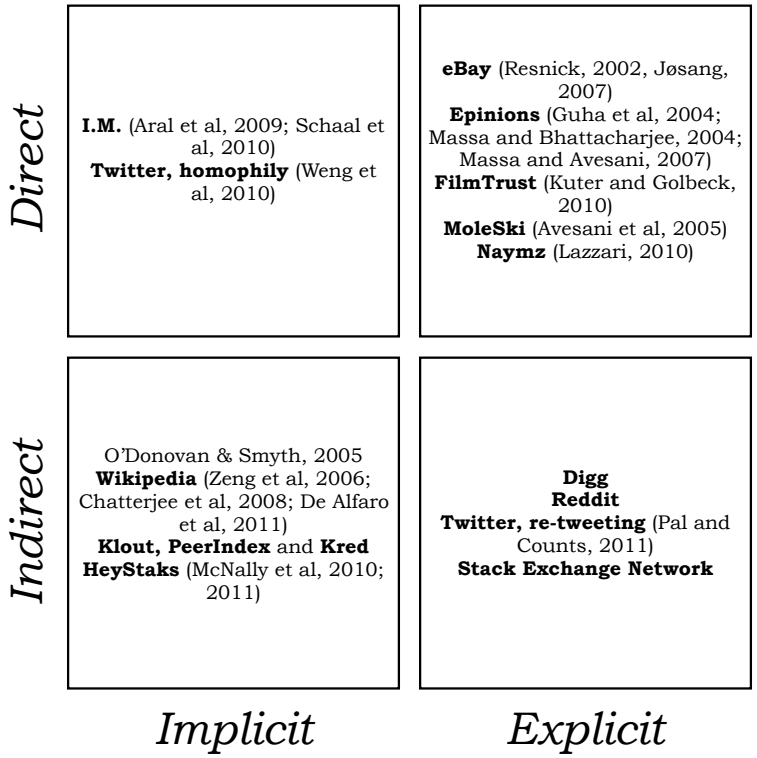


Fig. 3 An analysis of trust and reputation research, categorized into four distinct areas according to the nature of interaction between users

Fig. 4 An atomic representation of collaboration: The Collaboration Event.

see Figure 4. Or, if there are multiple producers, then the unit of trust can be shared between the k producers as in Figure 5. During a collaboration event, a single unit of trust is shared equally between producers, but in some circumstances trust could be weighted in favour of particular producers that have contributed more to this single collaboration. However, this is outside the scope of this paper, and we view that as an interesting matter for future work.

3.2 From Collaboration Event to Collaboration Graph

Over time sequences of collaboration events come to form a collaboration 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 trust scores flow along these edges. The trust score associated with each edge – based on the type of interaction and

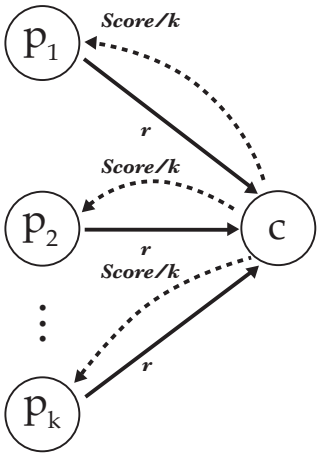


Fig. 5 A collaboration event with k producers. The consumer c acts on a piece of information, r, that has been produced by several producers, $p_1, ..., p_k$; a trust score is shared amongst these producers as a result of this collaboration event. One way to confer scores is to simply assign each user an equal share of the collaboration score, i.e. *score/k*.

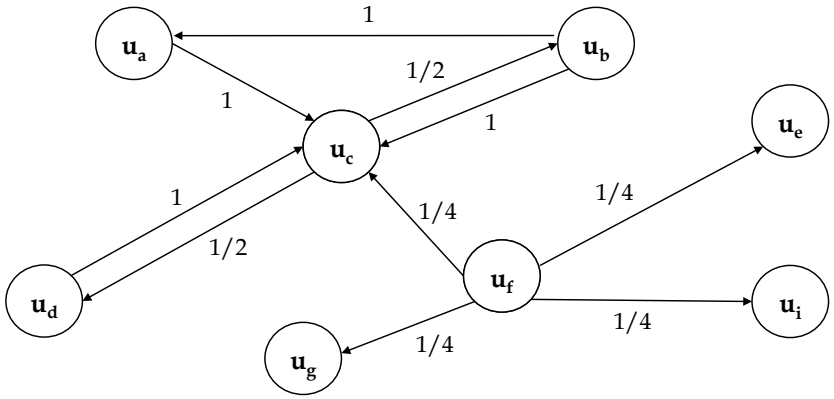


Fig. 6 An example of a collaboration graph with 8 users. The direction of the arrows correspond to the assignment of collaboration score. For example, user u_a has produced a piece of information that has been consumed by user u_b and thus an edge is drawn from u_b to u_a with a weight of 1. Likewise, user u_c has consumed information produced by users u_b and u_d and thus edges are drawn from u_c to each of these users with weights of 0.5 (assuming reputation is shared equally between producers).

domain – form the basis of user reputation which is accumulated at the nodes as per Section 4. For example, Figure 6 depicts a scenario where eight users of a system have interacted collaboratively with each other. Here it can be seen, for example, that user u_c has consumed information produced by users u_b and u_d and thus edges are drawn from u_c to each of these users with weights of 0.5.

So far we have described how we distill collaboration events and how each event can be interpreted as a set of directed, weighted edges in a collaboration graph. We can look at all incident edges per-user to get a sense of how users collaborate with not just another single user, but with their community. In that sense, the trust score illustrated in Figure 4 can be

viewed as a fundamental unit of reputation. In the next section we describe our approach for calculating user reputation based on users' propensity to collaborate with others, as well as outlining a number of alternatives using well-known link analysis techniques.

4 Modelling Reputation

Given a collaboration graph how can we calculate the reputation of the collaborating users? Moreover how can we use this reputation to positively influence the future dissemination of content? Both of these questions are addressed in this section where we describe three user-based reputation models, as well as four approaches to calculating item reputation based on the scores of the producing users.

4.1 User Reputation

We now present a series of methods to calculate user reputation by examining the collaboration graph: The "Weighted Sum" model (McNally et al, 2010) and the two well known link analysis techniques, PageRank (Brin and Page, 1998) and HITS (Kleinberg, 1999).

4.1.1 Reputation as Weighted Sum

The Weighted Sum model simply calculates the reputation of a producer p_i at time t, $rep(p_i,t)$, as the sum of the weights of incoming edges as follows:

$$rep(p_i,t) = \sum_{e \in E_{p_i}} w_e \tag{1}$$

where E_{p_i} is the incoming edge set of producer p_i and w_e is the weight of edge e. A user may gain a high reputation score from assuming the role of producer in many collaboration events. However, the growth of this user's reputation may be stymied if they are only one of many producers of a number of consumed items. This feature of the model may be used to encourage users to produce their own high-quality content, and thus discourage piggy-backing on the work of other producers.

4.1.2 Reputation as PageRank

PageRank is the well known algorithm used by Google to rank web search results (Brin and Page, 1998). The key intuition behind PageRank is that pages on the web can be modeled as vertices in a directed graph, where the edge set is determined by the hyperlinks between pages. PageRank leverages this link structure to produce an estimate of a relative importance of web pages, with inlinks from pages seen as a form of recommendation from page authors. Important pages are considered to be those with relatively large number of inlinks. Moreover, pages that are linked to by many other important pages receive higher ranks themselves. PageRank is a recursive algorithm, where the ranks of pages are a function of the ranks of those pages that link to them.

The PageRank algorithm can be readily applied to compute the reputation of collaborating 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 of the producers as described

above. Once all the collaboration events up to some point in time, t, have been captured on the graph, the PageRank algorithm is then executed and the reputation (PageRank) of each user p_i at time t is computed as:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{|L(p_j|)}, \tag{2}$$

where d s a damping factor, N is the number of users, $M(p_i)$ is the set of inlinks (from consumers) to (producer) p_i and $L(p_j)$ is the the set of outlinks from p_j (i.e. the other users from whom p_j has consumed results). In this paper, we use the Python library NetworkX[15] implementation of PageRank, which implements the algorithm according to Page et al (1999), and Langville and Meyer (2005).

4.1.3 Reputation as HITS

The HITS algorithm (Kleinberg, 1999) was also developed to rank web search results and, like PageRank, makes use of the link structure of the web to perform ranking. In particular, HITS computes two distinct scores for each page: an authority score and a hub score. The former provides an estimate of the value of a page's content while the latter measures the value of its links to other pages. Pages receive higher authority scores if they are linked to by pages with high hub scores, and receive higher hub scores if they link to many pages with high authority scores. HITS is an iterative algorithm where authority and hub scores are computed recursively.

As with PageRank, we use the collaboration graph and the HITS algorithm to estimate user reputation. In this regard, it may at first seem appropriate to consider producers as authorities and consumers as hubs. However, as we will discuss in Section 6, hub scores are useful when it comes to identifying a particular class of users which act both as useful consumers and producers of high quality information. Thus we model user reputation using both authority and hub scores, which we compute using the NetworkX implementation of the HITS algorithm. This implementation is based on work by Kleinberg (1999), and Langville and Meyer (2005). Briefly, the algorithm operates as follows. After initialization, repeated iterations are used to update the authority ($auth(p_i)$) and hub scores ($hub(p_i)$) for each user p_i. At each iteration, authority and hub scores are given by:

$$auth(p_i) = \sum_{p_j \in M(p_i)} hub(p_j) \tag{3}$$

$$hub(p_i) = \sum_{p_j \in L(p_i)} auth(p_j) \tag{4}$$

where as before $M(p_i)$ is the set of inlinks (from consumers) to (producer) p_i and $L(p_i)$ is the set of outlinks from p_i (i.e. the other users from whom p_j has consumed results).

In the above we have described reputation models for *users*. People accumulate reputation when content that they have produced is consumed in some way by other users. We have described how reputation is distributed between multiple producers during these collaboration events. In the following section we describe how this reputation information can be used to capture the quality of content these users produce. In order to achieve this, we propose a number of methods to map user to item reputation by considering the source of those items.

[15] http://networkx.lanl.gov.

4.2 Item Reputation

We mentioned in the previous section that often in online scenarios there can be more than one producer of a piece of information. Therein lies the challenge to calculating the reputation of that information: How do we deduce an item's reputation in a way that best reflects that of its producers? In this section we describe a number of approaches to model the reputation of information produced by users in an online community. In each case, the goal is to calculate the reputation score of some user-produced item r at time t based on the reputation scores of the item's producers, $\{p_1, ..., p_k\}$, at that point in time; see Equation 5.

$$rep(r,t) = f\big(rep(p_1,t), ..., rep(p_k,t)\big) \; . \tag{5}$$

For the purpose of illustration we will calculate each reputation score based on a hypothetical recommendation scenario for an item r which is associated with a set of 10 producers with the following reputation scores at time t: $\{0.003, 0.014, 0.023, 0.052, 0.089, 0.097, 0.154, 0.297, 0.348, 0.581\}$. This includes a cross section of producers including some with low reputation scores and some with high scores. Note that in practice producer reputation scores are first normalized by the maximum producer reputation in the corresponding community to ensure a score between 0 and 1, to facilitate comparisons across subsections of online communities (subreddits on Reddit, or Twitter topics for example).

4.2.1 Median Reputation

Perhaps the simplest way to translate user reputation into item reputation is to calculate the average reputation of the item's producers. As the distribution of reputation scores for a particular item may not always be parametric, the most useful average metric is the Median.

$$rep(r,t) = median\big(rep(p_1,t), ..., rep(p_k,t)\big) \; . \tag{6}$$

The advantage of this approach over a simple mean reputation is that the median statistic tends to better represent the central tendency of the set of user reputations. In the case of our hypothetical recommendation scenario the reputation of the item r is 0.093 according to this median model.

4.2.2 Max Reputation

Another simple way of scoring an item based on the reputation of its producers is to take the maximum reputation value from that set. Formally, Max Reputation is calculated thus:

$$rep(r,t) = max\big(rep(p_1,t), ..., rep(p_k,t)\big) \; . \tag{7}$$

Scoring pages in this way is advantageous as the reputation of an item will not be harmed if, for example, many new, not yet reputable users have selected the page. In this scenario, the reputation of item r is 0.581 by this approach.

4.2.3 Harmonic Mean Reputation

Harmonic Mean is an average measure that tends towards the lower bound of a set of numbers, and thus is more conservative than arithmetic mean or median. It is calculated by finding the reciprocal of the arithmetic mean of the reciprocals. Formally, the harmonic mean of a set of user reputation scores is calculated as:

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

In this case, the reputation of item r is 0.020. Harmonic mean may be a good indicator of the utility of a page in the sense that a page is only as reputable as its least reputable producer. However, rather than simply using the minimum producer reputation score available, harmonic mean permits the full range of producer reputation scores to influence the overall item reputation.

4.2.4 Hooper's Reputation

In order to reinforce a page's reputation according to the number of producers, keeping in mind their score, a different approach is required. Several approaches for carrying out such a task are available in past literature, for example by utilizing probability theory (Voorbraak, 1995) or by measuring the statistical distribution between samples (Bailey and Gribskov, 1998). A simpler technique is George Hooper's *Rule for Concurrent Testimony*, originally proposed as a technique to calculate the credibility of human testimony (Shafer, 1986). This is applicable in our case in the sense that users who have produced a piece of information are endorsing it, in the same way that a group of witnesses might attest to the same report. Hooper gives to the report a credibility of $1-(1-c)^k$, assuming k reporters, each with a credibility of c (where $0 \leq c \leq 1$). Shafer (1986) goes on to cite an extension to Hooper's Rule, proposed by Lambert, that calculates the probability of testimony being true if two witnesses act independently of each other. This is analogous to our reputation model where two producers of the same piece of content have different reputation scores. Because of the applicability of this approach, as well as its simplicity and effectiveness at deducing item reputation (as discussed in Section 6.3), we chose it over the other approaches mentioned.

We can extend the idea presented by Shafer (1986) to allow for any number of reporters with different values of c. In an online social environment, the quality of an item can be determined by performing the calculation formalized in Equation 9, across the reputation scores of its producers. For clarity, herein we refer to this equation as "Hooper".

$$rep(r,t) = 1 - \prod_{i=1}^{k} (1 - c_i) \ . \tag{9}$$

As per Shafer (1986), we can calculate the reputation of information r as $1-(1-0.003)(1-0.014)(1-0.023)$ ... and so on. The reputation of this particular item r is 0.878.

If we examine each score, we can clearly see the tendencies of these item reputation models: Harmonic Mean tends towards the lower bound of the set of user reputation scores, median tends towards the centre, and Max is always on the upper bound of the set. Hooper is similarly positively inclined, with concurrent positive scores yielding a result higher than even the maximum producer reputation score. Our aim is to evaluate each of these item reputation models in terms of their effectiveness at determining item quality; this analysis is carried out in the context of the HeyStaks collaborative web search platform as described in the next section.

5 Case Study: HeyStaks

HeyStaks is an approach to collaborative web search that is designed to work with mainstream search engines such as Google, Bing, and Yahoo; so users search as normal, on their favourite search engines, but benefit from search recommendations from people they trust. The HeyStaks system has been described previously in Smyth et al (2009), where the focus was on a description of its recommendation technique. The aim of this paper is to investigate the role of a novel approach to calculating user and item reputation during recommendation, whereby the search reputation of users is allowed to influence recommendation directly. We will return to the issue of reputation in following sections, but first we present a brief review of HeyStaks in order to provide sufficient technical context for the remainder of this paper.

5.1 System Architecture

Figure 8 presents the HeyStaks architecture. There are two key components: a client-side *browser toolbar/plugin* and a back-end *server*. The toolbar, as seen in Figure 7, provides users with direct access to the HeyStaks functionality. It allows users to record and share their search experiences in repositories called "staks". Users can create and share their own staks, according to a specific topic or community of their interest. In the instance shown in Figure 7, the user is searching for information on Cascading Style Sheets (CSS). HeyStaks recommends result pages from a set of items previously clicked on by other HeyStaks members. These recommendations appear on screen as well as the standard Google result-list. For more information on the HeyStaks toolbar and related interface, see McNally et al (2010). The toolbar also provides for the type of deep integration with mainstream search engines that HeyStaks requires. For example, the toolbar captures the routine search activities of the user (query submissions and result selections) and it 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 toolbar also manages the communication with the back-end HeyStaks server. Search activities (queries, result selections, tags, votes, shares etc.) are used by the server to update the HeyStaks stak indexes. And 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.

5.2 The Recommendation Engine

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, $S = \{r_1, ..., r_k\}$. Each result is also anonymously associated with a number of implicit and explicit interest indicators, based on the type of actions that users can perform on these pages, which include:

- *Selections* – a user selects a search result (whether *organic* (originating from normal search activity) or *recommended*);
- *Voting* – a user votes on a given search result or the current web page;
- *Sharing* – a user chooses to share a specific search result or web page with another user (via email or by posting to their Facebook Wall etc.);

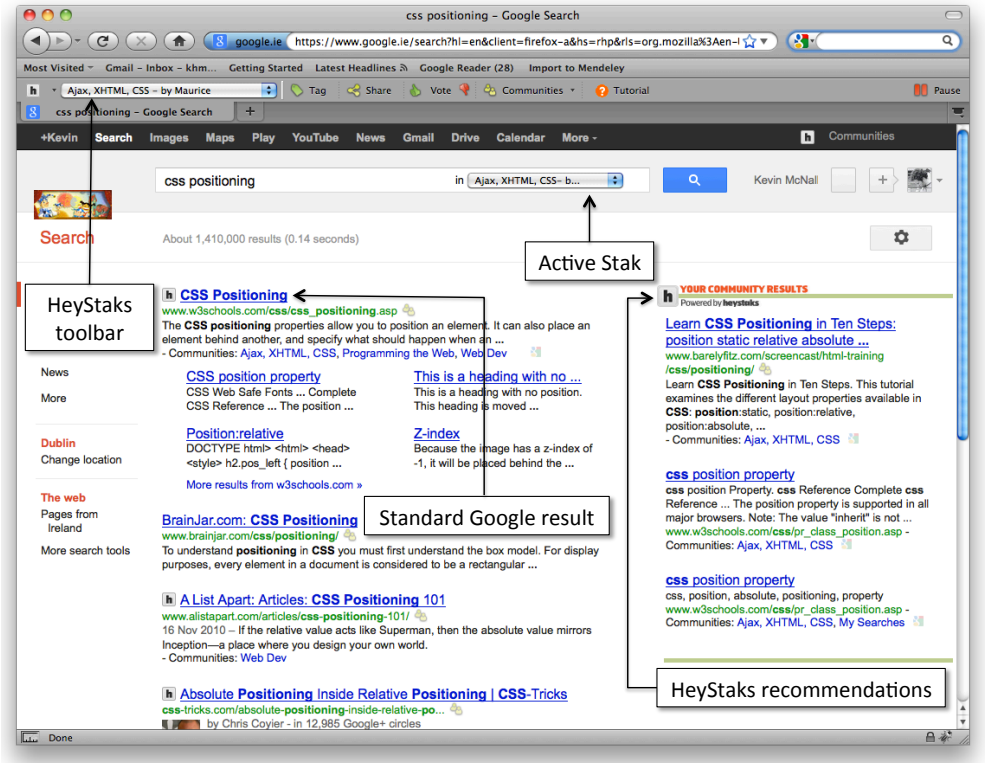


Fig. 7 A screenshot of HeyStaks in action. Here the searcher is looking for information relating to CSS positioning. HeyStaks recommends results relevant to the search that exist in the current stak - in this case, "Ajax, XHTML, CSS". These appear alongside standard Google results.

– *Tagging/Commenting* – the user chooses to tag and/or comment on a particular result or web page.

Each of these actions can be associated with a degree of confidence that the user finds the page to be relevant for a given query. Each result page r_i^S from stak S, is associated with these indicators of relevance, including the total number of times a result has been selected (Sl), the query terms $(q_1,...,q_n)$ that led to its selection, the terms contained in the snippet of the selected result $(s_1,...,s_k)$, the number of times a result has been tagged (Tg), 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 (Sh) as indicated by Equation 10.

$$r_i^S = \{q_1...q_n, s_1...s_k, t_1...t_m, v^+, v^-, Sl, Tg, Sh\} \,. \tag{10}$$

Importantly, this means 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[16] index, with each result indexed under its associated query and tag terms, and this 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.

[16] (http://lucene.apache.org)

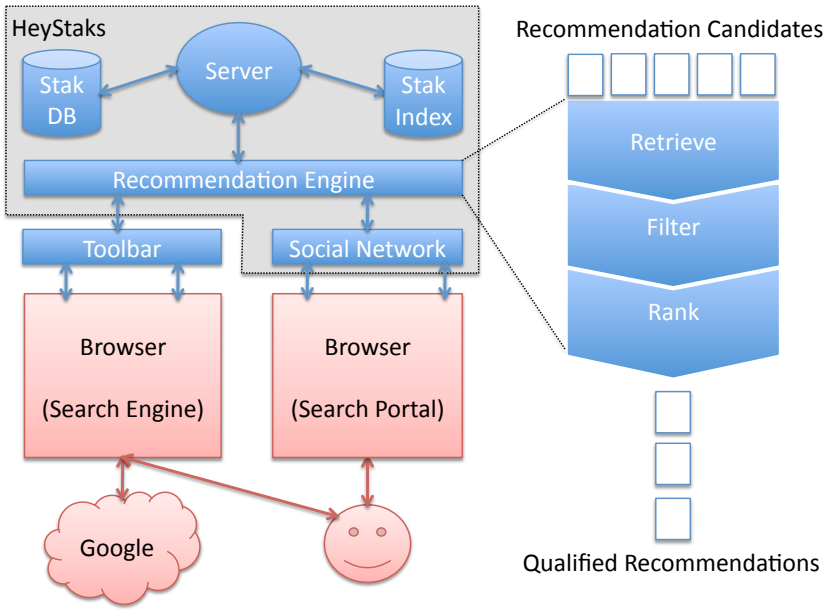


Fig. 8 The HeyStaks system architecture and outline recommendation model.

At search time, the searcher's query q_T and current stak S_T are used to generate a list of recommendations to be returned to the searcher. At search time, the source of the recommendation is not known to the searcher. For the purpose of this paper we will discuss recommendation generation from the current stak S_T only, although in practice recommendations may also come from other staks that the user has joined or created.

$$rel(q_T, r) = \sum_{\tau \in q_T} tf(\tau \in r) \times idf(\tau \in r) \,. \tag{11}$$

There are two key steps when it comes to generating recommendations. First, a set of *recommendation candidates* are retrieved from S_T by querying the relevant Lucene index using the target query q_T. This effectively produces a list of recommendation candidate based on the overlap between the query terms and the terms used to index each recommendation (query, snippet, and tag terms). Next, these recommendation candidates are filtered and ranked. Results that do not exceed certain activity thresholds are eliminated as candidates; such as, for example, results with only a single selection or results with more negative votes than positive votes (see Smyth et al, 2009). Each remaining recommendation candidate r is then ranked according to its relevance (*rel*) score, weighted with various interest indicators such as the past levels of users' result selections, votes and shares of the candidate, at time t as per Equation 11.

The relevance of a result r with respect to a query q_T is computed using *TF-IDF*, a well-known document retrieval function originating in work by Salton and McGill (1983); see Equation 11. This function 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 in-

dex terms. HeyStaks uses the default implementation given by Lucene, which also includes various optimization techniques such as query and document boosting[17].

5.3 Reputation in HeyStaks

The above relevance model pays no attention to the *source* of the recommendation; i.e. the users who originally contributed the page to a stak or whose subsequent activities resulted in the page being recommended. For this reason recent research has looked at the possibility of adding a reputation component to recommendation (McNally et al, 2010, 2011). Recommendation candidates can be scored by a combination of relevance and reputation according to Equation 12, where w is used to adjust the relative influence of relevance and reputation. As part of our evaluation, we attempt to find an optimal value of w, see Section 6. This provides the reputation of the source with influence on a page's overall score. Pages that have been contributed by many reputable users are considered more eligible for recommendation than those that have been contributed by fewer, less reputable users.

$$score(r,q_T,t) = w \times rep(r,t) + (1-w) \times rel(q_T,r)\ . \tag{12}$$

Specifically, a collaboration event in HeyStaks occurs when a user is recommended a result page and acts upon it. In this instance, the user or users who are responsible for the page's existence in the system become producers in the event, and the user who received the recommendation is the consumer. For more details on how the collaboration graph is built in HeyStaks, see McNally et al (2010). Although there are a range of possible actions a user can take on the result page (selection, voting, tagging, sharing) the reputation system does not discriminate according to action type. It would be possible to weight trust scores according to action type (for example, a vote up on a result page results in a greater degree of trust transferred onto producers). We leave this for future work.

The key idea in applying our collaboration-based approach to calculating reputation for HeyStaks is that reputation can be calculated by mining the indirect collaborations that occur between users as a result of their searches. For example, 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 a single collaboration event. The current searcher who chooses to act on the recommendation is the consumer and the original searcher (or searchers), whose earlier action on this result caused it to be added to the search stak, and ultimately recommended, is 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. Thus, we can apply any combination of user and item reputation models discussed in Section 4 to HeyStaks by integrating them into its recommendation engine according to Equation 12. Of course, the success of these models is determined by the extent to which they improve the effectiveness of the HeyStaks recommendation engine, which we analyse using a set of queries submitted to HeyStaks during the course of a live-user trial.

6 Evaluation

In previous work (Smyth et al, 2009) we have demonstrated how the standard relevance-based recommendations generated by HeyStaks can be more relevant than the top ranking

[17] http://lucene.apache.org/core/old_versioned_docs/versions/2_9_0/api/all/org/apache/lucene/search/Similarity.html

No.	*Question*	*Answer*
1	Who was the last Briton to win the men's singles at Wimbledon	Fred Perry
2	Which Old Testament book is about the sufferings of one man	Job
3	Which reporter fronted the film footage that sparked off Band Aid	Michael Buerk
4	Which space probes failed to find life on Mars?	All of them
5	in the general theory of relativity what causes space-time to be modified?	Mass/Matter/Energy

Table 1 A sample of the 20 questions presented to trial participants. The correct answers for each question are also shown.

results delivered by Google in a similar setting. In this work we wish to explore how an online social service such as HeyStaks can be enhanced by the integration of reputation information into its recommendation engine. We compare HeyStaks' standard, relevance-based recommendation technique to an extended version of HeyStaks that also includes user and item reputation models. Earlier work has demonstrated that reputation can indeed positively influence recommendations made by HeyStaks using a single pairing of user and item model – Weighted Sum and Max (McNally et al, 2011). Now we explore different approaches to both user and item reputation in terms of their effectiveness in influencing recommendations. Specifically we consider the relative benefits (or otherwise) of different combinations of user and item reputation models.

6.1 Dataset and Methodology

Our experiment involves 58 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. The students worked concurrently on the same set of questions, which were randomly ordered to avoid any learning bias effects on students. The questions were selected from a quiz book by Preston and Preston (2007), and were chosen specifically for their obscurity and difficulty, and lead users to perform queries that are informational in nature. A sample of the questions and their correct answers are shown in Table 1.

It was highly unlikely that students would be able to answer any significant number of these questions from their own general knowledge and so the purpose of this experiment was to look at how the students used HeyStaks and Google to help them answer these questions. Each user was allocated a desktop computer with Mozilla's 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. Users were made aware of the functionality provided by the HeyStaks toolbar, so if they found a page they liked they could either *tag* it or *vote* on it, having been informed in an introductory one hour lecture and demonstration of the HeyStaks system how this might affect future Google searches and the searches of others. Note however that users were not explicitly directed to use the HeyStaks toolbar, rather to avail of it as they saw fit.

The 58 students were randomly divided into search groups. Each group was associated with one newly created search stak, which would act as a repository for the groups' search knowledge. For this trial, we created 4 shared staks containing 5, 9, 19, and 25 users. The different sizes of shared staks provided an opportunity to examine the effectiveness of collaborative search across a range of different group sizes, which is explored in detail in

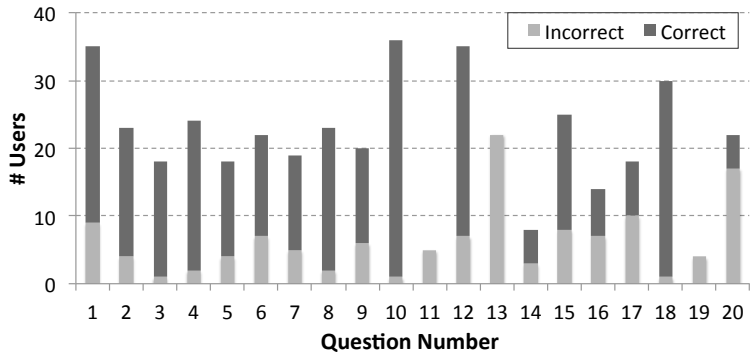


Fig. 9 Number of correct and incorrect attempts made by trial users per question (one attempt per user).

earlier work (McNally et al, 2011). In this previous work we found that stak size was not in itself a strong influencer of recommendation quality, rather the search expertise of stak members played a much more important role. As a result, the different staks can be seen as a legacy artifact of this previous experiment, and now we focus on performance across all trial participants. For this reason, in this evaluation we will focus our attention on reputation and recommendation performance averaged across these different staks rather than at the individual stak-level.

It is worth highlighting here that the setup used is such that under normal search conditions the users are likely to receive the same results for the same queries; unlike the *filter bubble* type of effect observed in Pariser (2011) which highlights how Google can return very different results to users for the same queries. The reason that the filter bubble does not apply here is that all users are searching in the same location, on the same network, and they were not using their personal Google accounts. This means that there was no personalization or customization of results to users.

During the 60 minute trial a total of 3,124 queries and 1,998 result activities (selections, tagging, voting, popouts) were logged, and 724 unique results were selected. All questions were attempted by at least one user during the trial, and so for the purpose of this evaluation there were no unattempted questions. Figure 9 shows the number of correct and incorrect attempts for each question, illustrating the range of user performance. Questions were answered correctly on average around 15 times, with a standard deviation of 5.3. For a more detailed analysis of overall user performance see McNally et al (2011). As expected, during the course of the trial, result selections — the typical form of search activity — dominated over HeyStaks-specific activities such as tagging and voting. Averaged across all staks, result selections accounted for just over 81% of all activities, with tagging accounting for just under 12% and voting for only 6%.

For the purpose of establishing a ground-truth for result relevance, each result page was manually examined post-trial by a number of people furnished with answers to the quiz questions and its relevance with respect to the appropriate quiz question was categorized as follows:

- not relevant (i.e. the result page content had no relevance with respect to a question);
- partially relevant (i.e. the result page contains an implicit reference to the answer or to a part of the answer to a question);
- relevant (i.e. the result page contains an answer to a question).

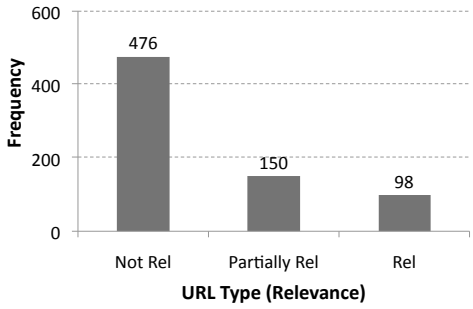


Fig. 10 The number of not relevant, partially relevant, and relevant pages found during the trial.

Figure 10 shows a relevance breakdown of the result pages logged during the course of the trial. 66% of result pages acted on were categorised as being not relevant with respect to the questions posed, while only 14% were deemed relevant. These findings demonstrate the difficulty of the questions presented, as mentioned above. We will return to this relevance information later in this section when we use it to evaluate the relevance of HeyStaks recommendations.

6.2 User Reputation Scores

We begin by analysing the reputation scores at the end of the 60 minute trial period that were produced by each of the four user reputation models described in Section 4. To facilitate comparisons, for each model user reputation scores are normalised by stak; i.e., for a given reputation model and stak, reputation scores are divided by the maximum user reputation score in the stak at the end of the trial. The range of normalized reputation scores across all users and all staks for each model are shown in Figure 11. In this figure, we can see the variation in scores assigned to users given by each model used.

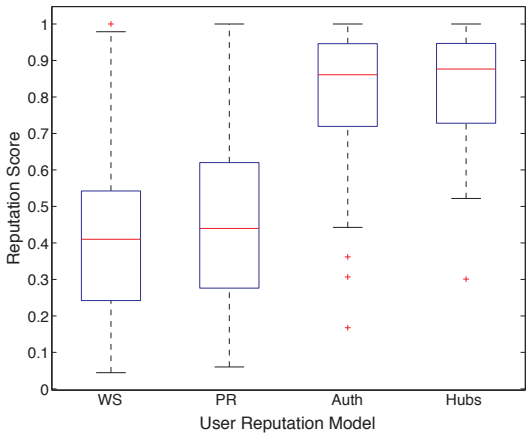


Fig. 11 Boxplot showing the range of scores assigned to users, across all staks, by each of the four user reputation models.

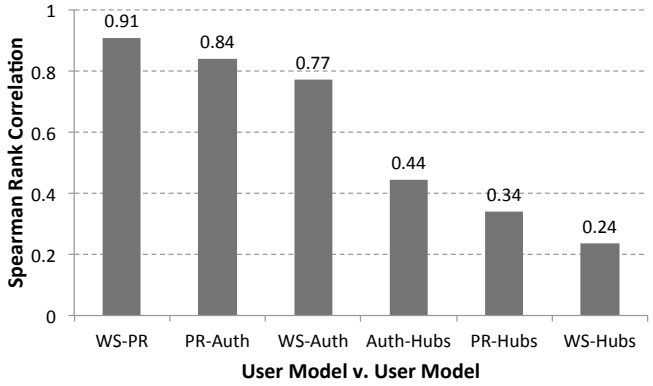


Fig. 12 Pairwise rank correlations between user reputation scores given by each reputation model across all users.

It is clear that similar distributions were observed for the Weighted Sum (WS) and PageRank (PR) models and likewise for the HITS Hubs and Authority (Auth) models, although significant differences between the pairs of models are apparent.

What should be obvious is that all of the test users acquired at least some reputation by the end of the trial across the four user reputation models. More importantly, we can see that each of these models results in a range of different reputation values spread across the 58 users. The Weighted Sum (WS) and PageRank (PR) models, for example, generate a wide range of reputation scores (from 0.05 to approximately 1) across the users. The median reputation is about 0.4–0.45 with most users acquiring reputation scores in the range of 0.25 to about 0.6. At least some users enjoy very high reputation (up to 1) while others accumulate very low reputation scores of about 0.05. What is important here is the range of reputation values rather than the actual values. At the very least it means that users are separable in terms of reputation scores which suggests that using reputation as part of a recommendation system is at least likely to generate different types of recommendations. In the extreme case if all users ended up with the same reputation then there would be little possible benefit to be derived from incorporating a reputation model into recommendation. For the two HITS based models (Auth and Hubs) we can see that there is also a spread of reputation scores, but that these scores are less diverse that those found for WS or PR. The median reputation is considerably higher than in the WS and PR models (approximately 0.9 for Auth and Hubs) with the majority of users obtaining scores in the range 0.75–0.95. Once again the actual values are less important here than the range of values reported.

The Spearman pairwise rank correlations between user reputation scores given by each reputation model are shown in Figure 12. The chart shows that, although the scores output by HITS Hubs and HITS Authorities have similar ranges and median values, they were not assigned to the same users. This is evidenced by the relatively low Spearman Rank correlation between HITS Hubs and Authorities scores—0.44. It can also be seen from the figure that high correlations are seen between the Weighted Sum, PageRank and Authority models, with pairwise correlations between these models in the range 0.77–0.91. In contrast, the correlations between the Hubs model and the other models are much lower. It is difficult to draw precise conclusions about the Hubs correlations given the constrained nature of the user-trial, but since the HITS Hubs metric is designed to identify pages that contain useful

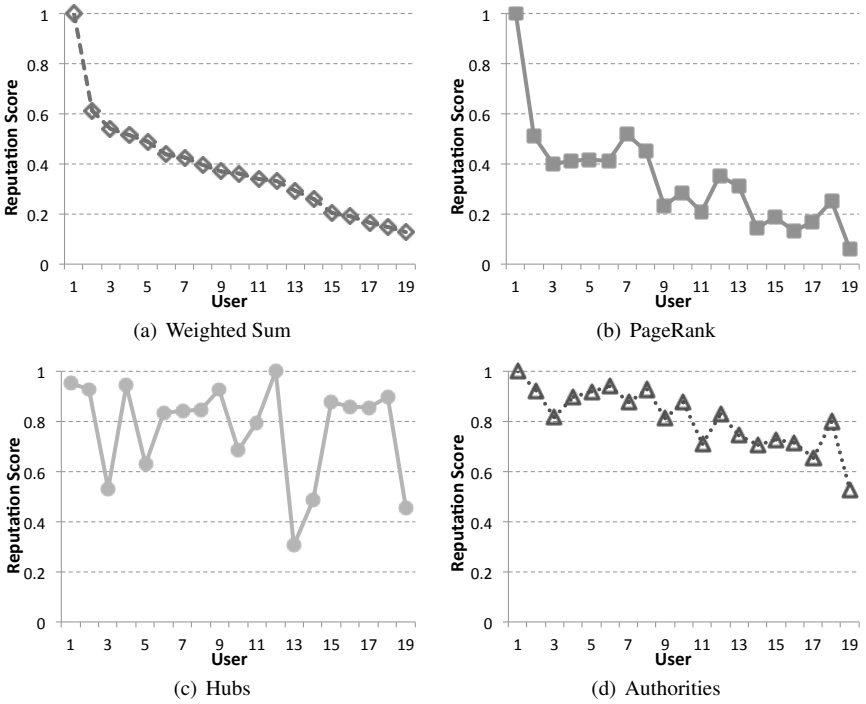
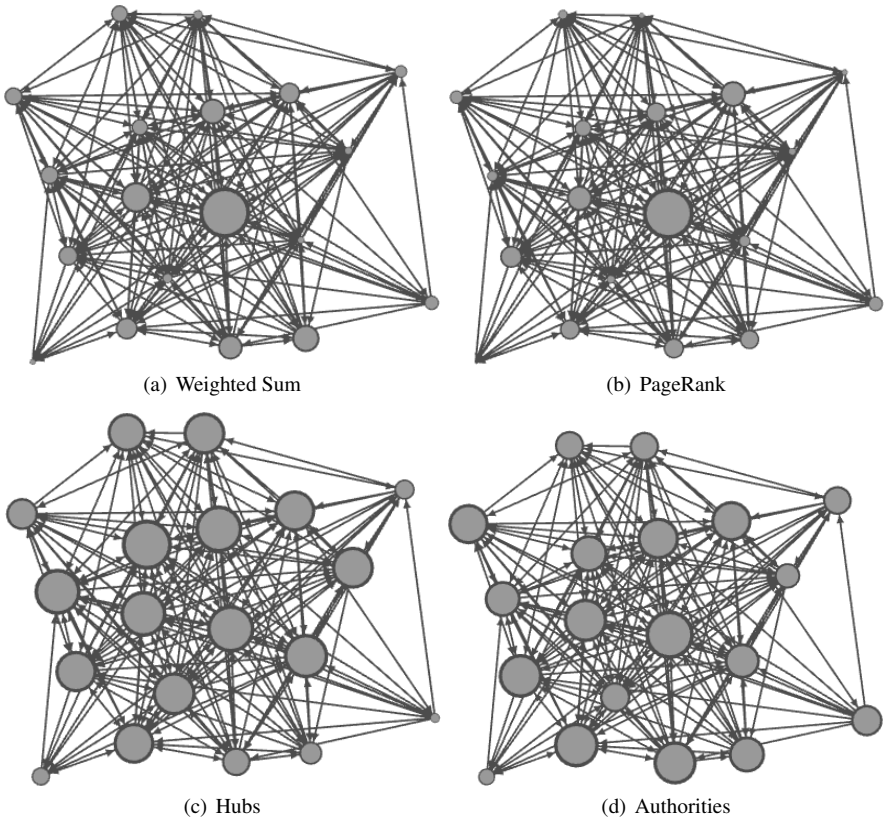


Fig. 13 Reputation scores for members of the 19-person stak: (a) Weighted Sum, (b) PageRank, (c) HITS Hubs and (d) HITS Authorities.

links towards authoritative pages in the web search domain (analogous to good consumers rather than producers in our context), such low correlations are to be expected with the other models which more directly focus on producer activity.

Figures 13 (a)–(d) illustrate the scores members of the 19-person stak had accumulated by the end of the trial. Here we present results from the 19-person stak as they are representative of those found in the three larger staks. Results from the 5-person stak were not as clear-cut, due to the small number of users. In the figures, users are ranked according to their Weighted Sum score in descending order, and this order is preserved across all 4 charts. The charts provide a visual example of the strong correlation between Weighted Sum, PageRank and HITS Authorities scores. However, as suggested by the correlation scores above, Hubs has scored these users differently. For example, users 14–18 have low reputation scores in all but the Hubs model, where these users were all assigned above-average reputation according to Hubs. The extent to which each of these users collaborated with their fellow stak members is illustrated in Figures 14 (a)–(d). These graphs reinforce the finding that the PageRank and Weighted Sum models are more discriminatory in determining reputable users, and thus the variation of reputation scores is greater compared to HITS Hubs and Authorities.

Although HITS Authorities and PageRank output similar scores to the Weighted Sum model, we know that the mechanism by which these scores are calculated is fundamentally different. Specifically, a user with a high PageRank score benefits from others with high scores linking in to them (i.e. collaborating with them), and a high HITS Authority user gains greater reputation when users with high Hubs scores endorse their content. Put simply, user



(a) Weighted Sum (b) PageRank

(c) Hubs (d) Authorities

Fig. 14 Graphs showing collaborations occurring in the 19-person stak. In each graph, nodes are sized according to their relative score given by a particular user reputation model: (a) Weighted Sum, (b) PageRank, (c) HITS Hubs and (d) HITS Authorities.

reputation scores are dependent on the scores of adjacent (collaborating) users, and updating one user's reputation score has a cascading effect throughout the network. This is not true in the Weighted Sum model, which involves scoring users independently of the reputation of any other users. In a scenario such as HeyStaks using PageRank or Authorities as a reputation scoring mechanism, a user could potentially provide a stak with a single useful result that another high-reputation user could then select, thus becoming highly reputable themselves. In theory, this means the system is easier to game as reputation can be gained much more quickly. Conversely, with Weighted Sum a user must provide useful results on a more consistent basis to achieve high reputation.

Weighted Sum promotes the idea that reputation is improved over time, and a user's reputation score according to this model is a direct function of the number of collaboration events they are involved in as producers. Thus, Weighted Sum is an inherently robust model of reputation, one in which users cannot significantly increase their reputation scores by producing less content that happens to be selected by other high reputation users as is the case with PageRank and HITS Authorities.

HITS Hubs captures a different kind of user activity compared to PageRank, HITS Authorities and Weighted Sum. Unlike those models, it measures the extent to which users

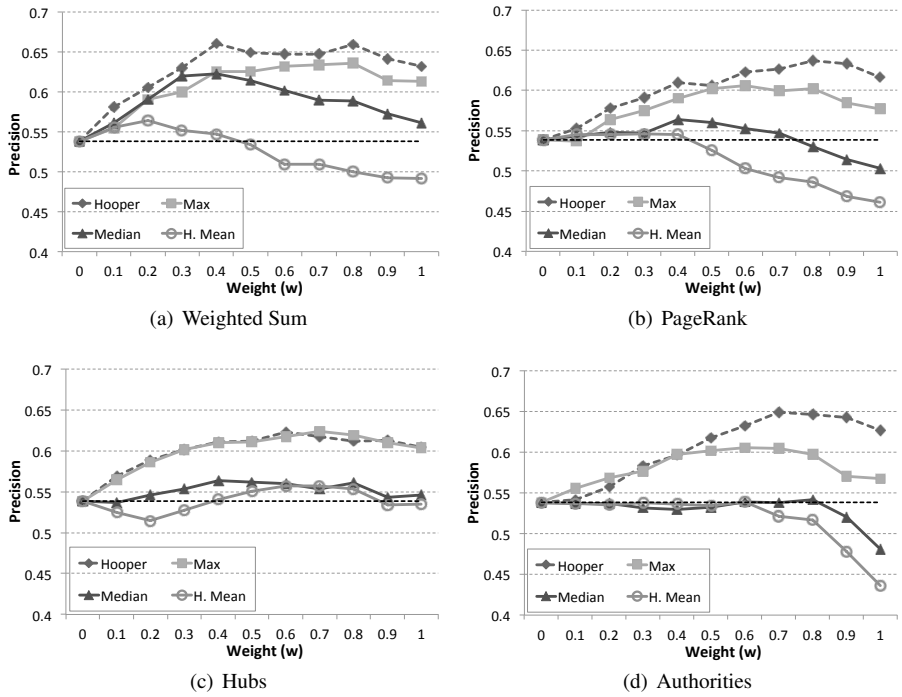
consume information. Intuitively this goes against our primary goal – to find a metric that best expresses how effective users are at not only producing content (in the case of HeyStaks, search results), but content that others deem useful. However it is entirely possible that HITS Hubs scores can also tell us something useful. For example a user could achieve a high Hubs score by facilitating the communication of quality content in his search community.

6.3 From Reputation to Recommendation

One strong test of the reputation models in this work is the extent to which they can improve the quality of results recommended by HeyStaks. Specifically, do the reputation models help by improving the relevance of the top ranked recommendations? To test this, in our evaluation we regenerate each of the recommendation lists produced during the trial using each of the item reputation models, and based on the user reputation scores calculated at the appropriate point in time. Since we have ground-truth relevance information for all of the recommendations (relative to the quiz questions), we can evaluate the objective quality of the resulting recommendations.

For the purpose of this work 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 condition, we count how often the top-ranked result is relevant for each query and then compute a simple precision metric by dividing this count by the total number of queries considered. Thus, precision returns a value between 0 and 1 and a precision of 0.5, for example, means that 50% of top-ranked results over all queries were relevant for a given condition. This precision metric provides a clear and concise approach to compare the number of relevant versus not relevant top-ranked search results returned by each of the combinations of user and item reputation models considered in this work. The rationale for focusing on the top recommendation instead of top k is that often during the trial HeyStaks made only a small number of recommendations. The manner in which HeyStaks operates is that it is limited to making no more than 3 recommendations as a practical way to avoid swamping the searcher with additional results. In many cases only single recommendations were made and so this was a practical approach given the sparsity of data. In a more open ended and longer-term trial it would be possible to consider the top 3 results but we leave this as a matter for future work.

In Section 4 we described four models for calculating user reputation and four different ways in which user reputation can be translated into item reputation scores. Thus, we have 16 different combinations of user and item models, and here we analyse the precision of each of these combinations when it comes to improving the quality of recommendations made by HeyStaks. Figures 15 (a)–(d) illustrate precision scores versus w (used in Equation 12 to control the relative influence of reputation) for each combination of user and item reputation model. The charts are organized according to user model, then item model. Remember that as a recommendation is made, its reputation score is calculated based on the reputation of its producers at recommendation time. Each chart shows the same value at $w = 0$ where reputation is not an influencing factor during the recommendation process. In each case, when $w = 0$, precision is 0.54. This is the baseline with which to measure the performance of each user and item reputation model combination; it represents the performance of HeyStaks using only relevance information to generate recommendations. As w increases, so too does the degree to which reputation influences recommendations, and each chart shows a similar trend. For each user reputation model there is at least one item model that brings about at least a 15% increase in precision – a precision of about 0.62 to the baseline's 0.54. For



(a) Weighted Sum (b) PageRank

(c) Hubs (d) Authorities

Fig. 15 Precision scores for all item models using the (a) Weighted Sum, (b) PageRank, (c) HITS Hubs and (d) HITS Authorities user reputation models.

example, Figure 15(b) shows that using PageRank paired with the Hooper item model brings about an optimal precision score of 0.64 at $w = 0.8$, a percentage increase of 19%. This means that 19% more relevant top ranked recommendations are made over the course of the trial.

For all models, a similar trend can be seen in that precision is observed to increase initially with w before decreasing as w approaches 1. Thus the relevance information HeyStaks uses to rank recommendations is needed in order to optimally rank recommendations; i.e. reputation alone does not provide best performance. Peak performance occurs in a broadly similar weighting range for each technique – for example, Max paired with HITS Authorities peaks at w=0.6 before declining in precision; see Figure 15(d). However the exact shape of this curve differs with each model combination. For example, some combinations such as Hooper paired with Authorities or PageRank does not reach an apex until $w = 0.7$ or $w = 0.8$ before dipping.

Each user model, when paired with one or more item models, achieved a significant level of increase in precision over the HeyStaks baseline. Overall, Weighted Sum, PageRank and HITS Authorities performed particularly well when paired with the Max and Hooper item models. Equally, each of these three user models were negatively affected when combined with the Harmonic Mean item model, achieving only marginal improvement on the baseline at low values of w before dipping below the baseline as w increased. Weighted Sum performed well when combined with Median with a top precision of 0.62 before descending back towards the baseline as w approaches 1. Although the trend is similar for HITS Author-

ities and PageRank, the top precision score was much lower when Median was used. This is indicative of the general trend showing Weighted Sum as the top performing user reputation model. Overall, HITS Hubs performed the least well of the user reputation models, achieving a maximum precision of 0.62 (combined with both Hooper and Max), the lowest across user models.

Examining the performance of item models, Hooper and Max consistently achieved the best precision across all user models. In the case of HITS Authorities for example, Hooper performed best by a considerable margin, with an optimum precision of 0.65. When combined with Weighted Sum, Hooper achieves the top precision observed in the experiment: At both $w = 0.4$ and $w = 0.8$ the combination achieved a precision score of 0.66 – a 22% improvement.

Not all item reputation models performed well. Median was an inconsistent performer, yielding a small degree of improvement when paired with PageRank, HITS Hubs and Weighted Sum, but little or no performance improvement in the case of HITS Authorities. The Harmonic Mean reputation model on the whole performed poorly regardless of the user reputation model it was paired with. In fact in many cases this item model yielded smaller precision values than the HeyStaks baseline. For example, a combination of Harmonic Mean and PageRank leads to a precision of only 0.47 when $w = 0.9$. So not only would users receive relevant recommendations less frequently than if standard HeyStaks was used (according to the baseline precision of 0.54), they would receive fewer relevant recommendations than not relevant ones.

Overall Figures 15 (a)–(d) show that including reputation information in the recommendation process can bring about considerable improvement in the system's capacity to deliver relevant recommendations to its users. In fact, when using Weighted Sum paired with Hooper's Reputation model we see up to a 22% percentage increase in precision over the HeyStaks baseline, a testament to the success of pairing reputation with the default HeyStaks' relevance-based scoring mechanism.

6.4 Analysis of Recommendations

The precision scores reported demonstrate different levels of benefit overall across the different approaches considered. But it is interesting to consider the source of these benefits. For example, do these benefits arise because of improvements in the recommendations associated with just one or two questions, or are they evident across a broader set of questions? To explore this we analyzed the data to identify those questions that enjoyed an improvement in precision and found that in the main, and depending on the particular approach used, an improvement was found across a range of questions. For example, in the case of our best performing combination – Weighted Sum paired with Hooper – precision was achieved across queries matched to 9 of the 20 questions posed; in other words for this combination 45% of the questions enjoyed recommendations that represented an improvement over those provided by HeyStaks alone. Figure 16 summarizes the results across the other combinations, using the optimal value of w in each case. We have also overlaid the precision achieved by each approach. A correlation can be seen between precision and number of questions over which an improvement was observed ($r = 0.74$).

Why did some questions enjoy benefits while others did not? It is likely that this is due to a combination of factors. Clearly the approach to reputation modeling is a factor in that different combinations of techniques deliver different overall precision improvement and across different questions. Question difficulty is also likely to play a role here in that there

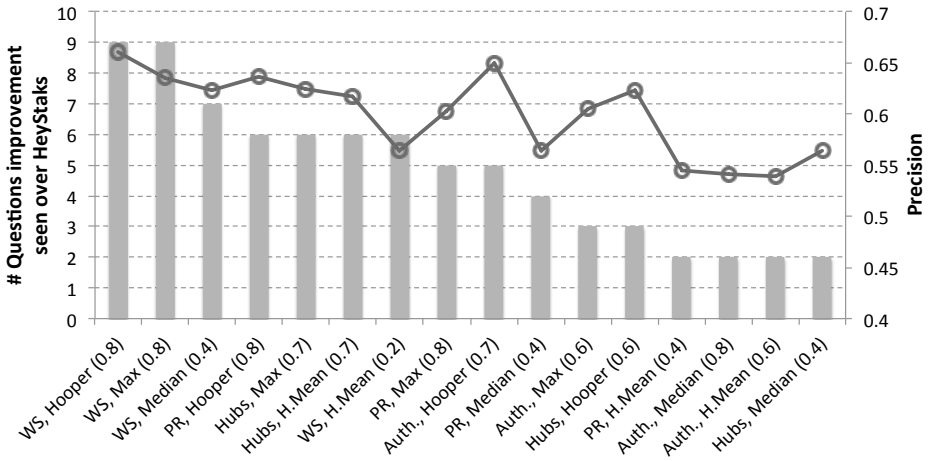


Fig. 16 Number of questions for which each reputation approach (using its optimal w value, shown in parentheses) achieved an improvement in relevant top-ranked recommendations, compared to HeyStaks. Overlaid is the precision score of each approach, also seen in Figure 15.

is more opportunity for improvement, and therefore an increase in precision, for more challenging questions. It is also likely impacted by the random ordering of questions presented to users in the trial.

6.5 Summary Analysis

To facilitate a more direct comparison of the performance of user and item reputation models, Figure 17 shows, for each item model, the median precision scores over w for each user model. Here, it is clear that the best performing item reputation model is Hooper, which outperforms all other item models irrespective of the particular user model employed. Further, it can be seen that the Weighted Sum user model outperforms the other user models (except when used in conjunction with Harmonic Mean). Overall, the best user-item model combination is Weighted Sum and Hooper, achieving a precision score of 0.66 compared to 0.65 (Figure 15) for the next best combination of Authorities and Hooper, and similar if not significantly greater performance improvements over the remaining user-item model combinations.

A Kruskal-Wallis one way analysis of variance test on precision scores across weights indicates there are statistically significant differences between the performance of the reputation model combinations at the 0.01 level. Using the Tukey-Kramer Method we examined pairwise differences between model combinations at the 0.01 level. A total of 12 pairwise comparisons yielded significant results. We observed that Weighted Sum, PageRank and Hubs, when paired with the Hooper item reputation model, were our top performers. These three combinations performed significantly better than four of the other user and item reputation model combinations: Weighted Sum with Harmonic Mean, PageRank with Harmonic Mean, and Authorities with Median and Harmonic Mean. No significant differences were observed between any of the other pairs of reputation model combinations. These findings confirm the strong performance achieved by three of the four user models and, in particular, the Hooper item reputation model.

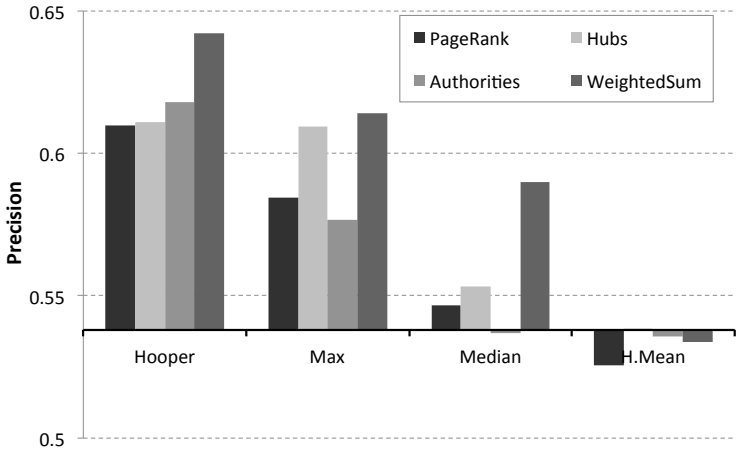


Fig. 17 Median precision score (over w) organized by item, then by user model.

6.6 Discussion

The results of this user trial clearly show the benefit that comes from integrating reputation information into the HeyStaks recommendation engine. We present our key findings in comparing combinations of user and item models to the HeyStaks baseline over the set of user queries obtained from the user trial:

1. Generally speaking a number of combinations of user and item reputation models are capable of delivering improvements in precision with respect to the HeyStaks baseline.
2. The precision scores recorded increase with respect to w to a point and typically apex in the $w = 0.6 - 0.8$ range.
3. Weighted Sum outperforms other user reputation models, regardless of item reputation model.
4. Hooper is the best performing item reputation model, delivering an average precision increase of 15% across all user reputation models, relative to standard HeyStaks.

The fact that Hooper and Max are significantly more effective at ranking pages than Median and Harmonic Mean is not surprising. Given a set of producer reputation scores for a candidate recommendation, Hooper and Max tend towards the upper bound of the distribution of scores, whereas median and harmonic mean provide a much more conservative reputation measure. For example, imagine a scenario where two very reputable users produce a result page, then two new users with low reputation select this page and become producers for future recommendations. In this case, it stands to reason that we would want to reward that page with a high score given its two more reputable producers rather than penalize it simply because two new users are also producers. However, Hooper's high performance may be peculiar to this trial due to the nature of the task assigned to participants: as the search task was the same for everyone, users tended to converge on the same pages. This meant that the number of producers of a page grew as the trial wore on. As such, the variation of producer reputation scores for a given page became greater. Hooper and Max still rewarded those pages with high reputation whereas Harmonic Mean penalized them, tending towards the lower bound of the set of scores. Anecdotally we have found that the context-aware nature of HeyStaks results in users converging on pages in staks, even in

a real-world setting, and thus we consider that Hooper or Max are the most suitable item reputation models going forward.

Differences between the user reputation models can be explained by examining how each user model scores users after each collaboration event. For example, one user may have contributed a single page to a stak that was recommended to and selected by a high-reputation consumer. In the case of Weighted Sum the reputation of this consumer is irrelevant from the point of view of updating the producer's reputation. In contrast for PageRank and both HITS Hubs and Authorities, the reputation of the consumer is taken into consideration when the link between consumer and producer is analyzed. Put simply, a previously low reputation user can gain a high degree of reputation by virtue of one high reputation user selecting that one recommendation. If we are scoring users based on their level of engagement with users on a collaborative platform and to prevent gaming, the degree with which they collaborate with other users should be the primary consideration. Since these criteria are best captured by the Weighted Sum model, we believe that this model is the most suitable of those considered.

Interestingly Hubs was a good performer, in fact Figure 17 shows Hubs outperformed both HITS Authorities and PageRank when coupled with Max, Median and Harmonic Mean item models. In a sense, this finding is counter-intuitive and highlights an interesting property of the HITS algorithm in this context. One might expect, for example, that the Authorities model would outperform Hubs, given that Authorities scores capture the extent to which users are good *producers* of quality search knowledge (i.e. users whose recommendations are frequently selected by other users), while Hubs captures the extent to which users are good *consumers* (i.e. users who select, tag, vote etc. HeyStaks recommendations deriving from the activity of good producers). However, given the manner in which the collaboration graph is constructed (Section 3.2), once a user has consumed (selected) a recommended result, then that user is also considered to be a producer of the result in question if it is subsequently recommended by HeyStaks and then selected by other users in the future. Thus, good consumers — who select recommended results from many good producers (i.e. producers with high Authority scores) — serve a "filter" for a broad base of quality search knowledge. Therefore re-ranking default HeyStaks recommendations using reputation scores from the Hubs model leads to the good recommendation performance observed above. However, as stated previously, a significant limitation of the Hubs model is that it is vulnerable to a simple form of gaming; for example, by adding many results to staks, users can accrue high reputation according to the Hubs model which is clearly not a desirable property from a robustness perspective.

6.7 Limitations of the Experiment

This article speaks to the benefit of utilizing reputation information to positively influence content presented by social recommender systems. We have shown this in the context of a live-user trial of the social search utility HeyStaks. Using reputation information gained by employing various user and item reputation models, we improved on the frequency at which HeyStaks makes relevant recommendations. We have presented generic models of user and item reputation, however in this study we have only showed how the models can be used in one of many possible platforms. The next important step is proving the utility of this approach in other domains. Once this is achieved, a key question to answer is can reputation scores from multiple domains be aggregated? We view such a question as an important one

to consider not just to augment this work, but as an important one for all future work in the area of online reputation.

Of course we acknowledge that the trial described in this paper has its limitations and that our results must be viewed in the context of these limitations. It is not a large-scale trial of thousands or millions of searchers. Such a trial might be possible in the context of conventional search engines but it is not feasible, at least not yet, for HeyStaks. Nevertheless the trial does involve a reasonable number of users and reflects a realistic search use-case. Of course this use-case — a fact-finding search task — also has its limitations. It is, for example, just one of the many reasons why users avail of search engines and there is clearly an opportunity for further work in order to broaden our evaluation to cover more open-ended search and discovery tasks; preliminary results for these open-ended style evaluations have been presented elsewhere in Smyth et al (2009). Nevertheless, our closed fact-finding search task does provide useful insights and facilitates a thorough evaluation with respect to an independent model of result relevance, in which the absolute relevance of individual results is known.

Finally, we demonstrated the benefits of our proposed approach in a very concrete, albeit offline, manner: by allowing reputation to influence recommendation ranking it was possible to significantly improve the relevance of the top-ranked recommendations made to users. Of course we are not able to conclude that this will mean that searchers are likely to benefit directly from this improved ranking, because we were not in a position to evaluate the actual responses of live users to these re-ranked recommendations. It is conceivable, for example, that searchers may avoid these more relevant results when they are ranked using reputation, while selecting them in the default HeyStaks ranking. However, this seems most unlikely and it is common practice in web search evaluations to acknowledge that there is an extremely strong bias between the position of results and their likelihood of selection (see Keane et al, 2008) and, as such, it is generally accepted that if one can produce rankings where top-ranked results are more relevant, then these rankings are likely to meet with a better user response. Hence we believe that the findings of the previous section have merit when considered from this viewpoint. Given that we have a ground-truth for relevance of recommended result pages, we can effectively replay queries that were entered by live-users, and this ground-truth allows us to measure relative recommendation efficacy with great confidence.

7 Conclusions

The ability to assess the reputation of online users is an important research area especially as we come to increasingly rely on social networks and connections for a wide range of tasks, from information finding to e-commerce and communication. The central contribution of this article is a generalized approach to calculating user and item reputation based on the collaboration (implicit or explicit, direct or indirect) that naturally occurs between users in a variety of online tasks and scenarios. Early work on computational reputation systems (Resnick et al, 2000; Shneiderman, 2000) asserts that users' past performance in online, interpersonal relationships can provide an indication of the extent to which others in their community trust them, and thus their overall reputation within that community. The reputation models put forward in this article are based on that assertion, leveraging information gained from examining past user collaborations to positively influence interactions in the future. This particular approach calculates reputation by using a collaboration graph as a harness. Past work has examined how a graph can be built using trust information explicitly

and directly given by users (Massa and Bhattacharjee, 2004; Massa and Avesani, 2007; Golbeck, 2006; Kuter and Golbeck, 2010). This article explores how a graph can be constructed using collaboration information that may be explicit or implicit, and direct or indirect. From this graph the reputation of the users within it can be calculated. We have shown how this approach can be developed in a social search context, proposed a variety of user and item reputation techniques, and demonstrated its overall effectiveness in the context of a live-user trial.

In this article we have explored different ways to translate user reputation scores into an overall page reputation/relevance score. We have described the results of a comparative evaluation based on real-user data, albeit in a constrained search setting, which highlights the ability of these techniques to improve overall recommendation quality, when combined with the relevance-based recommendation ranking metrics that are currently used by HeyStaks. For example, many of the page reputation models can improve precision in delivering relevant recommendations (compared to the standard HeyStaks benchmark) by over 15%. Moreover, we have found that by combining our Weighted Sum user reputation model with an enforcement-based item scoring mechanism based on Hooper's rule for Concurrent Testimony (Shafer, 1986), relative performance improvements of up to 22% are delivered. We believe that this work lays the ground-work for future research in this area which will focus on scaling-up the role of reputation in online social platforms and refining the combination of reputation and content utility.

Our focus in this work was on the role of reputation during the recommendation process, in order to maximise the relevance of the community recommendations made by HeyStaks. But this is just one use of reputation in a system such as HeyStaks. For example, in many social systems there is the risk that malicious users will attempt to manipulate the outcome of social processes; see relevant work in recommender systems research (Lam and Riedl, 2004; Mobasher et al, 2007; O'Mahony et al, 2002). In HeyStaks, for example, it is possible for malicious users to flood search staks with irrelevant or self-interested results, which could impact recommendation quality. By using reputation to mediate recommendation it will be possible to guard against this; these malicious users will have low reputation scores (assuming their contributions are rarely acted on by other users) and as such, their contributions will be unlikely to appear in future recommendation sessions. Furthermore, reputation can be exposed to users of systems like HeyStaks as an important social signal. For example, although HeyStaks' recommendations are anonymous (so users do not know the source of result recommendations at search time), it may make sense to explain recommendations with reference to the reputation of producers in the future.

References

von Ahn L (2006) Games With A Purpose. IEEE Computer Magazine pp 96–98

Amershi S, Morris MR (2008) CoSearch: a System for Co-located Collaborative Web Search. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08), Florence, Italy, ACM, New York, pp 1647–1656

Aral E, Muchnik L, Sundararajan A (2009) Distinguishing Influence-based Contagion From Homophily-driven Diffusion in Dynamic Networks. Proceedings of the National Academy of Sciences 106:21,544–21,549

Avesani P, Massa P, Tiella R (2005) A Trust-enhanced Recommender System Application: Moleskiing. In: Proceedings of the 2005 ACM Symposium on Applied Computing (SAC '05), Santa Fe, New Mexico, ACM, New York, pp 1589–1593

Bailey TL, Gribskov M (1998) Combining Evidence Using p-values: Application to Sequence Homology Searches. Bioinformatics, Oxford University Press 14:48–54

Bakshy E, Hofman JM, Mason WA, Watts DJ (2011) Everyone's an Influencer: Quantifying Influence on Twitter. In: Proceedings of the fourth ACM International Conference on Web Search and Data Mining (WSDM '11), Hong Kong, China, ACM, New York, pp 65–74

Basu C, Hirsh H, Cohen W (1998) Recommendation as Classification: Using Social and Content-based Information in Recommendation. In: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative applications of Artificial Intelligence (AAAI/IAAI '98), Madison, Wisconsin, USA, American Association for Artificial Intelligence, Palo Alto, USA, pp 714–720

Brin S, Page L (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings of the 7th International Conference on World Wide Web (WWW '98), Brisbane Australia, ACM, New York, pp 107–117

Cai K, Bao S, Yang Z, Tang J, Ma R, Zhang L, Su Z (2011) OOLAM: an Opinion Oriented Link Analysis Model for Influence Persona Discovery. In: Proceedings of the fourth ACM International Conference on Web Search and Data Mining (WSDM '11), Hong Kong, China, ACM, New York, pp 645–654

Canini K, Suh B, Pirolli P (2011) Finding Credible Information Sources in Social Networks Based on Content and Social Structure. In: Proceedings of the Third IEEE International Conference on Social Computing (SocialCom '11), Boston, Massachusetts, USA, pp 1 –8

Chatterjee K, de Alfaro L, Pye I (2008) Robust Content-driven Reputation. In: Proceedings of the 1st ACM Workshop on Artificial Intelligence and Security (AISec '08), Alexandria, Virginia, USA, ACM, pp 33–42

Cheng R, Vassileva J (2005) Adaptive Reward Mechanism for Sustainable Online Learning Community. In: Proceedings of the 2005 Conference on Artificial Intelligence in Education (AIED '05), Amsterdam, The Netherlands, IOS Press, pp 152–159

De Alfaro L, Kulshreshtha A, Pye I, Adler BT (2011) Reputation Systems for Open Collaboration. Communications of the ACM, ACM, New York 54:81–87

Dellarocas C (2003) The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. Management Science 49:1407–1424

Duan Y, Jiang L, Qin T, Zhou M, Shum HY (2010) An Empirical Study on Learning to Rank of Tweets. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10), Beijing, China, Association for Computational Linguistics, pp 295–303

Gambetta D, et al (2000) Can we Trust Trust? Gambetta, D (ed) Trust: Making and Breaking Cooperative Relations, University of Oxford 2000:213–237

Goh D, Ang R, Lee C, Chua A (2011) Fight or unite: Investigating game genres for image tagging. Journal of the American Society for Information Science and Technology 62(7):1311–1324

Golbeck J (2006) Generating Predictive Movie Recommendations from Trust in Social Networks. In: Proceedings of the 4th International Conference on Trust Management (iTrust'06), Pisa, Italy, Berlin, Heidelberg: Springer-Verlag, pp 93–104

Golbeck J, Hendler J (2004) Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networks. In: Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW '04), Whittlebury Hall, Northamptonshire, UK

Guha R, Kumar R, Raghavan P, Tomkins A (2004) Propagation of trust and distrust. In: Proceedings of the 13th international conference on World Wide Web (WWW '04), New

York, NY, USA, ACM, pp 403–412

Guy I, Perer A, Daniel T, Greenshpan O, Turbahn I (2011) Guess Who?: Enriching the Social Graph Through a Crowdsourcing Game. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11), Vancouver, Canada, ACM, New York, pp 1373–1382

Hacker S, von Ahn L (2009) Matchin: Eliciting User Preferences with an Online Game. In: In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09), Boston, USA, ACM, New York, pp 1207–1216

Joinson AN (2008) Looking at, Looking up or Keeping up with People?: Motives and Use of Facebook. In: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '09), Florence, Italy, ACM, New York, pp 1027–1036

Jones S, Wilikens M, Morris P, Masera M (2000) Trust Requirements in e-Business. Communications of the ACM, ACM, New York 43(12):81–87

Jøsang A, Golbeck J (2009) Challenges for Robust Trust and Reputation Systems. In: 5th International Workshop on Security and Trust Management (STM '09), Saint Malo, France, Springer LNCS

Jøsang A, Ismail R, Boyd C (2007) A Survey of Trust and Reputation Systems for Online Service Provision, elsevier. Decision Support Systems 43(2):618–644

Keane MT, O'Brien M, Smyth B (2008) Are People Biased in their Use of Search Engines? Communications of the ACM, ACM, New York 51:49–52

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the Spread of Influence Through a Social Network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03), Washington D.C., USA, ACM, New York, pp 137–146

Kleinberg JM (1999) Authoritative Sources in a Hyperlinked Environment. Journal of the ACM 46(5):604–632

Kuter U, Golbeck J (2007) SUNNY: a New Algorithm for Trust Inference in Social Networks using Probabilistic Confidence Models. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI '07), Vancouver, Canada, AAAI Press, Palo Alto, USA, pp 1377–1382

Kuter U, Golbeck J (2010) Using Probabilistic Confidence Models for Trust Inference in Web-based Social Networks. ACM Transactions on Internet Technologies, ACM, New York 10(2):1–23

Lam SK, Riedl J (2004) Shilling Recommender Systems for Fun and Profit. In: WWW '04: Proceedings of the 13th International World Wide Web Conference, ACM, New York, pp 393–402

Langville A, Meyer C (2005) A Survey of Eigenvector Methods for Web Information Retrieval. Society for Industrial and Applied Mathematics Review (SIAM), SIAM, Philadelphia, USA 47(1):135–161

Law E, von Ahn L (2009) Input-Agreement: A New Mechanism for Collecting Data Using Human Computation Games. In: In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09), Boston, USA, ACM, New York, pp 1197–1206

Lazzari M (2010) An Experiment on the Weakness of Reputation Algorithms Used in Professional Social Networks: The Case of Naymz. In: IADIS International Conference e-Society, Porto, Portugal, pp 519–522

Li H, Bhowmick SS, Sun A (2011) Casino: towards conformity-aware social influence analysis in online social networks. In: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '11, pp 1007–1012, DOI http://doi.acm.org/10.1145/2063576.2063721, URL `http://doi.`

acm.org/10.1145/2063576.2063721

Lih A (2004) Wikipedia as Participatory Journalism: Reliable Sources? Metrics for Evaluating Collaborative Media as a News Resource. In: Proceedings of the 5th International Symposium on Online Journalism, pp 16–17

Massa P, Avesani P (2007) Trust-aware Recommender Systems. In: RecSys '07: Proceedings of the 2007 ACM Conference on Recommender Systems, ACM, pp 17–24

Massa P, Bhattacharjee B (2004) Using Trust in Recommender Systems: An Experimental Analysis. In: In Proceedings of the 2nd International Conference on Trust Management (iTrust '04), pp 221–235

Mayer R, Davis J, Schoorman F (1995) An Integrative Model of Organizational Trust. Academy of Management Review, JSTOR pp 709–734

McKnight D, Chervany N (1996) The Meanings of Trust. Technical Report WP9604, University of Minnesota Management Information Research Center

McNally K, O'Mahony MP, Smyth B, Coyle M, Briggs P (2010) Towards a Reputation-based Model of Social Web Search. In: Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10), Hong Kong, China, ACM, New York, pp 179–188

McNally K, O'Mahony MP, Smyth B, Coyle M, Briggs P (2011) A Case-study of Collaboration and Reputation in Social Web Search. ACM Transactions on Intelligent Systems Technology (TIST), ACM, New York 3(1):4:1–29

Mobasher B, Burke R, Bhaumik R, Williams C (2007) Toward Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness. ACM Transactions on Internet Technology (TOIT), ACM New York 7(4):1–40

Morris MR, Horvitz E (2007a) S^3: Storable, Shareable Search. In: Proceedings of Human-Computer Interaction - INTERACT 2007, 11th IFIP TC 13 International Conference, Rio de Janeiro, Brazil, Springer Verlag, pp 120–123

Morris MR, Horvitz E (2007b) SearchTogether: an Interface for Collaborative Web Search. In: Proceedings of the 21st ACM Symposium on User Interface Software and Technology (UIST '07), Newport, Rhode Island, USA, ACM, New York, pp 3–12

Mui L, Mohtashemi M, Halberstadt A (2002) A Computational Model of Trust and Reputation. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS '02), Big Island, Hawaii, USA, IEEE, Palo Alto, USA, pp 2431–2439

O'Donovan J (2009) Capturing Trust in Social web Applications. In: Golbeck J (ed) Computing with Social Trust, Springer, pp 213–257

O'Donovan J, Smyth B (2005) Trust in Recommender Systems. In: Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05), San Diego, California, USA, ACM, New York, pp 167–174

Olson JS, Olson GM (2000) i2i Trust in e-Commerce. Communications of the ACM, ACM, New York 43(12):41–44

O'Mahony MP, Hurley NJ, Silvestre GCM (2002) Promoting Recommendations: An Attack on Collaborative Filtering. In: Proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA '02), Aix en Provence, France, Springer, Aix-en-Provence, France, pp 494–503

Page L, Brin S, Motwani R, Winograd T (1999) The PageRank Citation Ranking: Bringing Order to the Web. Technical Report

Pal A, Counts S (2011) Identifying Topical Authorities in Microblogs. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11), Hong Kong, China, ACM, New York, New York, NY, USA, pp 45–54

Pariser E (2011) The Filter Bubble: What the Internet is Hiding From You. Penguin Press HC

Phelan O, McCarthy K, Smyth B (2011) Yokie: A Curated, Real-time, Search and Discovery System using Twitter. In: 3rd Workshop on Recommender Systems and the Social Web, in association with The 5th ACM Conference on Recommender Systems (RecSys 2011)

Preston R, Preston S (2007) The Official Biggest Pub Quiz Book Ever! Carlton Books Ltd.

Raub W, Weesie J (1990) Reputation and Efficiency in Social Interactions: An Example of Network Effects. American Journal of Sociology, JSTOR pp 626–654

Recuero R, Araujo R, Zago G (2011) How Does Social Capital Affect Retweets? In: Adamic LA, Baeza-Yates RA, Counts S (eds) The Fifth International AAAI Conference on Weblogs and Social Media (ICWSM '11), Barcelona, Spain, AAAI Press, Palo Alto, USA

Resnick P, Zeckhauser R (2002) Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. Advances in Applied Microeconomics, Emerald Group Publishing Limited 11:127–157

Resnick P, Zeckhauser R, Friedman E, Kuwabara K (2000) Reputation Systems: Facilitating Trust in Internet Interactions. Communications of the ACM, ACM, New York 43(12):45–48

Rousseau D, Sitkin S, Burt R, Camerer C (1998) Not so Different After All: A Cross-discipline View of Trust. Academy of Management Review, Academy of Management 23(3):393–404

Salton G, McGill MJ (1983) Introduction to Modern Information Retrieval. McGraw-Hill

Schaal M, Fidan G, Müller RM, Dagli O (2010) Quality Assessment in the Blog Space. The Learning Organization, Special Issue on Web20: New and Challenging Practical Issues 17(6):529–536

Shafer G (1986) The Combination of Evidence. International Journal of Intelligent Systems, Wiley Online Library 1(3):155–179

Shneiderman B (2000) Designing Trust into Online Experiences. Communications of the ACM, ACM, New York 43(12):57–59

Siorpaes K, Hepp M (2008) Games with a Purpose for the Semantic Web. Intelligent Systems, IEEE, Palo Alto, USA 23(3):50 –60

Smyth B (2007) A Community-based Approach to Personalizing Web Search. IEEE Computer 40(8):42–50

Smyth B, Briggs P, Coyle M, O'Mahony MP (2009) Google Shared. A Case-Study in Social Search. In: The 17th conference on User Modeling, Adaptation, and Personalization (UMAP '09), Trento, Italy, Springer-Verlag, pp 283–294

Speretta M, Gauch S (2005) Personalized Search Based on User Search Histories. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI '05), IEEE Computer Society, Washington, DC, USA, pp 622–628

Voorbraak F (1995) Combining unreliable pieces of evidence. Tech. rep., University of Amsterdam

Walsh G, Golbeck J (2010) Curator: A Game With a Purpose for Collection Recommendation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10), Atlanta, Georgia, USA, ACM, New York, pp 2079–2082

Weng J, Lim EP, Jiang J, He Q (2010) TwitterRank: Finding Topic-sensitive Influential Twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM '10), New York, New York, USA, ACM, New York, pp 261–270

Wu JJ, Tsang ASL (2008) Factors Affecting Members' Trust Belief and Behaviour Intention in Virtual Communities. Behaviour Information Technology, Taylor & Francis, Inc

27:115–125
Yang J, Adamic LA, Ackerman MS (2008) Competing to Share Expertise: the Taskcn Knowledge Sharing Community. In: Proceedings of the 2nd AAAI Conference on Weblogs and Social Media (ICWSM '2008), Seattle, Washington, USA
Zeng H, Alhossaini MA, Ding L, Fikes R, McGuinness DL (2006) Computing Trust from Revision History. In: Proceedings of the 2006 International Conference on Privacy, Security and Trust (PST '06), Markham, Ontario, Canada, ACM, New York, pp 8:1–8:1