Computational Intelligence, Volume 99, Number 000, 2009

SPEAR: SPAMMING-RESISTANT EXPERTISE ANALYSIS AND RANKING IN COLLABORATIVE TAGGING SYSTEMS

Ching-man Au Yeung[1], Michael G. Noll[2]
Nicholas Gibbins[1], Christoph Meinel[2], Nigel Shadbolt[1]

[1]*School of Electronics and Computer Science, University of Southampton Southampton, SO17 1BJ, United Kingdom*
[2]*Hasso-Plattner-Institut, Unversity of Potsdam, 14440 Potsdam, Germany*

In this paper we discuss the notions of experts and expertise in resource discovery in the context of collaborative tagging systems. We propose that the level of expertise of a user with respect to a particular topic is mainly determined by two factors. Firstly, an expert should possess a high quality collection of resources, while the quality of a Web resource in turn depends on the expertise of the users who have assigned tags to it, forming a mutual reinforcement relationship. Secondly, an expert should be one who tends to identify interesting or useful resources before other users discover them, thus bringing these resources to the attention of the community of users. We propose a graph-based algorithm, *SPEAR (SPamming-resistant Expertise Analysis and Ranking)*, which implements the above ideas for ranking users in a folksonomy. Our experiments show that our assumptions on expertise in resource discovery, and SPEAR as an implementation of these ideas, allow us to promote experts and demote spammers at the same time, with performance much better than the original HITS algorithm and simple statistical measures currently used in most collaborative tagging systems.

Key words: Collaborative tagging, Expertise, Folksonomy, HITS, Ranking, Spamming.

1. INTRODUCTION

Collaborative tagging systems such as Delicious[1] and Flickr[2] have gained wide spread popularity in recent years. Web users post their favourite resources, such as bookmarks and digital photos, to these systems and assign descriptive keywords, usually referred to as tags, to the resources for the purpose of categorizing them or sharing them with other users (Ames and Naaman, 2007; Noll and Meinel, 2008). Such collaborative activity of tagging eventually produces user-generated categorization schemes now commonly known as *folksonomies* (Mathes, 2004; Golder and Huberman, 2006).

The rise of collaborative tagging and folksonomies provide users with a new means of searching for interesting or useful resources on the Web. The number of times that a tag has been assigned to a particular resource on the Web suggests how popular it is among the users and how relevant it is to the topic represented by the tag. On the other hand, identifying "experts" who are knowledgeable in a particular area and browsing their collection of resources can be another effective way of discovering useful resources (John and Seligmann, 2006). As most collaborative tagging systems are open to any user, users can subscribe to the collection of experts such that they will be notified when these experts have discovered new and useful resources.

However, the tasks of identifying resources which are of high quality—interesting, useful, or relevant—and identifying users who are knowledgeable with respect to a particular topic are not trivial. Existing tagging systems usually provide only a list of resources or users either in the order of how frequently or how recently they appear in the system. These two methods, however, do not necessarily result in useful rankings of resources and users due to a variety of reasons, one of which

[1]Delicious: http://delicious.com/
[2]Flickr: http://www.flickr.com/

being that there are spammers abusing the systems for their own malicious purposes. For example, Wetzker et al. (2008) find out that 19 out of the 20 most active users on Delicious are spammers.

In this paper, we discuss the notions of experts and expertise in the context of collaborative tagging systems. In general, an expert should be a user who is knowledgeable in a particular topic and possesses a collection of resources which are relevant to the topic, where the topic is represented by a tag or a set of tags. We propose two dimensions along which the expertise of a user can be measured in a collaborative tagging system. Firstly, an expert should possess a high quality collection of resources, while the quality of a Web resource depends on the expertise of the users who have assigned tags to it. In other words, there is a kind of mutual reinforcement between the expertise of a user and the quality of a resource. Secondly, an expert should be one who tends to identify interesting or useful resources before other users discover them. If a resource becomes very popular after a user has first discovered it, the user should be given credit for bringing this resource to the attention of the community.

We propose a graph-based algorithm, *SPEAR (SPamming-resistant Expertise Analysis and Ranking)*, which implements the above ideas for ranking users in a collaborative tagging system according to their expertise with respect to a particular topic. To overcome the problem of the lack of ground truths in the evaluation process, we design a novel way of injecting simulated users into real world data obtained from Delicious for the purpose of evaluation. We carry out experiments on both simulated data sets and real-world data sets to find out how the algorithm ranks different types of users. Our experiments show that our assumptions on expertise in resource discovery, and SPEAR as an implementation of these ideas, allow us to promote experts and demote spammers at the same time. The performance is much better than the original HITS algorithm and simple statistical measures such as ranking users by how many times they have used a tag.

The rest of the paper is structured as follows. Section 2.1 provides a brief review of collaborative tagging and folksonomies, and discusses related works which address ranking and expertise in collaborative tagging. Section 3 discusses the notions of expertise and experts in the context of collaborative tagging. We introduce and describe in detail SPEAR in Section 4 and describe our experiments in Section 5. Finally we give our conclusions and mention future research directions in Section 6.

2. BACKGROUND

2.1. Collaborative Tagging

Tagging, the act of assigning tags to online resources for the purposes of organization and sharing, is based on the simple idea of using descriptive keywords to describe and index resources. A *collaborative tagging system* (Golder and Huberman, 2006) takes this idea further by allowing arbitrary users to assign tags freely to any resources available on the Web. In most collaborative tagging systems such as Delicious and LibraryThing,[3] any user can maintain his own set of tags for a particular resource he posts to the system. Such an approach offers several advantages over traditional methods for organizing information. For example, users do not need to follow any predefined vocabulary and can use any keywords they like, allowing new terms or concepts to be used to provide more appropriate descriptions.

When the tags and resources contributed by different users are aggregated, a kind of user-generated classification scheme emerges. For example, the fact that tags such as `search`, `engine` and `tools` are assigned most frequently by users to the URL `http://www.google.com/` gives some idea as to what the page is about. Such bottom-up classification schemes have been given the name *folksonomies* (Golder and Huberman, 2006). A folksonomy basically involves three types of entities, namely users, tags and resources/documents. Since we focus our analysis on Delicious in this paper, resources will mostly be Web documents identified by their URLs. Formally, a folksonomy can be represented as a tripartite hypergraph of users, tags and documents (Mika, 2005; Lambiotte and Ausloos, 2006).

[3] http://www.librarything.com/

Definition 1: A folksonomy $\mathcal{F}$ is a tuple $\mathcal{F} = (U, T, D, R)$, where U is a set of users, T a set of tags, D a set of documents, and $R \subseteq U \times T \times D$ a set of annotations.

R is sometimes referred to as a set of *taggings* or *tag assignments*. It represents the fact that a particular user $u \in U$ has assigned a tag $t \in T$ to a document $d \in D$. In practice, a user usually does not assign tags to a particular document separately. Instead, the user creates a post to the system consisting of a set of tags that are assigned to the document. In the context of Delicious, such a post is referred to as a *bookmark*.

Since we are interested in ranking users by their level of expertise in a particular topic, we will focus on different subsets of the whole folksonomy. For example, if the topic is represented by the tag t, we can extract a subset $\mathcal{F}_t$ of $\mathcal{F}$ as follows:

$$\mathcal{F}_t = (U_t, D_t, R_t) \tag{1}$$

where

$$\begin{aligned} R_t &= \{(u,d)|(u,t,d) \in R\} \\ U_t &= \{u|(u,d) \in R_t\} \\ D_t &= \{d|(u,d) \in R_t\} \end{aligned}$$

This can be generalized to cases in which the topic is represented by a conjunction or disjunction of two or more tags $\{t_1, t_2, ..., t_n\}$:

$$R_{\{t_1 \wedge ... \wedge t_n\}} = \{(u,d)|(u,t_1,d) \in R \wedge ... \wedge (u,t_n,d) \in R\}$$

or

$$R_{\{t_1 \vee ... \vee t_n\}} = \{(u,d)|(u,t_1,d) \in R \vee ... \vee (u,t_n,d) \in R\}$$

2.2. Related Work

Expert identification and ranking have been studied extensively in the information retrieval community. The task mainly involves building candidate profiles by associating documents relevant to a certain topic with the candidates by co-occurrence analysis, and employing information retrieval techniques on the profiles to retrieve and rank the candidates (Macdonald et al., 2008). More recent approaches involve graph-based analysis of the network of users in a community. For example, Dom et al. (2003) study the performance of different graph-based ranking algorithms on expertise ranking in email exchanges. Zhang et al. (2007), on the other hand, apply an algorithm based on the PageRank algorithm to produce expertise ranking of users of a Java Developer bulletin board.

Although folksonomies can be easily represented as graphs, graph-based ranking methods such as HITS or PageRank cannot be directly applied to folksonomies due to their tripartite structure. Either the algorithms have to be adapted to handle tripartite graphs instead of simple or bipartite graphs, or folksonomies have to be reduced to simpler graph structures. John and Seligmann (2006) discuss expertise in collaborative tagging in the context of finding experts in the enterprise. The authors propose an iterative ranking algorithm based on PageRank which not only considers the number of times a user has used a particular tag, but also the number of times he has used other tags and how related these tags are to the tag which represents the topic in question. Other works in the literature focus on the more general issue of ranking any entities in a folksonomy. For example, Hotho et al. (2006) propose the FolkRank algorithm which is also based on the PageRank algorithm, for providing ranking of users, tags, and documents at the same time. The algorithm is a topic-specific and personalised ranking method which makes use of a preference vector. Along a similar line of thought, Bao et al. (2007) propose the SocialPageRank algorithm based on the mutual reinforcement of the levels of popularity between the three entities in a folksonomy. This can be considered as an adaptation of the HITS algorithm (Kleinberg, 1999) to the tripartite structure of a folksonomy.

While the above ranking methods are reported to produce satisfactory results, they are very likely to be vulnerable to spamming activities in collaborative tagging systems. This is because rankings produced by PageRank-based algorithms are highly dependent on the popularity of the entities being ranked. In addition, it is found that most of the highly active users in Delicious are actually spammers (Wetzker et al., 2008). Regarding spamming activities in collaborative tagging, Koutrika et al. (2007) are the first to discuss methods of tackling spams in collaborative tagging

systems. They propose that the "reliability" of users—whether their tags coincide with those of the others—should be taken into account to produce a ranking of documents which is more resistant to spammers. Reliability can be measured by the extent to which a user's tags are similar to those of other users. However, reliability is only one of the measures needed to tackle spammers. While a ranking based on reliability can demote spammers who deliberately assign wrong tags to resources, it is less likely to be able to demote more sophisticated spammers who for example try to game the system by posting a large number of resources to gain reputation.

Demotion of spammers in a ranking tries to reduce the prominence of the spammers in a system. There are other methods which can be used to tackle spamming activities. For example, some studies apply machine learning algorithms for detecting abnormal behavioral patterns to identify spammers in collaborative tagging systems (Krestel and Chen, 2008; Madkour et al., 2008). These approaches usually involve supervised learning, meaning that they require training data in which real spammers are manually labeled. However, training data might not be always available, and very often it requires considerable efforts to identify spammers for training a classifier. We therefore believe that using a ranking algorithm to reduce the prominence of spammers in an unsupervised manner is complementary to the weakness of detection methods. In addition, prevention methods such as challenging the users with hard AI problems when they perform tagging activities can also be used to avoid automated bots to spam tagging systems (Heymann et al., 2007).

3. EXPERTS IN COLLABORATIVE TAGGING SYSTEMS

In order to identify experts and to rank users according to their expertise, it is necessary to first have an idea of the characteristics we are looking for in an expert. In a general context, an *expert* is someone with a high level of knowledge, technique or skills in a particular domain. It implies that experts are individuals that we can consult for as reliable sources of relevant resources and information. This general idea can be readily applied to the context of collaborative tagging. In this section, we describe and justify two assumptions we have for experts in a collaborative tagging system.

3.1. User Expertise and Document Quality

The simplest way to assess the *expertise* of a user in a given topic is by the number of times he has used the corresponding tag (or set of tags) on some documents. This approach is most commonly seen in existing collaborative tagging systems. For example, on any page that is dedicated to a particular tag, LibraryThing, an online service to help people catalog and organize their books, presents a list of the top users of that tag. However, such an approach does not consider the obvious facts that quantity does not imply quality, and that spammers who indiscriminately tag a large number of documents may be mistaken as experts (Wetzker et al., 2008).

Studies in psychology have found that expertise involves the ability to select the most relevant information for achieving a goal (Feltovich et al., 2006). In the context of collaborative tagging, users assign tags to resources so as to facilitate retrieval in case the resources are useful to their information needs in the future. Therefore, we believe that an expert should be someone who not only has a large collection of documents annotated with a particular tag, but should also be someone who tends to add *high quality* documents to their collections. The quality of documents will in turn be determined by the number as well as the expertise of the users who have kept these documents in their collections. In other words, there is a relationship of mutual reinforcement between the expertise of a user and the quality of a document.

This approach is similar to the HITS algorithm (Kleinberg, 1999) for link structure analysis among Web pages, in which the concepts of *hubness* and *authority* of a page mutually reinforce each other. A major difference in our case is that collaborative tagging involves two different kinds of interrelated entities, namely human users and Web documents, instead of only Web pages in the case of HITS. Additionally, there are only links pointing from users to documents but not vice versa. Thus in our case users will only receive hub scores (expertise) whereas documents will only receive authority scores (quality). This, however, makes sense because experts act as hubs when we find

useful resources through them, and documents act as authority as they contain the information we need.

3.2. Discoverer vs. Follower

While the HITS approach for measuring expertise of users and quality of documents at the same time is a very intuitive and reasonable method, we have two concerns about whether it alone is sufficient to give good performance. Firstly, in the HITS approach, two users will be considered to have the same level of expertise even though one is the first to tag a set of documents and the other is simply tagging the documents because they are already popular in the community. Secondly, a spammer who wants promote some Web pages to other users can easily exploit this weakness and boost his expertise score by tagging lots of popular documents (Heymann et al., 2007).

Hence, in addition to knowing a lot of high quality documents per se, we believe an expert to be someone who is also able to recognize the usefulness of a document before others do (Chi, 2006), thus becoming the first to bookmark and tag it, and by doing so bringing it to the attention of other users of the collaborative tagging system. This aspect of expertise is similar to a distinguished researcher who not only has profound knowledge of existing publications and prior art in his area of expertise, but who is also able to advance the field by original research of his own.

In other words, experts should be the *discoverers* of high quality documents, in contrast to the *followers* who find these documents at a later time, for example because the documents have already become popular or they have been featured in the mass media in the meantime. Generally speaking, the earlier a user has tagged a document, the more *credit* he should receive.

With this assumption, we are introducing the *time* of tagging a document as an additional dimension for determining the expertise of a user. While we can never know how a user discovered a document (either by himself or by navigating within the collaborative tagging system), the time at which the user bookmarked the document is still a reasonable approximation of how sensitive he is to new information with respect to the topic.

The notion of discoverers and followers with differing credit scores is related to protection mechanisms against Sybil attacks (Yu et al., 2006) in information security. In a Sybil attack, a malicious user creates multiple user identities in order to boost his reputation or "trust score" within a system such as a peer-to-peer network. However, an attacker can create many identities but only few trust relationships, particularly with participants outside his fake user network. This aspect can be exploited to identify Sybil attacks. Similarly, a spammer that floods a collaborative tagging system for boosting his expertise score will end up being either just a follower (in case he focuses on documents that are already popular within the user community) or a discoverer without any followers (in case he introduces his own spam documents to the community that nobody else cares about). In both cases, he will not benefit much from his malicious activities.

We believe that the discoverer-follower assumption is both a reasonable and a desirable one because experts should be the ones who bring good documents to the attention of novices. In addition, this also makes our method of ranking expertise more resistant to the type of spammer mentioned above (more on this in Section 5).

4. SPAMMING-RESISTANT EXPERTISE ANALYSIS AND RANKING

We propose *SPEAR (SPamming-resistant Expertise Analysis and Ranking)* as an algorithm to produce a ranking of users with respect to a set of one or more tags based on the assumptions above.

Without loss of generality, we assume that the topic of interest is represented by a tag $t \in T$ (see section 2.1). We therefore focus on users who have used tag t for annotations, and documents which have been assigned tag t. The first step of the algorithm is to extract a set of taggings R_t from the folksonomy $\mathcal{F}$. As we also take into consideration the time at which a tagging is created, we extend the notion of tagging by associating a timestamp to each tagging. Hence, every tagging becomes a tuple of the form: $r = (u, t, d, c)$ where c is the time when user u assigned the tag t to document d, and $c_1 < c_2$ if c_1 refers to an earlier time than c_2.

Since our algorithm is based on the HITS (Hypertext Induced Topic Search) algorithm (Klein-

berg, 1999), we therefore first give a brief introduction of this algorithm before describing in detail our proposed SPEAR algorithm.

4.1. The HITS Algorithm

The HITS algorithm is an algorithm that performs link analysis in order to produce a ranking of Web documents. It measures two characteristics of documents, namely authority and hubness. Authoritative documents are those that provide good information with respect to a chosen topic, while hubs are documents that points to good authorities.

According to the assumptions of the algorithm, these two characteristics have a mutual reinforcement relationship: a document has high authority if many documents pointing to it have high hubness, and a document has high hubness if it points to many documents with high authority. Mathematically, the authority $a(d)$ and hubness $h(d)$ of a document d can be defined as follows:

$$a(d) \leftarrow \sum_{d' \in P(d)} h(d') \tag{2}$$

$$h(d) \leftarrow \sum_{d' \in C(d)} a(d') \tag{3}$$

where $P(d)$ is the set of documents with a link to d, and $C(d)$ is the set of documents pointed to by d.

The above operations can be represented using linear algebra. Let $\vec{a}$ be an n-dimensional vector of authority weights and $\vec{h}$ be another n-dimensional vector of hubness weights for n documents. In addition, let $\mathbf{A}$ be an $n \times n$ square matrix such that $\mathbf{A}_{i,j} = 1$ if document d_i has a link to document d_j, and $\mathbf{A}_{i,j} = 0$ otherwise. Then the algorithm at the kth iteration can be represented by the following equations:

$$\vec{a}_k = \alpha_k \mathbf{A}^T \vec{h}_{k-1} \tag{4}$$

$$\vec{h}_k = \beta_k \mathbf{A} \vec{a}_{k-1} \tag{5}$$

where α_k and β_k are normalization constants.

The authority and hubness vectors can be proved to converge. By solving the above two equations, we have the following equations after k iterations:

$$\vec{a}_k = \theta_k (\mathbf{A}^T \mathbf{A})^{k-1} \mathbf{A}^T \mathbf{1} \tag{6}$$

$$\vec{h}_k = \psi_k (\mathbf{A}\mathbf{A}^T)^k \mathbf{1} \tag{7}$$

where θ_k and ψ_k are normalization constants. Since $(\mathbf{A}^T\mathbf{A})$ and $(\mathbf{A}\mathbf{A}^T)$ are symmetric, we can obtain for each of the matrices a set of eigenvalues with full eigenspaces. According to theories in linear algebra, $\vec{h}$ would converge to the principle eigenvector (corresponding to the largest eigenvalue) of the matrix $(\mathbf{A}\mathbf{A}^T)$, and a similar case applies to $\vec{a}$. It is found that in practise the two vectors converge quite rapidly.

4.2. The SPEAR Algorithm

We now describe our proposed algorithm for ranking users in a collaborative tagging system by taking into the two assumptions of experts mentioned in Section 3.

Our first assumption of experts involves the level of expertise of the users and the quality of the documents mutually reinforcing each other. We define $\vec{E}$ as a vector of *expertise scores* of users: $\vec{E} = (e_1, e_2, ..., e_M)$ where $M = |U_t|$ is the number of unique users in R_t. In addition, we define $\vec{Q}$ as a vector of *quality scores* of documents: $\vec{Q} = (q_1, q_2, ..., q_N)$ where $N = |D_t|$ is the number of unique documents in R_t. $\vec{E}$ and $\vec{Q}$ are initialized by setting every element to 1. Basically, the exact value of the elements can be arbitrary as long as they are all equal, as the vectors will be normalized in later operations.

Mutual reinforcement refers to the idea that the expertise score of a user depends on the quality scores of the documents to which he tags with t, and the quality score of a document depends on the expertise score of the users who assign tag t to it. We prepare an adjacency matrix $\mathbf{A}$ of size $M \times N$ where $\mathbf{A}_{i,j} := 1$ if user i has assigned t to document j, and $\mathbf{A}_{i,j} := 0$ otherwise. Based on

this matrix, the calculation of expertise and quality scores is an iterative process similar to that of the HITS algorithm:

$$\vec{E}_k = \alpha_k \mathbf{A}^T \vec{Q}_{k-1} \quad (8)$$

$$\vec{Q}_k = \beta_k \mathbf{A} \vec{E}_{k-1} \quad (9)$$

To implement the idea of discoverers and followers, we prepare the adjacency matrix $\mathbf{A}$ in a way different from the above method of assigning either 0 or 1 to its cells. Before the iterative process we use the following equation to populate the adjacency matrix $\mathbf{A}$:

$$\mathbf{A}_{i,j} = |\{u|(u,t,d_j,c),(u_i,t,d_j,c_i) \in R_t \wedge c_i < c\}| + 1 \quad (10)$$

According to equation 10, the cell $\mathbf{A}_{i,j}$ is equal to 1 plus the number of users who have assigned tag t to document d_j after user u_i. Hence, if u_i is the first to assign t to d_j, $\mathbf{A}_{i,j}$ will be equal to the total number of users who have assigned t to d_j. If u_i is the most recent user to assign t to d_j, $\mathbf{A}_{i,j}$ will be equal to 1. The effect of such an initialization of matrix A is that we have a sorted timeline of any users who tagged a given document d_j.

The last step is to assign proper credit scores to users by applying a *credit scoring function* C to A:

$$\mathbf{A}_{i,j} = C(\mathbf{A}_{i,j}) \quad (11)$$

A first idea would be a linear credit score assignment such as $C(x) := x$. In this way, when the expertise scores are calculated by the iterative algorithm, users who tagged a document earlier will claim more of its quality score than those who tagged the document at a later time. One concern of such a linear credit score assignment is that the discoverers of a popular document will receive a comparatively higher expertise score even though they might have not contributed any other documents thereafter.

We believe that one criterion of a proper credit scoring function C is that it should be an increasing function with a decreasing first derivative: $C'(x) > 0$ and $C''(x) \leqslant 0$. In other words, the function should retain the ordering of the scores in A so that discoverers still score higher than followers but it should reduce the differences between scores which are too high. This is because it is undesirable to give high expertise scores to users who happened to be the first few to tag a very popular document but have not contributed any other high quality documents thereafter. For the context of this paper, we conduct our experiments with $C(x) := x^{0.5} = \sqrt{x}$. Overall, the above procedures of generating an adjacency matrix for the operation of SPEAR from the tagging data given a certain credit score function can be represented by the following function:

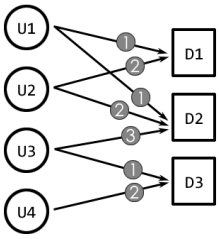
$$\mathbf{A} = GenerateAdjacencyMatrix(R_t, C) \quad (12)$$

The final SPEAR algorithm is shown in pseudocode in Algorithm 1, while Table 1 presents an example of running SPEAR on a simple case.

The SPEAR algorithm is different from the HITS algorithm in two aspects. Firstly, the adjacency matrix is not a square matrix. This is because, instead of considering a single set of documents, we now consider a set of users and a set of documents, and the number of users does not necessarily equal to the number of documents under consideration. Secondly, instead of having only 1 or 0 for the cells in the adjacency matrix $\mathbf{A}$, we initialize the matrix with different values depending on when the documents were tagged by the users. However, SPEAR can be proved to converge in the same way as HITS. This is because the proof involves the eigenvectors of the matrices $(\mathbf{A}^T\mathbf{A})$ and $(\mathbf{A}\mathbf{A}^T)$, instead of $\mathbf{A}$ (Farahat et al., 2006). Also, the proof is independent of the values in the cells of $\mathbf{A}$, as long as $\mathbf{A}$ is non-negative, which is also true in the case of SPEAR. Hence, SPEAR is guaranteed to converge under the same conditions as HITS.[4]

[4]In our experiments, it takes on average 160 iterations for the values in the vectors to stablize.

(a)



(b)

	D1	D2	D3
U1	1.4	1.7	0.0
U2	1.0	1.4	0.0
U3	0.0	1.0	1.4
U4	0.0	0.0	1.0

(c)

	Rank	Score
U1	1	0.422
U2	2	0.328
U3	3	0.212
U4	4	0.038

TABLE 1. A simple example of using SPEAR to rank users in a folksonomy. (a) shows the bipartite graph of four users and three documents. An arrow from a user to a document represents the fact that the user has assigned the tag concerned to the document. The numbers in circles represent the order of assigning the tag to the document. (b) shows the adjacency matrix after the credit score function is applied. Finally, (c) shows the final ranking of the users. In this example, `U1` is the discoverer of two popular documents (`D1` and `D2`), therefore `U1` is ranked first. `U4` is a mere follower of a single document (`D3`), and so `U4` is ranked last.

Algorithm 1 SPEAR: SPamming-resistant Expertise Analysis and Ranking

Input: Number of Users M
Input: Number of Documents N
Input: A set of taggings $R_t = \{(u, t, d, c)\}$
Input: Credit scoring function C
Input: Number of iterations k
Output: A ranked list L of users.

Set $\vec{E}$ to be the vector $(1, 1, ..., 1) \in \mathbb{Q}^M$
Set $\vec{Q}$ to be the vector $(1, 1, ..., 1) \in \mathbb{Q}^N$
$A \leftarrow GenerateAdjacencyMatrix(R_t, C)$
for $i = 1$ to k **do**
  $\vec{E} \leftarrow \vec{Q} \times \mathbf{A}^T$
  $\vec{Q} \leftarrow \vec{E} \times \mathbf{A}$
  Normalize $\vec{E}$
  Normalize $\vec{Q}$
end for
$L \leftarrow$ Sort users by their expertise score in $\vec{E}$
return L

5. EXPERIMENTS AND EVALUATION

5.1. Data Sets and Methodology

Evaluating the performance of SPEAR proves difficult due to the lack of a proper ground truth to compare experimental results. To mitigate this problem, we combine both real-world and simulated data to evaluate and compare the behavior and performance of SPEAR with alternative algorithms. Real-world data is used as the base input for our experiments. Here, it is important to realize that "real users" means "user accounts derived from real-world data", which may include real human users as well as real automated spam bots and other phenomena found in the wild. We then insert controlled, simulated data into the original real-world data by taking into account recent studies of collaborative tagging systems (Koutrika et al., 2007; Wetzker et al., 2008) and the characteristics of our real-world data sets. We can thus mitigate the lack of a proper ground truth by embedding controlled data into a real-world scenario, and analyze how the expected results compare to the experimental outcomes. The approach of our experimental setup follows the methodology of Caverlee et al. (2008) and Koutrika et al. (2007). In the context of combating spam in folksonomies, the latter work describes a method for ranking documents based on the tagging users' reliability and

Table 2. Statistics of real-world data sets retrieved from Delicious.com in February 2009.

`Total targeted tags`	110
`Total bookmarks`	15,987,386
`Total tag assignments`	52,435,158
`Total bookmarks with selected tags`	4,558,891
`Unique URLs`	132,165
`Unique users`	1,198,863
`Unique tags`	809,167

evaluate their proposed framework with a simulated tagging model. Similarly, the study of Heymann et al. (2007) discusses various spam models for social Web sites, in the context of which our approach is a hybrid of the so-called *trace-driven spam models* and *synthetic spam model*.

With regard to real-world data, we developed a crawler application which retrieved the most recent URLs posted on Delicious.com for 110 different tags, and then downloaded the bookmarking history of those URLs. The 110 tags are randomly selected from a pool of tags, which consists of all the 200 popular tags reported by Delicious[5] as well as over 200 other tags collected by monitoring the front page of Delicious. To obtain the data required for running the SPEAR algorithm, we had to crawl the Delicious.com Website directly because the official API did neither provide the volume of data nor all the required information. For each tag, we retrieved the most recent URLs that have been assigned the tag, with a maximum of 2,000 URLs per tag. This limit was the result of technical restrictions imposed by Delicious, which shows only up to 2,000 recent URLs per tag. After retrieving the list of URLs, we went on to collect up to 2,000 recent user bookmarks for each URL. While this procedure limited the data we can collect, we found that only a very small portion ($\sim$1%) of URLs had more than 2,000 bookmarks. Hence, for 99% of our URLs we had their full tagging histories. As we will describe later, our simulation mainly requires the timeline of an URL for generating users of different characteristics, so these data sets provided us with a good base for our simulation. In addition, the 110 tags we collected spanned a wide range of domains (e.g. `algorithm`, `economics`, `film`, `iphone`, `history`, `opera`). Hence, they also allowed us to test whether SPEAR behaves consistently across different topics in our experiments.

After the data collection process, we retrieved in total the tagging histories of 132,165 URLs, involving over 1 million users posting 15 million user bookmarks. A bookmark in our data set includes the Delicious username of the bookmarking user, the title and description given to the bookmark, any associated tags, and the creation timestamp of the bookmark. Among the bookmarks we collected, 4.5 millions of them involved one or more tags from our 110 selected tags. An overview of the real-world data sets is shown in Table 2.

With regard to simulated data, the basic idea was to insert simulated data properly into real-world data. For example, to simulate a discoverer-type user, we would have to insert a virtual bookmark in the early timeline of a document's "real" bookmarking history. All users with a later bookmark would automatically become followers of the simulated user for this document. Similarly, we would have to insert virtual bookmarks to popular documents in order to simulate experts because these users tend to tag only relevant information.

We wanted to create two different types of user profiles, expert-like and spammer-like users, in order to study the behavior of SPEAR. For each type of these users, we also wanted to model three variants in order to better match real-world scenarios and to improve the evaluation setup. An overview is shown in Table 3.

5.1.1. *Simulated Experts.* Simulated expert profiles are subdivided into geeks, veterans, and newcomers. A *veteran* is a user who bookmarks significantly more documents than the average user, following the reports of user behavior on Delicious described by Heymann et al. (2008); Noll and

[5]Delicious Popular Tags: http://delicious.com/tag/

TABLE 3. The simulated user profiles created for the evaluation of SPEAR.

User Type	**Variants**
Experts	Geek, Veteran, Newcomer
Spammers	Flooder, Promoter, Trojan

Meinel (2007). He tends to be among the first users to tag documents which usually become quite popular within the community. Hence, he is a discoverer with many followers. In the real-world, a veteran could be compared to an experienced researcher who has profound knowledge of his area of expertise, and advances the field by publications of his own.

A *newcomer* is an upcoming expert who is only sometimes among the first to "discover" a document. Most of the time, the documents are already quite well-known within the community at the time he tags them. In the real-world, a newcomer could be compared to a PhD student who already has knowledge about the state of the art in his area of expertise, but has yet to gain his reputation within the scientific community. He has just started with his own original research, so the number of publications is still low.

A *geek* is similar to a veteran but has significantly more bookmarks than a veteran. In the real-world, he could be a very distinguished researcher with the best knowledge of his area of expertise and a significant number of own publications. We can consider the geek profile as the "best" expert within our simulation.

In the experiments, geeks should generally be ranked higher than veterans, and the latter should in turn rank higher than newcomers. We must note though that the differences between geeks and veterans are more subtle compared to newcomers. Since we introduce the notion of document quality instead of document *quantity*, we expect veterans to compete with geeks for the top ranks even though the latter have better "odds" of success in the long run.

5.1.2. *Simulated Spammers.* Simulated spammer profiles are subdivided into flooders, promoters, and trojans. A *flooder* tags a huge number of documents which already exist in the system, most likely in an automated way. This spammer variant can often be found in the wild (Wetzker et al., 2008; Koutrika et al., 2007). He tends to be one of the last users in the bookmarking timeline[6]. Additionally, he tends to tag documents already known to the community rather than tagging new documents because he aims at gaining "reputation" through lots of bookmarks of existing, popular content.

A *promoter* is a spammer who focuses on tagging his own documents to promote their popularity, and does not care much for other documents. He tends to be the first to bookmark documents which attract few followers if any. This spammer type is quite common and we could find several on Delicious during our experiments. There were cooperating groups of them who had sequentially named user accounts of the form *iSpamYou001*, *iSpamYou002*, etc. who were possibly trying to perform a Sybil-type attack as discussed in Section 3.2. Such promoter-type spammers have recently been reported by Wetzker et al. (2008) and Krause et al. (2008). Wetzker et al. (2008) found that 19 of the top 20 most active Delicious users in their experimental data set were spammers who bookmarked ten thousands of URLs pointing to only few Web domains. In total, these 19 spammers alone accounted for 1.3 million bookmarks or around 1% of their data corpus. Likewise, Krause et al. (2008) observed spammers registering several accounts and publishing the same bookmark several times in a coordinated "attack". Similar to our anecdotal findings, Krause et al. (2008) also observed that the number of digits in a username is an indication of "spamminess", i.e. the more digits, the more likely the user is a spammer.

[6]This spammer behavior is not only caused by specific spamming strategies which try to boost expertise/reputation scores by spamming popular documents. In practice, such behavior can also be the result of the spam bot being created by its masters long after the Delicious service went online in 2003, so regular users have had a head start. Back in 2003, the eventual success of Delicious was not foreseeable, meaning that spamming it right away was not worth the risk and effort.

A *trojan* is a more sophisticated spammer in that his strategy is to mimic regular users in the majority of his tagging activities, thus sharing some traits with a so-called slow-poisoning attack. He disguises his malicious intents by tagging already popular pages, but at some point he adds links to his own documents which can be malware-infected or phishing Web pages. In other words, this spammer follows the "majority" opinion in the folksonomy most of the time to avoid detection. He tries to trick users into believing he is a knowledgeable, benevolent member of the community and then lures them into a trap – like a wolf in sheep's clothing. A recent study by Moore and Clayton (2008) discusses trojan-like spammers in the context of collaborative systems for reporting phishing Web sites.

As flooders and promoters can already be observed in existing collaborative tagging systems, an algorithm for telling experts from spammers should therefore be able to handle such spammer types. Trojan-type spammers could be seen as the next step in the evolution of malicious spamming techniques. For this reason we were interested in finding out how well SPEAR performs on these sneaky and potentially more harmful spammers.

It should be noted that our simulations were probabilistic so that even identical user profiles would produce variations in simulated data. On one hand, this means that even two users with the same profile would behave differently up to a certain extent (there can be some geeks who are "better" geeks than the others). On the other hand, we can expect overlaps in user behavior and experimental results between different user variants (a "good" newcomer might receive a higher expertise score than a "bad" veteran).

5.1.3. *Simulation Parameters.* We manipulate the following four parameters for modeling simulated users and their tagging behavior, and thus for generating simulated data in general.

- **P1**: *Number of a user's bookmarks.* For example, geeks and flooders would have a greater number of bookmarks than veterans or promoters, respectively.
- **P2**: *Newness* – Percentage of bookmarks to such documents which are not in the original real-world data. To make our experiments more realistic, we needed a feature which allows simulated users to bookmark new documents, i.e. documents that haven't been bookmarked by any real-world user yet. For example, trojans and promoters create links to their own Web documents. The actual URLs of such "new" documents are irrelevant in our experiments as long as they are unique.
- **P3**: *Document rank preferences* – A probability mass function (PMF) which specifies whether rather popular or rather unpopular documents tend to be selected when inserting simulated bookmarks. For example, the PMFs of veterans and trojans tend to select popular documents whereas the PMFs of flooders are more evenly distributed.
- **P4**: *Time preferences* – A probability mass function (PMF) which specifies where in the original timeline a simulated bookmark tends to be inserted into a given document's bookmarking history. For example, the PMFs of veterans tend to focus on the early stages of the bookmarking history, newcomers are rather evenly distributed, and flooders tend to be very late.

The actual configurations of the simulation parameters for each user type are shown in Table 4 (see also Figure 1 and 2 for the probability mass functions for **P3** and **P4**). Note that the number of bookmarks for promoters and trojans is set to absolute values (from 10 to 100), unlike that for flooders. Our reason for this decision is that promoters and trojans should exhibit behavior similar to that of real users (flooders are more likely to be bots that generate bookmarks automatically). The mean maximum number of bookmarks of real users in our data set is $\mu_{max} = 69$, therefore our chosen values cover a similar range.

5.2. General Behavior

We studied the performance of SPEAR by comparing its results with those returned by the HITS algorithm and a simple frequency count ranking algorithm, denoted *FREQ*, based on the number of user bookmarks. The latter is very popular on collaborative tagging systems in practice, and thus FREQ serves as the "baseline" of our experiments.

We first report the general behavior of SPEAR by an analysis of the resulting expertise scores. Figure 3 shows the normalized expertise score distributions of SPEAR, HITS and FREQ for

TABLE 4. Configuration of parameters P1-P4 for simulated user profiles. n_d is the total number of bookmarked documents in the relevant data set. $EQUAL()$ means that each document rank or time is selected with equal probability. The sequence of numbers in curly brackets denote multiple experiment runs with varying parameters as indicated.

Type	**P1**	**P2**	**P3**	**P4**
Geek	$2 * P1_{Veteran}$	0.10	*See figure 1*	*See figure 2*
Veteran	$\{0.01, 0.02, ..., 0.05\} * n_d$	0.10	*See figure 1*	*See figure 2*
Newcomer	$P1_{Veteran}$	0.10	*See figure 1*	EQUAL()
Flooder	$\{0.02, 0.04, ..., 0.20\} * n_d$	0.05	EQUAL()	*See figure 2*
Promoter	$\{10, 20, ..., 100\}$	0.95	EQUAL()	*See figure 2*
Trojan	$\{10, 20, ..., 100\}$	0.10	*See figure 1*	*See figure 2*

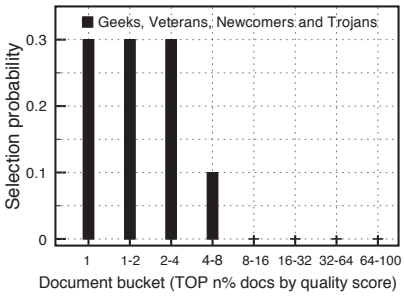


FIGURE 1. PMF for document rank preferences (P3) for geeks, veterans, newcomers and trojans. In contrast to these simulated users, flooders and promoters chose document ranks randomly. Lower bucket numbers refer to higher quality documents. We chose exponentially increasing bucket sizes here to account for power law/long tail effects in collaborative tagging systems such as Delicious (Noll and Meinel, 2007).

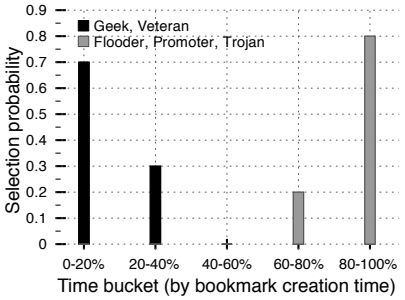
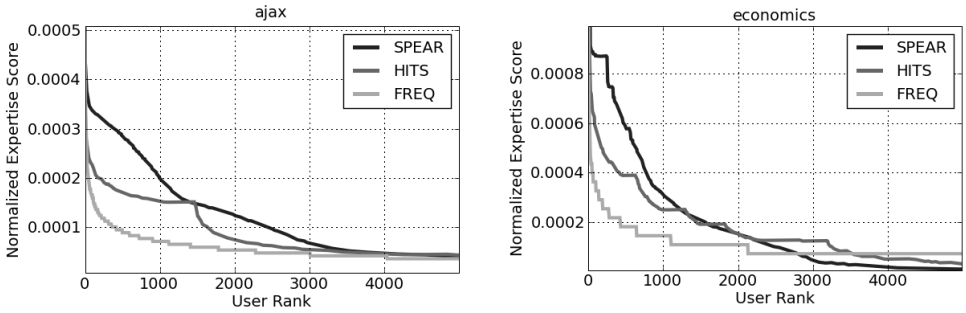


FIGURE 2. PMF for time preferences (P4) for geeks, veterans (black) and flooders, promoters, trojans (gray). Lower bucket numbers refer to earlier timestamps, e.g. the first bucket represents the first 20% of bookmarks in a URL's history. In contrast, newcomers chose timestamps randomly.

two exemplary data sets, namely `ajax` and `economics`. We observed that SPEAR generally produced more differentiated values than HITS and FREQ for top users, i.e. the difference in expertise scores between two ranks for SPEAR was generally larger than for HITS and FREQ, where the curves were flatter. We will see how SPEAR benefits from this characteristic in Section 5.4.

Another finding was the staircase-like shape of FREQ caused by the integer frequency counts on which it is based. This means FREQ tends to group users into buckets of equal expertise score instead of assigning an individual rank to each user. Both SPEAR and HITS also showed occasional



(a) (b)

FIGURE 3. Normalized expertise scores of the top 5000 users as returned by SPEAR, HITS and FREQ for two exemplary data sets `ajax` and `economics`.

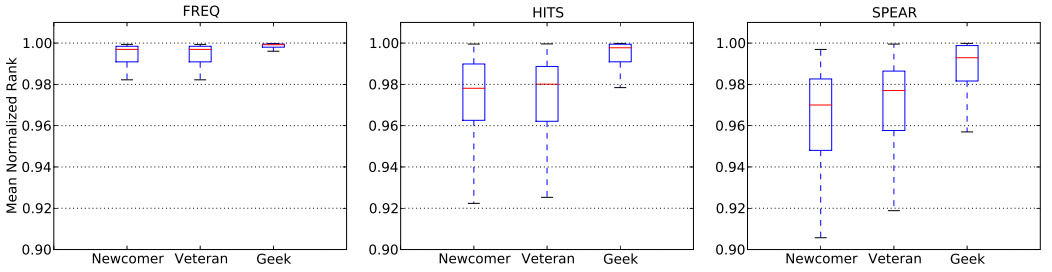
staircase steps. However, this was due to limitations in our real-world data sets as discussed in Section 5.1, as we could only retrieve the creation date of a bookmark from Delicious, not the time of day. This resulted in "time collisions", and coupled with our limited data sets—-only a snapshot view of the full data stored at Delicious—we could observe occasional plateaus of equal score values. Still, SPEAR was able to mitigate this problem better than HITS, which can be seen particularly in the score distributions for the tag `ajax`. In contrast, the plateaus of FREQ have structural reasons.

5.3. Promoting Experts

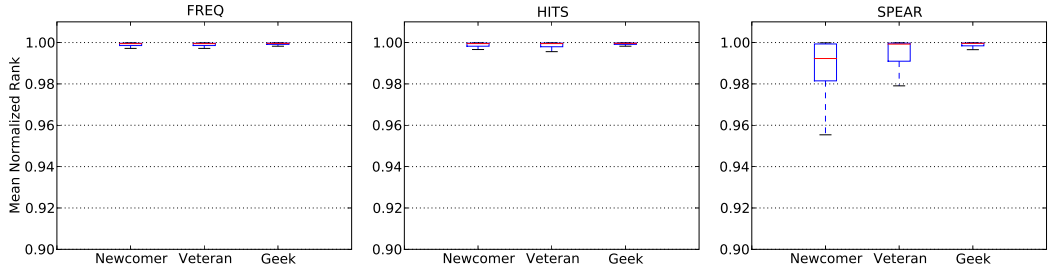
To study how different variants of experts are ranked by SPEAR, we generated, for each of the 110 real-world data sets, 20 experts of each type (60 total per data set) and added them to the corresponding data set. We then applied SPEAR, the original HITS algorithm and FREQ to these data sets comprising both real-world and simulated users. The results are shown in Figure 4. Note that some overlapping between the three expert variants are expected due to the PMF-based simulation setup as described in Section 5.1.

The plots show some major differences between SPEAR and the other two ranking algorithms. In SPEAR, geeks were generally ranked higher than veterans, which in turn were ranked higher than newcomers. We also observed that geeks and experts did compete for the top ranks even though geeks won in general. This means that some veterans, although having had fewer bookmarks than geeks in general, were ranked higher by SPEAR because they had some higher quality bookmarks. Another observation was that veterans were ranked higher than newcomers, though we expected an even stronger difference. This result suggests that better credit scoring functions than $C(x) := \sqrt{x}$ can be chosen, and we plan to study the effects of different credit scoring functions in the future. All in all however, SPEAR showed the expected and desired behavior.

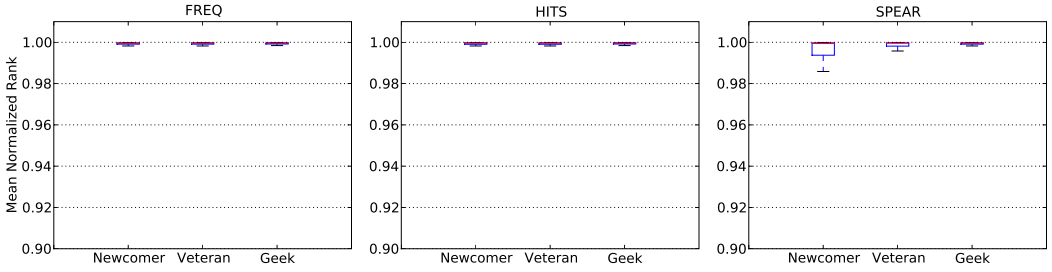
HITS and FREQ performed not as well. They did rank geeks higher than veterans and newcomers, but geeks were also the "easiest" expert variant because they had a very large number of high quality bookmarks. This means even the naive FREQ should and did perform reasonably for this user variant. However, both HITS and FREQ failed to differentiate between veterans and newcomers, which ended up being mixed together. This result suggests that only SPEAR succeeded in distinguishing veterans and newcomers by implementing the notion of discoverers and followers. In contrast, HITS still tended to return results which were heavily influenced and biased by the number of documents in a user's collection, even though it is also an implementation of a mutual reinforcement scheme. We conclude that in usage scenarios where quantity does not guarantee quality—and we believe collaborative tagging is one such scenario—SPEAR is expected to provide better ranking of experts. A more detailed example of rankings given by the three algorithms is given in Figure 8.



(a)



(b)



(c)

FIGURE 4. Boxplots of mean normalized ranks of simulated experts—**newcomers, veterans, geeks**—in direct comparison across all data sets for the three algorithms. Rank values of 1.0 and 0.0 represent the top-ranked user (highest expertise) and the bottom-ranked user (lowest expertise), respectively. The plots (a), (b) and (c) show the results for $P1_{Veteran} = 0.01$, $P1_{Veteran} = 0.03$ and $P1_{Veteran} = 0.05$, respectively. Some overlapping of simulated experts is expected due to the experimental setup as described in the text.

5.4. Demoting Spammers

Similarly, we generated and added 20 flooders, promoters and trojans, respectively, for each of the real-world data sets. The results are shown in Figures 5, 6, 7.

FREQ showed the weakest performance among the three algorithms. It was very vulnerable to all spammer types and gave them top ranks. This was true particularly for flooder-type spammers, which unfortunately are often found in today's collaborative tagging systems (Wetzker et al., 2008). This observation suggests that the (in practice) popular frequency count algorithm is not capable of mitigating the spammer problem.

HITS performed better than FREQ but was dominated in all experiments by SPEAR. While it was good at demoting promoters, HITS had problems to demote flooders with increasing numbers of spam bookmarks (see Figure 5), and was weak in general for handling trojans.

SPEAR showed the best performance among the three algorithms. Firstly, it correctly demoted both flooders and promoters by assigning them significantly lower ranks than HITS and FREQ.

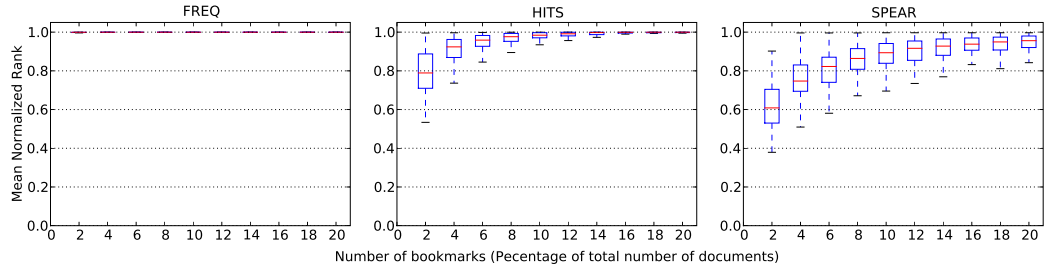


Figure 5. Boxplots of mean normalized ranks of simulated **flooders** across all data sets for the three algorithms in relation to the number of bookmarks generated per flooder (x-axis). Rank values of 1.0 and 0.0 represent the top-ranked user (highest expertise) and the bottom-ranked user (lowest expertise), respectively. Lower values are better.

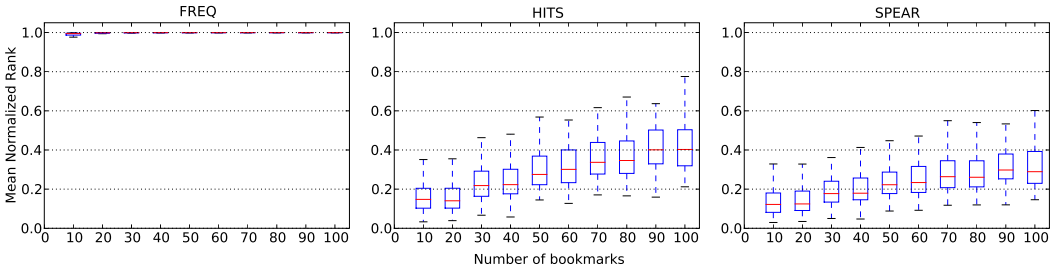


Figure 6. Boxplots of mean normalized ranks of simulated **promoters** across all data sets for the three algorithms in relation to the number of bookmarks generated per promoter (x-axis). Rank values of 1.0 and 0.0 represent the top-ranked user (highest expertise) and the bottom-ranked user (lowest expertise), respectively. Lower values are better.

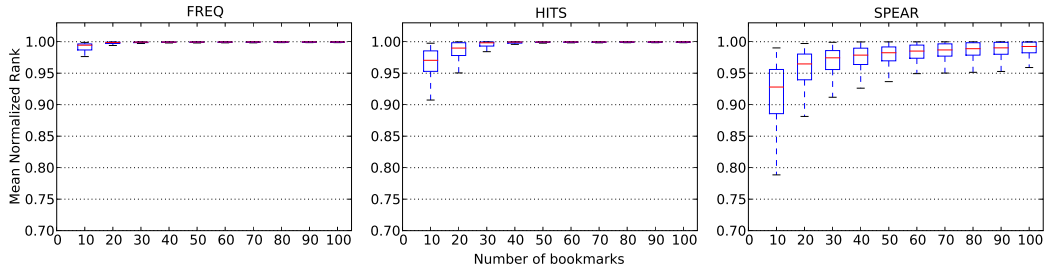
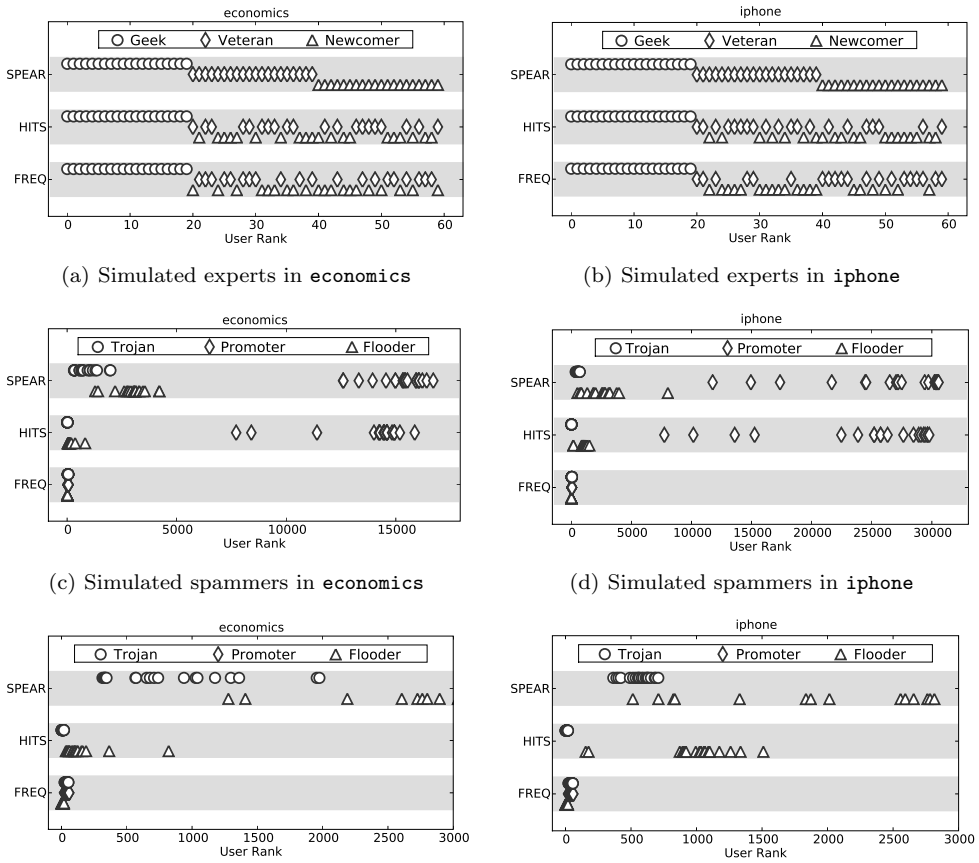


Figure 7. Boxplots of mean normalized ranks of simulated **trojans** across all data sets for the three algorithms in relation to the number of bookmarks generated per trojan (x-axis). Rank values of 1.0 and 0.0 represent the top-ranked user (highest expertise) and the bottom-ranked user (lowest expertise), respectively. Lower values are better.

Secondly, SPEAR was also able to demote trojans who use a much more sophisticated spamming scheme. While trojans were still ranked higher than the other two spammer variants, trojans were rarely ranked higher than rank #100 by SPEAR across the experiments. Given that in practice the TOP 10 to the TOP 50 experts should be the ones we are most interested in, SPEAR in its current form already performed reasonably well in getting rid of all trojans in the relevant rank range (see Figure 8). That being said, the problem with trojans is that it is tricky to demote them without demoting good users at the same time, because from a pragmatic point of view a trojan is still a rather good hub of resources. Users accessing documents in a trojan's collection may need to verify the quality score of the documents, which is also computed by SPEAR, to judge whether



(a) Simulated experts in `economics`

(b) Simulated experts in `iphone`

(c) Simulated spammers in `economics`

(d) Simulated spammers in `iphone`

(e) Simulated spammers in `economics` (zoomed)

(f) Simulated spammers in `iphone` (zoomed)

FIGURE 8. Ranks of simulated experts and spammers for two selected tags `economics` and `iphone`. In (a) and (b), SPEAR clearly distinguishes between the three types of expert users, while HITS and FREQ tend to mix up veterans and newcomers. (c) and (d) shows that SPEAR is better in demoting spammers than HITS, while FREQ always assigns high ranks to spammers. (e) and (f) focus on the top 2500 users in (c) and (d) respectively. It can be seen that SPEAR is able to demote trojans such that they are not ranked among the top 200.

they are really legitimate and useful resources before actually visiting them. Hence, we look forward to analyzing such spammers more thoroughly in the future and to studying how complementary techniques could help to demote or identify them.

Thirdly, SPEAR was the only algorithm that did not tend to "clump" spammers together in one spot in our experiments, i.e. it was better at differentiating and detecting nuances in spammer behavior compared to HITS and FREQ. We think this is a direct result of the different expertise score curves as described in Section 5.2.

5.5. Combined Evaluation: Experts plus Spammers

In the above experiments, we injected each type of simulated users separately into the real world data sets. As an overall evaluation, we now describe our final experiment which involved injecting all types of simulated users into the real world data sets to compare the performance of different algorithms.

Similar to the experiments described above, we first generated the six different types of simulated users using different parameters, and injected their profiles into our real world data sets. We then used the three algorithms, namely FREQ, HITS and SPEAR, to rank the users. Due to the large number

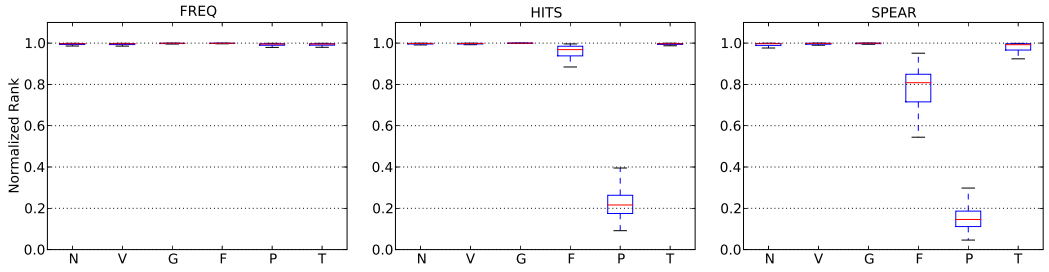


Figure 9. Boxplot of mean normalized ranks of different types of simulated users when they are injected together into our real data sets. In the figures, N=Newcomers, V=Veterans, G=Geeks, F=Flooders, P=Promoters, and T=Trojans.

Table 5. Summary of the result of overall evaluation with different types of simulated users being ranked at the same time. For spammers, the best (lowest) result is shown in bold font. As can be seen, only SPEAR was able to rank all three expert types at the top and retain the expected correct order. G=Geeks, V=Veterans, N=Newcomers, F=Flooders, P=Promoters, and T=Trojans.

	G	V	N	F	P	T	Order
FREQ	0.9873	0.9731	0.9747	0.9888	0.9797	0.9827	$F > G > T > P > N > V$
HITS	0.9943	0.9838	0.9842	0.9322	0.2286	0.9874	$G > T > N > V > F > P$
SPEAR	0.9914	0.9821	0.9774	**0.7687**	**0.1656**	**0.9707**	$G > V > N > T > F > P$

of possible combinations of parameters, we only report typical results in details, and we will discuss briefly situations when some extreme combination of parameters were used. Figure 9 shows a typical result of this experiment, with $P1_{Veteran} = P1_{Flooder} = 0.03$, and $P1_{Promoter} = P1_{Trojan} = 100$. With these parameters, the spammers always had their numbers of bookmarks great than those of the newcomers and veterans, but comparable to those of the geeks. Table 5 shows the mean normalized rank of each of the different types of users given by different algorithms.

From Figure 9 and Table 5, we can see that a combined simulation produce similar results as the separated simulations described above. FREQ ranked all spammers at the top due to their large collection of bookmarks. HITS was able to demote the flooders and promoters to a certain extend, but still ranked the trojans among the top users. SPEAR performed well by demoting the flooders and promoters more significantly than FREQ and HITS, and was able to remove the trojans from the top of the list.

Of course, given the same set of real world data sets, the ranks of spammers in SPEAR would still increase if they started to create more and more bookmarks, as we have shown in Section 5.4. However, it would, depending on the number of bookmarks and their distribution in the real world data sets, require a much larger number of bookmarks for spammers of any types to make SPEAR "fail", i.e. to make SPEAR rank these spammers as top users. In such case, it would then be rather easy to detect these spammers by observing abnormal behavior and activity patterns within the system. In other words, while it is possible that spammers could be ranked higher than all legitimate users in the system by SPEAR, these spammers by that time would be easily detected and removed. We will discuss further about the advantages and limitations of SPEAR in Section 5.8.

5.6. Qualitative Analysis

In addition to the quantitative analysis of the simulation results, it is worthwhile to take a look at the ranking of real users produced by SPEAR in a qualitative way so as to gain more insight into its effectiveness.

We run SPEAR on the data sets of four arbitrarily selected tags, namely `photography`, `semanticweb`, `javascript` and `programming`, where the last two are combined to form a conjunction as an example of running SPEAR on a more specific topic. We examined the top users who are given high ranks

by SPEAR in each of these data sets. While it would be difficult to provide an objective evaluation of the expertise of these users, we discovered that there were several things that were indicative of their expertise.

Firstly, many of these top users were more likely to provide optional personal information in their Delicious account, including for example their real names, address of personal Websites, links to their photos on Flickr, and links to their Twitter microblogging account. This implied that they were more involved in using Delicious. Secondly, many of them have a lot of other tags used together with the corresponding tag in which they attain high expertise scores. For example, a top user in `photography` has used 359 other tags together with `photography`, suggesting that he has an extensive collection of documents about the topic. Finally, we identified some "real" experts among the top users. For example, two users that were ranked in the top 10 in `semanticweb` turned out to be two researchers of Semantic Web technologies, while a third was found to be an active blogger of the same subject. The top two experts ranked by SPEAR in `javascript`∩`programming` were two professional software developers. In contrast, all the users mentioned above were ranked lower by FREQ and HITS, sometimes even outside the top 200.

As for spammers, we singled out the obviously heavily spammed tag in Delicious, `mortgage`, collected the bookmarking histories of the documents that were annotated with the tag[7], and run SPEAR, HITS and FREQ on it to rank the users. We wanted to find out whether spammers were really demoted by SPEAR and whether FREQ was vulnerable to spammers in this real setting. While we did not have a labeled list of the spammers as ground truth, we identified them manually by looking for several characteristics common to spammers. Spammers are usually automated bots. Hence, they either tend to extract words from the documents themselves (especially the title) and use them as tags, or use the same set of tags on a large number of documents even though the tags are not semantically related to the document content (Markines et al., 2009). Also, some spammers aim at promoting their own content, and therefore many of their bookmarks are likely to be documents from the same domain (which can usually be classified as spam at first glance).

By looking for these characteristics of users who used the tag `mortgage`, we successfully identified 30 spammers in the 50 most active users. Obviously, this meant that out of the top 50 users ranked by FREQ, 30 of them were found to be spammers. It is interesting that we even discovered a group of spammers whose usernames had the same prefix and were only different from each other in the numbers in the suffixes, suggesting that there exist spammers who submit spams in a more sophisticated way than merely flooding the system. As for the rankings produced by SPEAR and HITS, we observed similar results as we did in our simulations. All these 30 spammers were significantly demoted to below the 3000th rank by SPEAR and HITS, with ranks of these spammers in SPEAR much lower than those in HITS. We also observed that there were no spammers in the top 50 ranks returned by SPEAR and HITS.

In addition, we also run FREQ and SPEAR on arbitrarily selected tags and examined the differences between the top rank users. We found that very often users ranked at the top by FREQ were quite the opposite of experts, not to mention that many of them were spammers. For example, for the tag `bridge`, a user was ranked first by FREQ because he had a large number of bookmarks with the tag. However, a closer look at his collection of documents in Delicious revealed that the majority of them were not related to any conventional meanings of the word 'bridge'. In contrasts, SPEAR ranked this user much lower, at 2,088th out of the 3,144 users being ranked. The fact that this user was ranked low by SPEAR was that, despite the number of times he had used this tag, there were very few, if any, other users who would do the same thing as he did. In other words, although he was not necessarily a spammer, this user had few followers due to his idiosyncratic use of the tag. Arguably SPEAR gave a more sensible result because other users were quite unlikely to benefit from this user with respect to the topic in question.

By this qualitative study, we showed that SPEAR also works reasonably well in a real setting. On the one hand, it is able to identify real experts. On the other hand, it is able to solve problems in day-to-day operation of collaborative tagging systems by demoting real spammers.

[7]The data set of the tag `mortgage` was not among the 110 data sets we had collected at the beginning.

5.7. Analysis of Credit Score Functions

One important element of SPEAR is the credit score function $C(x)$ by which we assign higher scores to users who have tagged a document earlier and lower scores to users who have tagged the document at a later time. This credit score function actually directly affects the performance of SPEAR. If we do not apply the credit score function, SPEAR will be no different from the original HITS algorithm, in which every cell in the adjacency matrix will either be 1 or 0.

Intuitively, with a credit function of larger second derivative—credit scores for a user increases faster and faster when he has more and more followers, SPEAR should be more resistant to spammers. This is because the number of followers of a user is an important piece of information that allows us to distinguish between spammers from legitimate users. However, there is also a drawback when such an aggressive credit score function is used.

To give higher scores to users who have tagged a document at an earlier time will increase the chance of mistaking an inactive user as an expert. Consider a very popular document with 5,000 users, a certain user may happen to be the 100th user to tag this document, and therefore he has 4,900 followers with respect to this document. As a result, he will be assigned a an initial score of $x = 4,900$. Consider two credit score functions $C_1(x) = x^{0.2}$ and $C_2(x) = x^{0.8}$: $C_1(x)$ will return 5.47, while $C_2(x)$ will return 895.69. If C_2 is used, this user will receive an exceedingly high expertise score given this high credit score coupled with the probably very high quality scores of this popular document. Other expert users who have tagged many more high quality documents will find themselves ranked lower than this user only because they are followers of him in this particular document. This will be a problem because this inactive user is very unlikely to benefit other users.

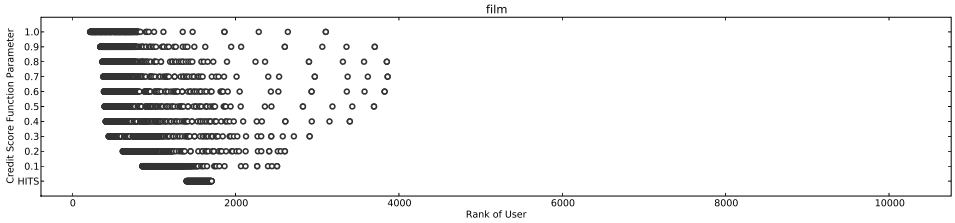
To investigate how the credit score function affects the ranks of these inactive users, we conducted experiments on some selected data sets with different credit score functions. Firstly, we randomly picked three tags from our data sets: `film`, `history` and `iphone`. For each of these data sets, we run SPEAR to obtain a ranking of the users involved by using different credit score functions of the form $C(x) = x^y$, where y ranged from 0 to 1.0 (in the case of y=0, the algorithm effectively became HITS). While it is true that there are many other types of functions that can be considered here, this class of functions should be sufficient in allowing us to have a better understanding of the behaviour of SPEAR, as it provides us with functions with different second derivatives, in which we are most interested. We then examined for each of the tags the ranks of the users who were found to have only tagged the most popular document in the respective data set.

Figure 10 shows the ranks of users who have only tagged the most popular document in each of the three data sets, with SPEAR operating under different settings of credit score function. We can see that the differences between credit score functions show similar effects on the ranking of these inactive users. Credit score functions with greater values of y tend to spread the users across a wider range. This is due to the fact that these credit score functions assign scores that spread a wider range of values. However, these functions also tend to rank some inactive users quite high, especially when they tagged the most popular document at a very early time.

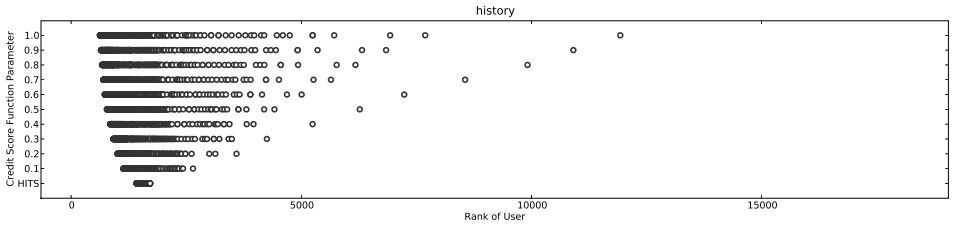
On the other hand, credit score functions with smaller values of y tend to clump users in small range of ranks. At the extreme end where $y = 0$, all of the users under consideration are assigned the same expertise score. A merit of these functions is that they tend to give lower range to these users on average. Therefore they also have a smaller chance of mistaking these users as expert users. However, as we have shown in our simulations, HITS, which is SPEAR with $y = 0$, performed relatively poorer than SPEAR where we set $y = 0.5$. In other words, smaller values of $y = 0$ would also make SPEAR more vulnerable to spammers.

Different credit score functions have different merits and weaknesses. Therefore there is no single correct choice of credit score function for SPEAR. In settings where spamming activities are commonly observed, functions with greater values of y or other functions with similar characteristics should be used. On the other hand, in settings where there are few spammers, one may consider to use functions with smaller values of y or other functions with similar characteristics.

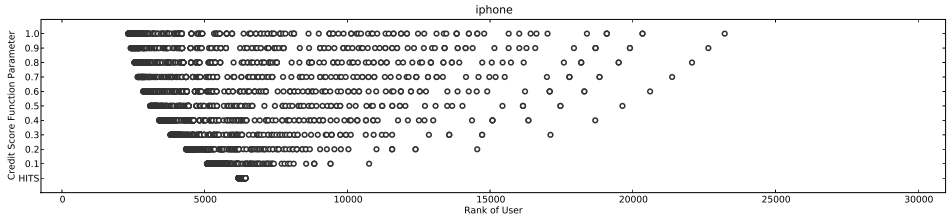
In fact, in addition to the method we propose in this paper for initializing the adjacency matrix for calculation in SPEAR, one may also consider assigning credit scores based on a certain time window. In other words, instead of assigning different users unique scores with respect to a certain document, we can assign the same scores to a group of users who have tagged this document within a certain time window. In addition, by creating, for example, windows of longer period for earlier



(a)



(b)



(c)

FIGURE 10. Ranks of users who have only tagged the most popular document for each of the three selected tags: `film`, `history` and `iphone`. Only these users are represented by the circular symbols. Other users in the data sets are not shown.

users and windows of shorter period for later users, it would be possible to mitigate the problem of mistaking inactive users as experts and at the same time retain the resistant of the algorithm to spammers. However, such extension to the algorithm is not trivial as this depends on the activity of the users. It can be foreseen that different systems such as Delicious and LibraryThing would have different user activity patterns, and would therefore require different settings. Hence, we expect to first conduct analysis of activity patterns in multiple systems in our future work, before investigating the usefulness of such extension to our algorithm.

5.8. Discussion

In summary, SPEAR produced better rankings than both the original HITS algorithm and simple frequency counting, the latter being a very popular ranking algorithm in collaborative tagging systems in practice. It distinguished reasonably well between different types of experts, and it consistently demoted different types of spammers and removed them from the top of the rankings. In other words, SPEAR was able to detect the subtle differences between good and bad users, and to demote spammers while still keeping the experts at the top of the ranking.

There are a number of reasons of why an expert ranking algorithm is needed in collaborative tagging. Firstly, with increasing number of documents for a given tag, it becomes increasingly difficult to retrieve documents which are useful and of good quality. One way to solve this problem is to first identify the experts and then *browsing* their collection which should contain good documents. On the

Table 6. TOP 5 documents returned by SPEAR for the `photography` data set.

1.	http://www.berniecode.com/writing/photography/beginners/
2.	http://www.diyphotography.net/
3.	http://strobist.blogspot.com/2006/07/how-to-diy-10-macro-photo-studio.html
4.	http://digital-photography-school.com/blog/
5.	http://www.krages.com/phoright.htm

other hand, by keeping an eye on the collection of an expert, we are able to benefit from *notification* when he adds new and useful documents to his collection.[8]

5.8.1. *Document Quality.* In fact, SPEAR also provides another piece of information which is a ranked list of documents sorted by their quality score. Although we did not pay much attention to this aspect in this paper, it can be very useful for providing a ranking of documents in the context of information retrieval and Web search in general. As an illustrating example, Table 6 shows the TOP 5 documents returned by SPEAR for the `photography` data set. Even at first glance, the list provides documents which are very relevant to photography in general, including quite a number of online tutorials on different aspects of photography. For instance, the first document is a very detailed technical tutorial of photography describing basic concepts and introducing different shooting techniques. However, as we have not analyzed in detail such document rankings in this paper, we have yet to draw any conclusions. We look forward to extend our study to this aspect of SPEAR and how it can be used together with expertise scores in the future.

In addition, although we only discuss expert ranking in the context of collaborative tagging, SPEAR is in fact applicable in many different situations because it assumes a very general model of user-document interactions. For example, it can be applied to collaborative filtering sites such as Last.fm[9] (music) and Digg[10] (news), which are very popular among Web users nowadays, to rank users by their expertise in a given topic. Studying how SPEAR can be extended to other Web applications is thus one of our future goals.

5.8.2. *Relationship to other Document Quality Measures.* Collaborative tagging systems and folksonomies are not isolated from the rest of the Web. Previous work such as that of Heymann et al. (2008) analyzed the interrelations of folksonomies derived from social bookmarking with Web search, and shown that folksonomies can serve as a data source for new Web pages which haven't been indexed by search engines yet.

In this context we were interested in finding out whether SPEAR might benefit from integrating information external to folksonomies such as a Web document's popularity as measured by its PageRank, and, similarly, whether there is already a correlation between SPEAR and PageRank in the first place. In the case of SPEAR, quality scores and thereby ranks are computed from folksonomy data, whereas the PageRank of a document is computed by an analysis of the hyperlink graph of the Web (Brin and Page, 1998).

We conducted a preliminary analysis on how the SPEAR quality score of documents is related to the Google PageRank for the same documents. Particularly, we studied how the subsets of the highest quality and lowest quality documents in our data sets, respectively, compare with the aggregation of *all* documents in terms of Pagerank information. We queried Google.com for PageRank information of all Web documents in our real-world data sets, and received PageRanks for 101,154 out of 132,165 documents (77%). We created two sets of Web documents for the PageRank analysis: *SPEAR-TOP* and *SPEAR-BOTTOM*. *SPEAR-TOP* and *SPEAR-BOTTOM* contained the top 100 quality

[8]Currently, Delicious allows users to subscribe to a particular tag or to become a fan of another user. However, there is neither a measure of a user's expertise nor a recommendation of related experts in your areas of interest given your own user profile.

[9]http://www.last.fm/

[10]http://www.digg.com/

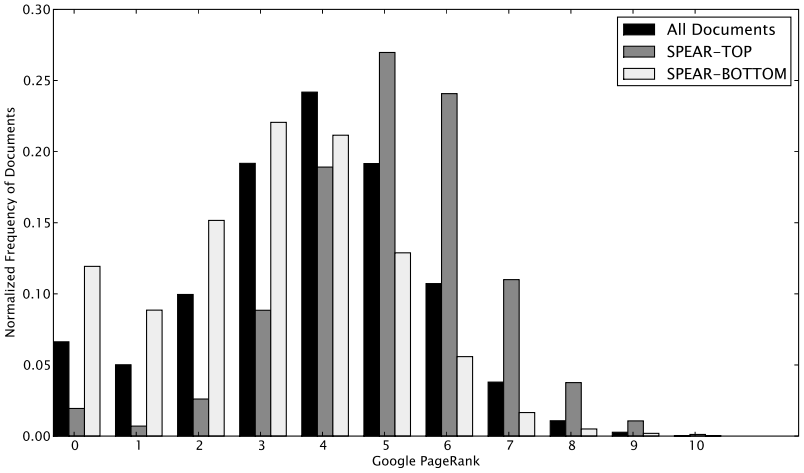


FIGURE 11. Google PageRank distribution for all documents, *SPEAR-TOP* and *SPEAR-BOTTOM*. The plot shows the shifts of high quality documents towards higher PageRanks, vice versa for low quality documents.

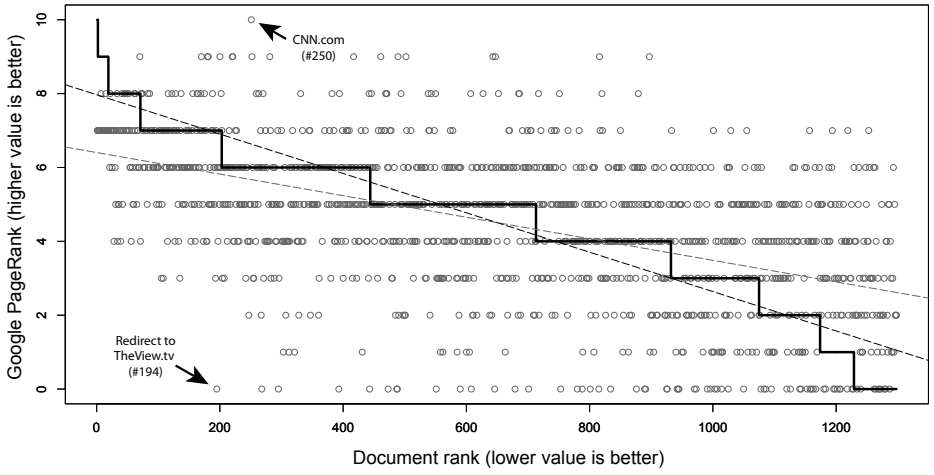


FIGURE 12. Exemplary Google PageRank distribution from *PR0* (lowest) to *PR10* (highest) for the data set `entertainment`. The solid, staircase-shaped line shows the PageRank distribution of documents when ranked by their *PR* value; the gray circles denote the PageRank distribution of documents when ranked by their SPEAR quality score. The dashed lines in black and gray show the least squares regression lines for ranking by PageRank and SPEAR, respectively.

documents and bottom 100 quality documents, respectively, from each of our 110 real-world data sets. We discarded 6 out of the 110 data sets because they were comprised of less than 200 documents. This step yielded a total of $104 \times 100 = 10,400$ documents for each of *SPEAR-TOP* and S*PEAR-BOTTOM*, respectively. We then compared the PageRanks of all documents with those in *SPEAR-TOP* and *SPEAR-BOTTOM*. The results are shown in Figure 11 and Table 7. We observed that high quality documents in SPEAR tend to have higher PageRanks (*PR*) than a random selection of documents, vice versa for low quality documents.

Table 7. Google PageRank (PR) statistics for all documents and those in *SPEAR-TOP* and *SPEAR-BOTTOM*, respectively. We observed clear shifts towards higher PageRanks for documents in *SPEAR-TOP* and towards lower PageRanks for documents in *SPEAR-BOTTOM*.

Documents	**Mean** PR	**Std. Dev.**	**Median** PR
All	3.71	1.81	4
SPEAR-TOP	5.05	1.61	5
SPEAR-BOTTOM	3.05	1.81	3

Additionally, for each real-world data set, we computed the Pearson-r correlation coefficient (Rice, 1995) of the complete (i.e. not only top and bottom) document rankings by SPEAR and PageRank. The mean Pearson-r correlation coefficient across all 110 data sets was $\bar{r}_{arithm} = +0.324$ ($\sigma = 0.146$), i.e. a weak positive correlation. The p-values were $\leqslant 0.05$ for all but eight data sets; most of the latter had less than 100 documents in total, i.e. the sample size was comparatively small. Under the assumption that SPEAR is reasonably able to measure the quality of a document within a folksonomy, this result suggests that there is a correlation between the "value" of document within a folksonomy and the hyperlink graph of the Web. It is also an indication that the algorithmic outcome of SPEAR is reasonable in general.

On the other hand, the behavior of SPEAR is still quite different from PageRank as is exemplarily shown in Figure 12 for the data set `entertainment`. In this data set for example, the PageRank #1 document with *PR10* was the well-known news site *CNN.com*. However, *CNN.com* was only ranked #250 by SPEAR, which is even lower than the highest-ranked *PR0* document for SPEAR at #194. Interestingly, the latter *PR0* document automatically redirected via an HTTP header `301 Moved Permantently` to the home page of *The View*, a popular ABC talk show, which itself has a high PageRank value of *PR8*. We could argue that this is an indication that SPEAR could identify the value of the document while PageRank failed. However, we must also consider that the *PR0* document in question did not display any content of its own but rather redirected to another Web document – which might be the reason why Google's PageRank implementation assigned it a low *PR0* value in the first place. Overall, only 2 documents from PageRank top 20 were present in the SPEAR top 20. For the record, the SPEAR #1 document was *eOnline.com*, a *PR7* website on entertainment news and celebrity gossip.

At this time, we cannot fully explain the relationship of SPEAR and PageRank yet. The dynamics and interactions of user-driven folksonomies with the link-based Web graph are still an open research question in general. Unfortunately, research in this area is hindered by the lack of adequate public data sets that can be studied, for instance data sets that provide historical PageRank information about Web documents in order to study the interaction of the Web's evolving link structure (and thus PageRank) with user behavior patterns in folksonomies over time. Nevertheless our preliminary observations suggest that SPEAR might be able to derive information about the "value" of Web documents from folksonomies (driven by content consumers) that PageRank and its analysis of the Web graph (driven by content creators) misses. Based on these first but encouraging results, we want to continue the study of SPEAR in non-folksonomy contexts in the future. Such studies might include how SPEAR could compliment or improve traditional techniques for Web search and ranking, or how SPEAR itself could benefit from information derived from data sources outside of folksonomies. For example, a user could be credited higher if he was the discoverer of a high quality but low PageRank document, i.e. of a valuable document that otherwise may not make it to the top of Web search results and thus be hidden from the views of ordinary users.

5.8.3. *Possible Improvements of SPEAR.* While we have seen that SPEAR performs well in giving us a ranking of users according to their expertise in a particular topic and is resistant to spammers, we do identify several limitations of SPEAR that deserve future investigations.

Firstly, SPEAR may mistake inactive users as expert users, especially when these users were once early discoverers of some documents that have become very popular afterwards, as we have shown in our analysis of the credit score function. A related idea is that of the "recency of information", i.e. how recent and up-to-date user-contributed information is. It is reasonable that a user who has been

more active recently should be given more credit than a user who only discovered several popular documents in the past and has ceased contributing thereafter (scenario of a "retired researcher"). Hence, it would be desirable to incorporate certain measures for reducing the weight of old user activities into SPEAR. This will make it easier for new users to rise to the top of the expert ranks and prevent older users to have an undue influence as reported by Guha (2004). On the other hand, it would also make SPEAR's user and document ranking scheme more trend-aware, for instance to the benefit of document recommenders.

Secondly, SPEAR focuses on user activity in a document's timeline. A tag-based analysis is only performed in a pre-processing stage for filtering information about documents and users by topic (where a topic is represented by a tag or a combination of tags). This leads to two limitations of SPEAR. The first limitation is that it overlooks users who have used related tags, such as synonyms, of the tag chosen for analysis. For example, when ranking users for `javascript`, should we also consider users who are ranked high in `programming`? While one can currently specify a disjunction of related tags for the filtering process required before SPEAR, it is usually difficult to know all or even the most important related tags of a particular tag beforehand. Hence, it is desirable to integrate some kind of tag co-occurrence analysis in SPEAR to produce a more comprehensive user ranking.

The other limitation of only focusing on the timelines is that SPEAR may be vulnerable to spammers who assign incorrect tags to documents. Consider a user who tagged the homepage of *Google Search* with `search`, `drugs`, `viagra` and `gifts`. If he happens to be among the users who tagged the page early in the timeline, he would be given a high rank by SPEAR with respect to the tag `search` – although we have good reasons to doubt that he is an expert in this topic given the other tags used by him. However, this type of spammers is less likely to be seen because they would have to *a)* be able to discover some documents that would eventually become popular (thereby competing with and beating other entities like search engines and regular human users to it), and *b)* be able to assign at least some correct tags to them at the same time (thereby solving a part of the open research question of tag recommendation for resources). Still, SPEAR would benefit from some analysis of the tagging vocabulary of users for increasing its robustness against this kind of spammers. As discussed in Section 2.2, there are other works that tackle spammers by focusing on analysis of tag usage of users (Koutrika et al., 2007; Markines et al., 2009; Neubauer and Obermayer, 2009). We believe these approaches and that of SPEAR are complimentary to each other and we look forward to studying how they can be combined to provide a better user ranking algorithm.

6. CONCLUSIONS AND FUTURE WORK

We proposed SPEAR for ranking experts in a collaborative tagging system and created several different variants of simulated experts and spammers and use them to study the behavior of SPEAR. Our experiments suggest that SPEAR is better at distinguishing various kinds of experts and is more resistant to different kinds of spammers than the original HITS algorithm and simple frequency analysis. We note that SPEAR measures expertise mainly based on a user's ability to discover (new) high quality content, which is but one aspect of an expert's skill set in the real world. However, a primary goal of collaborative tagging systems is to identify high-quality resources, so the expertise aspect analyzed by SPEAR is very relevant in such systems.

We believe this work opens up quite a number of research directions. Firstly, as we have mentioned in the previous section, SPEAR can be improved in several different aspects. We will therefore investigate, for example, how the notion of recency of information, reliability of the tags of the users, and the social networks established in collaborative tagging systems can be incorporated into our analysis in order to improve SPEAR.

Secondly, while we have discussed several common types of spammers in this paper, we still look forward to updating our simulation models after having conducted further research on user behavior in collaborative tagging systems, such that we can further improve SPEAR in the expertise ranking task.

Lastly, SPEAR also provides another piece of information which is a ranked list of documents sorted by their quality score. Although we did not pay much attention to this aspect in this paper, it can be very useful for providing a ranking of documents in the context of information retrieval

and Web search in general. We look forward to extend our study to this aspect of SPEAR and how it can be used together with expertise scores in the future.

Acknowledgements

This article discusses research work undertaken by Michael G. Noll in partial fulfillment of the requirements for the PhD degree of the University of Potsdam and the University of Luxembourg, following a joint thesis supervision agreement between the institutions (cotutelle de thèse).

Ching-man Au Yeung is supported by the R C Lee Centenary Scholarship, generously provided by the Drs Richard Charles and Esther Yewpick Lee Charitable Foundation, Hong Kong.

REFERENCES

M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. .

S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, New York, NY, USA, 2007. ACM.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

J. Caverlee, L. Liu, and S. Webb. Socialtrust: tamper-resilient trust establishment in online communities. In *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 104–114, New York, NY, USA, 2008. ACM.

M. T. H. Chi. Two approaches to the study of experts' characteristics. In *The Cambridge Handbook of Expertise and Expert Performance*, pages 21–30. Cambridge University Press, New York, NY, USA, 2006.

B. Dom, I. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *Proc. of 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, USA, 2003*, pages 42–48. ACM, 2003.

A. Farahat, T. LoFaro, J. C. Miller, G. Rae, and L. A. Ward. Authority rankings from hits, pagerank, and salsa: Existence, uniqueness, and effect of initialization. *SIAM J. Sci. Comput.*, 27(4):1181–1201, 2006.

P. J. Feltovich, M. J. Prietula, and K. A. Ericsson. Studies of expertise from psychological perspectives. In *The Cambridge Handbook of Expertise and Expert Performance*, pages 41–68. Cambridge University Press, New York, NY, USA, 2006.

S. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

R. V. Guha. Open rating systems. In *FOAF '04: Proceedings of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proc. of 1st ACM Int'l Conf. on Web Search and Data Mining (WSDM'08)*, pages 195–206. ACM, February 2008.

A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *Proc. of 3rd European Semantic Web Conference, Montenegro, 2006*, LNCS, pages 411–426. Springer, 2006.

A. John and D. Seligmann. Collaborative tagging and expertise in the enterprise. In *Proceedings of Collaborative Web Tagging Workshop, collocated at WWW2006, Edinburgh, Scotland, UK*, 2006.

J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *Proc. of 3rd Int'l Workshop on Adversarial information retrieval on the*

web, pages 57–64, New York, NY, USA, 2007. ACM.

B. Krause, C. Schmitz, A. Hotho, and G. Stumme. The anti-social tagger: detecting spam in social bookmarking systems. In *AIRWeb '08: Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 61–68, New York, NY, USA, 2008. ACM.

R. Krestel and L. Chen. Using co-occurence of tags and resources to identify spammers. In *Proceedings of ECML PKDD Discovery Challenge Workshop, collocated with ECML/PKDD 2008*, 2008.

R. Lambiotte and M. Ausloos. Collaborative tagging as a tripartite network. In *Computational Science – ICCS 2006*, volume 3993, pages 1114–1117, Heidelberg, 2006. Springer.

C. Macdonald, D. Hannah, and I. Ounis. High quality expertise evidence for expert search. In *Proc. of 30th European Conference on IR Research, UK, 2008.*, pages 283–295. Springer, 2008.

A. Madkour, T. Hefni, A. Hefny, and K. S. Refaat. Using semantic features to detect spamming in social bookmarking systems. In *Proceedings of ECML PKDD Discovery Challenge Workshop, collocated with ECML/PKDD 2008*, 2008.

B. Markines, C. Cattuto, and F. Menczer. Social spam detection. In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 41–48, New York, NY, USA, 2009. ACM.

A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. Technical report, University of Illinois Urbana-Champaign, USA, 2004.

P. Mika. Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web: Proceedings of 4th International Semantic Web Conference (ISWC)*, LNCS, pages 522–536, Heidelberg, 2005. Springer.

T. Moore and R. Clayton. Evaluating the wisdom of crowds in assessing phishing websites. In *FC '08: Proceedings of 12th International Conference on Financial Cryptography and Data Security (Revised Selected Papers)*, volume 5143 of *LNCS*, pages 16–30, Berlin, Heidelberg, 2008. Springer.

N. Neubauer and K. Obermayer. Hyperincident connected components of tagging networks. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 229–238, New York, NY, USA, 2009. ACM.

M. G. Noll and C. Meinel. Authors vs. readers: A comparative study of document metadata and content in the www. In *Proceedings of 7th International ACM Symposium on Document Engineering (DocEng'07)*, pages 177–186, Canada, 2007.

M. G. Noll and C. Meinel. Exploring social annotations for web document classification. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 2315–2320, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-753-7. .

J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, Canada, 1995.

R. Wetzker, C. Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Proc. of Mining Social Data Workshop, collocated with ECAI 2008*, pages 26–30, 2008.

H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. *SIGCOMM Comput. Commun. Rev.*, 36(4):267–278, 2006.

J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *Proc. of 16th WWW Conference, 2007*, pages 221–230, New York, NY, USA, 2007. ACM.