Telling Experts from Spammers: Expertise Ranking in Folksonomies

Michael G. Noll
Hasso Plattner Institute,
Potsdam, Germany
michael.noll@hpi.uni-potsdam.de

Ching-man Au Yeung
School of Electronics and
Computer Science
University of Southampton, UK
cmay06r@ecs.soton.ac.uk

Nicholas Gibbins
School of Electronics and
Computer Science
University of Southampton, UK
nmg@ecs.soton.ac.uk

Christoph Meinel
Hasso Plattner Institute,
Potsdam, Germany
meinel@hpi.uni-potsdam.de

Nigel Shadbolt
School of Electronics and
Computer Science
University of Southampton, UK
nrs@ecs.soton.ac.uk

ABSTRACT

With a suitable algorithm for ranking the expertise of a user in a collaborative tagging system, we will be able to identify experts and discover useful and relevant resources through them. 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 depends on the expertise of the users who have assigned tags to it. Secondly, an expert should be one who tends to identify interesting or useful resources before other users do. We propose a graph-based algorithm, *SPEAR (SPamming-resistant Expertise Analysis and Ranking)*, which implements these ideas for ranking users in a folksonomy. We evaluate our method with experiments on data sets collected from Delicious.com comprising over 71,000 Web documents, 0.5 million users and 2 million shared bookmarks. We also show that the algorithm is more resistant to spammers than other methods such as the original HITS algorithm and simple statistical measures.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms

Algorithms, Human Factors, Experimentation

Keywords

collaborative tagging, folksonomy, expertise, ranking, spam

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

1. INTRODUCTION

Collaborative tagging systems such as Delicious.com and Flickr.com provide Web users a new means of organizing and sharing resources such as bookmarks or photos on the Web [15]. Such systems also allow users to search for documents relevant to a particular topic or for other users who are experts in a particular domain. However, identifying relevant documents and knowledgeable users are not trivial tasks. 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 do not necessarily result in useful rankings of resources and users due to a variety of reasons such as the large volume of data and the presence of spammers [17].

In this paper, we propose a novel method to assess the expertise of a user in a collaborative tagging system. We propose that the level of expertise of a user with respect to a particular topic is mainly determined by two factors: (1) there should be a relationship of mutual reinforcement between the expertise of a user and the quality of a resource; and (2) an expert should be one who tends to identify useful resources before other users discover them. 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. We carry out experiments on both simulated and real-world data sets obtained from Delicious, and show that SPEAR is able to detect the difference between different types of experts, and is more resistant to spammers than other methods.

The rest of the paper is structured as follows. Section 2.1 provides a brief review of collaborative tagging. Section 2.2 mentions related work. We discuss expertise and experts in the context of collaborative tagging in Section 3. We then introduce SPEAR in Section 4 and describe our experiments in Section 5. Finally we give our conclusions in Section 6.

2. BACKGROUND

2.1 Collaborative Tagging

A *collaborative tagging system* [4] allows arbitrary users to assign tags freely to any resources available on the Web. When the tags and resources contributed by different users are aggregated, a kind of user-generated classification scheme emerges. Such bottom-up classification schemes have been given the name *folksonomies*.

A folksonomy basically involves three types of entities, namely users, tags and resources/documents and can be represented formally as a tripartite hypergraph of users, tags and documents [13].

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*. It represents the fact that a particular user $u \in U$ has assigned a tag $t \in T$ to a document $d \in D$. 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: $\mathcal{F}_t = (U_t, D_t, R_t)$, where $R_t = \{(u, d) | (u, t, d) \in R\}$, $U_t = \{u | (u, d) \in R_t\}$, and $D_t = \{d | (u, d) \in R_t\}$.

This can be generalized to cases in which the topic is represented by a conjunction or disjunction of two or more tags $\{t_1, t_2, ..., t_n\}$: $R_{\{t_1,...,t_n\}} = \{(u, d) | (u, t_1, d) \in R \wedge ... \wedge (u, t_n, d) \in R\}$ or $R_{\{t_1,...,t_n\}} = \{(u, d) | (u, t_1, d) \in R \vee ... \vee (u, t_n, d) \in R\}$.

2.2 Related Work

Expert identification traditionally involves building candidate profiles by associating documents with the candidates and employing IR techniques on the profiles [11]. Recent approaches involve graph-based analysis of user networks in a community [2]. For example, Zhang et al. [19] apply an algorithm based on PageRank to produce expertise ranking of users of a Java Developer bulletin board. Some studies propose ranking users and documents together, such as by considering the social network of users and the citation network of documents at the same time [20], or by analysing similar relations in a Web log [16].

While folksonomies can be represented as graphs, their tripartite structure requires certain modifications to either the existing graph-based algorithms or the way the data in the folksonomies are modeled before graph-based ranking methods can be applied. For example, Hotho et al. propose the FolkRank algorithm [7], which is based on the PageRank algorithm, for providing ranking of users, tags, and documents at the same time. However, as FolkRank is based on PageRank, users, tags and URLs which appear frequently in the folksonomy usually dominate the results. This also makes the method prone to the influence of spammers.

Koutrika et al. [9] are the first to discuss methods of tackling spamming activities in collaborative tagging. 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. There are also proposals for detecting spammers in tagging systems based on machine learning approaches [10, 12]. Compared with these approaches, our proposed algorithm aims at, in addition to finding experts, demoting spammers in the ranked list of users instead of detecting their presence. We believe that different types of methods, including detection, demotion, and also prevention are complementary in tackling spammers [5].

3. EXPERTS IN COLLABORATIVE TAGGING SYSTEMS

An *expert* is generally someone with a high level of knowledge, technique or skills in a particular domain. This implies that experts are reliable sources of relevant resources and information. Here, we describe 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 used by most existing collaborative tagging systems today. However, such an approach does not consider the facts that quantity does not imply quality, and that spammers who indiscriminately tag a large number of documents may be mistaken as experts [17].

Studies in psychology have found that expertise involves the ability to select the most relevant information for achieving a goal [3]. 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 tends to add *high quality* documents to their collections, i.e. such documents which are identified in turn by the number as well as the expertise of the users who have the same 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 is similar to the HITS algorithm [8] for ranking Web pages, in which the *hubness* and *authority* of a page mutually reinforce each other. The differences in our case are that collaborative tagging involves two different kinds of interrelated entities, namely users and documents, and that only users can point 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 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.

While mutual reinforcement relationship between users and documents have been discussed in the literature [16, 20], we do not believe this assumption alone is enough in our study. In collaborative tagging, users usually notice new documents after some other users have tagged them and introduced them to the community. In other words, there is a great chance that users learn from each other instead of discovering information by themselves as in performing a Web search. Hence, we also introduce the following second assumption of expertise in collaborative tagging.

3.2 Discoverer vs. Follower

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. In addition, 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 [5].

Hence, in addition to knowing a lot of high quality documents per se, we believe an expert to be someone who is also be able to recognize the usefulness of a document before

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
    $\vec{E} \leftarrow \vec{Q} \times A^T$
    $\vec{Q} \leftarrow \vec{E} \times A$
    Normalize $\vec{E}$
    Normalize $\vec{Q}$
end for
$L \leftarrow$ Sort users by their expertise score in $\vec{E}$
return $L$
```

others do [1], thus becoming the first to tag it, and by doing so bringing it to the attention of other users. 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 because the documents have become popular already. 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 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 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.

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. SPEAR ALGORITHM

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 $t \in T$. 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.

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.

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 A of size $M \times N$ where $A_{i,j} := 1$ if user i has assigned t to document j, and $A_{i,j} := 0$ otherwise. Based on this matrix, the calculation of expertise and quality scores is an iterative process similar to that of the HITS algorithm:

$$\vec{E} = \vec{Q} \times A^T \quad (1)$$

$$\vec{Q} = \vec{E} \times A \quad (2)$$

To implement the idea of discoverers and followers, we prepare the adjacency matrix 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 A:

$$A_{i,j} = |\{u|(u, t, d_j, c), (u_i, t, d_j, c_i) \in R_t \wedge c_i < c\}| + 1 \quad (3)$$

According to equation 3, the cell $A_{i,j}$ 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, $A_{i,j}$ 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, $A_{i,j}$ 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:

$$A_{i,j} = C(A_{i,j}) \quad (4)$$

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 $C''(x) \leq 0$. 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 $C(x) := x^{0.5} = \sqrt{x}$.

The final SPEAR algorithm is shown in pseudocode in listing 1.

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 with. To mitigate this problem, we adopt the following strategy. Firstly, we collect real-world data as the basis of our experiments. Based on these real-world data sets, as well as reported in recent studies of collaborative tagging [9, 17], we generate simulated users who exhibit certain tagging behaviors and inject these users and their generated bookmarks into the real-world data sets. Hence, we are able to mitigate the lack of a ground truth as we know how the simulated users should be ranked.

Since we required data sets of different topics to evaluate SPEAR, we first collected a large number of tags from Delicious by monitoring its list of popular tags and its front page. We then sampled at random a total of 50 tags, used a crawler to retrieve the most recent URLs posted to Delicious under these tags, and downloaded the bookmarking history of all these URLs. A bookmark includes the Delicious username of the bookmarking user, the title and description given to the bookmark, any associated tags, and its creation timestamp. Due to technical restrictions imposed by Delicious, we retrieved up to a maximum of 2,000 user bookmarks per URL. The set of 50 tags contains tags representing a wide range of topics, including for example `photography`, `semanticweb`, `economics` and `fashion`. The 50 data sets altogether involve 515,024 unique usernames, 71,300 unique URLs, and 2,189,978 unique bookmarks.

With regard to simulated data, the basic idea was to insert controlled 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. To simulate experts, we would have to insert virtual bookmarks to popular documents because these users tend to tag only relevant information.

We created two different types of user profiles: expert-like and spammer-like users. For each type, we also wanted to model three variants to better match real-world scenarios and to improve the evaluation setup. The three variants for experts are geeks, veterans and newcomers, and those for spammers are flooders, promoters and trojans. The detailed descriptions of these user types and the method we used to generate them are presented in the following sections.

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 in [6, 14]. He tends to be among the first users to tag documents which will eventually become quite popular within the community. Hence, he is a discoverer with many followers.

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.

A *geek* is similar to a veteran but has significantly more bookmarks than a veteran. We can consider the geek profile as the "best" expert within our simulation.

Type	**P1**	**P2**	**P3**	**P4**
Geek	$0.10 * n_d$	0.10	*See figure 1*	*See figure 2*
Veteran	$0.05 * n_d$	0.10	*See figure 1*	*See figure 2*
Newcomer	$0.05 * n_d$	0.10	*See figure 1*	EQUAL()
Flooder	$0.10 * n_d$	0.05	EQUAL()	*See figure 2*
Promoter	50	0.95	EQUAL()	*See figure 2*
Trojan	$\mu_d * 1.1$	0.10	*See figure 1*	*See figure 2*

Table 1: Configuration of parameters P1-P4 for simulated user profiles. n_d is the total number of bookmarked documents in the relevant data set, μ_d is the average number of bookmarked documents per user. $EQUAL()$ means that each document rank or time was selected with equal probability.

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 (cf. [17, 9]). However, he tends to be one of the last users in the bookmarking timeline.

A *promoter* is a spammer who focuses on tagging his own documents to promote their popularity, and does not care much about 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 quite a number 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 [18].

A *trojan* is a more sophisticated spammer. His strategy is to mimic regular users for most 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.

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

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 (a "good" geek might receive a higher expertise score than a "bad" geek). 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).

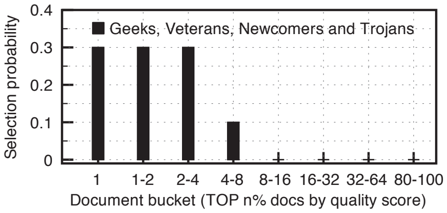


Figure 1: PMF for document rank preferences (P3) for geeks, veterans, newcomers and trojans. In contrast, flooders and promoters chose document ranks indiscriminately. Lower bucket numbers refer to higher quality documents. We chose exponentially increasing bucket sizes to account for power law effects in collaborative tagging systems [14].

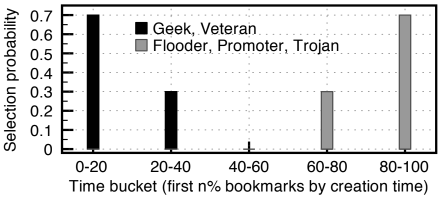


Figure 2: PMF for time preferences (P4) for geeks, veterans (black) and flooders, promoters, trojans (gray). Lower bucket numbers refer to earlier timestamps. In contrast, newcomers chose timestamps indiscriminately.

5.1.3 *Simulation Parameters*

We manipulate the following four parameters to model the users in our simulation.

P1: *Number of a user's bookmarks.* For example, geeks and flooders would have a higher 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 need 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 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 configuration of simulation parameters for each user type is shown in Table 1 (see also Figure 1 and 2 for the probability mass functions for **P3** and **P4**).

5.2 General Behavior

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

We single out three cases for a closer examination of the performance of SPEAR in the following analyses. These cases include `semanticweb`, `photography`, and a combination (conjunction) of the two tags `javascript` and `programming`.

By running the three ranking algorithms on the users and documents in the three selected data sets, we obtain Figure 3, which shows the resultant normalized expertise score curves. It shows that SPEAR produces more differentiated values than HITS and FREQ, i.e. the difference in expertise scores between two ranks in SPEAR is generally larger than in HITS and FREQ, where the curves are flatter.

Another finding is 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. While, SPEAR and HITS also show occasional staircase steps, this is 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 results in "time collisions", and coupled with only a snapshot view which we could create of the full data stored at Delicious, we see occasional plateaus of equal score values. 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 generate, for each of the 50 real-world data sets, 20 experts of each type (60 total per data set) and insert them to the corresponding data set. We then apply SPEAR, the original HITS algorithm and FREQ to these data sets comprising both real-world and simulated users. Figure 4(a) shows the average normalized ranks of the different types of simulated users given by the different algorithms. We observe that the major difference between SPEAR and the two other baseline algorithms is consistent among all the 50 data sets. In SPEAR, geeks are generally ranked higher than veterans, which are in turn ranked higher than newcomers. The other two algorithms, HITS and FREQ, however, cannot distinguish between veterans and newcomers.

To have a closer look at the differences, we visualise the ranks of the simulated experts in the three selected data sets. The results are shown in Figure 5 (other data sets show very similar results). Only ranks assigned to simulated experts are marked with symbols, real-world users are not marked (as we have no ground truth for the latter as described above, we cannot evaluate these). Note that some overlapping between the three expert variants are expected due to the PMF-based simulation setup.

Here, we can clearly see that SPEAR is able to detect the differences between the three types of experts. We observe that geeks and experts do compete for the top ranks even though the geeks win in general. This means that some veterans, although having fewer bookmarks than geeks in general, are sometimes ranked higher by SPEAR because they have some higher quality bookmarks. On the other hand, while HITS and FREQ do rank geeks higher than veterans

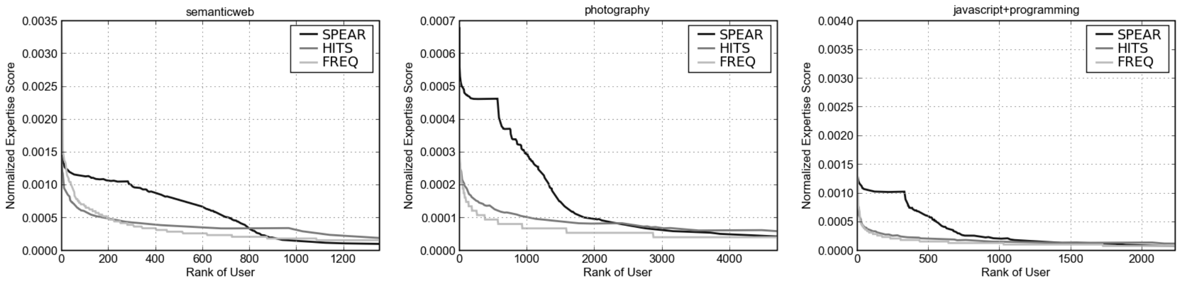
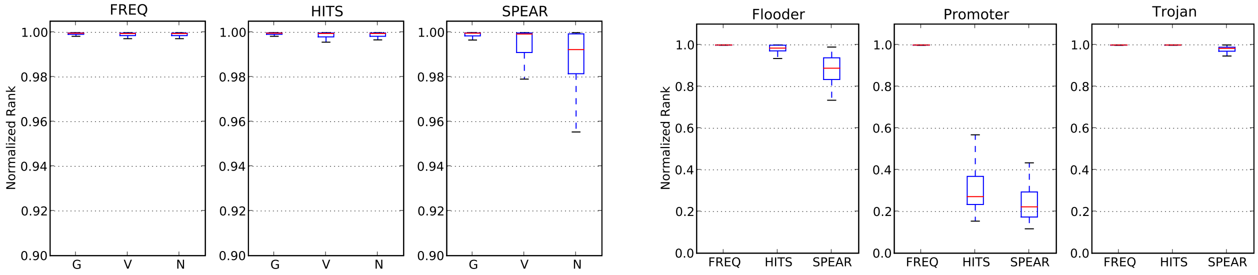


Figure 3: Normalized expertise scores of users returned by SPEAR, HITS and FREQ for the three selected data sets: `semanticweb`, `photography` and `javascript ∧ programming`. Note that the difference in scale of the y-axis for `photography` is caused by the significantly higher number of users in this data set.



(a)

(b)

Figure 4: Rankings of simulated users returned by SPEAR, HITS and FREQ for all the data sets. The y-axis represents the average normalized rank of the users: the user ranked first has a value of 1, while the last user has a value of 0. (a) shows how different types of experts are ranked by the three algorithms. G stands for geeks, V for veterans, and N for newcomers. (b) shows how different types of spammers are ranked. SPEAR can be observed to demote all types of spammers much better than the other two.

and newcomers, geeks are also the "easiest" expert variant because they have a very high quantity of good bookmarks. This means even the naive FREQ should and does perform reasonably for this user variant. However, both HITS and FREQ fail to differentiate between veterans and newcomers, which end up being mixed together. This result suggests that SPEAR succeeds in distinguishing veterans and newcomers by implementing the notion of discoverers and followers. In contrast, HITS still tends to return results which are 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 verified this simulation outcome with a manual analysis of the TOP 10 experts ranked by SPEAR for `photography`, `semanticweb` and `javascript∧programming`. A first observation was that these users seemed to be more involved or serious about their activity and participation on Delicious: they were significantly more likely to provide optional personal information such as their real name or personal website, e.g. links to their photos on Flickr.com or microblog on Twitter.com. Their number of bookmarks had a wide range from some hundreds to ten thousands. Interestingly, we could identify a Semantic Web researcher as one of SPEAR's experts for `semanticweb`. Similarly, the TOP 2 experts for `javascript∧programming` were professional software developers. FREQ in comparison ranked most of SPEAR's experts not even in the TOP 200.

We can 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.

5.4 Demoting Spammers

Similarly, we generate and add 20 flooders, promoters and trojans, respectively, to each of the 50 data sets. The overall results are shown in Figure 4(b), and the visualization of the ranks of users in the three selected data sets in Figure 6.

FREQ is observed to be very vulnerable to spammers, as all spammers are given top ranks simply because they have a large number of bookmarks. HITS performs better than FREQ as it tends to demote promoters to low ranks, although is not able to demote flooders and trojans. Unfortunately, flooder-type spammers in particular are often found in existing collaborative tagging systems [17].

SPEAR gives the best performance among the three algorithms. Firstly, it correctly demotes both flooders and promoters, and in every case it assigns the spammers much lower ranks than HITS and FREQ. Secondly, SPEAR is also able to demote trojans who use a much more sophisticated spamming scheme. While they are still ranked much higher than the other two variants of spammers, no trojans are ranked higher than rank #100 by SPEAR (see Figure 6b). 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 performs reasonably well in getting rid of all trojans in the relevant range. In fact, the problem with

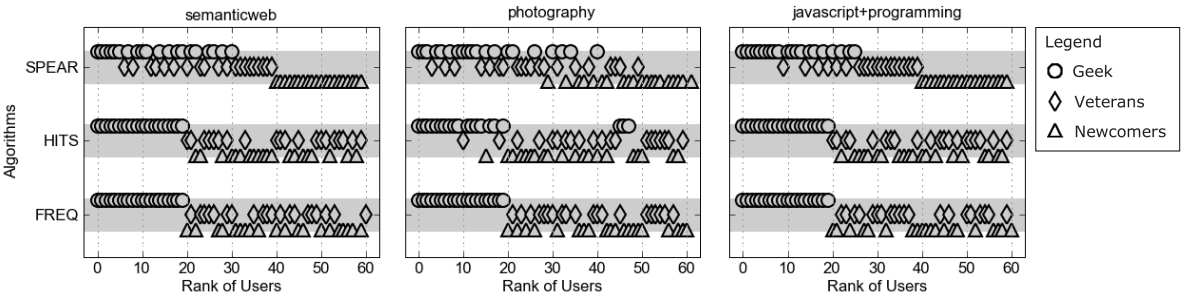


Figure 5: Ranks of real-world users and simulated experts as returned by the three algorithms (only the TOP 60 ranks are shown). Symbols mark the ranks assigned to simulated experts. For readability, symbols are enlarged beyond the x-axis scale and plotted on three different levels per bar. Bar areas without symbols on any level represent real-world users. The algorithms should rank geeks before veterans before newcomers. Some overlapping of simulated experts is expected due to the experimental setup as described in the text.

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 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.

We verified this simulation outcome with another data set containing documents tagged with `mortgage`. Without generating any simulated users, we ranked real users by the number of their bookmarks, and manually analyzed the TOP 50 users. We found that 30 out of 50 users were (real) spammers of either flooder or promoter type. Compared to FREQ, both SPEAR and HITS were able to remove these spammers from the TOP 50 in this case, and SPEAR demoted the spammers significantly more than HITS.

In summary, SPEAR produces better rankings than both the original HITS algorithm and simple frequency counting. It is able to distinguish between different types of experts, and it is also able to consistently demote different types of spammers and remove them from the top of the ranking. In other words, SPEAR is 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.

6. CONCLUSIONS AND FUTURE WORK

We propose SPEAR for ranking experts in a collaborative tagging system, and study its behavior by using a combination of simulation and real-world data. Our experiments suggest that SPEAR is better at distinguishing various kinds of experts and is more resistant to different kinds of spammers than HITS and a simple statistical measure. 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, we will further conduct experiments using different credit score functions and study how they affect the performance of SPEAR. In addition, we want to study how expertise in closely related tags – e.g. measured by co-occurrence – can be taken into consideration when ranking users for a particular tag. For example, when ranking users for `javascript`, can we also consider users who are ranked highly in `webdev` (aka "web development")? Moreover, we plan to incorporate the idea of "recency of knowledge" into SPEAR. In other words, we believe a user who is 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"). We will study how this notion can be implemented in our algorithm.

Lastly, SPEAR also provides another piece of information: a ranked list of documents sorted by their quality score. Although we do not pay much attention to this aspect, it can be very useful in providing a ranking of documents in a folksonomy. We look forward to extending our study to this aspect of SPEAR in the future.

7. ACKNOWLEDGMENTS

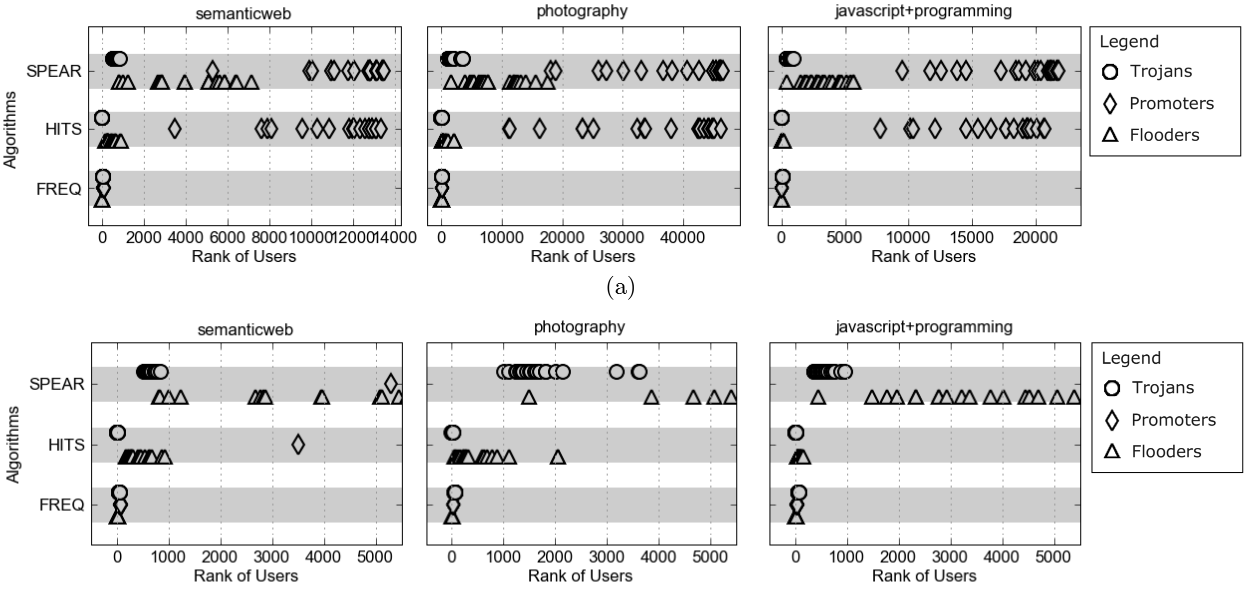
This paper discusses research work undertaken by the first author in partial fulfillment of the requirements for the PhD degree of the University of Potsdam and the University of Luxembourg/LIASIT, following a joint thesis supervision agreement between the institutions (cotutelle de thèse). The second author is supported by the R C Lee Centenary Scholarship, Hong Kong.

8. REFERENCES

[1] 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, USA, 2006.

[2] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *Proc. of ACM SIGMOD Workshop on Research issues in Data Mining and Knowledge Discovery*, pages 42–48. USA, 2003.

[3] 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, USA, 2006.



(a)

(b)

Figure 6: Ranks of real-world users and simulated spammers as returned by the three algorithms. (a) presents plots of all users in a data set, while (b) focuses on the first 5000 ranks of users. Symbols mark the ranks assigned to simulated spammers. For readability, symbols are enlarged beyond the x-axis scale and plotted on three different levels per bar. Bar areas without symbols on any level represent real-world users.

[4] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), April 2005.

[5] 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.

[6] 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*, pages 195–206. Palo Alto, USA, 2008.

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

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

[9] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *Proc. of Int'l Workshop on Adversarial information retrieval on the web*, pages 57–64. 2007.

[10] 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.

[11] 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.

[12] A. Madkour, T. Hefni, A. Hefny, and K. S. Refaat. Using semantic features to detect spamming in social bookmarking systems. In *Proc. of ECML PKDD Discovery Challenge Workshop*, Belgium, 2008.

[13] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15, 2007.

[14] M. G. Noll and C. Meinel. Authors vs. readers: A comparative study of document metadata and content in the www. In *Proc. of 7th Int'l ACM Symposium on Document Engineering*, pages 177–186, Canada, 2007.

[15] M. G. Noll and C. Meinel. Exploring social annotations for web document classification. In *Proc. of ACM Symposium on Applied Computing*, pages 2315–2320, Fortaleza, Brazil, 2008.

[16] J. Wang, Z. Chen, L. Tao, W.-Y. Ma, and L. Wenyin. Ranking user's relevance to a topic through link analysis on web logs. In *WIDM '02: Proceedings of the 4th Int'l workshop on Web information and data management*, pages 49–54, USA, 2002. ACM.

[17] 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.

[18] 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.

[19] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *Proc. of WWW Conference*, pages 221–230. Banff, Canada, 2007.

[20] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles. Co-ranking authors and documents in a heterogeneous network. In *Proc. of 7th IEEE Int'l Conference on Data Mining*, pages 739–744, Washington, USA, 2007.