On Measuring Expertise 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
SO17 1BJ, United Kingdom
[2]Hasso-Plattner-Institut, University of Potsdam
14440 Potsdam, Germany

cmay06r@ecs.soton.ac.uk, michael.noll@hpi.uni-potsdam.de,
nmg@ecs.soton.ac.uk, meinel@hpi.uni-potsdam.de, nrs@ecs.soton.ac.uk

1. INTRODUCTION

Collaborative tagging systems such as Delicious.com provide a new means of organizing and sharing resources. They also allow users to search for documents relevant to a particular topic or for other users who are experts in a particular domain. Nevertheless, identifying relevant documents and knowledgeable users is not a trivial task, especially when the volume of documents is huge and there exist spamming activities [13].

In this paper, we discuss the notions of experts and expertise in the context of collaborative tagging systems. We propose that the level of expertise of a user in 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 document; and (2) an expert should be one who tends to identify useful documents 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 data sets and real-world data sets obtained from Delicious, and show that SPEAR is more resistant to spamming than other methods such as the HITS algorithm and simple statistical measures.

2. BACKGROUND

2.1 Collaborative Tagging

A *collaborative tagging system* [1] allows arbitrary users to assign tags freely to any documents available on the Web. When the tags and documents contributed by different users are aggregated, a kind of user-generated classification scheme – commonly known as *folksonomies* [12] – emerges.

A folksonomy basically involves three types of entities, namely users, tags and documents, and can be formally represented as a tripartite hypergraph [10].

Definition 1. A folksonomy 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 $\mathcal{F}$ as $\mathcal{F}_t = (U_t, D_t, R_t)$, 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 profiles by associating documents with the candidates and employing IR techniques on the profiles [8]. Recent approaches involve graph-based analyses of user networks. For example, Zhang et al. [14] apply an algorithm based on PageRank to produce expertise ranking of users in an online forum. While folksonomies can be represented as graphs, their tripartite structure requires modifications to either existing graph-based algorithms or the data model before graph-based ranking methods can be used. For example, Hotho et al. [4] propose FolkRank, which is based on the PageRank algorithm, for ranking users, tags and documents.

Koutrika et al. [6] discuss methods of tackling spamming activities in collaborative tagging systems, and propose that the "reliability" of users – whether their tags coincide with those of the others – should be taken into account to produce document rankings which are resistant to spammers. There are also proposals of detecting spammers based on machine learning approaches such as [7, 9]. 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 [2].

3. EXPERTS AND EXPERTISE

In general, an *expert* is someone who possesses a high level of knowledge 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.

*Partly supported by EPSRC (Grant No. EP/F013604/1).

3.1 User Expertise and Document Quality

The simplest way to assess the *expertise* of a user is by the number of times he has used a tag (or a set of tags) on some documents. However, this does not take into consideration the facts that quantity does not imply quality, and that there exist spammers who indiscriminately tag a large number of documents [13].

We believe that an expert should be someone who not only has a large collection of documents, but also tends to add to their collections *high quality* documents, which are identified in turn by the number as well as the expertise of the users who have it 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 [5] for link structure analysis among Web pages, in which the *hubness* and *authority* of a page mutually reinforce each other. A major difference in our case is that collaborative tagging involves two different types of interrelated entities, namely human users and Web pages. There are also only links pointing from users to documents in a folksonomy because only users tag 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 much 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

In the HITS approach, two users will be considered to be of 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 of their popularity. In addition, a spammer who wants to bring some Web pages to the attention of other users can easily exploit this weakness and boost his expertise score by tagging several popular documents.

Hence, in addition to knowing a lot of high quality documents per se, we believe an expert should also be someone who is able to recognizes the usefulness of a document before the others do, thus becoming the first to bookmark it, assign tags to it and bring it to the attention of other users. 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 they have become popular already. Generally speaking, the earlier a user has tagged a document, the more *credit* he should receive for his actions.

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, as spammers will probably not be discoverers but mostly followers.

4. SPEAR ALGORITHM

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

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 $\mathcal{F}$. We extend the notion of tagging to accommodate the creation timestamp of each tagging: every tagging is 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 does.

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 he tags with t, and the quality score of a document depends on the expertise score of the users who assigns 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 the expertise and quality scores involves 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 populate the adjacency matrix A in a way different from the above method of assigning either 0 or 1 to its cells by using the following equation:

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

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. 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 have assigned t to d_j, $A_{i,j}$ will be equal to 1. In this way, earlier users will be credited more by the quality score of the documents in the iterative process than later users.

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

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

While a simple linear credit score assignment such as $C(x) := x$ can be used, we believe that the function should somehow reduce the differences between high scores while retaining the ordering of the scores in A. 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 thereafter. Hence, a proper credit scoring function C

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

should be an increasing function with a decreasing first derivative: $C'(x) > 0$ and $C''(x) \leq 0$. 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 Algorithm 1.

5. EVALUATION

5.1 Data Sets and Methodology

Evaluation is difficult due to the lack of a proper ground truth. To mitigate this problem, we combine both real-world and simulated data to compare the behavior and performance of SPEAR with other algorithms. Real-world data are used as the base input for our experiments which "sets the stage" for inserting simulated data. In addition, simulation allows us to control the characteristics of generate users based on recent studies of collaborative tagging systems [6, 13].

With regard to real-world data, we developed a crawler to retrieve the most recent URLs of several tags with their bookmarking history from Delicious.com. We retrieved up to a maximum of 2,000 bookmarks per URL (due to restriction of Delicious.com). A bookmark in our data set includes the Delicious username of the user, the title and description of the bookmark, any associated tags, and the creation timestamp of the bookmark. An overview of the data sets is shown in Table 1.

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 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. For each type, we also wanted to model three variants to better match real-world scenarios and to improve the evaluation setup. An overview is shown in Table 2.

We manipulate the following four parameters for modeling simulated users and their tagging behavior and thus simulated data. The detailed descriptions of the user variants will be presented after this list.

Tag	Users	Documents
`javascript` ∧ `programming`	22,329	887
`photography`	47,043	942
`semanticweb`	13,527	1,232

Table 1: Statistics of real-world data sets retrieved from Delicious.com in October 2008.

User Type	Variants
Expert	Geek, Veteran, Newcomer
Spammer	Flooder, Promoter, Trojan

Table 2: The simulated user profiles created for the evaluation of SPEAR.

- **P1**: *Number of a user's bookmarks.*
- **P2**: *Newness*: The percentage of bookmarks to documents which are not in the real-world data.
- **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, respectively.
- **P4**: *Time preferences*: A probability mass function (PMF) which specifies at which point in time a simulated bookmark tends to be inserted. For example, the PMFs of veterans tend to focus on the early stages in the bookmarking history, newcomers are rather evenly distributed, and flooders tend to be very late.

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 [3, 11]. He tends to be among the first users to tag documents which would usually become quite popular within the community. Hence, he is a discoverer with many followers.

A *newcomer* represents a new user 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 also 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.

Geeks should generally be ranked higher than veterans in terms of expertise, 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

We simulate three types of spammers, namely flooders, promoters, and trojans. A *flooder* tags a huge number of documents which already exist in the system, most likely in an automated way [13, 6]. However, he tends to be one of the last users in the timeline.

A *promoter* 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. This kind of spammers is quite common and quite a number were found on Delicious during our experiments. There were even groups of them who had sequentially named user accounts of the form *iSpamYou001*, *iSpamYou002*, etc.

A *trojan* is more sophisticated in that his strategy is to mimic regular users in the majority of his tagging activities. He disguises his malicious intents by tagging already popular pages, but at some point adds links to his own documents which can be malware-infected or phishing web pages.

It should be noted that our simulations were probabilistic so that even identical user profiles will produce variations in simulated data. On the one hand, this means that even two users with the same profile will 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).

5.2 Experiments

We study 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 bookmarks of the users. The latter is very popular on collaborative tagging systems in practice, and thus FREQ serves as the "baseline" for our experiments.

5.2.1 Promoting Experts

To study how different variants of experts are ranked by SPEAR, we generate, for each of the data sets of `semanticweb`, `photography` and `javascript` $\wedge$ `programming`, twenty simulated users for each of the variants. Figure 1(a) shows the ranks of the simulated users returned by SPEAR, the original HITS algorithm and FREQ.

In SPEAR, geeks are generally ranked higher than veterans, which are in turn ranked higher than newcomers. We can also 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 less bookmarks than geeks in general, are sometimes ranked higher by SPEAR because they have some higher quality bookmarks. All in all, this is the expected and desired behavior.

As for HITS and FREQ, while they do rank geeks higher than veterans and newcomers, geeks are actually the "easiest" expert variant because they have a very high quantity of good bookmarks. This means even the naive FREQ should 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 tends to return results which are heavily influenced by the number of documents of a user, even though it is also an implementation of a mutual reinforcement scheme. 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.

5.2.2 Demoting Spammers

Similarly, we generate twenty flooders, promoters and trojans, respectively, for each of the three data sets and apply the three different ranking algorithms to them. The results are shown in Figure 1(b).

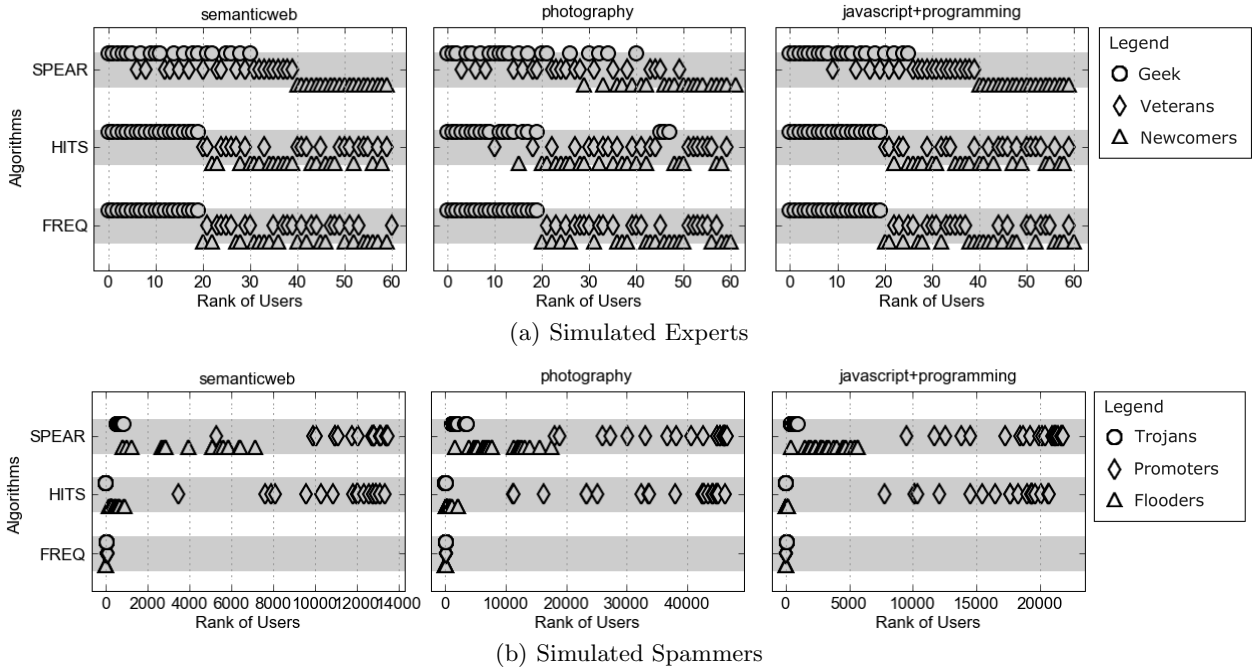
FREQ is very vulnerable to spammers, as all spammers are given top ranks due to their large number of bookmarks. HITS performs better than FREQ as it tends to demote promoters to low ranks, although it is not able to demote flooders and trojans. Unfortunately, flooder-type spammers in particular are often found in existing collaborative tagging systems [13].

SPEAR gives the best performance among the three algorithms. Firstly, it correctly demotes both flooders and promoters by assigning them much lower ranks than HITS and FREQ. Secondly, SPEAR is also able to demote trojans who use a much more sophisticated spamming technique. While they are still ranked much higher than the other two types of spammers, no trojans are ranked higher than rank #100 by SPEAR. 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 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. 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.

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 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 collabo-



(a) Simulated Experts

(b) Simulated Spammers

Figure 1: Ranks of different variants of simulated users. The gray bar represents users in the real data sets, while simulated users are represented by different symbols.

rative tagging system and create 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. In the future, we will further conduct experiments using different credit score functions and study how they affect the performance of SPEAR. We would also like to study how expertise in closely related tags can be taken into consideration when ranking users for a particular tag.

7. REFERENCES

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

[2] 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.

[3] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proc. of WSDM'08*, pages 195–206. ACM, 2008.

[4] A. Hotho, R. Ja"schke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *Proc. of ESWC'06, Budva, Montenegro*, LNCS, pages 411–426. Springer, 2006.

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

[6] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *Proc. of the 3rd Int'l Workshop on Adversarial IR on the web*, pages 57–64. ACM, 2007.

[7] R. Krestel and L. Chen. Using co-occurence of tags and resources to identify spammers. In *Proc. of ECML PKDD Discovery Challenge*, 2008.

[8] C. Macdonald, D. Hannah, and I. Ounis. High quality expertise evidence for expert search. In *Proc. of ECIR'08*, pages 283–295. Springer, 2008.

[9] 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*, 2008.

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

[11] M. G. Noll and C. Meinel. Authors vs. readers: A comparative study of document metadata and content in the www. In *Proc. of ACM DocEng'07*, pages 177–186, 2007.

[12] T. V. Wal. Folksonomy definition and wikipedia. *http://www.vanderwal.net/random/entrysel.php?blog=1750*, November 2, 2005. Retrieved on 13 Feb 2008.

[13] 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.

[14] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities. In *Proc. of WWW'07*, pages 221–230, 2007.