Combating Spam in Tagging Systems: An Evaluation

GEORGIA KOUTRIKA
Computer Science Department, Stanford University
FRANS ADJIE EFFENDI
Computer Science Department, Stanford University
ZOLTÁN GYÖNGYI
Computer Science Department, Stanford University
PAUL HEYMANN
Computer Science Department, Stanford University
HECTOR GARCIA-MOLINA
Computer Science Department, Stanford University

Tagging systems allow users to interactively annotate a pool of shared resources using descriptive strings, which are called tags. Tags are used to guide users to interesting resources and help them build communities that share their expertise and resources. As tagging systems are gaining in popularity, they become more susceptible to *tag spam*: misleading tags that are generated in order to increase the visibility of some resources or simply to confuse users. Our goal is to understand this problem better. In particular, we are interested in answers to questions such as: How many malicious users can a tagging system tolerate before results significantly degrade? What types of tagging systems are more vulnerable to malicious attacks? What would be the effort and the impact of employing a trusted moderator to find bad postings? Can a system automatically protect itself from spam, for instance, by exploiting user tag patterns? In a quest for answers to these questions, we introduce a framework for modeling tagging systems and user tagging behavior. We also describe a method for ranking documents matching a tag based on taggers' reliability. Using our framework, we study the behavior of existing approaches under malicious attacks and the impact of a moderator and our ranking method.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General

General Terms: Tagging, Spam

1. INTRODUCTION

Tagging systems allow users to interactively annotate a pool of shared resources using descriptive strings, which are called tags. For instance, in Flickr, a system for sharing photographs, a user may tag a photo of his Aunt Thelma with the strings "Thelma", "Aunt", and "red hair". In Del.icio.us, users annotate web pages of interest to them with descriptive terms. In these and other tagging systems, tags are used to guide users to interesting resources. For instance, users may be able to *query* for resources that are annotated with a particular tag. They may also be able to look at the most popular tags, or the tags used by their friends, to discover new content they may not have known they were interested in. Tagging systems are gaining in popularity since they allow users to build communities that share their expertise and resources.

In a way, tags are similar to links with anchor text on the web. That is, if page p contains a link to page q with the anchor text "Aunt Thelma", this implies that somehow page q is related to Aunt Thelma. This would be analogous to tagging page q with the words "Aunt Thelma" (in a tagging system where web pages were the resources). However, a tagging system is different from the web. The latter is comprised of pages and links, while the former is comprised of resources, users and tags. These resources can be more than web pages, e.g., they can be photos, videos, slides, etc. Typically, in a tagging system, there is a well defined group of users and resources that can be tagged.

As we know, the web is susceptible to *search engine spam*, that is to content that is created to mislead search engines into giving some pages a higher ranking than they deserve [Gyöngyi and Garcia-Molina 2005]. In an analogous fashion, tagging systems are susceptible to *tag spam*: misleading tags that are generated to make it more likely that some resources are seen by users, or generated simply to confuse users. For instance, in a photo system, malicious users may repeatedly annotate a photo of some country's president with the tag "devil", so that users searching for that word will see a photo of the president. Similarly, malicious users may annotate many photos with the tag "evil empire" so that this tag appears as one of the most popular tags. In a system that annotates web pages, one shoe company may annotate many pages (except the page of its competitor) with the string "buy shoes", so that users looking to buy shoes will not easily find the competitor's page. Taggers can also use popular tags to drive traffic to their sites. For instance, a blogger might add the tag "Britney Spears" to his or her blog, even when the content on the blog site has nothing to do with Britney Spears, trying to ride the celebrity wave and drive traffic to this blog [Adlam 2006].

Given the increasing interest in tagging systems, and the increasing danger from spam, our goal is to understand the problem better, to examine to what extent tagging systems can be manipulated by spammers and to try to devise schemes that may combat spam. In particular, we are interested in answers to questions like the following:

—How many malicious users can a tagging system tolerate before results significantly degrade? One or two bad guys are unlikely to bias results significantly, but what if 1% of the users are malicious? What if 10% are malicious? What if the malicious users collude? The answers to these questions, even if approximate, may give us a sense of how serious a problem tag spam is or could be. The answers may also help us in deciding how much effort should be put into the process of screening bad users when they register with the system.

—What types of tagging systems are more prone to spam? Are smaller systems more vulnerable to spammers? On the other hand, a popular system attracts more users but also possibly more spammers. Is popularity a double edged sword? Should users be allowed to use arbitrary tags or is it better to enforce more controlled vocabularies? Studying tagging systems can help reveal and understand both their strengths and their weaknesses with respect to potential spammers and is a step towards designing more robust systems.

—There are several potential countermeasures for limiting the impact of bad users on the system. For instance, one could apply a limit on the number of postings

each user is allowed to contribute. Would this shield the system from attacks? At the other extreme, what would be the impact of encouraging users to tag documents already tagged? Does encouraging people to post more tags help a system better cope with spam?

—Another possible way to combat spammers is to use a moderator that periodically checks the tags of users to see if they are "reasonable." This is an expensive and slow process; how effective can it be? How much effort would the moderator need to place in order to achieve a significant impact? How many users, documents or tag postings would it need to check?

—Could we build methods that leverage the collective knowledge of taggers to identify misused tags? For instance, we could take advantage of user correlations: if we notice that a user always adds tags that do not agree with the tags of the majority of users, we may want to give less weight to the tags of that user. Would this make the system more resilient to bad users? What would be the downside of such social anti-spam strategies for detecting bad users?

As the reader may suspect, answering these questions is extremely hard for a number of reasons. First, the notion of a "malicious tag" is very subjective: for instance, one person may consider the tag "ugly" on Aunt Thelma's photo to be inappropriate, while another person may think it is perfect! There are of course behaviors that most people would agree are inappropriate, but defining such behaviors precisely is not easy. Second, malicious users can mount many different "attacks" on a tagging system. For instance, if they know that correlations are being used to detect attacks, they can try to disguise their incorrect tags by posting some fraction of reasonable tags. What bad users do depends on their sophistication, on their goals, and on whether they collude. Bad users being a moving target, it is hard to know what they will do next. Third, tagging systems constantly evolve too, so taking a representative snapshot of a system at any time is not easy. Moreover, such snapshots do not naturally lend themselves to experimenting with all parameters that characterize a tagging system. For instance, it is hard to evaluate the effect of the number of postings contributed per user or to see how vulnerable are tags by modifying their popularity.

Our goal is to *study how spam affects tag-based search and retrieval in a tagging system.* Given the difficulties described above, we define an *ideal tagging system* where malicious tags and malicious user behaviors are well defined. In particular, we use two complementary techniques to generate scenarios:

—*Data Driven.* We use a real data set of documents and tags, and inject spam tags based on a bad user model.

—*Synthetic.* We generate documents and their tags based on data distributions, and then again inject spam tags.

The synthetic model lets us experiment with diverse forms of tag popularity and levels of user tagging, while the data-driven model lets us study an actual set of documents and tags under attack.

In each case we assume that malicious users use a particular, fixed strategy for their tagging, and that there are no ambiguous tags. We define a variety of malicious user strategies and we explore which are the most disruptive for different

query answering and moderating schemes. We perform a detailed evaluation of tag spam using different combinations of the models and the parameters involved. In this evaluation, the query answering and moderating schemes do not directly know which tags are misused, but when we evaluate them we will know which answers were correct and which were not.

Given that we are using an ideal model, our results cannot accurately predict how any one particular tagging system may perform but rather can yield insights into the relative merits or limitations of the protection schemes studied. Similarly, understanding the level of disruption in an ideal system may educate us on what may happen in a real system, where the distinction between an incorrect result and a correct one is less clear cut.

In summary, the contributions of this paper are:

—We define an ideal tagging system that combines legitimate and malicious tags. This model allows studying a range of user tagging behaviors, including the level of moderation and the extent of spam tags, and comparing different query answering and spam protection schemes (Section 3).

—We describe a variety of query schemes and moderator strategies to counter tag spam. Particularly, we introduce a social relevance ranking method for tag search results that takes into account how often a user's postings coincide with others' postings in order to determine their "reliability" (Sections 4 and 5).

—We define a metric for quantifying the "spam impact" on results (Section 6).

—We compare the various schemes under different models for malicious user behavior. We try to understand weaknesses of existing systems and the magnitude of the tag spam problem. We also make predictions about which schemes will be more useful and which malicious behaviors will be more disruptive in practice (Section 7).

2. RELATED WORK

We are witnessing a growing number of tagging services on the web, which enable people to share and tag different kinds of resources, such as: photos (Flickr), URLs (Del.icio.us), blogs (Technorati), people (Fringe [Farrell and Lau 2006]), research papers (CiteULike), slideshows (Slideshare), and so forth. 3spots is a web site providing links to several systems. Companies are also trying to take advantage of the social tagging phenomenon inside the enterprise [John and Seligmann 2006; Millen et al. 2005; Dogear].

The increasing popularity of tagging systems has motivated a number of studies [Xu et al. 2006; Sen et al. 2006; Golder and Huberman 2006; Kumar et al. 2006; Marlow et al. 2006; Brooks and Montanez 2006] that mainly focus on understanding tag usage and evolution. An experimental study of tag usage in My Web 2.0 has shown that people naturally select some popular and generic tags to label Web objects of interest [Xu et al. 2006]. In general, three factors seem to influence personal tagging behavior [Sen et al. 2006]: *people's personal tendency* to apply tags based on their past tagging behaviors, *community influence* of the tagging behavior of other members, and the *tag selection* algorithm used by the system for recommending "good" tags for a document to the candidate tagger. Community

influence has been shown in experimental studies of del.icio.us [Golder and Huberman 2006] and Flickr [Marlow et al. 2006]. Brooks and Montanez [2006] discuss what tasks are suitable for tags, how blog authors are using tags, and whether tags are effective as an information retrieval mechanism. In this paper, we take a step towards understanding the magnitude and implications of spamming in tagging systems. Although spamming is directly related to tag usage, existing studies have not explicitly dealt with it. In our earlier work [Koutrika et al. 2007], we have introduced a framework for modeling tagging systems and user tagging behavior and we performed a limited study on the performance of tag searches. In this paper, we extend this work and we perform a thorough study of the behavior of tagging systems based on refined user and system models. For this purpose, we use two datasets, a synthetic dataset and a real one containing postings from del.icio.us, and we study the consequences of injecting different attacks into them.

Harvesting social knowledge in a tagging system can lead to automatic suggestions of high quality tags for an object based on what other users use to tag this object (*tag recommendation*) [Xu et al. 2006; Mishne 2006; Ohkura et al. 2006], characterizing and identifying users or communities based on their expertise and interests (*user/ community identification*) [John and Seligmann 2006], building hierarchies of tags based on their use and correlations (*ontology induction*) [Schmitz 2006], and so forth. We argue that leveraging social knowledge may help fighting spam. The Coincidence-based query answering method, which we will describe in Section 4.2, exploits user correlations to that end. To the best of our knowledge, Xu et al. [2006] take into account spam by proposing a reputation score for each user based on the quality of the tags contributed by the user. In that work, reputation scores are used for identifying good candidate tags for a particular document, i.e., for automatic tag selection. This problem is somehow the inverse of ours, tag-based searching, i.e., finding good documents for a tag.

A tagging system is comprised of resources, users and tags. These elements have been studied independently in the past. *Link analysis* exploits the relationship between resources through links and is a well-researched area [Henzinger 2000]. Analysis of social ties and *social networks* is an established subfield of sociology [Wasserman and Faust 1994] and has received attention from physicists, computer scientists, economists, and other types of researchers. Recently, the aggregation and semantic aspects of tags have also been discussed [John and Seligmann 2006; Xu et al. 2006]. To what extent existing approaches may be carried over to tagging systems and, in particular, help tackle tag spam is an open question. For instance, link analysis has been suggested to help fight web spam [Gyöngyi et al. 2004; Wu et al. 2006] by identifying trusted resources and propagating trust to resources that are linked from trusted resources. An alternative way is to identify spam pages [Gyöngyi et al. 2006]. However, in a tagging system, documents are explicitly connected to people rather than other documents. Moreover, due to this association, tags have the potential to be both more comprehensive and more accurate than anchor-text based methods. Alternatively, tagging systems could utilize the information and trust in the social network, as in [Guha et al. 2004]. Again, they may need to consider the links from users to resources to reason about the importance and trust of users and resources and make the system more resilient to spam.

3. TAGGING FRAMEWORK

3.1 System Model

A tagging system is made up of a set $\mathcal{D}$ of documents (e.g., photos, web pages, etc), which comprise the *system resources*, a set $\mathcal{T}$ of available tags, which constitute the system *vocabulary*, a set $\mathcal{U}$ of users, who participate in the system by assigning tags to documents, and a *posting relation* $\mathcal{P}$, which keeps the associations between tags, documents and users. We call the action of adding one tag to a document a *posting*. Given our goals, we do not need to know the content of the documents nor the text associated with each tag. All we need is the association between tags, documents and users. Therefore, all entities, i.e., documents, tags and users, are just identifiers. We use the symbols d, t and u to denote a document in $\mathcal{D}$, a tag in $\mathcal{T}$ and a user in $\mathcal{U}$, respectively. We consider that a posting is a tuple $[u, d, t]$ in $\mathcal{P}$ that shows that user u assigned tag t to document d. Note that we have no notion of when documents were tagged, or in what order. Such information could be useful, but is not considered in this paper.

To capture the notion that users have limited resources, we introduce the concept of a *tag budget*, i.e., a limit on how many postings a user can add. Bad users may intentionally invest a specific amount of effort that could be different or beyond the effort of ordinary, good, users, therefore, we assume that good users have a tag budget p_g and bad users have a tag budget p_b. For simplicity, we assume that any given user makes exactly p_g (or p_b) postings.

Each document $d \in \mathcal{D}$ has a set $\mathcal{S}(d) \subseteq \mathcal{T}$ of tags that correctly describe it. For example, for a photo of a dog, "dog", "puppy", "cute" may be the correct tags, so they belong to the set $\mathcal{S}(d)$. All other tags (e.g., "cat", "train") are incorrect and are not in $\mathcal{S}(d)$. We do not consider the quality of tags. For instance, for a photo of Mont Blanc, the tag 'Mont Blanc" may be a high quality tag whereas "mountain" may be a lower quality tag, but they are both correct and therefore belong to the set $\mathcal{S}(d)$ for this photo. Any tag from $\mathcal{T}$-$\mathcal{S}(d)$ is a spam tag for document d.

We are using strings like "dog" and "mountain" in the examples above, but we are not interpreting the strings; they are just tag identifiers for us.

3.2 Basic Tagging Model

To populate a particular instance of a tagging system, we need to: (i) populate the $\mathcal{S}(d)$ sets and (ii) generate the actual posting relation $\mathcal{P}$. For the latter, we define a good user and a malicious user model (or generator) that simulates tagging behavior. We assume that there is a clear distinction between malicious and good users and that both good and malicious users use a particular, fixed strategy for tagging. That is, we consider good users in set $\mathcal{G}$ and bad (malicious) users in set $\mathcal{B}$, such that $\mathcal{U} = \mathcal{G} \cup \mathcal{B}$ and $\mathcal{G} \cap \mathcal{B} = \oslash$.

To create what we call an *instance* of a tagging system, we use two types of generating models:

—*Data-Driven.* We use a set of documents and their tags from a popular tagging site (details in Section 7). The documents and the tags used constitute our sets $\mathcal{D}$ and $\mathcal{T}$. We assume that all recorded tags are good, so for each document d, its $\mathcal{S}(d)$ set contains all the tags actually used.

—*Synthetic.* We generate the $\mathcal{S}(d)$ sets and the actual postings by good users using data distributions. For instance, for each document d we can select s tags (from the set $\mathcal{T}$ of available tags) at random to populate $\mathcal{S}(d)$. To generate the actual good user postings, we can use the following generating model:

Random Good-User Model:
  for each user $u \in \mathcal{G}$ do
    for each posting $j = 1$ to p_g do
      select at random a document d from $\mathcal{D}$;
      select at random a tag t from $\mathcal{S}(d)$;
      record the posting: user u tags d with t.

Note that in Sections 3.3 and 3.4 we present other synthetic generating models that rely on other distributions.

To add spam tags to the system, we rely on a bad user model to generate spam postings. For example, the following is a random generating model of bad user postings:

Random Bad-User Model:
  for each user $u \in \mathcal{B}$ do
    for each posting $j = 1$ to p_b do
      select at random a document d from $\mathcal{D}$;
      select at random an incorrect tag t from $\mathcal{T} - \mathcal{S}(d)$;
      record the posting: user u tags d with t.

The random bad user model assumes that each user acts independently, that is, the bad users are "lousy taggers" but not malicious. However, in some cases malicious users may collude and mount more organized attacks. We consider a particular form of targeted attack behavior assuming that a set of users attacks a particular document d_a with some probability r. This model is defined as follows.

Targeted Attack Model:
  select a particular document d_a from $\mathcal{D}$;
  select a particular incorrect tag t_a from $\mathcal{T} - \mathcal{S}(d_a)$;
  for each user $u \in \mathcal{B}$ do
    for each posting $j = 1$ to p_b do
      with probability r record the posting:
        user u tags d_a with t_a;
      else:
        select at random a document d from $\mathcal{D}$;
        select at random an incorrect tag t from $\mathcal{T} - \mathcal{S}(d)$;
        record the posting: user u tags d with t.

Observe that for $r = 0$, the targeted attack model coincides with the random bad user model. Also note that both good and bad users may submit duplicate tags: Even if document d already has tag t, a user can tag d with t (and even if the first t tag was added by the same user).

Fig. 1. An example of a targeted attack.

Example. A tag cloud is a visual representation of the tags that are used in a tagging site, where popular tags are depicted in a larger font or otherwise emphasized. Figure 1 depicts a screenshot of Amazon's tag cloud. One of the largest tags shown is "Defectivebydesign". Browsing the results for this tag reveals that they refer to the same product that has been attacked by a group of people. This is an example of a targeted attack.

One can extend this basic tagging model we have presented in two ways: (a) by changing the synthetic generating model to use other distributions and to vary the number of good tags per document or postings per good user; and (b) by changing the bad user model. Note that the data-driven model cannot be modified, since its behavior is fixed by the input data. Thus, in the following subsections, when we refer to good user behavior, we are focusing on the synthetic generating model.

3.3 Skewed Tag Budget Distribution

In a tagging system, (normal) users display different levels of activity (see also Section 7). Typically, most of them contribute a small number of postings, and only a few are very active. In order to capture that behavior, we consider two types of good users: very active and less active. In particular, there is a set $\mathcal{G}' \subseteq \mathcal{G}$ of very active users with a tag budget p'_g, while the rest of them have a tag budget $p_g < p'_g$. Based on that, we define the following 2-level-activity random good user model.

2-Level-Activity Random Good-User Model:
  for each user $u \in \mathcal{G}'$ do
    for each posting $j = 1$ to p'_g do
      select at random a document d from $\mathcal{D}$;
      select at random a tag t from $\mathcal{S}(d)$;
      record the posting: user u tags d with t.
  for each user $u \in \mathcal{G} - \mathcal{G}'$ do
    for each posting $j = 1$ to p_g do
      select at random a document d from $\mathcal{D}$;
      select at random a tag t from $\mathcal{S}(d)$;
      record the posting: user u tags d with t.

We can define a similar model for bad users. Since in this paper we focus on the impact of malicious attacks on tagging systems, we will assume a constant tag budget for bad users. In reality, a bad user could sign in the system using different usernames and post a different number of postings from each account in order to make his presence in the system less easily detectable. However, this is not significant for the purposes of our study.

3.4 Skewed Tag Distribution

People naturally select some popular and generic tags to label web objects of interest [Xu et al. 2006]. For example, the word "dog" is more likely to be used as a tag than "canine", even though they may be both appropriate. In a tagging system, popular and less frequent tags co-exist peacefully. Therefore, we consider that there is a set $\mathcal{A} \subseteq \mathcal{T}$ of popular tags. In particular, we assume that popular tags may occur in the postings m times more often than unpopular ones. However, when we generate the appropriate $\mathcal{S}(d)$ set for a document d, we disregard popularity, because an unpopular tag like "canine" has the same likelihood to be relevant to a document as a popular tag like "dog". So, members of each $\mathcal{S}(d)$ are chosen *randomly* from $\mathcal{T}$.

A *Biased Good User* selects a correct tag for a document d taking into account tag popularity. For instance, for a cat photo, the set of correct tags may be $\mathcal{S}(d)$={"cat", "$feline$"}, with "cat" being more popular than "feline". Thus, "cat" is more likely to be selected for a posting. This good user model is defined as follows:

Biased Good-User Model:
  for each user $u \in \mathcal{G}$ do
    for each posting $j = 1$ to p_g do
      select at random a document d from $\mathcal{D}$;
      select a tag t from $\mathcal{S}(d)$ with probability proportional to tag popularity;
      record the posting: user u tags d with t.

Then, for bad users, we consider three models that capture different ways bad users may try to exploit or influence tag popularity depending on their goals.

Extremely Biased Bad Users use only popular tags for the wrong documents. Their objective is to ride the popularity wave and achieve high visibility. For instance, in a particular tagging system, the tag "travel" may be very popular. This means that this tag will also appear in tag searches often and it will possibly be visible in the tag cloud. Hence, using this tag to label one's documents will make them more "viewable." Moreover, search engines, such as Google, index the pages that contain the results for some tag searches from popular tagging sites. These pages often appear high in the search engine results, offering high visibility to the postings contained in them. Popular tags that are often misused include company and product names, celebrity names, or any other tag that may be currently popular among taggers in a social system. This tagging behavior based on misusing popular tags is captured by the following model.

Extremely Biased Bad-User Model (The Exploiter):
  for each user $u \in \mathcal{B}$ do

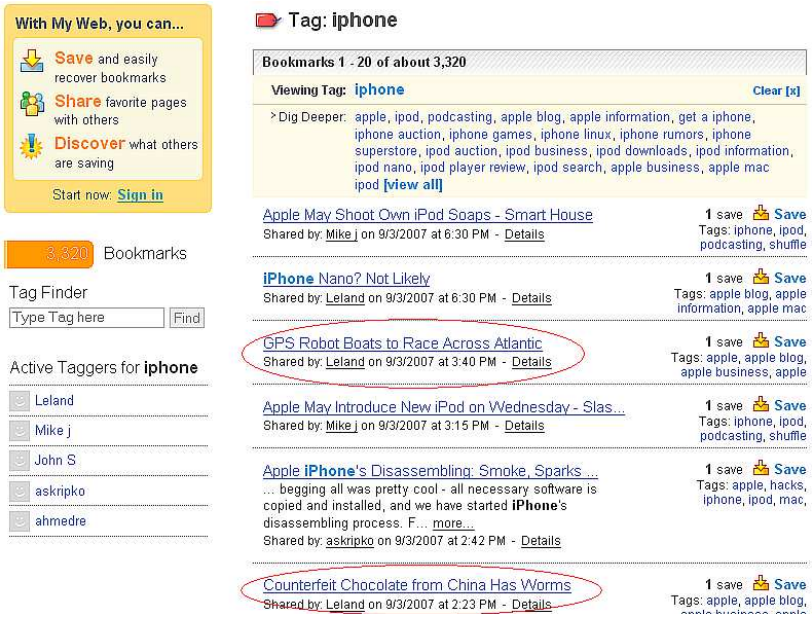


Fig. 2. An example of a popular tag attack.

for each posting $j = 1$ to p_b do
  select at random a document d from $\mathcal{D}$;
  select a *popular* tag t from $\mathcal{A} - \mathcal{S}(d)$;
  record the posting: user u tags d with t.

Example. Figure 2 depicts the first page of results for the tag "iphone" taken from MyWeb's site. This tag has been very popular among users in this system making a tempting target for bad users. This figure shows two cases, circled, of tag misuse. Clicking on the first one leads to the site depicted in Figure 3, which is not related to iphones. One can also observe that this site has been assigned many other popular but irrelevant tags.

Outlier Bad Users use tags that are not very popular among good users to label unrelated documents. Their objective typically is to gain a steady number of views for their postings. A posting that is tagged with a popular tag may achieve many views for a period of time because it will appear in a high position in the search results for this tag. But soon newer related posts will take the high, more visible, positions in the results. On the other hand, a post assigned an unpopular tag will be possibly viewed by fewer users but it may stay at the top of the tag's results for a longer period of time. Furthermore, by choosing an unpopular tag, a bad user's postings may more easily dominate the results for this tag. Then there is an increased probability that when another user views the results for this tag, he will select one of the "bad" postings. This model is defined as follows.

Outlier Bad-User Model (The Atypical):

Small **ROBOTIC BOATS** from all by the world are set to race each other next year across the Atlantic, some 4,000 miles from Brittany, France, to the Caribbean. The race, called Microtransat 2008, was conceived by a computer scientist at the University of Wales, Aberystwyth. Entry boats must be "fully autonomous" (they can use GPS), self-sufficient in terms of energy (via solar panels) and no longer than 13 feet.

Original post by *Mike*

everything ipod ipod photos ipod link iphone hack ipod work iphone news ipod video 60 gig iphone price drop iphone business iphone rumors buy ipod iphone linux itunes podcasting iphone search iphone link iphone user ipod bose mac ipod auction apple the apple store ipod ibook iphone ipod nano ipod stories songs apple reveal apple information apple work ipod update apple company iphone deals itunes download apple superstore apple search ipod player review iphone software apple mac ipod iphone information macworld ipod downloads itunes search ipod generation ipod stereo iphone games iphone wifi iphone blog ipod information

Fig. 3. An example of a misleading posting.

for each user $u \in \mathcal{B}$ do
  for each posting $j = 1$ to p_b do
    select at random a document d from $\mathcal{D}$;
    select an *unpopular* tag t from $\mathcal{T} - \mathcal{A} - \mathcal{S}(d)$;
    record the posting: user u tags d with t.

Example. Figure 4 depicts the first page of results for an uncommon tag, "crabs std", taken from MyWeb 's site. We observe that this page contains postings by a single user, all of them containing links to the same site.

Biased Bad Users use more often popular tags and less frequently unpopular ones for mislabeling documents. By using a mix of tags, their objective is to combine the best of both worlds and also try to disguise themselves by acting like normal users, which may use a variety of tags depending on their interests. This bad user

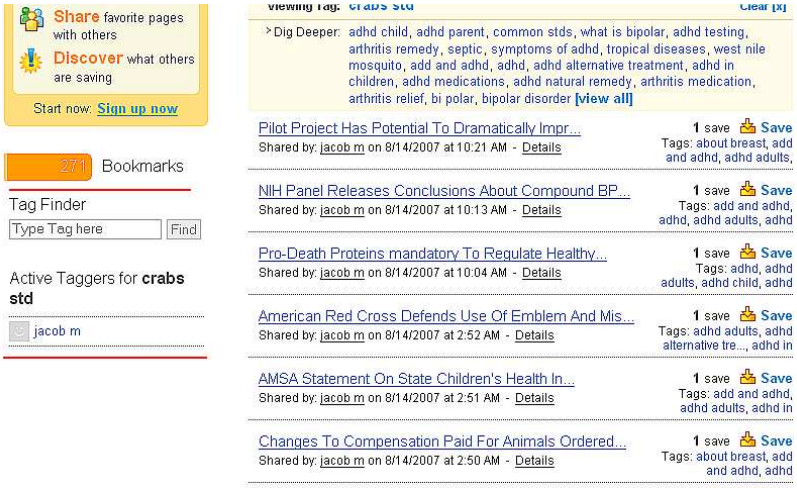


Fig. 4. An example of an unpopular tag attack.

model is defined below.

Biased Bad-User Model (The Imitator):
  for each user $u \in \mathcal{B}$ do
    for each posting $j = 1$ to p_b do
      select at random a document d from $\mathcal{D}$;
      select a tag t from $\mathcal{T} - \mathcal{S}(d)$ with probability proportional to tag popularity;
      record the posting: user u tags d with t.

4. TAG SEARCH

In a tagging system, users may be able to *query* for resources that are annotated with a particular tag. Given a query containing a single tag t, the system returns documents associated with this tag. We are interested in the top $\mathcal{K}$ documents returned, i.e., documents contained in the first result pages, which are those typically examined by searchers. So, although all search algorithms can return more than $\mathcal{K}$ results, for the purposes of our study, we consider that they generate only the top $\mathcal{K}$ results.

4.1 Existing Search Models

The most commonly used query answering schemes are the Boolean (e.g., Slideshare) and the Occurrence-based (e.g., Rawsugar). In Boolean searches, the query results contain $\mathcal{K}$ documents randomly selected among those associated with the query tag, i.e.,:

Boolean Search:
  return random $\mathcal{K}$ documents assigned t in $\mathcal{P}$.

The Boolean search model also provides a measure of comparison for other search schemes. That is, we can compare the other schemes with a baseline that selects matching documents at random.

In occurrence-based searches, the system ranks each document based on the number of postings that associate the document to the query tag and returns the top ranked documents. This search model is described as follows:

Occurrence-Based Search:
rank documents by decreasing number of postings in $\mathcal{P}$ that contain t;
return top $\mathcal{K}$ documents.

We have also experimented with variants of this ranking model, such as ordering documents based on the number of a tag's occurrences in a document's postings divided by the total number of this document's postings, i.e., based on tag frequency (e.g., Diigo). In this paper, we consider only the basic occurrence-based ranking scheme, since our experiments have shown that variants of this model exhibit a similar behavior with respect to spamming.

4.2 Coincidences

Common search techniques in tagging systems do not take into account spamming. In particular, in Boolean search, a document that has been maliciously assigned a specific tag may be easily included in the results for this tag. The underlying principle of occurrence-based search is that a document is relevant to a tag depending on the number of postings that claim so. Although this seems to be quite reasonable and to make searches more "spam-proof", bad users may still easily promote their documents to the top results as the following example shows.

Example. Consider the following postings:

user	document	tag
1	d_1	a
2	d_1	a
3	d_1	b
4	d_1	b
5	d_1	b
3	d_2	a
3	d_2	c
4	d_2	c

We assume that correct tags for document d_1 and d_2 belong to the sets $\{b, c\}$ and $\{a, c\}$, respectively. Different users may assign the same tag to the same document. For instance, users 3, 4 and 5 have all assigned tag b to document d_1. Since we use a small number of documents and postings in order to keep the example compact, let's assume that the system returns the top $\mathcal{K}$=1 document for a query tag. Users 1 and 2 are malicious, since tag a is not a correct tag for d_1. The system does not know this information. Therefore, for tag a, based on occurrences, it will erroneously return d_1.

The example above shows that the raw number of postings made by users in a tagging system is not a safe indication of a document's relevance to a tag. Postings'

reliability is also important. We observe that user 3's posting that associates d_2 with tag a seems more trustable than postings made by users 1 and 2, because that user's postings are generally in accordance with other people's postings: the user agrees with user 4 in associating d_2 with tag c and with users 4 and 5 in associating d_1 with b.

Based on the above intuition, we propose an approach to tag search that takes into account not only the number of postings that associate a document with a tag but also the "reliability" of taggers that made these postings.

In order to measure the reliability of a user, we define the *coincidence factor* $c(u)$ of a user u as follows:

$$c(u) = \sum_{d,t:\exists \mathcal{P}(u,d,t)} \sum_{\substack{u_i \in \mathcal{U} \\ u_i \neq u}} | \mathcal{P}(u_i, d, t) | \tag{1}$$

where $\mathcal{P}(u_i, d, t)$ represents the set of postings by user u_i that associate d with t.

Example (cont'ed). The coincidence factors for the users of this example are:

$$c(1) = 1, c(2) = 1, c(3) = 3, c(4) = 3 \text{ and } c(5) = 2.$$

The coincidence factor $c(u)$ shows how often u's postings coincide with other users' postings. If $c(u)$=0, then u never agrees with other people in assigning tags to documents. Our hypotheses is that the coincidence factor is an indication of how "reliable" a tagger is. A high factor signifies that a user agrees with other taggers to a great extent; thus, the user's postings are more "reliable." The lower $c(u)$ is, the less safe this user's postings become.

Given a query tag t, coincidence factors can be taken into account for ranking documents returned for a specific query tag. The score of a document d with respect to t is computed as follows:

$$score(d, t) = \frac{\sum_{u \in users(d,t)} c(u)}{c_o} \tag{2}$$

where $users(d, t)$ is the set of users that have assigned t to d and c_o is the sum of coincidence measures of all users. The latter is used for normalization purposes so that a score ranges from 0 to 1.

Example (cont'ed). The sum of coincidence measures of all users is $c_o = c(1) + c(2) + c(3) + c(4) + c(5) = 10$. Then, the document scores with respect to each of the posted tags are:

$$\begin{aligned} score(d_1, a) &= (c(1) + c(2))/c_o = & 2/10 \\ score(d_1, b) &= (c(3) + c(4) + c(5))/c_o = & 8/10 \\ score(d_2, a) &= c(3)/c_o = & 3/10 \\ score(d_2, c) &= (c(3) + c(4))/c_o = & 6/10. \end{aligned}$$

In words, a document's importance with respect to a tag is reflected in the number and reliability of users that have associated t with d. A document's score is high if it is tagged with t by many reliable taggers. Documents assigned a tag by few less reliable users will be ranked low.

Example (cont'ed). For tag a, document d_2 gets the highest score, $score(d_2, a) = 3/10$ compared to $score(d_1, a) = 2/10$, and comprises the system answer.

5. TRUSTED MODERATOR

In order to reduce the impact of bad postings, a trusted moderator can periodically check user postings to see if they are "reasonable." This moderator is a person that can "conceptually" identify good and bad tags for any document in the collection. Search engine companies typically employ staff members who specialize in web spam detection, constantly scanning web pages in order to fight web spam [Gyöngyi et al. 2004]. Such spam detection processes could be used in tagging systems too. The moderator examines a fraction f of the documents in $\mathcal{D}$. For each incorrect posting found, the moderator could simply remove this posting. But she can go a step further and remove all postings contributed by the user that made the incorrect posting, on the assumption that this user is bad. The moderator function could be described as follows:

Trusted Moderator:
  let $\mathcal{D}_f \subseteq \mathcal{D}$ containing a fraction f of $\mathcal{D}$'s documents;
  for each document $d \in \mathcal{D}_f$ do
    for each incorrect posting [u, d, t]
      eliminate all entries [u, *, *].

6. SPAM FACTOR

We are interested in measuring the impact of tag spam on the result list. For this purpose, we define a metric called $SpamFactor(t)$ as follows. Given a query tag t, the system returns a ranked sequence $\mathcal{D}_\mathcal{K}$ of $\mathcal{K}$ documents ranked, i.e.,:

$$\mathcal{D}_\mathcal{K} = [d_1, d_2, ...d_\mathcal{K}], \text{ with } rank(d_{i-1}, t) \geq rank(d_i, t), \quad 2 \leq i \leq \mathcal{K}.$$

Then, $SpamFactor(t, \mathcal{K})$ for tag t in the results $\mathcal{D}_\mathcal{K}$ is given by the formula:

$$SpamFactor(t, \mathcal{K}) = \frac{\sum_{i=1}^{\mathcal{K}} w(d_i, t) * \frac{1}{i}}{\sum_{i=1}^{\mathcal{K}} \frac{1}{i}} \tag{3}$$

where $w(d_i, t) = \begin{cases} 1 & \text{if } d_i \text{ is a bad document for } t; \\ 0 & \text{if } d_i \text{ is a good document for } t. \end{cases}$

In what follows, we explain each part of the definition above. A document d_i is "bad" if it is included in the results for tag query t, but t is not a correct tag for d_i, i.e., $t \notin \mathcal{S}(d_i)$. SpamFactor measures the spam in the result list introduced by bad documents. This is captured by the factor $w(d_i, t)$ in the formula, which returns 1 if d_i is a bad document and 0 otherwise. SpamFactor is affected by both the number of bad documents and their position in the list. The higher the position of a bad document in the result list is, the higher the numerator in formula (3) is. The maximum numerator value is $1 + \frac{1}{2} + \ldots + \frac{1}{\mathcal{K}}$, and it is used as denominator in the calculation of SpamFactor in order to normalize values between 0 and 1. Higher SpamFactor represents greater spam in the results. In order to illustrate the significance of different SpamFactor values, we consider the following example.

Example. Consider the following postings:

user	document	tag
1	d_1	a
1	d_1	c
3	d_1	c
2	d_1	a
2	d_1	b
1	d_2	a
2	d_2	a
3	d_2	a
3	d_2	c
4	d_2	c
3	d_3	a
6	d_3	a
1	d_3	b
5	d_3	b
6	d_3	b
4	d_4	b
5	d_4	b
5	d_4	c
5	d_5	a
5	d_5	c
1	d_5	b

The sets of correct tags for the documents in this example are: $S(d_1) = \{a, b, c\}$, $S(d_2) = \{a, c, d\}$, $S(d_3) = \{a, c\}$, $S(d_4) = \{b\}$ and $S(d_5) = \{b\}$. We assume that the system returns the top $\mathcal{K} = 4$ documents for a query tag based on the number of occurrences. For query tag a, the system returns d_2, d_1, d_3 and d_5 in that order. Only d_5 is a bad document and it is at the bottom of the result list. So, for this tag, the spam introduced by malicious users is limited. This is indicated by the low value of SpamFactor, which is $SpamFactor(a, 4) = 0.12$. The results for query tag b are d_3, d_4, d_1 and d_5. In this case, d_3 is a bad document, and it is ranked first. This fact results in higher SpamFactor, i.e., $SpamFactor(b, 4) = 0.48$. Finally, for tag c, the system returns d_1, d_2, d_4 and d_5. There are two bad documents in the results, i.e., d_4 and d_5, but these are found at the bottom of the results. So, it is $SpamFactor(c, 4) = 0.28$, i.e., it is between $SpamFactor(a, 4)$ and $SpamFactor(b, 4)$.

7. EXPERIMENTS

7.1 Experimental Framework

We have developed a simulator in Java for evaluating the behavior of a tagging system under malicious attacks. For our experiments, we proceeded in three steps:

(1) For the first step, we used the data-driven generating model to build our $\mathcal{S}(d)$ sets and create good user postings. We used data collected from del.icio.us. This set contains $8,792,717$ postings from $10,000$ users over $380,923$ documents

Table I. Parameters in *RealS* and *SimS*

Symbol	*Description*	*Value*
$\|\mathcal{D}\|$	*number of documents in the system*	380,923
$\|\mathcal{T}\|$	*size of the system vocabulary*	319,387
$\|\mathcal{G}\|$	*number of good users in the system*	10,000
p'_g	*tag budget per active user*	7500
p_g	*tag budget per less active user*	743
$\|\mathcal{G}'\|$	*number of good active users in the system*	200
s	*(average) size of* $\mathcal{S}(d)$	12

using 319,387 tags. As mentioned in Section 3, we considered all users good, i.e., $|\mathcal{G}| = 10,000$, the documents in the set as the system resources, i.e., $|\mathcal{D}| = 380,923$, and the tags provided as the system vocabulary, i.e., $|\mathcal{T}| = 319,387$. Furthermore, for each document d, its $\mathcal{S}(d)$ set was populated with all distinct tags assigned to it in the postings. We use *RealS* to refer to this instance of the good posting relation $\mathcal{P}$ (and associated $\mathcal{S}$ sets.) Using *RealS* we consider one bad user model.

(2) Before using synthetic models extensively, in the second step we "calibrate" the accuracy of synthetic generation to data-driven generation. For this comparison we generate a synthetic instance, *SimS*, that approximates *RealS*. We considered $|\mathcal{D}| = 380,923$ documents, $|\mathcal{T}| = 319,387$ tags and $|\mathcal{G}| = 10,000$ good users, as in *RealS*. Furthermore, the total number of different tags provided per document in the real set is 4,802,782 (less than the number of postings because different users may give the same tag to the same document), which gives on average $4,802,782/380,923 \approx 12$ distinct tags per document. Hence, the size of each $\mathcal{S}(d)$ was $s = 12$ and its members were *randomly* chosen from $\mathcal{T}$. In order to generate the postings for the good users, we used the 2-level-activity good user model considering 2% of the users as very active with a tag budget $p'_g = 7500$ while the rest having a tag budget $p_g = 743$. Table I summarizes the parameters considered in *SimS* (and *RealS*).

(3) As we will see, our results show that a synthetic model can emulate a data-driven one fairly well. Thus, we are relatively confident that the synthetic model can be used to explore interesting scenarios that are not covered by the real data. Thus, in our third step we explore a wide variety of tagging systems, using different combinations of good and bad user tagging models and varying the parameters involved. We use *HypS* to refer to these hypothetical instances. Table II summarizes all parameters considered and their default values.

7.2 SpamFactor Variation

In our experimental analysis, we use SpamFactor to evaluate the spam impact taking into account not only the number but also the positions of bad documents in the results. Other metrics, such as precision and recall, could be also adapted but we have found that they do not provide additional insights to tag spam impact. In the results that follow we will encounter a variety of spam values. In order to get a sense of how these factors translate to the "desirability" of an answer, we illustrate in Figure 5 how SpamFactor changes depending on the number of bad documents and their positions in the results. In the figure we show SpamFactor for a result

Table II. Parameters in the hypothetic systems

Symbol	*Description*	*Value*
$\|\mathcal{D}\|$	*number of documents in the system*	$10,000$
$\|\mathcal{T}\|$	*size of the system vocabulary*	500
$\|\mathcal{U}\|$	*number of users in the system*	$1,000$
$\|\mathcal{B}\|$	*number of malicious users*	10%
p_g	*tag budget per good user*	10
p_b	*tag budget per bad user*	10
s	*size of* $\mathcal{S}(d)$	25
f	*fraction of documents checked by moderator*	5%
r	*probability of targeted attack*	0
$\|\mathcal{A}\|$	*number of popular tags*	0

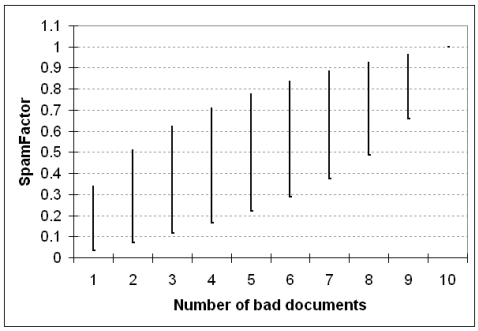


Fig. 5. Spam Factor Variation.

list of size 10 ($\mathcal{K} = 10$), which is the size we use in our experiments. For instance, having two documents at the top two positions in the results ($SpamFactor = 0.51$) is worse than having four documents in the last positions ($SpamFactor = 0.163$). Even for the same number of bad documents, SpamFactor depends on the document positions in the results. For instance, SpamFactor for results containing three bad documents ranges approximately from 0.11 to 0.62. SpamFactor ≤ 0.1 indicates that at most two bad documents exist at low positions in the results, which may be considered as tolerable spam. Greater values of SpamFactor indicate the existence of more bad documents in higher positions in the list. For instance, SpamFactor equal to 0.2 may correspond to up to 4 bad documents in the list, which means that almost half of the list is spammed. Therefore, SpamFactor values equal to or greater than 0.2 will be considered excessive in our experimental analysis. In other words, SpamFactor equal to 0.2 is considered as a threshold and whenever we observe such values, we consider that the system has been considerably affected by spam.

7.3 Evaluation Results

The objective of our experimental evaluation was to gain insights into the tag spam problem, to study the behavior of tagging systems under malicious attacks and to highlight promising directions for the design of appropriate countermeasures. In particular, we are interested in shedding light on the following issues:

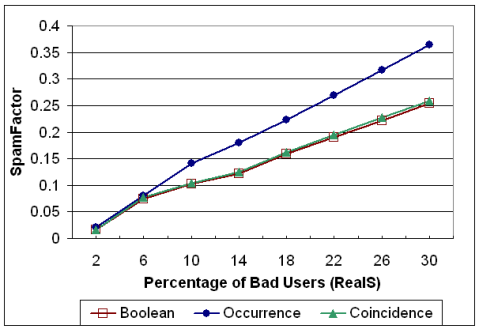


Fig. 6. The impact of bad users in a tagging system.

[Q1] How many bad apples (users) can spoil the batch?
[Q2] How much effort is required by bad users to affect the system?
[Q3] How useful is a synthetic model for studying tag spam?
[Q4] Can free vocabulary be an entrapment for tagging systems?
[Q5] Does leveraging social knowledge help in fighting spam?
[Q6] Are smaller players (sites) more susceptible to spam?
[Q7] Should users be encouraged to tag documents tagged by others?
[Q8] Does limiting the number of tags per user help in combating spam?
[Q9] What is the effort of a trusted moderator to block spammers?
[Q10] How effective can a trusted moderator be in blocking spammers?
[Q11] What happens when users collude?
[Q12] What may be the "Achilles' heel" of social countermeasures?
[Q13] How can malicious users exploit tag popularity?

We address these questions in turn.

[Q1] *How many bad apples (users) can spoil the batch*?

A vital question is how many malicious users can a tagging system tolerate before results significantly degrade. For this purpose, we start with a "clean" system, which has not been infected by spam, i.e., all users in the system are good. We use *RealS* and we then inject into it different attacks. For each attack, we consider a different number of participating bad users and we generate their postings using the random bad user model with tag budget p_b=600. Figure 6 shows SpamFactor as a function of the percentage of bad users over the good user population $(100 * |\mathcal{B}|/|\mathrm{G})|)$. SpamFactor grows linearly as more bad users enter the system because the number of bad postings increases linearly as well. For occurrence-based searches, SpamFactor exceeds the threshold 0.2 when the percentage of bad users in the system goes beyond 14%. Coincidence-based searches tolerate bad users better: SpamFactor exceeds the threshold when the percentage of bad users is 22%. However, coincidence-based searches do not perform any better than Boolean searches, meaning that social knowledge is not very useful in this case. Subsequent experiments will help us explain this phenomenon (see [Q4], [Q5]).

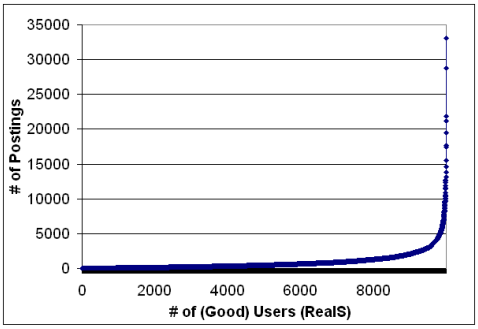
We point out that SpamFactor growing only linearly with the number of bad users is good news: bad users do not have a multiplicative effect. However, we have only examined the case of lousy taggers, and results may be significantly different when users mount more sophisticated attacks or when they collude (for instance, see discussion for the question [Q11].) Furthermore, we see that even a moderate number of these bad users can generate sufficient spam in tag search results. In practice, many users may accidentally assign incorrect tags (lousy or non-expert taggers), therefore unintentionally generating spam.

[Q2] *How much effort is required by bad users to affect the system*?

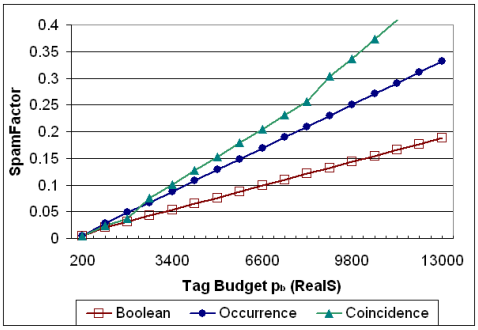
Figure 7(a) shows the cumulative tag budget distribution for good users in *RealS*. Note that most of them ($> 8,000$) have contributed fewer than 2,000 postings, and there are few users that are very active (with over $5,000$ postings.) If bad users behave like the very active good users, they can achieve an overall impact much larger than their number would imply. Figure 7(b) shows how much spam can be generated by a small set of bad users ($|\mathcal{B}|$ equal to 2% of $|\mathcal{G}|$) with an increasing tag budget p_b. SpamFactor exceeds our 0.2 threshold when each bad user provides more than $8,000$ postings.

One could possibly argue that as there is no actual limit on the number of postings a user can contribute, a single bad user could generate an unlimited number of postings in order to achieve his/her goals. In practice, such abnormal user activity is easily spotted and offending postings can be removed from the system. Bad users are less easily detectable when their activity does not go beyond good user activity. For instance, in those cases in Figure 7(b) that SpamFactor exceeds 0.2, there are over 300 "suspicious" (bad and good) users in the system based on their activity. A trusted moderator could examine them in order to find the actual bad users. However, a tagging system is constantly changing, new users, documents, and postings being added in large numbers. Bad users may create new accounts in the system when their old ones become inactive. Special programs (bots) are also used in order to upload a very large number of postings in a short time. The above make the moderator's job more difficult. A possible countermeasure could be to limit the number of postings a user is allowed to contribute within a period of time in order to reduce the frequency of malicious postings appearing in the system. Bot detection mechanisms can also be helpful (e.g., NoBot Control .)

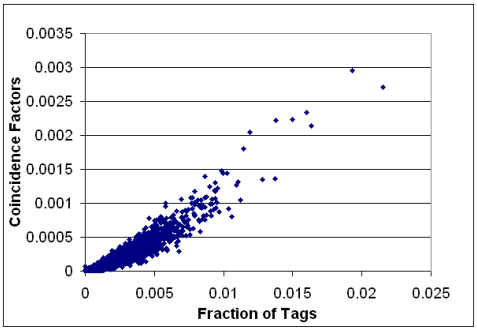
In Figure 7(b), when the budget p_b is greater than about $2,600$, coincidence-based SpamFactor is actually worse (larger) than occurrence-based SpamFactor. Coincidences do not work well because, as bad users contribute increasingly more postings, they agree more often with each other and hence they tend to reinforce each other's reliability. This is an example of how spam countermeasures, such as using coincidences, can be vulnerable to bad users. Very active good users have high coincidence factors too. This fact raises the question of how much influence these users have on tag search results. Figure 7(c) shows how coincidence factors correlate to tag usage. For instance, users with high coincidence factors (~ 0.003) use only a very small fraction of the tags that are available in the system (< 0.025). Hence, active users may influence the results for some tags but for most of the tags the opinions of other, less active, users will be taken into account.



(a) Tag budget distribution



(b) Bad user tag budget



(c) Tag coverage

Fig. 7. The effect of the bad users' tag budget in a tagging system.

[Q3] *How useful is a synthetic model for studying tag spam?*

A real dataset, such as *RealS*, has the potential to produce more realistic results. However, it is often difficult and time consuming to create such dataset. Moreover, the data may under-represent content due to sample bias or topic drift in the underlying collection. Tagging systems also tend to experience rapid growth, which may promote topic drift as the user base changes. Synthetic models, on the other hand, make it possible to study a wide range of current and future systems. However,

it is important to validate a synthetic model, e.g., in our case by checking that it produces results that are consistent with the data-driven model.

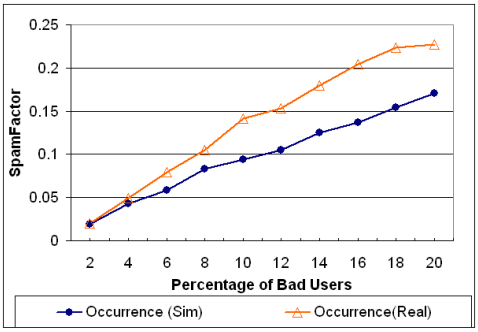
For our validation, we synthetically generate an instance *SimS* that is similar to *RealS*, and then compare the results. While the results are not identical, the differences are relatively small. More importantly, the *trends* observed with either the real or synthetic instance are the same. For example, Figures 8(a) and 8(b) show two of the many comparisons we performed. The figures present SpamFactor for occurrence-based tag searches using *RealS* and *SimS*, as a function of the percentage of bad users (relative to the good user population) and as a function of the bad users' tag budget, respectively. We observe that with *SimS* we can approximate quite well the behavior of the real instance *RealS*. For instance, Figure 8(a) shows that for 8% of bad users in the system, SpamFactor is around 0.83 in *SimS* and around 1.05 in *RealS*. SpamFactor measured in *SimS* is lower than in *RealS* because the former is generated using a random user model, which selects documents and tags in a random way thus creating postings that are uniformly distributed over tags and documents. In this way, there is sufficient good information for the tags in the system making them more tolerant to bad postings. On the other hand, in *RealS*, some tags are very popular, i.e., they can be found in many postings, but most of the tags are less frequently used. (We evaluate the consequences of tagging models that do not use a uniform tag distribution later in the paper in [Q13].) Less popular tags make an easier target for spammers because, in the absence of enough good postings, bad postings will surface in the results for these tags. The above also explain why the difference in SpamFactor between *RealS* and *SimS* increases with the percentage of bad users in the system and their tag budget increasing.

Based on the observations above, we are relatively confident that synthetic models can be used to study the behavior of tagging systems under different attacks. And as mentioned earlier, because of the flexibility that synthetic models provide, we need them to address the remaining questions on our list. Therefore, at this point we switch to synthetic generating models, *HypS*.

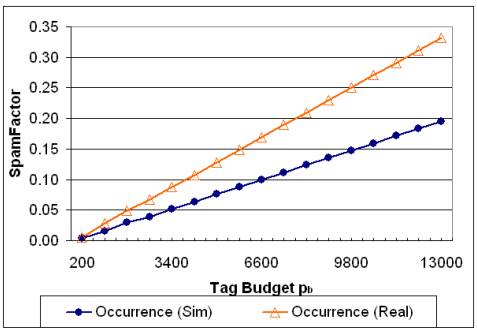
[Q4] *Can free vocabulary be an entrapment for tagging systems*?

A critical characteristic of tagging systems that promotes social navigation is their vocabulary, i.e., the set of tags used by members of the community. Instead of imposing controlled vocabularies or categories, tagging systems' vocabularies emerge organically from the tags chosen by individual members [Sen et al. 2006]. On the other hand, some systems prefer to impose their own categorization. For instance, in RealTravel, contributors place their content into the destination hierarchy.

Figure 9 shows SpamFactor as a function of the vocabulary size. As $|\mathcal{T}|$ increases and the overall tagging power of good users is constant, the number of correct documents per tag decreases, giving incorrect documents an opportunity to appear in the results. Also, the number of postings that associate a correct tag to a document drops, hence, incorrect postings may more easily prevail. These observations explain why occurrence- and coincidence-based tag search results are more spammed with growing vocabulary. (Coincidence-based results are more tolerant because coincidences take into account not only local information regarding the query tag, but also global information regarding the users that have made the postings.)



(a) The impact of the bad users



(b) The effect of bad users' tag budget

Fig. 8. Using synthetic data (*SimS*) vs. using real data (*RealS*).

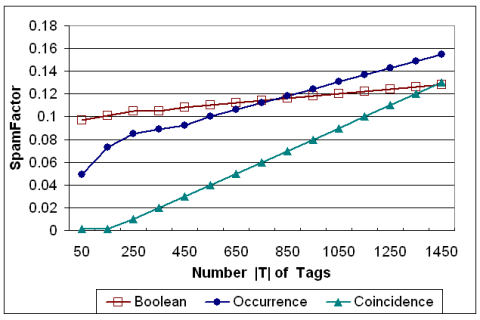


Fig. 9. The effect of free vocabulary.

Consequently, when there is a large number of tags in the system, very often tags will be associated with very few good documents. Then these tags can be more easily spammed: in the absence of enough good documents, bad documents will make it into search results. This is what happens in *RealS*, where the number of tags is almost equal to the number of documents. In such cases, social countermeasures may not be helpful, as Figures 6 and 7(b) revealed for coincidences. This effect is

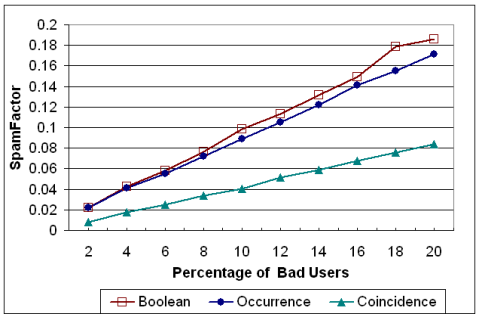


Fig. 10. The effect of bad users on systems with a smaller vocabulary.

not seen when the vocabulary size is relatively small with respect to the number of documents. For instance, Figure 10 shows SpamFactor in *HypS* as a function of the percentage of bad users in the system. In this case, the number of documents is much larger than the number of tags (10,000 documents vs. 500 tags.) Then, tag searches are more robust to attacks and combining social knowledge, such as coincidences, greatly helps fight spam.

The effect of a very large vocabulary may be reduced if more users enter the system, assuming that they do not enlarge significantly the vocabulary, and if the users contribute more postings. We investigate these two cases in [Q6]-[Q8].

[Q5] *Does leveraging social knowledge help in fighting spam?*

In a tagging system, users not only share their documents but they also share their expertise and personal opinions when tagging resources. Harvesting the collective wisdom of taggers can potentially help tag searches be less influenced by spammers and return results that are useful to the searcher. Using tag coincidences is an example of a social relevance ranking method.

Figure 10 demonstrates that using tag coincidences (for *HypS* with random bad and good user models) works substantially better than the other search schemes, cutting SpamFactor by a factor of two (see also [Q13] for a discussion on the performance of coincidences). The reason behind this improvement is that more informed decisions regarding which objects to select are possible when relying not only on object tags, but also on the number and the "reliability" of the users who have tagged. In effect, using coincidences, a greater number of postings is exploited for generating results for a tag than in the case of Boolean or occurrence-based searches.

Consequently, leveraging social knowledge can help fight spam. It is possible that not all social schemes work equally well for any kind of system but the particularities of each system should be evaluated first. For instance, the particular characteristics of *RealS* do not lend themselves to an advantageous use of social knowledge. Of course, when bad users proliferate all search schemes become gradually less effective. For instance, in the case of using coincidences, a high coincidence factor could actually correspond to a bad user. (See also question [Q12] ahead for a related discussion on the subject of relying on social knowledge.)

[Q6] *Are smaller players (sites) more susceptible to spam*?

Part of the power of tagging systems lies in their user base. This observation brings up the issue of how large the user base should be in order to have a positive impact on the system's performance. Are systems with many users better shielded from malevolent users? Does this mean that small players are in danger? For instance, Jots was totally overwhelmed by tag spam before starting to rebuild their site [EbiquityBlogger].

Although one would expect that a tagging system is in danger when bad users in the system proliferate or when they mount sophisticated attacks, Figure 11 reveals another situation in which the system may be also vulnerable. This figure measures SpamFactor as a function of the number of users in the system, ranging from 200 to 4200. The percentage of bad users is constant, therefore one might not expect significant variation in SpamFactor. However, for a "small" number of users, occurrence-based results become increasingly more affected by spam postings, similarly to Boolean results. The reason is that a small group of not very active users, i.e., having a relatively small tag budget, may contribute a small number of postings. Hence, the actual pool of good documents for a particular tag will be undersized and bad documents will surface in the results for this tag. Also, few, if any, duplicate postings exist in a small number of postings, so occurrence-based results are essentially based on random decisions. Moreover, due to the relatively "poor" collective knowledge in the system, search schemes such as the one based on tag coincidences also exhibit initially high SpamFactor.

As more users register in the system, a sufficiently large number of postings can be collected for helping search schemes identify good users and good content. We observe in Figure 11 that after a point ($|\mathcal{U}| = 600$), using occurrences, or even better coincidences, generates increasingly more spam-free results because a sufficient number of (re-occurring) postings is collected. A second interesting point in the figure is around $|\mathcal{U}| = 2400$, where SpamFactor returns to its starting value (~ 0.07), thus the effect of insufficient number of postings in the system is canceled. Given that the number of good users in the system is $|\mathcal{G}| = 90 * |\mathcal{U}|/100$, and their budget is $p_g = 10$, there are $21,600$ postings for $10,000$ documents, i.e., approximately 2 postings per document. Hence, in order to find this turning point, a rule of thumb is the following: $|\mathcal{G}| * p \geq 2 * |\mathcal{D}|$. Consequently, a tagging system requires a critical mass of users in order to cancel side-effects of sparse postings and fight spammers. This critical mass depends on the number of documents, and the number of good users vs. the bad users.

[Q7] *Should users be encouraged to tag documents tagged by others*?

Some systems allow for a multiplicity of tags for the same resource which may result in duplicate tags from different users (e.g., Del.icio.us). Alternatively, they may ask users to collectively tag an individual resource, thus denying any repetition (e.g., YouTube, Flickr). These two tagging models are referenced as the bag-model vs. the set-model for tag entry in [Marlow et al. 2006]. Furthermore, a tagging system can be restricted to "self-tagging", where users only tag the resources they created (e.g., Technorati) or allow free-for-all tagging, where any user can tag any resource (e.g., Yahoo! Podcasts).

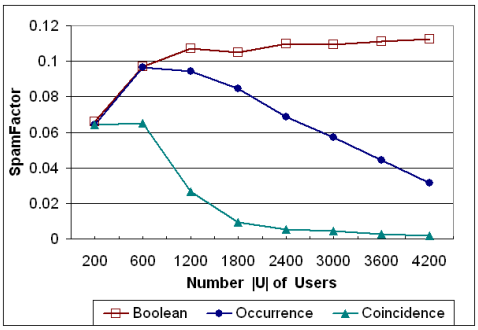


Fig. 11. The effect of the number of users in a tagging system.

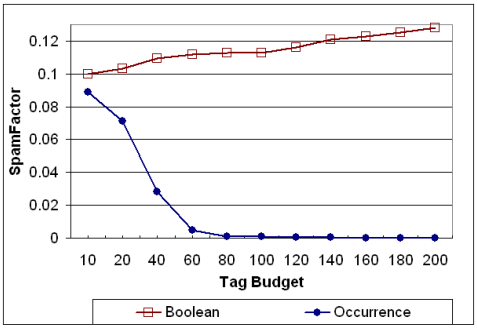
Figure 12(a) shows how occurrence-based tag search results are affected by varying the number of tags allowed per user (tag budget) from 2 to 500 considering the number of bad users in the system to be equal to 10% of the overall user population. We consider that all (good and bad) users have the same tag budget, i.e., p_g=p_b. Hence, this figure shows SpamFactor as both good and bad users provide more postings.

Note that our user models allow tags to be repeated. When users provide more postings in the system, duplicate good postings accumulate. Counting the number of tag occurrences in a document's postings in order to decide whether it will be included in the results leads to increasingly more spam-free results than relying only on single postings as in the case of Boolean results. Consequently, systems that follow a set-model (i.e., Boolean model) for tag entry may be more vulnerable to malicious users. This is especially true for self-tagging systems. For instance, when tags for a resource come from the user who has contributed this resource, as in the case of many photo or video sharing systems, then tag searches are more vulnerable to inappropriate content. On the other hand, systems that try to motivate users to tag resources that are already tagged by others can potentially learn more about the usefulness of these resources by aggregating user opinions regarding descriptive tags for the resources.

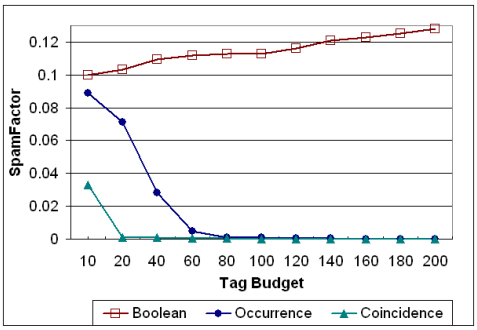
[Q8] *Does limiting the number of tags per user help in combating spam*?

A reasonable measure for preventing bad users from flooding the system with bad postings could be to apply a limit on the number of postings each user is allowed to contribute.

Figure 12(b) shows a counter-example: spam in coincidence-based tag search results is lower than in results generated by other types of searches as the number of tags allowed per user (tag budget) increases. As users contribute more tags, tag coincidences occur more often, thus reliable users can be identified more easily. Consequently, active good users are beneficial for tag searches because they can provide more evidence on the goodness of documents and help promote the good ones. Forcing a tag budget per user to limit the negative impact of malicious users also constrains the positive impact on the system due to good user activity (as long as there are plenty of good users.)



(a) The effect of multiple tag occurrences



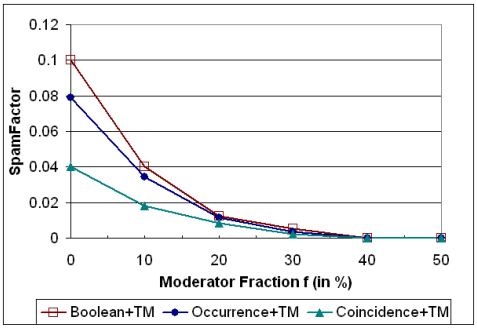
(b) The effect of unrestricted tag budget

Fig. 12. The effect of tag budget.

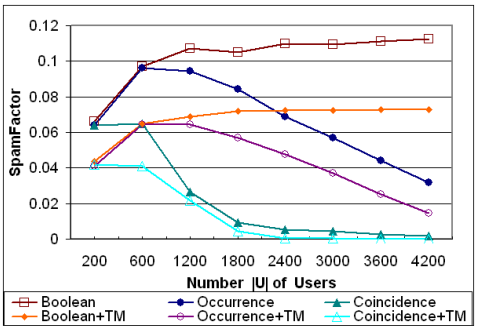
Combining the observations for questions [Q4], [Q6] and [Q8], we see that a tagging system can collect enough good knowledge about the association of tags with documents when there is a sufficient number of relatively active users w.r.t. to the number of the resources they tag and the size of their vocabulary is relatively small compared to the number of documents tagged. Then, the useful social knowledge accumulated in the system can be leveraged in order to provide spam-free tag searches.

[Q9] *What is the effort of a trusted moderator to block spammers*?

A trusted moderator can examine a fraction of the postings in order to identify malicious tags and malicious users. Figure 13(a) presents SpamFactor as a function of the percentage f of documents examined by the moderator, for f ranging from 1% to 50%. We observe that spam postings are completely removed from the system by scanning almost half the document collection, but this is impractical. A substantial reduction in spam is achieved after 10% of the documents have been examined. At this point, SpamFactor is around 0.04. Referring to Figure 5, this value indicates the existence of only one bad document in a low position in the results. The gain of having a moderator decreases as f grows. For example, increasing f from 10% to 20% achieves a slower reduction in SpamFactor than increasing f from 1% to 10%.



(a) Moderator effort



(b) Moderator effectiveness with the number of users

Fig. 13. Effort and effectiveness of a trusted moderator.

This means that the moderator needs to invest increasingly more effort in order to achieve a steady improvement on SpamFactor. In other words, as bad postings are eliminated from the system, it becomes more difficult to discover the remaining ones.

Consequently, a trusted moderator greatly helps reducing the spam in the system but it takes considerable effort in order to have a significant impact. Also, aiming at progressively more spam-free results means that the moderator needs to place increasingly more effort.

[Q10] *How effective can a trusted moderator be in blocking spammers*?

Figure 13(b) shows SpamFactor as a function of the number $|\mathcal{U}|$ of users in the system, when a trusted moderator examines $f = 5\%$ of the documents. For Boolean results, the moderator can cut SpamFactor almost by a factor of 2. This improvement does not change with the number of users in the system, because we consider the random bad generator model, which creates bad postings that are uniformly distributed over all documents. Thus, the number of bad postings coming from different users found in the same fraction of documents does not change significantly. On the other hand, the moderator's relative effectiveness for occurrence-based searches slowly decreases with $|\mathcal{U}|$. The reason is that, after a

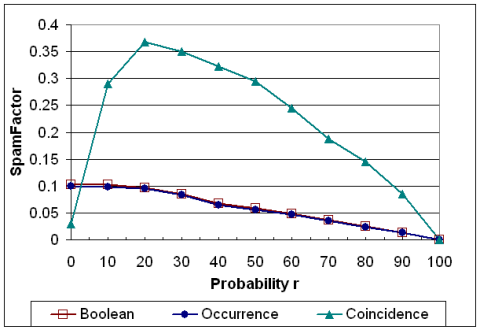


Fig. 14. The effect of targeted attacks.

certain point in the figure, unmoderated occurrence-based results greatly benefit from the increasing number of users in the system thus reducing the gap between the moderated and unmoderated curves. Overall, moderated coincidence-based searches return results that are the least affected by spam.

The moderator scheme we used in our evaluation may seem a bit harsh for adopting in a real-life setting, since it removes all tags created by a user that was found to provide bad postings. However, it is appropriate for our evaluation setting, which assumes a clear distinction between bad and good users. In a more realistic scenario, a moderator will remove tags only of users that are judged to be malicious, for instance based on their tag coincidences.

[Q11] *What happens when users collude*?

Users may act individually or they may collude. It is even possible for the same person to sign in the system assuming different personas in order to mount targeted attacks.

Figure 14 shows SpamFactor as a function of the probability r that bad users attack the same document (Targeted attack model). If $r = 0$, then we observe the random bad user tagging behavior, while $r = 1$ means that all users attack the same document. With r growing, targeted bad postings proliferate resulting in an amplified SpamFactor for the tag used in the targeted attacks. However, the number of bad postings for the rest of the documents and tags is reduced. Consequently, Boolean and occurrence-based SpamFactor decrease with r. Coincidence-based SpamFactor initially degrades fast with r, because coincidence factors of bad users are boosted, which means that all bad postings (apart from the targeted attack ones) are promoted in searches. However, as r increases, the number of different bad postings decreases, so the influence of bad users is restricted to fewer documents and tags. Therefore, Coincidence-based SpamFactor starts shrinking after a certain point.

Under targeted attacks, it may be easier for a moderator to locate spammed documents. For instance, the moderator may examine documents that have an unusually high number of tags, or postings by users with unusually high coincidence factors. We expect such a focused moderator approach to work very well in this scenario.

[Q12] *What may be the "Achilles' heel" of social countermeasures*?

We have seen that building social methods that leverage the collective wisdom of taggers seems promising for protecting tag searches from spammers. However, even these methods may have their weaknesses and spammers may find ways to abuse social knowledge.

Figure 14 reveals a potential downside of techniques that build on social knowledge. For instance, we see that while using coincidences was a good strategy with "lousy but not malicious" users, it is not such a good idea with colluding bad users. This is just an example of how anti-spam techniques may be manipulated by malicious users. Another case could be that a malicious user "camouflages" spam postings among a number of good postings that are copies of legitimate postings made by other users in order to boost his reliability. In fact, sophisticated anti-spam and evaluation strategies have a habit of breeding more sophisticated adversaries. In what follows, we will see another case of how malicious users may try to exploit tag usage in the system.

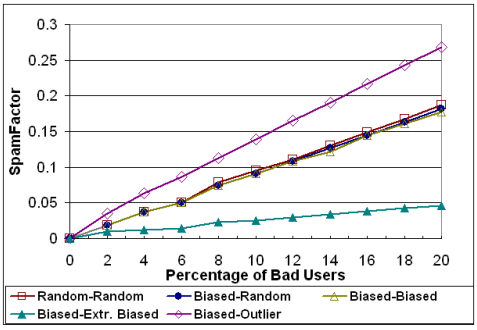
[Q13] *How can malicious users exploit tag popularity*?

In a tagging system, the existence of popular and unpopular tags may provide many opportunities for malicious users to misuse tags and spam tag searches. Taggers often use popular tags to drive traffic to their sites [Adlam 2006].

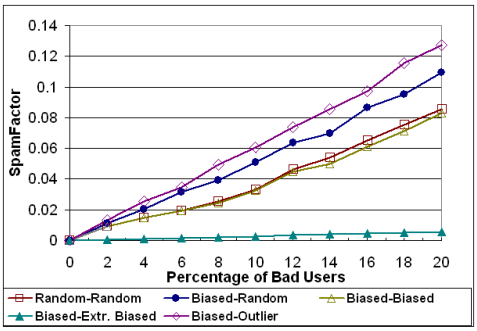
We considered all meaningful combinations of good/bad user models we have defined based on tag popularity (Section 3.4): (Good = *random*, Bad = *random*), (Good = *biased*, Bad = *random*), (Good = *biased*, Bad = *biased*), (Good = *biased*, Bad = *extremely biased*) and (Good = *biased*, Bad = *outlier*). We assume that popular tags may occur in the postings $m = 4$ times more often than unpopular ones.

Figure 15 shows the effect of varying the percentage of different types of bad users in the system ($100 * |\mathcal{B}|/|\mathrm{U})|$) on tag searches. We observe that unpopular tags may be more easily abused and they may be the worst source of spam. Given a biased good user model, misuse of unpopular tags can be performed by two types of bad users: the random and the outlier. On the other hand, tags that are popular among good users can be more spam resistant. In particular, bad users mimicking good users and using popular tags for their postings (i.e., the biased bad model) have a smaller impact on the system being as disruptive as the lousy taggers. Bad users that use only popular tags to mislabel content (the extremely biased user model) achieve the smallest distortion in the system. Comparing Figures 15(a) and 15(b), we also observe that, for any combination of user models, using coincidences cuts SpamFactor almost by a factor of 2. Revisiting question [Q5] about using social knowledge for fighting spam, we conclude that, in the presence of more elaborate spam efforts, leveraging social knowledge can still help tag searches.

Consequently, malicious users can exploit popular tags. Tags that are popular among good users seem more "protected" against spammers compared to the unpopular ones. There are many ways to fight attacks that exploit tag (un)popularity. Taking into account the collective usage of tags (for example by considering tag coincidences) is certainly one way. Another possible way is by estimating the relevance of a tag to a document in a posting based on content analysis.



(a) Occurrence-based searches



(b) Coincidence-based searches

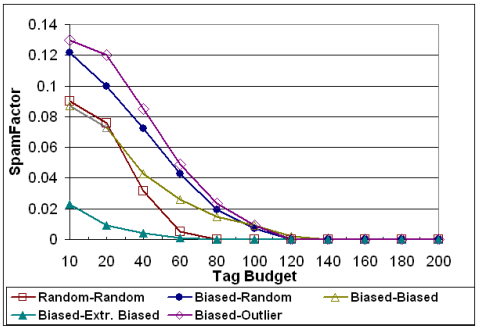
Fig. 15. The impact of bad users for different combinations of bad and good user models.

Figure 16 shows how tag search results are affected by varying the number of tags allowed per user (tag budget) from 2 to 500 for a bad user population that is 10% of the overall user population. We consider that all (bad and good) users have the same tag budget. We observe that SpamFactor shrinks in the presence of active good users. Particularly, relying on social knowledge, such as tag coincidences, makes searches more tolerant to spammers. These results in combination with the results shown in Figure 12 show that even in the presence of more elaborate spam efforts, leveraging social knowledge can help fight tag spam when there are very active good users.

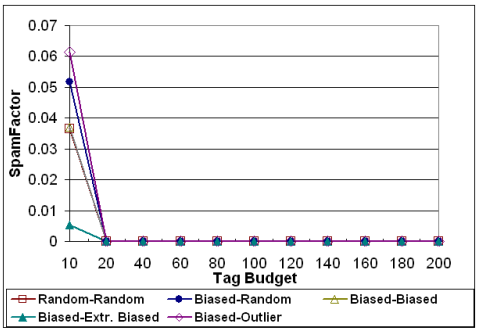
Finally, Figure 17 shows the effect of varying the percentage of popular tags $(100 * |\mathcal{A}|/|\mathrm{T})|)$ on tag searches. We observe that the impact of each bad user model on the system varies with the number of popular tags in the system.

8. CONCLUSIONS AND FUTURE WORK

Given the increasing popularity of tagging systems and the increasing danger from spam, we have proposed an ideal tagging system that combines legitimate and malicious tags. We have used two complementary techniques to generate scenarios: a data driven, where we use a real data set of documents and tags, and inject spam



(a) Occurrence-based searches



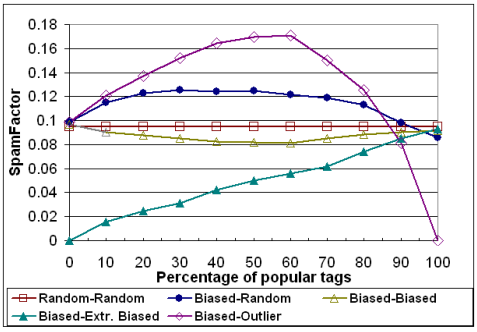
(b) Coincidence-based searches

Fig. 16. The effect of the tag budget for different combinations of bad and good user models.

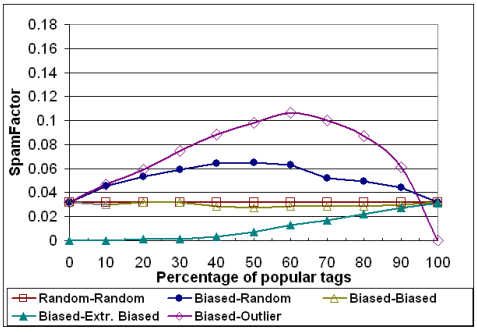
tags based on a bad user model, and a synthetic one, where we generate documents and their tags based on data distributions, and then again inject spam tags. We described and studied a variety of user tagging models as well as query schemes and moderator strategies to counter tag spam. Our objective was to gain insight into the tag spam problem, to examine important characteristics of tagging systems and to highlight promising directions for the design of appropriate countermeasures. For instance, we have seen that existing tagging systems, e.g., ones using the number of occurrences of a tag in a document's postings for answering tag queries, are threatened not only by malicious users but also by "lousy ones". In our evaluation, the synthetic model allowed us to experiment with diverse forms of tag popularity and levels of user tagging, while the data-driven model allowed us to study an actual set of documents and tags under attack.

We believe that the model we proposed helps one understand the dangers of tag spam and the effectiveness of counter-measures, and it can provide useful insights on how to wage the ongoing "spam wars." In particular, some useful "take-away messages" for designers include:

—Use of certain protective measures can be a double-edged sword. For instance, instituting a tag budget per user may limit the negative impact of malicious



(a) Occurrence-based searches



(b) Coincidence-based searches

Fig. 17. The effect of popular tags for different combinations of bad and good user models.

users but may also constrain the positive impact on the system due to good user activity. Therefore, it is very important for a tagging system, in terms of viability and good operation, to fight bad users without constraining the interaction of good users.

—Sophisticated spam measures may have their "Achilles' heel" soon to be discovered by spammers. In fact, sophisticated measures have a habit of breeding more sophisticated adversaries. A representative example is the case of social wisdom in a tagging system: although promising anti-spam measures can be built upon it, spammers will always try to exploit or influence it for their own purposes (as in the case of exploiting tag popularity).

This evaluation has a take-away message for users of tagging systems as well:

—A great fraction of the power and popularity of tagging systems lies with their users. Their interactions can shield to a great extent a system against spammers. Therefore, the more tags generated by more responsible users the better.

Based on our work, there are many avenues for future work. We plan to incorporate time in our models and study the evolution of spam attacks over time. We are also interested in schemes that would make a system more resistant to spam.

For instance, if tags are related, e.g., there is a tag hierarchy, can we devise smart algorithms that take into account tag relationships? If we track the time at which postings are made, can we better deal with spammers? For example, would it help to give more weight to recent tags as opposed to older tags? Also, if users can also use negative tags, e.g., this document is *not* about "cars", what would be the impact on searches?

9. ACKNOWLEDGEMENTS

We would like to thank Manuel Deschamps for taking part in the initial stages of the simulator development.

REFERENCES

3SPOTS. url: http://3spots.blogspot.com/2006/01/all-social-that-can-bookmark.html.

ADLAM, T. 2006. Tag and ping phenomenon. *url: http://www.optiniche.com/blog/174/tag-and-ping/*.

BROOKS, C. AND MONTANEZ, N. 2006. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th International Conference on World Wide Web*.

CITEULIKE. url: http://www.citeulike.org/.

CONTROL, N. *url: http://asp.net/ajax/control-toolkit/live/NoBot/NoBot.aspx*.

DEL.ICIO.US. url: http://del.icio.us/.

DIIGO. url: http://www.diigo.com/.

DOGEAR. url: http://domino.research.ibm.com/comm/research-projects.nsf/ pages/dogear.index.html.

EBIQUITYBLOGGER. url: http://ebiquity.umbc.edu/blogger/2007/01/24/tag-spam-on-the-rise.

FARRELL, S. AND LAU, T. 2006. Fringe contacts: People tagging for the enterprise. In *Proceedings of the Collaborative Web Tagging Workshop in conjunction with the 15th WWW Conference*.

FLICKR. url: http://www.flickr.com/.

GOLDER, S. AND HUBERMAN, B. A. 2006. Usage patterns of collaborative tagging systems. *Journal of Information Science 32,* 2, 198–208.

GUHA, R., KUMAR, R., RAGHAVAN, P., AND TOMKINS, A. 2004. Propagation of trust and distrust. In *Proceedings of the 13th WWW Conference*. 403–412.

GYÖNGYI, Z., BERKHIN, P., GARCIA-MOLINA, H., AND PEDERSEN, J. 2006. Link spam detection with mass estimation. In *Proceedings of the 32nd International Conference on Very Large Databases (VLDB)*. 439–450.

GYÖNGYI, Z. AND GARCIA-MOLINA, H. 2005. Web spam taxonomy. In *Proceedings of the 1st Intl. Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*. 39–47.

GYÖNGYI, Z., GARCIA-MOLINA, H., AND PEDERSEN, J. 2004. Combating spam with TrustRank. In *Proceedings of the 30th Intl. Conference on Very Large Databases (VLDB)*. 576–587.

HENZINGER, M. 2000. Link analysis in web information retrieval. *IEEE Data Engineering Bulletin 23,* 3, 3–8.

JOHN, A. AND SELIGMANN, D. 2006. Collaborative tagging and expertise in the enterprise. In *Proceedings of the Collaborative Web Tagging Workshop in conjunction with the 15th WWW Conference*.

JOTS. url: http://www.jots.com/.

KOUTRIKA, G., EFFENDI, F., GYÖNGYI, Z., HEYMANN, P., AND GARCIA-MOLINA, H. 2007. Combating spam in tagging systems. In *AIRWeb*.

KUMAR, R., NOVAK, J., AND TOMKINS, A. 2006. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*. 611–617.

Marlow, C., Naaman, M., Boyd, D., and Davis, M. 2006. Position paper, tagging, taxonomy, flickr, article, toread. In *Proceedings of the Hypertext Conference*. 31–40.

Millen, D., Feinberg, J., and Kerr, B. 2005. Social bookmarking in the enterprise. *Social Computing 3,* 9, 5–10.

Mishne, G. 2006. Autotag: collaborative approach to automated tag assignment for weblog posts. In *Proceedings of the 15th International Conference on World Wide Web*.

MyWeb. *url: http://myweb2.search.yahoo.com/*.

Ohkura, T., Kiyota, Y., and Nakagawa, H. 2006. Browsing system for weblog articles based on automated folksonomy. In *Proceedings of the WWW 2006 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, at WWW 2006*.

Rawsugar. url: http://rawsugar.com/.

RealTravel, X. url: http://realtravel.com/.

Schmitz, P. 2006. Inducing ontology from flickr tags. In *Proceedings of the Collaborative Web Tagging Workshop in conjunction with the 15th WWW Conference*.

Sen, S., Lam, S., Rashid, A., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., and Riedl, J. 2006. Tagging, communities, vocabulary, evolution. In *Proceedings of the CSCW'06*.

Slideshare. url: http://slideshare.net/.

Technorati. url: http://www.technorati.com/.

Wasserman, S. and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge.

Wu, B., Goel, V., and Davison, B. 2006. Topical trustrank: Using topicality to combact web spam. In *Proceedings of the 15th WWW Conference*. 63–72.

Xu, Z., Fu, Y., Mao, J., and Su, D. 2006. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative Web Tagging Workshop in conjunction with the 15th WWW Conference*.

YouTube. url: http://www.youtube.com/.