Combating Spam in Tagging Systems

Georgia Koutrika, Frans Adjie Effendi,
Zoltán Gyöngyi, Paul Heymann, Hector Garcia-Molina
Computer Science Department
Stanford University
{koutrika, franse}@stanford.edu,
{zoltan, heymann, hector}@cs.stanford.edu

ABSTRACT

Tagging systems allow users to interactively annotate a pool of shared resources using descriptive tags. 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. 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.3.3 [**Information Search and Retrieval**]: [Retrieval models, Search process]; H.3.5 [**Online Information Services**]: [Data sharing, Web-based services]

Keywords

Tag spam, tagging systems, social systems

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 [4], 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 [3], 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.

AIRWeb '07, May 8, 2007 Banff, Alberta, Canada.

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 [10]. Web spam is a big problem for search engines, as well as a big opportunity for "search engine optimization" companies that for a fee generate spam to boost the customer's pages. 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.

Given the increasing interest in tagging systems, and the increasing danger from spam, our goal is to understand the problem better 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? Is a tagging system with a large number of resources less susceptible to spam than a system with a limited number of resources? A popular system attracts more users and possibly more spammers. Is popularity a blessing or a curse for tagging systems? Revealing and understanding weaknesses of existing systems is a step towards designing more robust systems.

- What is the impact of encouraging users to tag documents already tagged? Does encouraging people to post more tags help a system better cope with spam? In other words, assume users are discouraged from tagging a document with a tag, if it already has that tag. Will things then be easier for spammers?

- What can be done to reduce the impact of malicious users? For example, one could 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? What would the moderator effort be in order to achieve a positive impact?

- Is there a way to use correlations to identify misused tags? For instance, 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 using correlations to detect 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 bad users will try next.

Given these difficulties, our approach here is to define an *ideal tagging system* where malicious tags and malicious user behaviors are well defined. In particular, we generate tags with a synthetic model, where some fraction of the tags are malicious, and there are no ambiguous tags. Based on this model, we study how tag spam affects tag-based search and retrieval of resources in a tagging system. Of course, our query answering algorithms will not directly know which tags are misused, but when we evaluate query answering schemes we will know which answers were correct and which were not. Similarly, we will assume that malicious users use a particular, fixed strategy for their tagging. Again, the protection schemes will be unaware of the malicious user policy. A proper understanding of tag spamming can guide the development of appropriate countermeasures. For this purpose, we start with simple user models and see how a system behaves under naive attacks and how it can protect itself against them. Once we introduce safeguards against these naive bad users, the bad users may respond with more sophisticated attacks. But these sophisticated attacks can only be defined once we know the safeguards.

Given that we are using an ideal model, our results will *not* be useful for predicting how any one particular tagging system may perform. Nevertheless, our results can yield insights into the relative merits of the various protection schemes we study. That is, if scheme A is significantly better than scheme B at protecting against tag spam in the ideal system, then it is reasonable to expect that system A will perform better in practice. Similarly, understanding the level of disruption malicious users can introduce in an ideal system, may provide insights into what they can do in a real system: That is, one can interpret the ideal results as an "upper bound" on disruption, since in a real system the distinction between an incorrect result and a correct one will be less clear cut.

In summary, the contributions of this paper are:

- We define an ideal tagging system that we believe is useful for comparing query answering schemes and we model user tagging behavior (Section 3).

- We propose a variety of query schemes and moderator strategies to counter tag spam (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 and we make predictions about which schemes will be more useful in practice (Section 7).

Due to space limitations, we are unable to include here all of our work. We refer the reader to our extended technical report [14] where additional references, examples, variations to our model, and experimental results can be found.

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 [4]), URLs (Del.icio.us [3]), people (Fringe [8]), research papers (CiteULike [2]), and so forth. Reference [1] provides links to several systems. Companies are also trying to take advantage of the social tagging phenomenon inside the enterprise [13]. The increasing popularity of tagging systems has motivated a number of studies [23, 20, 9, 15, 16, 7] that mainly focus on understanding tag usage and evolution. In this paper, we take a first 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. We believe that this fact underlines the importance and uniqueness of our study.

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* [23, 17, 18], characterizing and identifying users or communities based on their expertise and interests (*user/community identification* [13]), building hierarchies of tags based on their use and correlations (*ontology induction* [19]),

and so forth. We argue that leveraging social knowledge may help fighting spam. The Coincidence-based query answering method that we will describe in Section 4.2 exploits user correlations to that end. To the best of our knowledge, only reference [23] takes into account spam by proposing a reputation score for each user based on the quality of the tags contributed by the user. 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 relationships between resources through links and is a well-researched area [12]. Analysis of social ties and *social networks* is an established subfield of sociology [21] and has received attention from physicists, computer scientists, economists, and other types of researchers. The aggregation and semantic aspects of tags have also been discussed [13, 23]. 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 [11, 22] by identifying trusted resources and propagating trust to resources that are linked from trusted resources. 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.

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. For simplicity, we assume that any given user makes exactly p 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 are using strings like "dog" and "cat" in the example 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 postings of users. The members of each $\mathcal{S}(d)$ are *randomly* chosen from $\mathcal{T}$. In order to populate the posting relation, we need to define bad and good user tagging models to simulate user tagging behavior. For our purposes, 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 $\mathcal{G} \cap \mathcal{B} = \emptyset$. In the subsequent subsections, we define several models of good and bad taggers.

Assuming that users randomly pick documents (*uniform document distribution*) and tags (*uniform tag distribution*) for their postings, we define this random good user model:

Random Good-User Model:
  for each user $u \in \mathcal{G}$ do
    for each posting $j = 1$ to p 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.

Likewise, we define a random bad user model. The only difference from the above definition is that: Given a randomly selected document d, a *Random Bad-User* picks at random an incorrect tag t from $\mathcal{T} - \mathcal{S}(d)$.

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 colluding users attack 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 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). Some systems may disallow such duplicate tags. We have experimented with a no-duplicates-per-user policy but do not report the results here (the conclusions are not significantly different). Moreover, a person may sign in the system using different usernames and express a number of duplicate opinions.

One can extend this basic tagging model we have presented in many directions, e.g., changing the distributions that are used to select tags, queries, documents, and so on; or by introducing non-determinism in parameters such as

the tag budget or the size of the $\mathcal{S}(d)$ sets; or by defining additional good/bad user models. We have experimented with a number of these variations. Due to space constraints, here we discuss one interesting variation considering tag popularity.

3.3 Skewed Tag Distribution

People naturally select some popular and generic tags to label web objects of interest [23]. 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 per 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. Then, for bad users, we consider three different behaviors:

Biased Bad Users may try to disguise themselves by acting like normal users, i.e., using more often popular tags and less frequently unpopular ones, but for mislabeling documents.

Extremely Biased Bad Users use only popular tags for the wrong documents. 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. Then, these bad users may use this tag to label particular documents in order to make them more "viewable."

Outlier Bad Users use tags that are not very popular among good users. For instance, in a publications tagging system, these users may try to promote their pages selling particular products, so they may use tags such as "offer" or "buy", which are not popular among good users.

The definitions of these models can be found in [14].

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 [6]) and the Occurrence-based (e.g., Rawsugar [5]). In *Boolean* searches, the query results contain $\mathcal{K}$ documents randomly selected among those associated with the query tag. 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, i.e.,:

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. 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 Boolean search, a document that has been maliciously assigned a specific tag may be easily included in the results for this tag. The following example illustrates how occurrence-based search may be susceptible to spamming.

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, but the system does not know this information. For tag a, based on occurrences, the system 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:

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

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

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. Then, the rank of a document d with respect to t is computed as follows:

$$rank(d,t) = \frac{\sum_{\forall u \in users(d,t)} c(u)}{c_o} \quad (2)$$

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 rank ranges from 0 to 1. 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. d is ranked 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 rank, $rank(d_2, a) = 3/10$ compared to $rank(d_1, a) = 2/10$, 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 [11]. 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, i.e.,:

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

Table 1: Parameters used in Experiments

Symbol	Description	Value
$\lvert\mathcal{D}\rvert$	number of docs. in $\mathcal{D}$	10,000
$\lvert\mathcal{T}\rvert$	size of the vocabulary $\mathcal{T}$	500
$\lvert\mathcal{U}\rvert$	number of users in $\mathcal{U}$	1,000
$\lvert\mathcal{B}\rvert$	number of malicious users	10%
p	tag budget per user	10
s	size of $\mathcal{S}(d)$	25
f	frac. docs. checked by moderator	5%
$\mathcal{K}$	number of docs. in results	10
r	probability of targeted attack	0
$\lvert\mathcal{A}\rvert$	number of popular tags	0

Then, $SpamFactor(t)$ for tag t is given by the formula:

$$SpamFactor(t) = \frac{\sum_{\forall d_i \in D_K} w(d_i) * \frac{1}{i}}{H_K} \quad (3)$$

where

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

and H_K is the K^{th} harmonic number, i.e., it is the sum of the reciprocals of the first $\mathcal{K}$ natural numbers, i.e.,

$$H_K = \sum_{i \in [1..K]} \frac{1}{i} \quad (4)$$

A document d is "bad" if it is included in the results for tag query t, but t is not a correct tag for d, i.e., $t \notin \mathcal{S}(d)$. SpamFactor measures the spam in the result list introduced by bad documents. This is captured by the factor $w(d_i)$ 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. Higher SpamFactor represents greater spam in the results. The K^{th} harmonic number is used as denominator in the calculation of SpamFactor in order to normalize values between 0 and 1.

7. EXPERIMENTS

We have developed a simulator in Java that simulates the behavior of a tagging system based on the model described in this paper. We have conducted many experiments under several modifications of the parameters involved. Table 1 summarizes all parameters considered and their default values. Due to space constraints, here we only highlight some of our results. All experimental results along with detailed explanations can be found in [14].

7.1 Experimental Results

7.1.1 Random Attacks

For these experiments, we have used a set of $1,000$ tag queries that follow a uniform distribution. Figure 1 illustrates the effect of varying the number $|\mathcal{B}|$ of bad users in the system on Boolean, Occurrence-based and Coincidence-based tag searches. SpamFactor grows linearly, because the number of bad postings increases linearly with $|\mathcal{B}|$ while the number of good postings decreases. In [14], we argue that SpamFactor less than 0.1 is "tolerable" in the sense that the

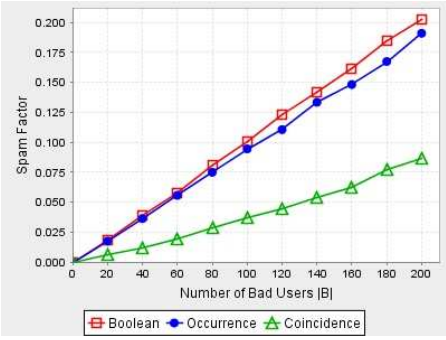


Figure 1: Impact of the number of bad users

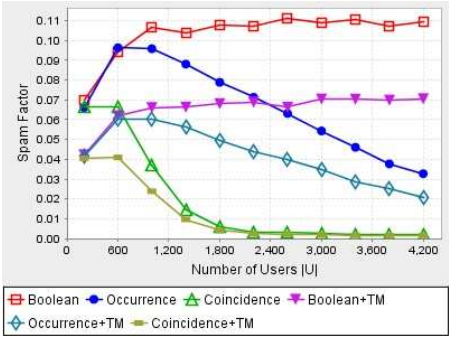


Figure 2: Impact of the number of users

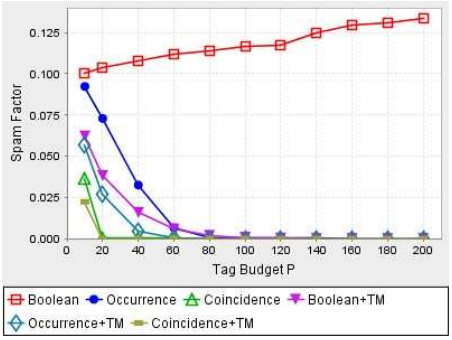


Figure 3: Impact of the tag budget

spam documents will be few and towards the bottom of the result list. Thus, looking at Figure 1, we conclude that for Boolean and Occurrence-based searches, a very small percentage of malicious users (e.g., $< 2\%$ of $|\mathcal{U}|$) with limited tagging power (p=10) as compared to the document collection size ($\mathcal{D}$=10,000) does not bias results significantly. Excessive SpamFactor is observed for growing bad user populations ($> 12\%$). This observation is critical because in practice many users may accidentally assign incorrect tags (lousy taggers), therefore unintentionally generating spam.

SpamFactor for Boolean results is higher because they are randomly selected from documents associated with a query tag, thus they may include more bad documents. Using coincidences works substantially better than using occurrences, cutting spam by a factor of 2. The reason behind this improvement is that coincidence factors are computed taking into account not only the postings that associate a document to a query tag, but also information about the users that have made these postings. Thus, they exploit a greater number of postings in order to generate results for a tag. This leads to more informed decisions regarding which documents to return and justifies low Coincidence-based SpamFactor. However, as bad users proliferate, effectiveness of this scheme also deteriorates. A high coincidence factor could actually correspond to a bad user. Still, using coincidence factors retains its factor-of-2 advantage.

A trusted moderator helps reduce spam in the system, but it may take a significant effort in order to have a positive impact. Figure 2 shows SpamFactor as a function of $|\mathcal{U}|$ 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 with $|\mathcal{U}|$ growing, bad postings 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 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, the best results are returned in moderated coincidence-based searches.

Note that the initial degradation of Boolean and Occurrence-based results shown in Figure 2 is due to sparse postings generated by the few users in the system. So, the pool of documents matching any tag is undersized, and bad documents also surface in the results. As $|\mathcal{U}|$ increases, so does the number of malicious postings, causing more bad documents to appear in the results. Once a sufficient number of postings has accumulated, Boolean results do not degrade any further, while Occurrence-based results improve with $|\mathcal{U}|$ due to re-occurring postings.

Another interesting outcome of the experiments is that the moderator's impact is not always the same on all search schemes. Figure 3 shows SpamFactor for moderated and unmoderated results as a function of tag budget, ranging from 2 to 500 for a moderate bad user population ($|\mathcal{B}| = 10\%$ of the overall user population). We first observe that unmoderated Boolean and Occurrence-based results are not affected in the same way: SpamFactor increases for the former and decreases for the latter. The reason is that when users provide more postings in the system, duplicate good postings accumulate, helping Occurrence-based searches to generate better results, while Boolean still make random decisions. Coincidence-based results are even better because the coincidence factors are computed by taking into account common postings over the whole collection, thus they are boosted as the tag budget grows. However, the intervention of a moderator has a dramatic impact on Boolean searches: with p growing, moderated SpamFactor improves and the gain from having a moderator grows. This effect is due to the fact that when users contribute more tags, once a moderator finds a bad posting, then a possibly larger number of bad postings is eliminated at once by removing the corresponding tagger.

Consequently, using a moderator can have a different impact depending on the underlying search scheme used. Over-

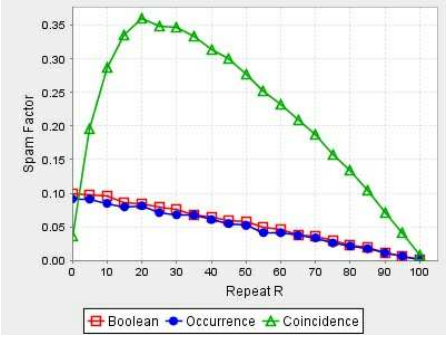


Figure 4: Impact of targeted attacks

all, a system that supports coincidences-based search backed up by a moderator is more tolerant to spamming.

7.1.2 Targeted Attacks

In this subsection, we study the effect of colluding users. Figure 4 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 decreasing after a certain point.

Consequently, under targeted attacks, there is little one can do to protect searches for the attacked tag, but all other searches actually fare better. Moreover, 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. However, with focused 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.

7.1.3 Attacks Based on Tag Popularity

In this subsection, we study how vulnerable tag searches are to malicious attacks that exploit tag popularity in a tagging system. As an example, we will focus on Occurrence-based searches. We studied all meaningful combinations of good and bad user models: (Good = *random*, Bad = *random*), (Good = *biased*, Bad = *random*), (Good = *biased*, Bad = *biased*), (Good = *biased*, Bad = *extremely biased*) and (Good = *biased*, Bad = *outlier*). Also, we considered two different searcher models: A *naive searcher* may use any tag in his searches. We simulate this behavior with a set of random queries. A *community member* may query popular tags more often. We simulate this behavior by a set of queries that follow the biased tag distribution.

For each combination of user models, we study the effect of varying the number $|\mathcal{A}|$ of popular tags on SpamFactor. We assume that popular tags may occur in the postings $m = 4$ times more often than unpopular ones. Figure 5 summarizes the corresponding experimental results. Random good and bad user models do not generate popular tags and thus do not depend on $|\mathcal{A}|$. Moreover, for $|\mathcal{A}| = 0\%$ and $|\mathcal{A}| = 100\%$, the biased tag distribution becomes random. Thus, the biased (good/bad) user models become random (good/bad) user models. Therefore, SpamFactor curves corresponding to all combinations but the ones that involve the Extremely Biased and the Outlier bad user models coincide at these two points. For the Extremely Biased behavior, SpamFactor is 0 for $|\mathcal{A}| = 0\%$, because there are no popular tags to use in bad postings. For the Outlier model, SpamFactor is 0 for $|\mathcal{A}| = 100\%$, since there are no unpopular tags to use.

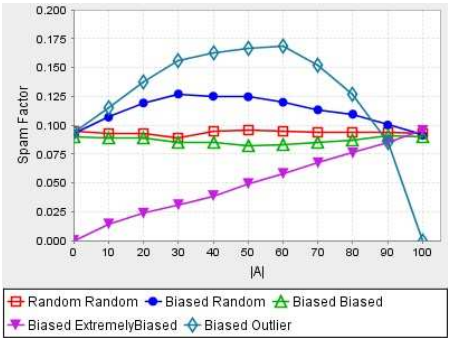
A general observation on Figure 5 is that random searches are more vulnerable to spam. In particular, random and outlier malicious attacks are the worst sources of spam for this category of searches. Let us discuss the Outlier model in more detail. A discussion of the other models is analogous and can be found in [14].

With the Outlier model, we observe that for random searches (Figure 5(a)), two conflicting phenomena take place with $|\mathcal{A}|$ growing: *On one hand*, unpopular tags receive increasingly more spam postings. This results in re-occurring bad postings multiplying, and thus in SpamFactor increasing. *On the other hand*, with $|\mathcal{A}|$ growing, there are fewer unpopular tags. Consequently, spam is confined to a smaller set of tags, while the "healthy" tags proliferate. These two conflicting phenomena reach a balance point, where maximum SpamFactor is observed. From this point forward, SpamFactor decreases to zero. In comparison, SpamFactor for community searches (Figure 5(b)) always decreases, because these searches consider spam postings only when unpopular tags are queried and this happens less often as $|\mathcal{A}|$ increases.

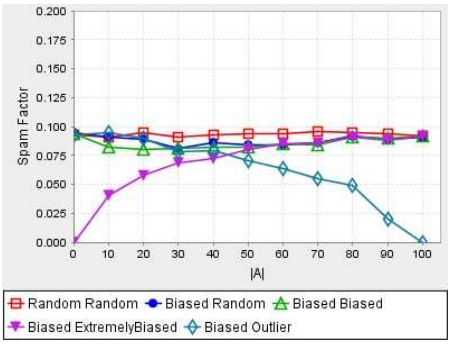
Overall, the existence of popular tags provides many opportunities for malicious users to misuse tags and spam searches. Naive or first-time users are most vulnerable. Random noise in postings as well as misused unpopular tags constitute the worst sources of spam for naive searches. Bad users mimicking good users and using popular tags for their postings can have a smaller impact on the system, in the worst case being as disruptive as lousy taggers (given a moderate number of bad users in the system). Community members may be less confused by spam postings, since they more often query about tags contributed by their community.

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 where malicious tags and malicious user behaviors are well defined, and we described and studied a variety of query schemes and moderator strategies to counter tag spam. 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. A countermeasure like our coincidences algorithm can be defeated by focused spam attacks. As a countermeasure for that situation, we proposed a focused moderator to detect the focused attacks. This is just an example of the measure-



(a) Occurrence-based naive searches



(b) Occurrence-based community searches

Figure 5: Impact of the number of popular tags

countermeasure battles that must be constantly fought to combat spam. Undoubtedly, the bad guys will counter-attack this proposal, and so on. We hope that the model we have proposed here, and the results it yields, can provide useful insights on how to wage these ongoing "spam wars." We also believe that our approach helps one quantify (or at least bound) the dangers of tag spam and the effectiveness of countermeasures.

There are also other interesting aspects of the problem and possible future directions to look into. 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 users can also use negative tags, e.g., this document is *not* about "cars", what would be the impact on searches?

Acknowledgements. We thank M. Deschamps for taking part in the initial development stages of the simulator.

9. REFERENCES

[1] 3spots: url: http://3spots.blogspot.com/2006/01/all-social-that-can-bookmark.html.

[2] CiteUlike: url: http://www.citeulike.org/.

[3] Del.icio.us: url: http://del.icio.us/.

[4] Flickr: url: http://www.flickr.com/.

[5] Rawsugar: url: http://rawsugar.com/.

[6] Slideshare: url: http://slideshare.net/.

[7] C. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th International Conference on World Wide Web*, 2006.

[8] S. Farrell and T. Lau. Fringe contacts: People tagging for the enterprise. In *Collab. Web Tagging Workshop in conj. with WWW2006*.

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

[10] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *1st Intl. Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 39–47, 2005.

[11] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating spam with TrustRank. In *30th Intl. Conf. on Very Large Databases (VLDB)*, pages 576–587, 2004.

[12] M. Henzinger. Link analysis in web information retrieval. *IEEE DE Bulletin*, 23(3):3–8, 2000.

[13] A. John and D. Seligmann. Collaborative tagging and expertise in the enterprise. In *Collab. Web Tagging Workshop in conj. with WWW2006*.

[14] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. Technical report, available at http://dbpubs.stanford.edu/pub/2007-11, 2007.

[15] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD*, pages 611–617, 2006.

[16] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Position paper, tagging, taxonomy, flickr, article, toread. In *Hypertext*, pages 31–40, 2006.

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

[18] T. Ohkura, Y. Kiyota, and H. Nakagawa. 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*, 2006.

[19] P. Schmitz. Inducing ontology from flickr tags. In *Collab. Web Tagging Workshop in conj. with WWW2006*.

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

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

[22] B. Wu, V. Goel, and B. Davison. Topical TrustRank: Using topicality to combact web spam. In *WWW*, pages 63–72, 2006.

[23] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Collab. Web Tagging Workshop in conj. with WWW2006*.