The Art of Tagging: Measuring the Quality of Tags

R. Krestel and L. Chen

L3S Research Center
Universität Hannover, Germany
{krestel,lchen}@L3S.de

Abstract. Collaborative tagging, supported by many social networking websites, is currently enjoying an increasing popularity. The usefulness of this largely available tag data has been explored in many applications including web resources categorization,deriving emergent semantics, web search etc. However, since tags are supplied by users *freely*, not all of them are useful and reliable, especially when they are generated by spammers with malicious intent. Therefore, identifying tags of high quality is crucial in improving the performance of applications based on tags. In this paper, we propose TRP-Rank (Tag-Resource Pair Rank), an algorithm to measure the quality of tags by manually assessing a seed set and *propagating the quality* through a graph. The three dimensional relationship among users, tags and web resources is firstly represented by a graph structure. A set of seed nodes, where each node represents a tag annotating a resource, is then selected and their quality is assessed. The quality of the remaining nodes is calculated by propagating the known quality of the seeds through the graph structure. We evaluate our approach on a public data set where tags generated by suspicious spammers were manually labelled. The experimental results demonstrate the effectiveness of this approach in measuring the quality of tags.

1 Introduction

With the recent rise of Web 2.0 technologies, many social media applications like *Flickr*, *Del.ici.ous*, and *Last.fm* provide features which allow users to assign tags [1] to a piece of information such as a picture, blog entry, video clip etc. Web users from different backgrounds annotate (tag) resources on the Web at an incredible speed, which results in a large volume of tag data obtainable from the Web today. The hidden value of tag data has been explored in many applications. For example, Tso-Sutter et al [2] incorporated tags into collaborative filtering algorithms to enhance recommendation accuracy. In [3], the authors discussed using tags to lighten the limitation of the amount and quality of anchor text to improve enterprise search. The usage of tags in Web search has also been investigated in Bao et al [4].

One notable reason which supports the increasing popularity of collaborative tagging is that users are permitted to enter tags at will, without referring to

J. Domingue and C. Anutariya (Eds.): ASWC 2008, LNCS 5367, pp. 257–271, 2008.

any pre-specified taxonomy or ontology. On the one hand, this easy and flexible utility boosts the spreading of collaborative tagging systems. On the other hand, allowing users to *freely* choose tags sometimes leads to poor quality of the tag data. For example, ambiguity and synonymy are two frequently cited problems. The tag “XP” is used to annotate both web pages about “Extreme Programming” and pages about “Windows XP”. Synonymous tags, like “RnB” and “R&B”, are also widely used. Such problems hamper the applications built upon tags. Another problem which even damages the performance of applications using tags is tag spam, which refers to misleading tags generated maliciously in order to increase the visibility of some resources or simply to confuse users. *Therefore, measuring the quality of tags is an important issue and discriminating high quality from low quality tags improves the effectiveness of different tag-based applications.*

In [5], the authors discussed some properties a good tag combination (e.g., the set of tags annotating a common resource) should possess. For example, a good tag combination should cover multiple facets of the tagged resource; the set of tags should be used by a large number of people; and the number of resources identified by the tag combination should be small etc. They further proposed a tag suggestion algorithm based on these properties. In contrast to suggesting new tags to users based on existing tags so that a good tag combination can be achieved, our objective here is to assess the quality of tags assigned by users. Koutrika et al [6] proposed to combat tag spam by ranking the results returned from a query tag, based on the co-occurrence frequency between the tag and each resource. Thus, their approach is specially designed for tag based search. Our research objective is more general so that the results can be used in various applications of tags.

Note that, whether a tag is good or bad can only be assessed with respect to a particular resource. Hence, our investigation is based on the unit of a tag-resource pair. We aim to measure the quality of each individual pair of tag and resource. For this purpose, we firstly construct a graph which models tag-resource pairs as nodes and co-user relationship as edges. We then select a set of seed nodes whose qualities are assessed manually. The qualities of the remaining nodes are calculated by propagating the qualities of seed nodes through the graph. In order to improve the performance of this approach, a set of various seed selection strategies are employed. We evaluate the effectiveness of our approach on a bibsonomy data set[1] labelled manually.

The rest of this paper is organized as follows. We discuss the background knowledge by reviewing related work in Section 2. In Section 3, we describe the approach which propagates the quality of tag-resource pairs and discuss improving the performance by employing different strategies to select a set of seeds. The evaluation results conducted on a public data set are presented and analyzed in Section 4. Finally, Section 5 concludes this paper with some summary remarks and future work discussions.

[1] http://www.kde.cs.uni-kassel.de/ws/rsdc08/dataset.html

2 Related Work

In this section, we review related work in two areas, collaborative tagging systems and spam detection.

A collaborative tagging system allows users of a web site to freely attach to a particular resource arbitrary tags which, in the opinion of the user, are somehow associated with the resource in question. The commonly noted structure of collaborative filtering systems is a tripartite model consisting of users, tags and resources. This model is developed as a theoretical extension of the bipartite structure of ontologies with an added "social dimension" in [7]. The dynamics of collaborative systems are examined in [8] using the tag data at the bookmarking site Del.ici.ous. According to this work, tag distributions tend to stabilize over time. Halpin et al. confirm these results in [9] and show additionally that tags follow a power law distribution. Considering the structure and stable dynamics of collaborative tagging systems, it seems likely that tag data would be a reliable source of semantic information reflecting the cultural consensus of a particular system's users. As a result, various applications of tag data have been researched. Mika [7] investigates the automatic extraction of ontological relationships from tag data and proposes the use of such emergent ontologies to improve currently existing ontologies which are less capable of responding to ontological evolution. Dmitriev et al. [3] explore the use of "annotations" for enterprise search to compensate for the lack of sufficient anchor text in intranet environments. In [4], tag data is exploited for the purpose of web search through the use of two tag based algorithms: one exploiting similarity between tag data and search queries, and the other utilizes tagging frequencies to determine the quality of web pages. Tso et al [2] incorporate the tag data into the collaborative filtering systems. Berendt and Hanser [10] demonstrate the benefits of using tag data for weblog classification by treating it as content instead of meta data. For searching and ranking within tagging systems, [11] proposes the exploitation of co-ocurrence of users, resources, and tags. This is done using a graph model to represent the *folksonomy.*

Everywhere in the internet where information is exchanged, malicious individuals try to take advantage of the information exchange structure and use it for their own benefit. The largest amount of spam and historically the first field where spam was generated is the electronic communication system (e-mail). Afterwards, various internet applications were attacked by spammers such as search engine spam, blog spam, wiki spam etc, which triggered numerous research efforts in spam combating. For example, TrustRank [12] separates spam pages from non-spam pages based on the intuition that trustworthy pages usually link to also trustworthy pages and so on. They select a seed set of highly trusted pages first and then propagate the trust score of seed pages by following the links from these pages through the Web. A survey of approaches fighting spam on social web sites can be found in [13]. Comparing to spam detection from other web applications, studies on detecting spam from collaborative tagging systems are very limited. Koutrika et al [6] propose to combat spam in the particular situation when users query for resources annotated with certain tags. Their method ranks a resource higher if more users annotated it with the queried tags, based on the assumption that tag spam may not be used by the majority. Our work is different in the way that our approach is not designed for a particular application. Consequently, the output of our algorithm can be used by any application based on tags. Xu et al [5] assign authority scores to users, and measure the quality of each tag with respect to a resource by the sum of the authority scores of all users who have tagged the resource with the tag. Then, the authority scores of users are computed via an iterative algorithm similar to HITs [14]. Their approach treats every tag-resource pair used by a user equally even if a spam user may use good tag-resource pairs frequently and bad ones occasionally. Our approach addresses this problem by measuring the quality of a tag-resource pair more independently from a particular user.

3 Measure Tag Quality

The hidden value of tag data has been explored by a wide range of applications. However, as mentioned before, since there is no limitation on the vocabulary users are allowed to use for taggging, the quality of tags varies. In other words, tags are not equally useful for a particular application. For example, recovery and discovery of resources on the web is one of the main uses of tags. Although tags describing the general topics of resources might be useful for search engines, *personal* or *subjective* (see [15,16] for a taxonomy of tags) tags such as "myFavorite", "funny", "home" do not seem to be promising for this task. Furthermore, it is common that tags which describe one resource very well may not be suitable for another resource. Consequently, measuring the quality of tags is critical for applications to exploit the positive usage of tag data. The quality of a tag should be measured with respect to the resource to which it is assigned.

In this section, we first formally define the problem we focus on in this paper. Then, the data structure which models the relationship among tags, resources, and users is described. Next, we illustrate our algorithm, called *TRP-Rank* (Tag-Resource Pair Rank), which iteratively assesses the quality of each pair of tag-resource in the data set. Finally, several strategies which select various sets of seed nodes, serving as the input of TRP-Rank, are discussed.

3.1 Problem Specification

Let $\mathcal{T}$ be a set of tags, $\mathcal{R}$ be a set of resources, and $\mathcal{U}$ be a set of users. We denote a tag assignment of a tag $t \in \mathcal{T}$ to a resource $r \in \mathcal{R}$ as a tag-resource pair tr. All tag assignments in the data $\mathcal{T} \times \mathcal{R}$ is a set of tag-resource pairs denoted as $\mathcal{TR} = \{tr|t \in \mathcal{T}, r \in \mathcal{R}\}$. Each tag-resource pair is assigned by at least one user $u \in \mathcal{U}$. We define the function $getU(tr)$ to retrieve the set of users who assigned t to r. Note that, $getU(tr) \neq \emptyset$. Then, given the complete set of tag-resource pairs $\mathcal{TR} = \{tr_1, \cdots, tr_n\}$, and associated users of each tag-resource pair $getU(tr_i) \subseteq \mathcal{U}$, our goal is to find a function $Q(tr_i)$ which assigns a score to each tag-resource pair tr_i such that the higher the value of $Q(tr_i)$, the better the quality of the pair tr_i. The value of $Q(tr_i)$ ranges in $[-1, 1]$ (the reason why negative values are involved will be explained later in Section 3.3).

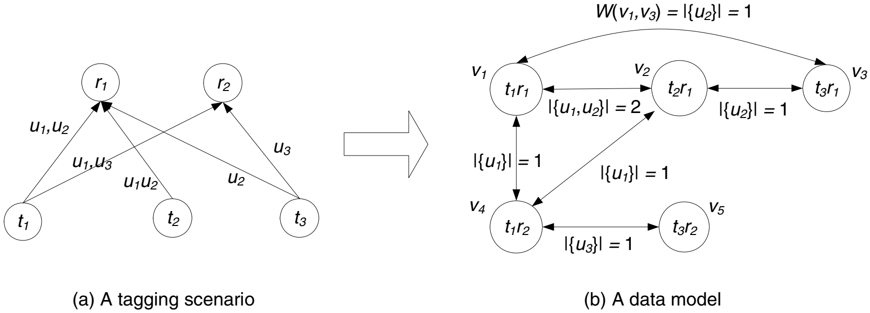
3.2 Tagging System Model

Given a set of data including tags $\mathcal{T}$, resources $\mathcal{R}$ and users $\mathcal{U}$, we model the data as a bidirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of vertices with each $v \in \mathcal{V}$ represents a $tr \in \mathcal{TR}$. $\mathcal{E}$ is a set of edges such that each edge (v_i, v_j) indicates that the two corresponding tag-resource pairs tr_i and tr_j are assigned by at least one common user. That is, $|getU(tr_i) \cap getU(tr_j)| \geq 1$. Additionally, we associate a weight to each edge so that the weight of an edge is the number of common users who assigned the tag-resource pairs corresponding to the two end nodes of this edge, $W(v_i, v_j) = W(tr_i, tr_j) = |getU(tr_i) \cap getU(tr_j)|$.

In Figure 1 (a), we present a very simple tagging scenario: Suppose we have three users $\mathcal{U} = \{u_1, u_2, u_3\}$, three different tags $\mathcal{T} = \{t_1, t_2, t_3\}$ and two resources $\mathcal{R} = \{r_1, r_2\}$. Each user has annotated the resources with certain tags. For example, the leftmost link in Figure 1 (a) indicates that both users u_1 and u_2 have supplied the tag t_1 with the resource r_1. Observing the tag assignments in this figure, we notice that there are a total 5 tag-resource pairs $\mathcal{TR} = \{t_1r_1, t_2r_1, t_3r_1, t_1r_2, t_3r_2\}$. Hence, as shown in Figure 1 (b), there are five nodes involved in the data model where each node represents a particular tag-resource pair. An edge connects two nodes if the two corresponding tag-resource pairs are supplied by at least one common user. For example, there is an edge between $v_1 : t_1r_1$ and $v_3 : t_3r_1$ because they are supplied by the common user u_2. Accordingly, the weight of this edge, as shown in the figure, is $|\{u_2\}| = 1$.

Based on this graph model, we introduce a right stochastic transition matrix T, which is defined as:

$$T(i,j) = \begin{cases} 0 & \text{if } (v_i, v_j) \notin \mathcal{E} \\ \frac{W(v_i,v_j)}{\sum_{v_k \in \mathcal{V}} W(v_i,v_k)} & \text{if } (v_i, v_j) \in \mathcal{E} \end{cases}$$



(a) A tagging scenario

(b) A data model

Fig. 1. A tagging scenario and its data model

Figure 2 shows the adjacency matrix and the transition matrix for the example in Figure 1. Note that, the adjacency matrix is symmetric since the graph model is bidirected, while the transition matrix is asymmetric.

	v_1	v_2	v_3	v_4	v_5
v_1		2	1	1	
v_2	2		1	1	
v_3	1	1			
v_4	1	1			1
v_5				1	

$$T = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Fig. 2. Adjacency (left) and transition (right) matrixes of the example in Figure 1

3.3 Quality Propagation

Similar to TrustRank [12], which semi-automatically separates web pages from spam, the basic idea of TRP-Rank is to manually assign quality scores to a subset of $\mathcal{TR}$ first, and propagate these quality values through the graph. As the TrustRank algorithm is based on the well-known PageRank [17] algorithm, we briefly review PageRank and TrustRank in the following before illustrate TRP-Rank.

PageRank. PageRank is an algorithm that assigns scores to web pages based on link information. When important pages point to a particular page, this page should also be considered important as well. Thus importance information is propagated through the web graph via an iterative process:

$$\text{p-rank}_{i+1} = \alpha \cdot T \cdot \text{p-rank}_i + (1 - \alpha) \cdot \frac{1}{N} \cdot 1_N. \tag{1}$$

where α is a decay factor, T is the transition matrix and N is the number of web pages. The transition matrix is not weighted and all web pages get the same initial value of p-rank. The iteration process goes on until the difference between two consecutive runs' results is below a certain threshold.

TrustRank. TrustRank extends the Equation (1) to identify web spam. Therefore the original PageRank algorithm was altered to be biased towards a seed set of high quality sites, where each site x was manually assessed with an oracle function $\mathrm{O}(x)$. Then, the column vector $\frac{1}{N} \cdot 1_N$ in Equation (1) is replaced with a vector d, such that elements corresponding to manually assessed sites are set as $O(x)$ and the remaining elements are set as 0. d is then normalized, $\mathsf{d} = \mathsf{d}/|\mathsf{d}|$, and feed as t-rank_0.

$$\text{t-rank}_{i+1} = \alpha \cdot T \cdot \text{t-rank}_i + (1 - \alpha) \cdot \mathsf{d}. \tag{2}$$

The set of seed sites is selected using an inverse PageRank algorithm. Particularly, nodes from where lots of other nodes can be reached are identified and ranked accordingly, similar to the idea of Hubs [14]. Then, the top-k nodes are manually assigned values 1, or 0 in case of a spam web site, and these initial values are stored in d.

TRP-Rank. For TRP-Rank, the quality of each tag-resource pair, $Q(tr)$, is computed similarly as the Equation (2) in TrustRank. That is, we propagate initial quality scores of seed tag-resource pairs through the graph. In addition to TrustRank which propagates only trust information, we adopt the distrust propagation idea described in [18] to allow the propagation of scores for not only good tag assignments but also explicitly bad ones. Consequently, in TRP-Rank, we extend the manual seed set assessment to include both tag-resource pairs of high quality and those of low quality. We populate the initial vector d with:

$$\mathsf{d}(tr_i) = \begin{cases} \mathrm{O}(tr_i) & \text{if } tr_i \in SEED \\ 0 & \text{if } tr_i \notin SEED \end{cases} \tag{3}$$

where $\mathrm{O}(tr_i) \in \{-1, 0, 1\}$ is the oracle function which assigns initial quality score 1 to good tag-resource pairs, -1 to bad ones and 0 to the rest. $SEED \subseteq \mathcal{TR}$ is a set of seed nodes, which will be defined in Section 3.4.

Consider the running example shown in Figures 1 and 2, the results of TRP-Rank (i.e. quality of tag-resource pairs) after 10 iterations are shown in Figure 3, where v_3 and v_4 are selected as seed nodes and the decay factor α is set as 0.85.

$$\text{trp-rank}_{i+1} = 0.85 \cdot \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \text{trp-rank}_i + (1 - 0.85) \cdot \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}$$

$i = 10$	v_1	v_2	v_3	v_4	v_5
trp-rank(10**)**	-0.03341879	-0.03341879	-0.16368952	0.180295	0.05023218

Fig. 3. TRP-Rank computation and results for the example in Figure 1

3.4 Seed Selection Strategies

In our approach, we experiment with three different seed selection strategies, whose performance will be presented and discussed in Section 4.3. The two main challenges for seed set selections are: 1) finding an appropriate size for the seed set. A small seed set may not be enough to reach most nodes in the graph, while a large seed set means an expensive manual assessment process; 2) picking the *right* set of tag-resource pairs as seeds. On the one hand, the seed set should contain not only good tag-resource pairs but also pairs of low quality, so that explicit information of both good and bad quality can be propagated. On the other hand, the seed set should contain nodes from which many of the remaining nodes can be reached.

Algorithm 1. Different Seed Selection Strategies

```
Input:
  $N$: a set of graph nodes, $K$ ($K < |N|$): the number of seeds
Output:
  $SEED$: A set of selected seed nodes
order N as $\hat{N} =< v_1, v_2, \cdots, v_n >$ such that $PR(v_i) \geq PR(v_{i+1})$
for each $v_i \in \hat{N}$ do
  if Top-K Seed Selection then
    if $|SEED| < K$ then
      $SEED = SEED \cup \{v_i\}$
    end if
  end if
  if Exponential Base Seed Selection then
    if $i \in \{a_n\};\ a_n = n + \lfloor b^n \rfloor;\ b = e^{\frac{\ln(|N|-K-1)}{K-1}};\ \forall n \in \{0, \ldots, K-1\}$ then
      $SEED = SEED \cup \{x_i\}$
    end if
  end if
  if Constant Base Seed Selection then
    if $\exists n \in \mathbb{N} \mid \lfloor an = i \rfloor; a = \frac{|N|}{K}$ then
      $SEED = SEED \cup \{x_i\}$
    end if
  end if
end for
```

We first compute PageRank scores for each tag-resource node to examine the connectivity of each node in the graph. The resulting list, with the nodes ordered according to PageRank, is the starting point for the three strategies we evaluated. Algorithm 1 shows the three seed selection processes.

1. **Top-k seed set.** TrustRank also employed the (inverse) top-k PageRank selection to find highly connected nodes whose quality influences a lot of neighboring nodes. However, since our data model is a bidirected graph, we consider the top-k PageRank directly without computing the inverse PageRank scores. This strategy can be easily adjusted to satisfy the first requirement of the seed set size, while it may not be able to select the right seed set which includes both good and bad tag-resource nodes. The reason is that, as will be shown in the next section, bad tag-resource nodes usually have lower PageRank values.
2. **Exponential base seed set.** Motivated by the observation that the top-k strategy mainly select the good tag-resource nodes, this strategy aims to include more bad tag-resource nodes in the seed set. However, in order to propagate quality scores through the graph as far as possible, nodes with high PageRank values (i.e., high connectivity) are favored. Hence, after ordering nodes based on their PageRank scores, seed nodes are selected with an increasing interval, such as $\{v_1, v_2, v_4, v_8, \cdots\}$.
3. **Constant base seed set.** In contrast to exponential base seed selection which favors nodes with high connectivity to those less connected, so that more good tag-resource nodes are selected, this strategy selects good and bad tag-resource nodes with equal chances. For example, let the constant base be 10, then every 10th node will be selected. The inclusion of more bad

tag-resource nodes may be able to discover more tag-resource nodes with inferior quality, while the propagation may not be as extensive as before.

4 Evaluation

Since there is no manually annotated corpus – of which we are aware of – that could be used to compare our results for the quality of tags with a gold standard, we have to resort to an indirect approach. Particularly, we use the tag data compiled for a competition[2] to detect spam users. In this section, we first describe the data set. Then, an indirect approach to evaluate TRP-Rank is discussed. Next, we evaluate the performance of TRP-Rank, with different seed selection strategies. Finally, we examine the performance of our approach when applied to a larger dataset.

4.1 Data Set

The data set used by us consists of $221,354$ tag assignments by $1,328$ users of the BibSonomy[3] system for publications. Out of these users, 118 were marked manually as spammers and $1,210$ as non-spammers. The size of the set of unique tag-resource pairs $\mathcal{TR}$ is $195,198$. We discarded tag-resource pairs which were made by users having only one tag assignment (these tag-resource pairs would be disconnected nodes in our data model). And we only picked the first 1000 tag assignments of users whose number of tag assignments exceed this threshold. The remaining set has $132,520$ trs.

In order to show the connectivity of tag-resource nodes, Table 1 summarizes the numbers of pairs of tag-resource nodes, $\{tr_i, tr_j\}$, and their associated common users. For example, the second column of the table indicates that there are $175,619$ pairs of trs that are used by only one common user. In other words, in the adjacency matrix of our data model, there are $2 * 175,619$ elements with value 1. Although these numbers seem to imply that the graph is not highly connected, as we will show in Section 4.3, a rather small seed set is sufficient to reach most of the nodes in the graph.

Table 1. Number of pairs of trs assigned by common users

Number of pairs of trs	175619	15767	2664	641	197	115	55	41	24	75
Shared by # of Users	1	2	3	4	5	6	7	8	9	≥10

Some necessary preprocessing has been done before using the data. For example, since the data set consists of the raw BibSonomy data, we have to give IDs to each individual tr. To identify the semantic relationship between certain tags, we use stemming and ignore capital letters to assign one ID to a group of tags (e.g. "Book", "book", or "Books").

[2] `http://www.kde.cs.uni-kassel.de/ws/rsdc08/`

[3] `http://www.bibsonomy.org`

4.2 Indirect Evaluation Method

The TRP-Rank algorithm aims to measure the quality of each tag-resource pair, while the data set contains only the spammer information. Hence, an indirect evaluation method needs to be used. Basically, we need to consider the following two issues: 1) The input of TRP-Rank needs manually assessed quality scores of a set of seed tag-resource nodes. How to assign the initial quality using the spammer information in the data? 2) The output of TRP-Rank is the converged quality scores of all tag-resource pairs. How to map the quality scores of tag-resource pairs to some score which could reflect whether a user is a spammer or not? We discuss the solutions of the two problems respectively as follows.

For assigning the initial quality scores to seed tag-resource nodes, we make use of the available spammer information in the dataset by defining a function $notSpammer(u) \in \{1, -1\}$. When a user u is not a spammer, the function returns value 1; otherwise, it returns value -1. Thus, the oracle function $O(tr)$ assigns the scores to each $tr \in SEED$ as:

$$\mathrm{O}(tr) = \begin{cases} 1 & \text{if } \frac{1}{|getU(tr)|} \sum_{u \in getU(tr)} \text{notSpammer}(u) > 0 \\ -1 & \text{if } \frac{1}{|getU(tr)|} \sum_{u \in getU(tr)} \text{notSpammer}(u) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

That is, when a tag-resource pair is assigned by more normal users than spammers, it is deemed as a good tag-resource node and assigned a positive quality score. Otherwise, a negative score is given to reflect the inferior quality of the tag-resource node.

For mapping the result quality scores $Q(tr)$ of all tag-resource pairs, returned by TRP-Rank, to the scores indicating whether a user is a spammer or not, we aggregate the quality of all tag-resource pairs assigned by the user. Let $getTR(u)$ return the set of tag-resource pairs used by u, $getTR(u) = \{tr_1, \cdots, tr_n\}$. We define the function $isSpammer(u)$ as:

$$isSpammer(u) = \begin{cases} 1 & \text{if } \frac{1}{|getTR(u)|} \sum_{tr_i \in getTR(u)} Q(tr_i) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

4.3 TRP-Rank Performance

We first examine the maximum performance which can be achieved theoretically with our approach. Namely, the performance generated when the complete set of tag-resource nodes are used as seeds. As shown by the top confusion matrix in Table 2, the accuracy is approximately 97.66% (1210/1239). It is actually promising considering that our algorithm is not designed for spammer detection. We further investigate the theoretically achievable maximum by using all nodes with positive initial quality scores and all nodes with negative initial quality scores as seeds respectively. The middle and bottom confusion matrixes in Table 2 show the results. We notice that, compared with using only the nodes

Table 2. Confusion matrixes for theoretically achievable maximum using different seeds

Positive and Negative spread information			
True Positives:	**1210**	True Negatives	**89**
False Positives:	**29**	False Neagatives	**0**
Only positive spread information			
True Positives:	**1079**	True Negatives	**114**
False Positives:	**4**	False Neagatives	**131**
Only negative spread information			
True Positives:	**1210**	True Negatives	**91**
False Positives:	**27**	False Neagatives	**0**

with positive initial scores as seeds, using all nodes with negative initial quality scores is able to detect more spammers correctly.

Then, we investigate the performance of TRP-Rank which uses a combination of good and bad nodes as seeds. We conduct the experiments by varying the size of seed sets. As discussed in Section 3.4, the PageRank of nodes is used as the starting point to select seeds. Figure 4 shows the PageRank scores for all *tr*s in our data set. By examining the PageRank scores of nodes, we notice that nodes related to spammers usually have lower PageRank values. That observation implies that the *top-k* method probably will not include as many negative nodes related to spammers as seeds as the exponential base and constant base seed selections do. The results for the different selection strategies are shown in Table 3, which verify the previous hypothesis. The top-k approach is not comparable to the other two seed selection approaches. It also could not outperform the method

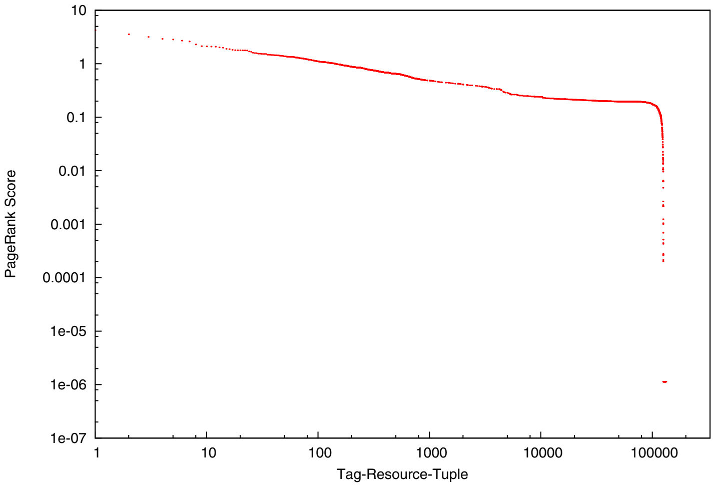


Fig. 4. Log-log graph of PageRank scores for the whole data set

Table 3. Accuracy for different seed set selection strategies with seed set size 10000/20000

Strategy	Accuracy Seed Set Size	
	10000	20000
Top-k	91.11 %	91.11 %
ExponentialBase	94.58 %	96.39 %
ConstantBase	94.88 %	96.31 %

which uses all nodes with negative initial scores as seeds. In contrast, the other two seed selection methods exhibit similar good performance.

We further investigate how the performance of TRP-Rank varies with respect to the seed set size. The seed set size is a crucial factor for the algorithm. Since we need an oracle function that gives us $O(tr)\ \forall tr \in SEED$, and the oracle function usually invokes human assessing procedures, a large seed set could be expensive. However, a smaller seed set may be not able to propagate the quality through the graph wide enough. As shown in Table 4, which are the performance of TRP-Rank with constant base seed selection running on seed sets with different size, we notice that our approach can achieve an accuracy as good as 93.75% even if only 3.7% $(5000/132,250)$ of the nodes are selected as seeds, which equals roughly the manual assessment of 50 users.

Table 4. Results for different sized seed sets using constant base TP=true positives, TN=true negatives, FP=false positives, FN=false negatives

Seed Set Size	TP	FP	TN	FN	Accuracy
132520	1210	29	89	0	97.82 %
50000	1210	29	89	0	97.82 %
20000	1210	49	69	0	96.31 %
10000	1210	68	50	0	94.88 %
5000	1210	87	31	0	93,45 %

4.4 Data Reduction

For large data sets the matrix of our algorithm can become very large. To reduce the amount of data to process, we examine the effect of considering only trs where tags were used by at least x $(x > 1)$ users. This seems to be justifiable at least for the case of measuring the quality of a certain tag for a certain resource. For detecting spam users, this filtering scheme is also an option. We examine the performance by using the whole data set as seed set and setting the parameter x as 3 and 10 respectively. The results are shown as below. We observe that the performance drops by only 2.94 % when considering only tags that were used by at least 10 users (compared with the performance where $x = 1$), while the transition matrix size is reduced by more than 50%.

- Minimum 10 Users → Accuracy 94.80 %
- Minimum 3 Users → Accuracy 95.63 %

- Minimum 1 Users → Accuracy 97.67 %

4.5 Discussion

The experimental results demonstrate that our algorithm performs quite well on distinguishing spammers from normal users based on the quality of their tag-resource pairs. After looking at the data into more detail, it seems that our approach could perform even better when modifying the notion of "spammer". For example, users with only one "test" tag assignment are considered as non spammers in the data set. Since they are not malicious users, this might be an acceptable classification. Nevertheless, from the tag quality point of view, these users would be considered unreliable because they use bad quality tag-resource pairs.

As observed from the experiments, an appropriate seed set should be well representative so that it contains not only good tag-resource pairs but also bad one. However, in a real-world tagging system, the majority are usually good/non-spam tags. Thus, the negative seeds are ranked rather low by PageRank which makes them hard to be found. The constant base seed selection method is generally applicable and has shown to be effective.

Regarding the size of the whole data set, we saw that the accuracy drops only little when putting some restrictions on the tags which are allowed for valid tag-resource pairs. Filtering out tag-resource pairs with tags used by few users is useful under the assumption that tags that are regarded valuable are used by a lot of users.

5 Conclusions and Future Work

In this paper, we focus on the problem of measuring the quality of tags which are supplied by users to annotate resources on the Web. Due to the intrinsic feature of existing collaborative tagging systems that users are allowed to supply tags freely, the resulting tags can have great disparity in quality. Consequently, measuring the quality of tags appropriately is important towards effectively exploiting the usefulness of tags in many applications. The main characteristics of our algorithm are represented by the data model we adopt and the seed selection functions we investigate. By decoupling the relationship between users and tag-resource pairs, we model the tag-resource pairs as nodes and co-user relationship as edges of a graph. Different from existing models, this structure allows every two tag-resource pairs used by the same user to have different quality, which complies with the practical situation better. Our algorithm, which propagates quality scores iteratively through the graph, needs to be initialized with the scores of a set of seed nodes. We investigate various seed selection strategies with the aim to not only minimize the size of the seed set but also minimize the error of the resulting quality scores. The effectiveness of our algorithm is evaluated on a manually labelled data set and demonstrated by the promising experimental results.

For future work, we are interested in pursuing the following problems:

- We currently assign the three distinct values $\{-1, 0, 1\}$ to the set of seeds. However, finer initial quality scores such as 0.2, 0.5 might be able to dissect the quality of tag assignments better.
- The manually assessment of the quality of seed nodes is expensive. How to make use of Web 2.0 and let users generate the seed set is an interesting issue which is worthwhile to consider.
- Since TRP-Rank demonstrated good performance of detecting spammers in tagging systems, we are considering to revise our approach to specifically address combatting tag spam. For example, our current model represents tag-resource pairs as nodes in order to measure the quality of tag-resource pairs. We can alternatively model users as nodes and common tag-resource pairs as edges to directly find spam users.

Acknowledgements

This work is supported by the EU project IST 45035 - Platform for searcH of Audiovisual Resources across Online Spaces (PHAROS).

References

1. Marlow, C., Naaman, M., Boyd, D., Davis, M.: Ht06, tagging paper, taxonomy, flickr, academic article, to read. In: Wiil, U.K., Nürnberg, P.J., Rubart, J. (eds.) Hypertext, pp. 31–40. ACM, New York (2006)
2. Tso-Sutter, K.H.L., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: Wainwright, R.L., Haddad, H. (eds.) SAC, pp. 1995–1999. ACM, New York (2008)
3. Dmitriev, P.A., Eiron, N., Fontoura, M., Shekita, E.J.: Using annotations in enterprise search. In: [19], pp. 811–817 (2006)
4. Bao, S., Xue, G.R., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing web search using social annotations. In: [20], pp. 501–510 (2007)
5. Xu, Z., Fu, Y., Mao, J., Su, D.: Towards the semantic web: Collaborative tag suggestions. In: WWW2006: Proceedings of the Collaborative Web Tagging Workshop, Edinburgh, Scotland (2006)
6. Koutrika, G., Effendi, F., Gyöngyi, Z., Heymann, P., Garcia-Molina, H.: Combating spam in tagging systems. In: AIRWeb (2007)
7. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
8. Golder, S.A., Huberman, B.A.: The structure of collaborative tagging systems. CoRR abs/cs/0508082 (2005)
9. Halpin, H., Robu, V., Shepherd, H.: The complex dynamics of collaborative tagging. In: [20], pp. 211–220 (2007)

10. Berendt, B., Hanser, C.: Tags are not metadata, but just more content - to some people. In: ICWSM (2007)
11. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)
12. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.O.: Combating web spam with trustrank. In: Nascimento, M.A., Özsu, M.T., Kossmann, D., Miller, R.J., Blakeley, J.A., Schiefer, K.B. (eds.) VLDB, pp. 576–587. Morgan Kaufmann, San Francisco (2004)
13. Heymann, P., Koutrika, G., Garcia-Molina, H.: Fighting spam on social web sites: A survey of approaches and future challenges. IEEE Internet Computing 11(6), 36–45 (2007)
14. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM 46(5), 604–632 (1999)
15. Sen, S., Lam, S.K., Rashid, A.M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F.M., Riedl, J.: Tagging, communities, vocabulary, evolution. In: Proceedings CSCW, New York, NY, USA, pp. 181–190. ACM, New York (2006)
16. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. Journal of Information Science 32, 198–208 (2006)
17. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. techreport (1998)
18. Wu, B., Goel, V., Davison, B.D.: Propagating trust and distrust to demote web spam. In: Finin, T., Kagal, L., Olmedilla, D. (eds.) MTW of CEUR Workshop Proceedings, vol. 190 (2006), CEUR-WS.org
19. Carr, L., Roure, D.D., Iyengar, A., Goble, C.A., Dahlin, M. (eds.): Proceedings of the 15th international conference on World Wide Web. In: Carr, L., Roure, D.D., Iyengar, A., Goble, C.A., Dahlin, M. (eds.) WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006. ACM Press, New York (2006)
20. Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.): Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007. ACM, New York (2007)