Robust Expert Ranking in Online Communities - Fighting Sybil Attacks

Khaled A. N. Rashed, Cristina Balasoiu, Ralf Klamma
Advanced Community Information Systems (ACIS)
RWTH Aachen University
Ahornstr. 55, D-52056 Aachen, Germany
Email: {rashed|balasoiu|klamma}@dbis.rwth-aachen.de

***Abstract*—Nowadays, many online communities provide means for users to contribute in the evaluation of community created media by tagging, commenting and rating. Judging the users expertise in such collaborative systems is an important issue. As these systems are becoming increasingly popular, they are attackable, e.g. by Sybil Attacks. Thus, an effective expert ranking strategy must be robust to such attacks. In this paper, we propose MHITS, an algorithm to rank users' expertise by exploiting the number of users' fair ratings and direct trust users gain in the online community. We integrate SumUp, a Sybil-resilient algorithm, into MHITS algorithm as a robust ranking strategy. Experimental results show the effectiveness of the proposed method, which can ensure that the highly ranked experts are highly trusted users and provide the high number of fair ratings for the relevant media. We contribute to the experimental evaluation of algorithms for online systems, fighting malicious behavior.**

***Index Terms*—MHITS Algorithm, Robust Expert Ranking, Collaborative Fake Detection, Fighting Sybil Attacks, Trust-Awareness**

I. Introduction

Online social media communities and web-based reputation systems enable users to rate, tag and comment media (e.g. news, images, articles, products, etc.). Recently, such collaborative systems are becoming more and more popular media and knowledge sharing platform. Identifying the expertise of users in such environments and ranking them in a proper order become an important issue. As the size of these systems enlarges, the number of involved entities (media) and users increases, the behavior of some users changes what makes users' ratings questionable, especially when a set of malicious users are trying to game the system by boosting or downgrading the entity's reputation. Consequently, this brings issues to the media authenticity evaluation as well as to the ranking quality in such dynamic environments. A common malicious behavior in reputation systems is the Sybil attack [1], where an attacker controls a large number of adversary entities that are used to gain influence in the community by feeding the system with bogus information (ratings in our case) and affect the correct functioning of the system as well as hurt the ranking algorithm. Thus, an effective ranking strategy must be robust to such attacks.

The task of experts ranking has been addressed in previous research and various approaches have been proposed. The link analysis approaches, including PageRank [2] and HITS [3], have been adopted to find and rank experts. Most of these approaches (e.g. [2], [3], [4], [5]) represent the relationship of users and entities, and not much attention has been made to the trust and reputation of experts that can also be derived from link analysis.

In this paper, we investigate the expertise of users in web-based collaborative systems and offer a ranking strategy that is capable to capture user's expertise in such online communities. We propose a new expert ranking algorithm, called *MHITS*, a modified HITS algorithm to rank community users based on the number of fair ratings for a particular entity category and direct trust that users gain in the community. We built an expert network based on user interaction information to model a domain independent information, and use MHITS algorithm to calculate the hubness scores of experts. Unlike existing expert ranking approaches, we also deal address the Sybille attacks problem. Considering the Sybil attacks, we integrate SumUp [6], a Sybil-resilient algorithm, into our MHITS algorithm as a robust ranking strategy. To evaluate our proposed algorithm, we create a synthetic dataset that corresponds to the real scenario of a fake media detection system. Experimental results show that our algorithm is an effective expert ranking algorithm, which can ensure that the highly ranked experts are highly trusted users and provide the high number of fair ratings for relevant media. Furthermore, the algorithm is capable to limit the number of Sybil attacks as a result of its integration with SumUp. Our contributions in this paper include:

- Developing the MHITS algorithm, a new algorithm to rank users' expertise in online web-based collaborative systems.
- Ensuring the robustness of the MHITS by limiting the effects of Sybil attacks by incorporating SumUp into MHITS.
- Evaluating the MHITS and integrated MHITS with SumUp on a fake media detection system, an online trust-based community aiming to detect fake media with the help of users' ratings.

The rest of this paper is structured as follows: Section 2 defines the problem and briefly reviews the related work. Section 3 provides a brief description of a case study, which will be used to evaluate our algorithm. Section 4 describes

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250439

our ranking algorithm in details. In Section 5, the evaluation methods and the experimental results are depicted. Section 6 concludes the paper with some remarks of future direction.

II. Related Work

The *expert search* refers to the way of finding a group of authoritative users with special skills and knowledge for a specific category [7]. *Expertise* is defined as "*the ability to discriminate meaningful classes of domain features and patterns, and to take decisions or actions that are appropriate to the class at hand*" [8]. Expert search and ranking problem has become an active research area in various application domains. However, it was studied in different contexts including the TREC enterprise track [9], question answering (QA) Websites [10], [5], [11], enterprises such as email communication [12], and scientific networks in digital libraries [13], [14].

Various methods exist for ranking users in networks. They can be classified into: Formal Models such as vector space models [15], Topic Models and probabilistic models e.g. [13], rank candidates according to the probability of a candidate being an expert [16]. Voting Models [17], [18], where an expert search problem modelled as a voting process: entities, documents for instance in the collection are ranked in response to the query then each retrieved document associated to a candidate is seems as a vote for that candidate to be retrieved in response to the query [18]. *Link-based analysis approaches*, aim at capturing the structural properties of the networks. PageRank [2] and HITS [3] and their variations are the most popular link-based algorithms recently applied for ranking nodes in networks, and have been adopted for ranking user expertise in web-based communities. They also called eigenvector centrality measures, they measure the centrality of a node as a linear combination of the centralities of the nodes to which it is connected [19].

According to the source of the expertise evidence, the expert search can be classified as: *Candidate-centric approaches* [4], they are also called profile-based approaches (query independent profiles), this involves building candidate profiles by associating entities with the candidates and then applying information retrieval (IR) techniques on the profiles [18]. All documents related to a candidate expert are collected together composing a single personal profile, the profiles represent the system's knowledge of expertise of each candidate. Afterwards, profiles are ranked using IR techniques. *Document-centered approaches* [17], [16] analyze the content of each document separately, and ranking is based on a subset of documents obtained using a query. Most of these methods basically utilize a simple probabilistic model with the assumption that the probability of expertness of a person is a sum of relevance probabilities of all related documents.

Our approach shares similarities with the existing link-based approaches, it represents the entities as a connected graph and then ranking scores are refined by some variation of PageRank or HITS. Similar to recent link-based approaches, we try to incorporate as much information into the graphs as possible. Campbell et.al [12] use link-based ranking method, namely, HITS and content-based analysis to rank users' expertise in an email network. They exploit links between senders and receivers of emails and email content analysis to rank users' expertise levels. Although they apply these algorithms in small artificial and email networks, they argue that using a link-based algorithm is effective in that it extracts more information comparing with content alone. Similarly to Campbell et.al, Dom et al. [20] studied PageRank and HITS to estimate expertise of users by analyzing email conversations. Despite they used HITS method for building the social network on email relations, they do not provide an explanation of the relationship between hubs in HITS and experts in an e-mail communication enterprise.

Zhang et al. [10] investigate also PageRank and HITS for finding users with high expertise in Java forum as a large scale online community, using social network analysis methods. They developed an algorithm called *ExpertisRank*, which is based on PageRank to produce ranking of users's expertise levels, considering only link structure. They find that various network structures affect the performance of these algorithms. Yang et al. [11] investigate the user influences ranking on dark web forums. Besides to link analysis, they consider features that reflect user influence in particular, *message content similarity* and *response immediacy* and developed *UserRank* algorithm. The Weighted social network is used for integrating these features to social network structure.

Jurczyk and Agichtein [5] adapt link analysis techniques, the focus is on estimating the authority of users that could be exploited for ranking, incentive mechanism design, and spam detection. They consider only the links between user nodes. Network is modeled by HITS, authors create edges themselves by answering questions. Similarly, Zhu et al [7] apply a link analysis approach for authority ranking in QA portals. They exploit the information in both target and relevant categories for authority ranking. First, they used topic models for inferring topics' category relevancy, then link analysis to rank user authority for a given category.

Close to our work is the *SPEAR* algorithm [21], a variation of HITS for a collaborative tagging system, similar to our work, it uses the mutual reinforcement that exists between the expertise of the users and the quality of the resources that the users possess. [21] argue that experts should identify valuable resources before other users do. They introduce the notion of discoverer and follower, that is, experts having the role of discovering valuable resources while other users will act as followers by finding the documents at later time. A timestamp of the tagging is considered for this algorithm.

Jiao et al. [22] propose the *ExpertRank*, an algorithm for ranking users in online communities. Two different methods for expert finding are combined in this algorithm: a domain knowledge driven method and a domain knowledge independent method. In the former, the expert relevance between an input query and a user profile that was built from all previous posts submitted by that user is being computed. In the later, a network based on user interaction is created and then PageRank is used for computing the authority scores of users. Two ways in which the two strategies are combined are tested: a linear combination and a cascade strategy in which first the expert relevance is used to rank users and then a percentage of these users are selected and are re-ranked by using the domain independent method. The algorithm enforces also countermeasures against spamming by weakening the impact of small group discussions by adding weights to the PageRank equation. Hence, they separate the expert relevance and expert authority. Similar to [22], we integrate domain driven information and domain independent information. For the domain driven information, we suggest to compute the similarity coefficient between the tags a user used in the system and the tags a certain entity has associated with it, while for the domain independent method we use the MHITS algorithm. HITS algorithm is known to be a domain dependent method of ranking web pages but we consider MHITS a domain independent method since we do not restrict our data to a root set when applying it (as it discussed in the evaluation case study). We consider that the restriction of the root set to a topic is still general enough to consider MHITS a domain independent method.

Regarding the ranking robustness problem, there are some related work that attempt to address this problem. Most of the proposed solutions are machine learning based algorithms. For instance, Bian and Liu [23] address the problem of noise/spaming in online social media environment and propose a machine learning algorithm that integrates user interactions and content relevance and improve ranking relevance in such noisy environment by adding noise to training data. Xin Li et al. [24] argue that not only relevance, but also ranking robustness should be considered in web ranking function selection. In contrast to these machine learning based algorithms, we came up with a completely different approach to expert ranking, which we will explain in Section IV.

III. A case study - Fake Media Detection System

We intent to study our approach in the context of a collaborative fake media detection system in a real-time media distribution network as the one proposed in [25]. The purpose of the collaborative fake media detection system is to detect fake media by means of the community. The target application areas are e.g. press agencies where the need to ensure the authenticity of media that originates from sources whose trustworthiness is hard to assess; the networked group of users that publish and rate media communicate via the near real-time XMPP protocol [26]. All group members can rate the authenticity of a media uploaded by a particular group member and should be published in the near future. Because of the openness of the system, we do not expect that all the ratings submitted are reliable. Therefore, in this system, we distinguish between three types of users as follows:

- *Honest users*, who rate media files in an honest way, but their competence is not good enough to judge well a media file authenticity.
- *Experts*, users who judge fairly and also have the competence to give a good feedback regarding the media file authenticity.
- *Malicious users*, users who intent to game the system and behave in an unfair way, trying to modify the outcome of the media file's authenticity for their profit.

We find it intuitive that the user's expertise in the context of a topic depends on the number of correctly rated by him media files belonging to that topic and on the trust she gains from his direct neighbors in the community. The users in the community can rate media with either fake (a value of 0.5) or authentic (a value of 1). Each topic is identified by a certain tag. When a media file is uploaded, more than one topic can be chosen to describe the media content.

In our approach, for obtaining a robust ranking, we extract the ratings, trust and user tags. Then, we apply the algorithm from Section IV and compute the expertise of a user with respect to a media category (topic) by using a linear combination of the *MHITS algorithm integrated with SumUp* as part of the domain independent model and the *tag relevance* as part of the domain dependent model.

IV. Expert Ranking Algorithm

This section proposes and discusses the expert ranking algorithm MHITS and the way it is integrated with SumUp method to limit Sybil attacks in order to ensure its robustness.

A. MHITS

In the original HITS algorithm, web pages are organized in a bipartite graph where the vertices correspond to the web pages and edges correspond to the links between them. Web pages act either as hubs or as authorities for a certain web search topic specified by one or more query terms. Authoritative pages are those that provide good information with respect to a given topic while hubs are pages that point to good authorities. The algorithm assumes that there is a mutually reinforcement relation between hubs and authorities (a page has high authority if many pages pointing to it have high hubness and a page has high hubness if many pages pointing to it have high authority) that results by applying the algorithm for several iterations until the hub and authority score values converge to their steady-state values.

In the context of web-based social networks, we aim to rank users according to their expertise with respect to a topic. A common practice in social networks is rating different entities, which according to the type of social network can be media files, products, movies, etc. We find it intuitive that the user expertise in the context of a topic will depend on the actions of that user and on the trust that other users of the community have in him/her.

In this work, we propose MHITS, a ranking algorithm approach that is inspired by the HITS algorithm. Considering the general way of applying HITS, we model a social network as a weighted bipartite graph. Figure 1 shows an abstract representation of the expert ranking network. The network has two types of nodes, *users* and *resources*. In addition, two types

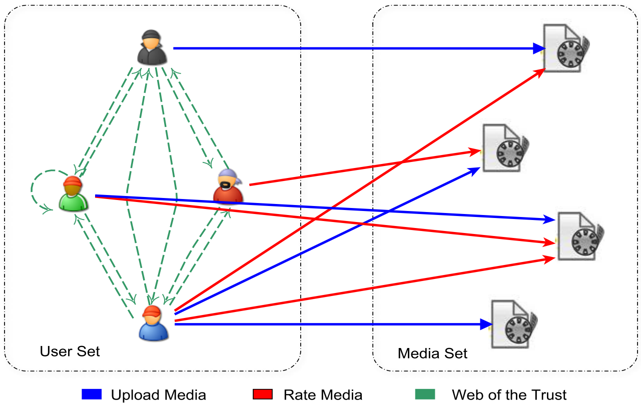

Fig. 1. Expert finding network

of edges, *ratings edges* that are directed edges representing the ratings users assign to these resources, and *trust edges*, which are edges between users in the community that represent the set of local trust values (in the range $[0,1]$).

In order to rank nodes (experts) in this network, we can see the network as a combination of two networks. The first network only contains users and resources nodes and ratings edges between them, so MHITS can be used here for authority transfer from resources to their raters. The second network is the trust network which contains only the trust edges. Our algorithm operates on both networks, passing information back and forth between the networks. For users, only the hubness scores are considered while for resources (media files in our case) the authority scores are taken into consideration. The mutual reinforcement relation refers to the fact that the expertise of a user depends on the way she rates and the authority of a rated resource comes from the way it is rated by users. This means that the authority of a media file is influenced by the ratings users assign to it and by the trust the users receive from their neighbors, at the same time, the expertise of users comes from the authorities that they rate. The authority and the hubness are computed by the following formulations.

$$a(m) = \sum_{u \in U(m)} h(u) * r(u) \tag{1}$$

$$h(u) = \beta * \sum_{m \in M(u)} a(m) * r(u) + (1 - \beta) * t(u) \tag{2}$$

where $a(m)$ is entity (media file) m's authority score; $U(m)$ denotes the set of users voting for a media file m; $r(u)$ represents the vote of user u for the given media file m. $h(u)$ is user u's hubness score; $M(u)$ denotes the set of media files to which a user u point; $t(u)$ denotes the average trust of directly connected users to the user u; and β is a parameter $\in [0,1]$ used to adjust the relative importance of hubness score and authority score, giving the percentage of the two terms in the hubness formula. By varying the value of β, a higher importance can be given to any of the two parts. In our algorithm, we set the value β at 0.3. In contrast to HITS, in MHITS the hubness and the authority values are weighted by the rating value and the average of trust as it shown in the hubness formula.

We start by considering the whole network of users and media files voted by the users. Accordingly, the initial steps of HITS, which compute the *root set* and extend it further to a *base set* are skipped. We also consider that more meaningful results can be obtained when applying the MHITS algorithm for a topic. In this case, the root set is obtained by considering only the voted entities that belong to that topic together with the users who rated them, and the base set is obtained by extending the user set to the whole trust network. The algorithm is applied iteratively as in the original HITS algorithm until the values are converging. The hubness values will represent the expertise of users and by ordering the users according to the expertise values we obtain the ranking of the users in the community.

B. Combining MHITS and a Tag Relevance Method

As mentioned in the related work section in this paper, we intent to mix a domain knowledge driven method and a domain knowledge independent method. The domain knowledge independent method was explained in detail in Section IV-A. The domain knowledge driven method will be used to find the relevance of a user's knowledge with regard to a given entity by using tags. The knowledge of an user can be reflected by all previous tags the user submitted to different entities. As consequence, a profile describing what a user knows is built by merging all tag posts. Afterwards, the similarity coefficient between the candidate profile and the tags assigned to a specific entity can be computed.

The algorithm is applied iteratively until the values are converging. The hubness scores will represent the users' expertise and by ordering the users according to the expertise values, the ranking of users in the community is obtained.

The two methods can be combined as follow: after applying the domain independent method to classify users in the system according to their expertise with regard to a topic, the domain dependent method is further applied to get a better classification of users for each rated entity. So the expertise of a user according to a specific rated entity will be a linear combination between the expertise deducted using the domain independent method algorithm and the user's expert relevance.

However, in this work we are interested in evaluating the MHITS algorithm part and its robustness, while the tag relevance method and the combination between the two methods are suggested as further work.

C. Attack Modeling

In this paper, we also tackle the challenge of how to ensure the robustness of MHITS algorithm. Robustness in this case, means returning fair ranking results despite the attacks that are being used against the algorithm. A very popular attack against online content rating systems is the Sybil attack. The Sybil attack can be modeled as follow: an attacker controls one or multiple user IDs, and each of these user IDs represents a malicious user. The malicious users controlled by the same attacker collude with each other composing a group, and try to boost or downgrade the media file's reputation. Therefore, all ratings provided by malicious users are considered as unfair ratings. To ensure the robustness of our proposed ranking algorithm, the defense solution should be able to detect Sybil attacks and exclude them from the media file's reputation computation and expert ranking results. Moreover, it should be able to get accurate media file's reputation even when it is under attack.

D. The Robust Expert Ranking: Integration of MHITS and SumUp

In this section, we present a way to limit the effect of Sybils in a online content rating system by using the users' trust network. We have studied several methods for this purpose such as SybilLimit [27], SybilGuard [28] and SumUp [6]. While SybilLimit and SybilGuard are two decentralized algorithms, SumUp is a centralized vote collector model and is the only one that directly addresses the vote aggregation problem in content rating systems. It was proved that when all nodes vote, SumUp leads to much lower attack capacity than SybilLimit even though both have the same $O(logn)$ asymptotic bound per attack edge [6]. Therefore, we chose the SumUp method for mitigating the impact of Sybils on expert rankings.

Formally, SumUp is a Sybil resilient online content rating system that uses the trust network among users to defend against Sybil attacks. It uses the concept of *max-flow*. According to [6], the flow concept is critical to limit the number of ratings that malicious users can propagate for a media file. When we apply it in the context of the media votes aggregation problem, the objective is to compute the max-flow in the given trust network from the votes collector to the set of voters. Vote-flow paths to trusted voters are congested at links close to the collector while paths to Sybil voters are also congested at far-away attack edges. The *adaptive vote flow* technique is used to collect as many as trusted votes and as few as potentially bogus votes.

SumUp uses the *adaptive vote flow* technique to collect as many as trusted votes and as few as potentially bogus votes. Three key ideas are used in the adaptive vote flow computation. First, the algorithm restricts the maximum number of votes collected on a media file to a value C_{max}. Notice that as C_{max} is used to assign the overall capacity in the trust graph, a small C_{max} results in less capacity for the attacker. SumUp can adaptively adjust C_{max} to collect a large fraction of trusted votes on any given media file. The second key of SumUp is a *capacity assignment*, i.e. how we assign capacities to each trust link to collect a large fraction of honest votes and only a few bogus votes. To this end, the vote collector distributes C_{max} tickets downstream in a breath-first search (BFS) manner within the trust graph. The capacity assigned to a link is the number of tickets distributed along the link plus one. The basic idea of capacity assignment is to construct a *vote envelope* around the source, which contains C_{max} nodes that can be viewed as entry points. Beyond the envelop, all links have capacity value 1. An edge attack beyond the envelope can propagate at most 1 vote regardless of the number of Sybil IDs behind that edge.

Three key ideas are used in the adaptive vote flow computation. First, the algorithm restricts the maximum number of votes collected on a media file to a value C_{max}. Notice that as C_{max} is used to assign the overall capacity in the trust graph, a small C_{max} results in less capacity for the attacker. SumUp can adaptively adjust C_{max} to collect a large fraction of trusted votes on any given media file. The second key of SumUp is a *capacity assignment*, i.e. how we assign capacities to each trust link to collect a large fraction of honest votes and only a few bogus votes. To this end, the vote collector distributes C_{max} tickets downstream in *a breath-first search* (BFS) manner within the trust graph. The capacity assigned to a link is the number of tickets distributed along the link plus one. The basic idea of capacity assignment is to construct a *vote envelope* around the source, which contains C_{max} nodes that can be viewed as entry points. Beyond the envelop, all links have capacity value 1. An edge attack beyond the envelope can propagate at most 1 vote regardless of the number of Sybil IDs behind that edge. Finally, a user voting history is leveraged to restrict the voting power of adversarial nodes, who continuously propagate bogus votes.

Figure 2 presents the integration of the MHITS and SumUp algorithms. Given as input the community trust network and the ratings network, the MHITS algorithm is run first. The resulted ranking of the experts is used to choose the first expert in the ranking as being the source node for the SumUp algorithm. Starting form the source node, levels are assigned to each node in the trust network in a BFS manner. The next step is the pruning of the network that will take care that no node in the trust graph will have more than a given threshold of in nodes. Then, the adaptive vote flow mechanism is made for each rated content present in the system in order to limit the number of bogus votes, without affecting the number of honest votes that can be gathered. This is done by.

- Restricting the maximum number of votes collected for a resource to a certain capacity value.
- Keeping negative history for nodes, which is used further to adapt the voting capacity in the trust network.
- After computing the capacity value, the given voting capacity is distributed in the network starting from the source node in a BFS manner.
- The votes are collected and an aggregated vote is computed.
- The votes that are far from the value of the aggregated vote are marked as being bogus votes, and the negative history for the path from the voter to the source node is increased.
- Those links for which the accumulated negative history exceeds five times the assigned capacity will be deleted and new links will be added from the trust network before pruning.

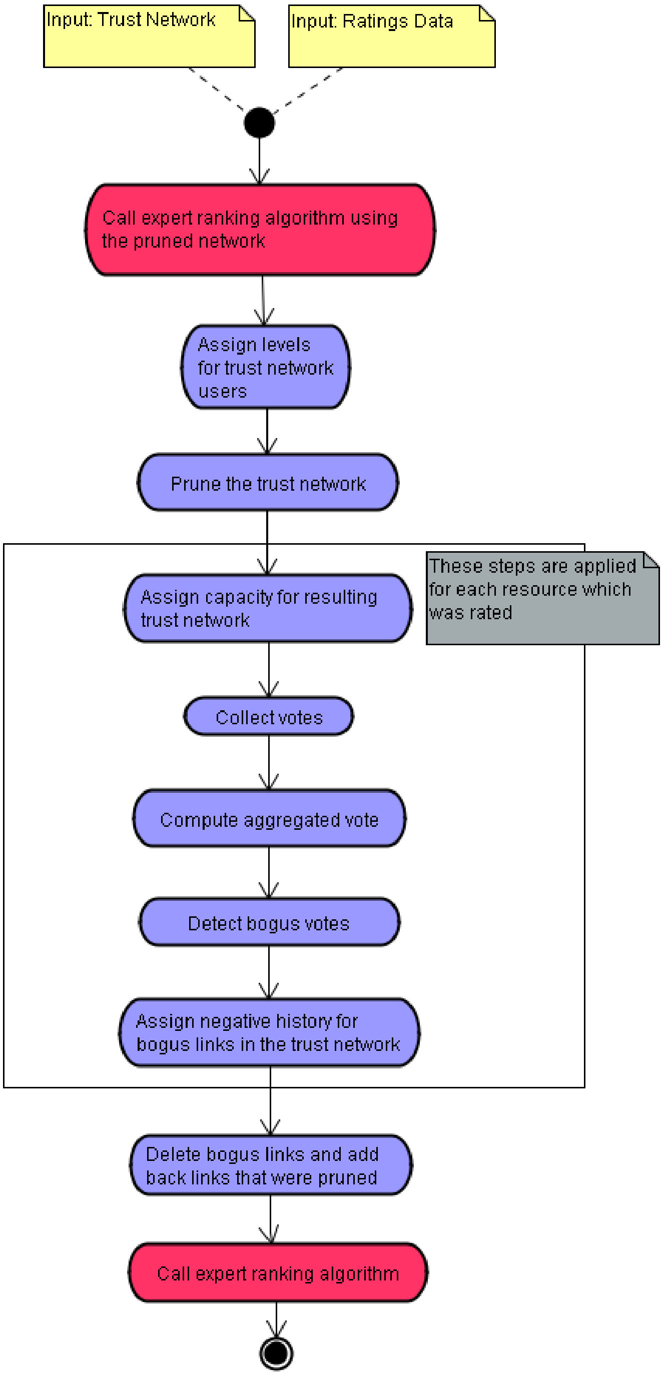
Input: Trust Network
Input: Ratings Data
Call expert ranking algorithm using the pruned network
Assign levels for trust network users
Prune the trust network
These steps are applied for each resource which was rated
Assign capacity for resulting trust network
Collect votes
Compute aggregated vote
Detect bogus votes
Assign negative history for bogus links in the trust network
Delete bogus links and add back links that were pruned
Call expert ranking algorithm

Fig. 2. Activity diagram presenting the integration of MHITS with SumUp

After all these steps on the modified trust network are completed, the MHITS is run again to recompute the expertise ranking of users.

V. Evaluation

In this section, we evaluate the algorithm presented in Section IV with respect to the community-based fake media detection scenario and report the obtained results.

A. Evaluation Metrics

To evaluate our ranking strategy, we adopt the following metrics.

- *Precision@K* computes for a given result of ranked users, the fraction of relevant results in the top K results. The higher the precision, the better the performance is. We use Precision@K for characterizing the precision of top K ranking lists, i.e. the accuracy of the users listed among the first K users of a ranking. We compare the results of the expert ranking algorithms with the ranking of experts resulted by counting the number of fair votes.
  Assume $TopK$ and $TopK'$ are the retrieved users of ranking lists r and r' respectively, then the Precision@K is defined as:
  $$P@K = \frac{|TopK' \cap TopK|}{K} \tag{3}$$
- *Spearman's rank correlation coefficient*, Spearman's correlation coefficient is a non-parametric measure of statistical dependence between two ranked lists. It assesses how well the relationship between the two lists can be described using a monotonic function. If there are no repeated data values, a perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other. The 0 value stands for no correlation between the two compared ranked lists. Spearman's rank correlation coefficient is described by the following formula:
  $$\rho = 1 - \frac{6 * \sum d_i^2}{n(n^2 - 1)} \tag{4}$$
  where $d_i = x_i - y_i$ represents the distance between the ranks of each observation on the two ranking lists and n is the size of the sample. x_i, y_i denote the ranking of i in x and y.

For computing this measure, the first n ranked users from the two ranking lists are chosen. Then, only users that appear in both lists are considered. We considered only the first 30 ranked users for all the ranked lists. From these 30, we keep only those common in all ranking lists and re-rank the users. It is important to note that the ranking of users is done by counting the number of fair ratings in the community, assigns the same position to all users that have the same number of fair votes (we have tied values). Therefore, the rank of the users (that have the same number of fair votes) is computed as the mean of what their ranks would otherwise be. After assigning a rank for each common user in the all considered ranking lists, the correlation between the ranking considering the fair votes and any of the other rankings is computed by calculating d_i and then ρ.

B. Evaluating the Robustness of Ranking

During the evaluation phase, we intend to analyze the influence of the MHITS approach on the ranking of experts in the system and to verify if the modifications that we have made (using trust in MHITS formula and the integration with

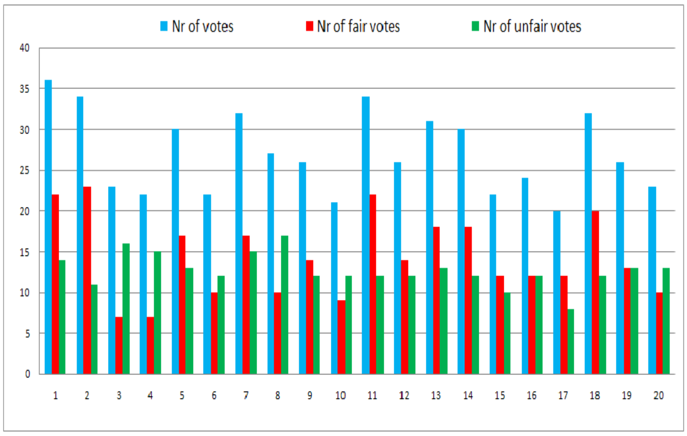

Fig. 3. Media ratings distribution

SumUp) show better results. We mainly address the following questions in the experiments:

- Is the introduction of trust in the MHITS formula influencing the algorithm in a positive way?
- Is MHITS in combination with SumUp capable to limit the Sybil attack?

C. Experiment settings

The MHITS and its integration with SumUp was developed for a collaborative fake media detection system, whose purpose is to detect fake media by means of community (cf. Section III). In order to conduct the evaluation on this system, a critical mass of users and media files should be present in the system. Still at the time being, the community did not gain enough members yet. Because of this reason and also because of simulating attacks on a real system is difficult to achieve, we conduct our experiments on a synthetic dataset (synthetic social network) that was created using the BarabasiAlbert model [29], which is an algorithm, for generating random scale-free networks using a preferential attachment mechanism. We further use a rating dataset that follows a power law distribution. For this purpose, an implementation of the Zipf's distribution [30] was used.

Before explaining the way trust was assigned in the network, we want to remind the reader the expert definition in our system from Section IV. In order to check that the algorithm is ranking experts according to the conditions mentioned, the trust was assigned in the network considering the number of fair ratings the user has done. In this way, users that rate fairly many media files, gain high trust values from their neighbors.

The created dataset is composed of 300 users out of which 250 are expose a honest behavior, 3000 trust edges and 800 ratings. The set of media files used in the evaluation is made out of 20 media files, out of which half are known to be authentic and half are known to be fake. These media files are randomly assigned to several topics (sports, nature, science, fashion, war). The vote distribution for the media files is shown in Figure 3. The initial dataset only contains the honest users and the trust network characteristics are presented in Table I.

Nr Vertices	Nr Edges	Nr Comp.	% of Main Comp.	Clustering. Coeff.	Avgerage shortest path
250	1482	1	100%	0.18	2.61

TABLE I
SYNTHETICALLY CREATED USER TRUST NETWORK CHARACTERISTICS

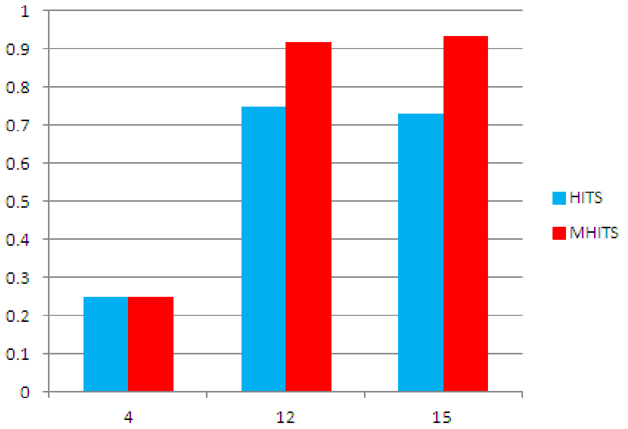

Fig. 4. Precision@K. Our MHITS significantly outperforms HITS for $K = 12$ and $K = 15$

D. Experimental Results

To be able to answer the first question, we assumed that all users in the network behave fairly. We compared the performance of HITS and MHITS algorithms with the result obtained with the ranking of users according to the number of fair ratings each user had in the system. It is worth to note that in fact, the original HITS algorithm has one additional parameter added, which is the ratings of the users.

To evaluate our MHITS algorithm and its integration with SumUp, we employed the following evaluation metrics: *Precision@*K and *Spearman's rank correlation coefficient*. As mentioned above, precision @ K computes for a given result of ranked users the fraction of relevant results in the top K results while the Spearman's rank correlation coefficient measures a statistical dependence between two ranked lists and assesses how well the relationship between the two list can be described using a monotonic function. The comparison between the two algorithms can be seen in Figure 4 by using the Precision@$K = 4, 12, 15$ metric. Table II presents the Spearman's correlation for the top K users ranked by the HITS and MHITS algorithms. From these results, one can observe that the MHITS is more precise than HITS for $K = 12, 15$ as it also has the trust incorporated into its formula.

To answer the second question, i.e to evaluate the effects of MHITS integrated with SumUp, we simulated different Sybil

	HITS	MHITS
Spearman n=15	0.87	0.93

TABLE II
SPEARMAN CORRELATION COEFFICIENTS

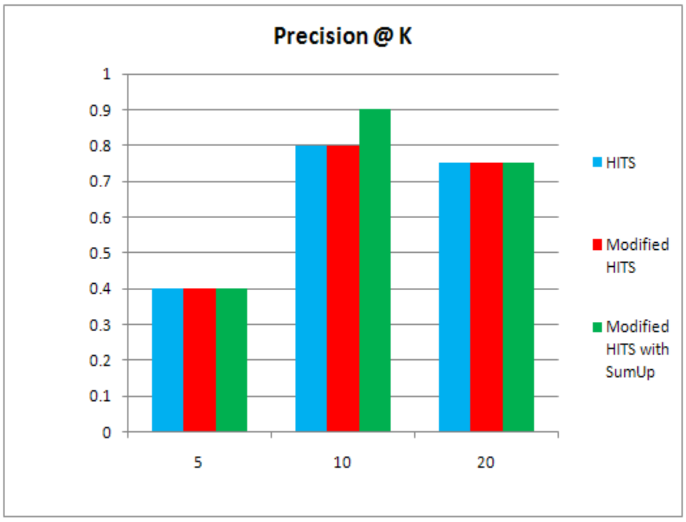

Fig. 5. The Precision@K comparison of HITS, MHITS and MHITS integrated with SumUp

	HITS	MHITS	MHITS & SumUp
Spearman n = 20	0.52	0.68	0.58

TABLE III
SPEARMAN CORRELATION COEFFICIENTS

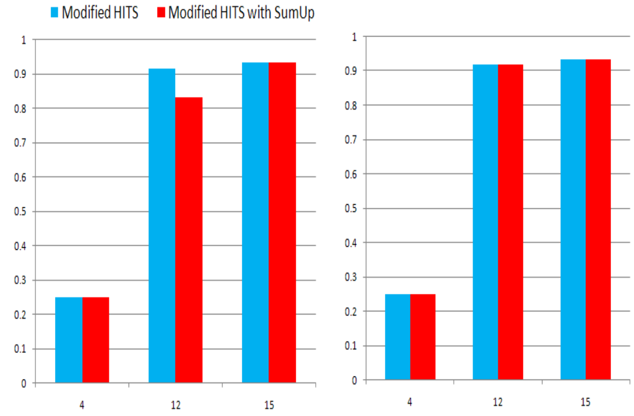

Fig. 6. The Precision@K comparison of MHITS and MHITS integrated with SumUp

K	MHITS 20%	MHITS & SumUp 20%	MHITS 50%	MHITS& SumUp 50%	MHITS 100%	MHITS & SumUp 100%
12	0.91	0.91	0.27	0.33	0.08	0.08
15	0.93	0.93	0.33	0.40	0.06	0.06

TABLE IV
PRECISION@$K = 12, 15$ FOR DIFFERENT SYBIL VOTES AMOUNTS COMPUTED FOR MHITS AND MHITS WITH SUMUP

attacks by varying the following parameters:

1) The number of Sybils in a group
2) The number of attack edges
3) The number of Sybil groups

Malicious users are introduced into the system by injecting it by: First, 10% more nodes representing a group of Sybils. This group of Sybils is connected to the initial trust network by 4 attack edges. The comparison can be seen in Figure 5. From the Figure 5, we can see that the MHITS in combination with SumUp outperforms HITS and MHITS for $K = 10$. For $K = 20$ the precision decreases rapidly. This is due to the fact that some Sybil users have already entered the ranking for $K = 20$, because of their high local trust values. Therefore, the precision is decreased. The same results can be seen from the Speraman's correlation coefficients presented in Table III.

We compared the results obtained when the Sybil number is increased from 10% to 20% with 8 attack edges and second, with triple (24) attack edges. From Figure 6, one can see that the precision in the second case is not decreased.

Next, when the number of Sybil votes was increased from 3% to 17%, the outcome on the ranking results was the same. We also tried an extreme case by increasing the Sybil votes from 17% out of the total number of fair votes to 50% and then to 100% keeping the number of attack edges and the Sybil numbers constant. The result is depicted in table IV. As it is expected, due to the high number of Sybil rates, the Precision@K decreases dramatically as more Sybil ratings are introduced into the system, the precision@K reaches close to zero.

VI. Conclusions and Outlook

In this paper, we have presented a robust and effective approach which uses entity (media) rating and trust social networks to rank users' expertise levels in online media communities. In particular, we have developed a modified HITS (MHITS) and integrated it with SumUp algorithm to support its robustness to Sybil attacks. We have performed an empirical evaluation of this method on community-based fake media detection scenario, demonstrating its effectiveness for discovering users with high expertise levels. Since our experiments only study the problem of expert ranking in the community-based fake media detection system, we can not claim that our approach will work similarly in other domains, but the approach will be useful in many contexts such as e-commerce communities. Some issues still remain to be addressed in the future: we have to investigate our proposed approach on other online communities. It would be desirable to carry out a number of experiments on real datasets. We think that different networks structures and ratings ways and scopes may have an impacts in our proposed approach's effectiveness. We have to consider the dynamics of the web-based community system hence, temporal analysis such as time series analysis could be applied. For instance, change detectors such those applied in quality control and Spam filtering systems to identify changes in probability distribution of a random process, could be applied for determining the media under attack, identifying the suspicious time intervals, and then clustering techniques could be used to cluster groups of malicious users and removing unfair ratings. Moreover, further attack strategies and corresponding robust ranking methods are to be explored.

Acknowledgment

This work was supported by B-IT Research School.

References

[1] J. R. Douceur, “The sybil attack,” in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS ’01. Springer-Verlag, 2002, pp. 251–260.

[2] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Comput. Netw. ISDN Syst.*, vol. 30, pp. 107–117, April 1998.

[3] J. M. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[4] X. Liu, W. B. Croft, and M. Koll, “Finding experts in community-based question-answering services,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, ser. CIKM ’05. ACM, 2005.

[5] P. Jurczyk and E. Agichtein, “Discovering authorities in question answer communities by using link analysis,” in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, ser. CIKM ’07. ACM, 2007, pp. 919–922. [Online]. Available: http://doi.acm.org/10.1145/1321440.1321575

[6] N. Tran, B. Min, J. Li, and L. Subramanian, “Sybil-resilient online content voting,” in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, ser. NSDI’09, 2009, pp. 15–28.

[7] H. Zhu, H. Cao, H. Xiong, E. Chen, and J. Tian, “Towards expert finding by leveraging relevant categories in authority ranking,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*, ser. CIKM ’11. ACM, 2011, pp. 2221–2224.

[8] J. Freeman, W. Stacy, J. Macmillan, and G. Levchuk, “Capturing and building expertise in virtual worlds,” in *Proceedings of the 5th International Conference on Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience: Held as Part of HCI International 2009*, ser. FAC ’09. Springer-Verlag, 2009, pp. 148–154.

[9] N. Craswell, A. de Vries, and I. Soboroff, “Overview of the TREC-2005 Enterprise Track.” in *TREC 2005*, 2006.

[10] J. Zhang, M. S. Ackerman, and L. Adamic, “Expertise networks in online communities: structure and algorithms,” in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW ’07. ACM, 2007, pp. 221–230. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242603

[11] C. C. Yang, X. Tang, and B. M. Thuraisingham, “An analysis of user influence ranking algorithms on dark web forums,” in *ACM SIGKDD Workshop on Intelligence and Security Informatics*, ser. ISI-KDD ’10. ACM, 2010, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/1938606.1938616

[12] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom, “Expertise identification using email communications,” in *Proceedings of the twelfth international conference on Information and knowledge management*, ser. CIKM ’03. ACM, 2003, pp. 528–531. [Online]. Available: http://dx.doi.org/10.1145/956863.956965

[13] Y. Tu, N. Johri, D. Roth, and J. Hockenmaier, “Citation author topic model in expert search,” in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING ’10. Association for Computational Linguistics, 2010, pp. 1265–1273.

[14] S. D. Gollapalli, P. Mitra, and C. L. Giles, “Ranking authors in digital libraries,” in *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, ser. JCDL ’11. ACM, 2011, pp. 251–254.

[15] G. Demartini, J. Gaugaz, and W. Nejdl, “A vector space model for ranking entities and its application to expert search,” *Advances in Information Retrieval*, vol. 5478, pp. 189–201, 2009.

[16] K. Balog, L. Azzopardi, and M. de Rijke, “Formal models for expert finding in enterprise corpora,” in *SIGIR*. ACM, 2006, pp. 43–50.

[17] C. Macdonald and I. Ounis, “Voting for candidates: adapting data fusion techniques for an expert search task,” in *Proceedings of the 15th ACM international conference on Information and knowledge management*, ser. CIKM ’06. ACM, 2006, pp. 387–396.

[18] C. Macdonald, D. Hannah, and I. Ounis, “High quality expertise evidence for expert search,” in *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ser. ECIR’08. Springer-Verlag, 2008, pp. 283–295.

[19] S. Kameshwaran, V. Pandit, S. Mehta, N. Viswanadham, and K. Dixit, “Outcome aware ranking in interaction networks,” in *Proceedings of the 19th ACM international conference on Information and knowledge management*, ser. CIKM ’10. ACM, 2010, pp. 229–238.

[20] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang, “Graph-based ranking algorithms for e-mail expertise analysis,” in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, ser. DMKD ’03. ACM, 2003, pp. 42–48.

[21] M. G. Noll, C.-m. Au Yeung, N. Gibbins, C. Meinel, and N. Shadbolt, “Telling experts from spammers: expertise ranking in folksonomies,” in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR ’09. ACM, 2009, pp. 612–619. [Online]. Available: http://doi.acm.org/10.1145/1571941.1572046

[22] J. Jiao, J. Yan, H. Zhao, and W. Fan, “ExpertRank: An Expert User Ranking Algorithm in Online Communities,” in *Proceedings of the 2009 International Conference on New Trends in Information and Service Science*. IEEE Computer Society, 2009, pp. 674–679.

[23] J. Bian, Y. Liu, E. Agichtein, and H. Zha, “A Few Bad Votes Too Many? Towards Robust Ranking in Social Media,” in *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, ser. AIRWeb ’08. ACM, 2008, pp. 53–60. [Online]. Available: http://portal.acm.org/citation.cfm?id=1451983.1451997

[24] X. Li, F. Li, S. Ji, Z. Zheng, Y. Chang, and A. Dong, “Incorporating robustness into web ranking evaluation,” in *CIKM*. ACM, 2009, pp. 2007–2010.

[25] D. Renzel, K. A. N. Rashed, and R. Klamma, “Collaborative Fake Media Detection in a Trust-Aware Real-Time Distribution Network,” in *Proceedings of the 12th International Workshop of the Multimedia Metadata Community, the 2nd Workshop focusing on Semantic Multimedia Database Technologies 2010, collocated with the 5th International Conference on Semantic and Digital Media Technologies (SAMT2010)*, H. Kosch, R. Klamma, M. Lux, M. Spaniol, and F. Stegmaier, Eds., vol. 680. CEUR-WS.org, Dec 2010, pp. 17–28.

[26] P. Saint-Andre, “RFC 3920 – Extensible Messaging and Presence Protocol (XMPP): Core,” Jabber Software Foundation, Tech. Rep., 10 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3920.txt

[27] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, “SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks,” in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2008, pp. 3–17.

[28] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, “Sybilguard: defending against sybil attacks via social networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 267–278, 2006.

[29] A. Barabási and R. Albert, “Emergence Of Scaling In Random Networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[30] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA), 1949.