A Framework for Unsupervised Spam Detection in Social Networking Sites

Maarten Bosma, Edgar Meij, and Wouter Weerkamp

ISLA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
maarten@bosma.de, edgar.meij@uva.nl, w.weerkamp@uva.nl

Abstract. Social networking sites offer users the option to submit user spam reports for a given message, indicating this message is inappropriate. In this paper we present a framework that uses these user spam reports for spam detection. The framework is based on the HITS web link analysis framework and is instantiated in three models. The models subsequently introduce propagation between messages reported by the same user, messages authored by the same user, and messages with similar content. Each of the models can also be converted to a simple semi-supervised scheme. We test our models on data from a popular social network and compare the models to two baselines, based on message content and raw report counts. We find that our models outperform both baselines and that each of the additions (reporters, authors, and similar messages) further improves the performance of the framework.

1 Introduction

In the last decade there has been a shift on the Internet from the static, editor-controlled Web 1.0 to the user-driven Web 2.0 paradigm. Web 2.0 platforms offer the possibility to share, exchange, and create any kind of content which makes these platforms an obvious target for spammers. Among the most popular social media platforms are social networking sites, like Facebook, Google+, and many local variants, in which users can connect to other users (e.g., "become friends") and post messages to each other and on other pages within the network (e.g., group pages for users with shared interests). Most social networking sites rely on their users not only to generate content, but also to fight spam and other inappropriate content. As Caroline Ghiossi of Facebook puts it: "With billions of pieces of content being shared on Facebook [...] preventing spam isn't easy. Just as a community relies on its citizens to report crime, we rely on you [our users] to let us know when you encounter spam." [7]

Within social networks, spam can be found in publicly visible pages (groups, celebrity profiles, etc.), on profile pages, and in private inboxes. Most networking sites allow their users to issue a report on all of these levels when they feel a message is spam or otherwise inappropriate (e.g., violent or sexist language). These *user spam reports* are the main ingredient of this paper. We explore the usefulness of user spam reports in classifying spam in social networks. More precisely, we present a framework that indicates the likelihood of a message being spam, based on user spam reports.

The simplest spam detection method that makes use of user spam reports is one that uses the raw number of reports as the spam score of a message. The main disadvantage

of this method, however, is that many user spam reports will not be acted upon due to a lack of reports, or acted upon too late as it will take a while for messages to accumulate sufficient reports. We require a more elaborate method that can deal with these issues.

Our framework tries to improve over the simple counting method by taking into account not only the number of spam reports, but also the spam reporters and the authors of the messages. The core assumption that underlies our framework is that spam messages are more likely to be spam when they have been reported as spam by several "trustworthy" users. We define a trustworthy user as one who issued spam reports for a large number of messages that are actually spam messages. An obvious way of formalizing this assumption is to use (a variation of) the HITS link analysis algorithm (see Section 3), in which hubs and authorities are used to propagate scores. We refine our initial model by propagating spam scores not only via spam reporters, but also via messages with similar content and messages written by the same author.

We believe that an unsupervised framework as introduced above is more suitable for identifying spam messages than a static, supervised classifier. The unsupervised framework does not depend on the content of spam messages and can therefore detect new types of spam, without the need for retraining. However, supervised methods have been shown to perform well on the task of classifying spam and we acknowledge that fact by (i) including it as a baseline and (ii) combining the two approaches using a semi-supervised variant of our framework.

We find that the framework's scalability is one of its main advantages: given the sheer amount of messages being created every day (e.g., 30 billion pieces of content were created on Facebook every month in 2010[1]), scalability is a key aspect of any spam classification method for social networking sites. A disadvantage of any report-based spam detection system is that it is prone to abuse, e.g., when a large group of users tries to get rid of certain messages by reporting these as being spam. This problem can be partially prevented by ranking reports according to their trustworthiness, which is supported by our framework.

We try to answer the following research questions in this paper: (i) Given a set of (spam) messages and user spam reports, can we use a HITS-based algorithm to improve over a supervised baseline as well as over a count-based method? (ii) Given our initial HITS-based model, can we improve spam detection performance by including information from the author of the message? (iii) Based on the observation that certain spam messages look alike, can we improve performance further by including similarity links between messages? Finally, (iv) can we improve performance of our models by making them semi-supervised?

We find that the three instantiations of our framework, based on reports alone, reports and authors, and reports, authors, and similar messages, all improve spam ranking performance over the two baselines. Moreover, performance improves with each additional piece of information, resulting in the Similarity-Author-Reporter Model being the best performing model. In the semi-supervised setting we find that only the model using the similar messages improves over its unsupervised variant and this model gives the best overall results.

[1] http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers/

The remainder of the paper is organized as follows. In Section 2 we discuss related work. The main innovation of this paper is in the models in Sections 3 and 4. We test our models using the setup of Section 5 and present the results in Section 6. Finally, we analyze the results in Section 7 and conclude in Section 8.

2 Related Work

To the best of our knowledge no work has been done aimed at weighting individual user spam reports to identify spam messages in social networks. However, some researchers address the general topic of using machine learning algorithms to classify users or resources as spam [1, 4, 9, 12, 16, 19]. Most of this previous work uses supervised machine learning methods to classify messages based on features related to the message contents or features of the network of the spammer. Below we discuss work related to spam detection and resource ranking in social networks, link analysis methods, and trust frameworks for social networks.

DeBarr and Wechsler [4] use centrality in the social graph of a social networking site to predict if a user is likely to post spam in a social network. Wang [16] uses graph-based metrics to improve spam classification on a microblogging platform. Mehta et al. [14] study statistical methods to identify voting papers, which are intended to deceive collaborative filtering systems. These supervised methods, however, have the weakness that they are static and therefore more easily deceivable than unsupervised methods.

For our framework we took inspiration from resource ranking and classification in social networks, and from link analysis methods. Bian et al. [2] present a semi-supervised framework based on logistic regression that uses the mutual reinforcement principle to rank resources in a social network. Lu et al. [13] propose a linear regression framework, which predicts the quality of a review in an e-commerce portal, based on language features. They extend the framework by incorporating constrains to the formalism based on properties of the social network of the reviewer. Two prominent algorithms for the analysis of link structure between documents are (i) HITS [11] and (ii) PageRank [15]. Modified versions of both HITS and PageRank have been used to model expertise, authority, reputation and trust in social networks.

Zhang et al. [17] analyze data from an online forum to find expert users using both HITS and PageRank. Jurczyk and Agichtein [10] apply the HITS algorithm to a cQA portal. They show that there is a positive correlation between the HITS score and quality of the answer. Campbell et al. [3] and Dom et al. [5] study the performance of the link-based methods in order to identify expert authors in a network of email exchanges. A spam report can be seen as a token of distrust. Guha et al. [8] and Ziegler and Lausen [18] introduce frameworks that model the propagation of trust and distrust in a social network using HITS and PageRank. Guha et al. [8] note that a distrust relationship is fundamentally different from a trust relationship: while it is plausible to assume a trust relation is transitive, this does not hold for a distrust relation.

In our work we use a modified version HITS to model the relation between messages, authors, user spam reports, and reporters. We introduce the HITS-based models in the next section.

3 Spam Detection Framework

Our spam detection framework is based on HITS and uses the links between messages and other objects to propagate spam scores. We have several choices as to which objects play a role in our framework. In this section we introduce three instantiations of our spam detection framework, each of which builds on the previous instance. The models assign spam scores to each message in the dataset, allowing us to rank messages.

In Section 3.1 we introduce our Reporter Model, using only messages and reporters. The Author-Reporter Model in Section 3.2 builds on the previous model by mixing in the authors of messages. Finally, in Section 3.3 we introduce the Similarity-Author-Reporter Model, which uses the similarity between messages as additional evidence.

3.1 Reporter Model

In the Reporter Model we interpret the problem space as a bipartite graph. The graph has two node types: (i) reporters and (ii) messages. Directed edges go from reporters to messages, indicating that the reporter issued a user spam report regarding this particular message. Figure 1 shows the Reporter Model graph, to which we apply HITS. This

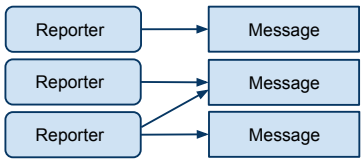

Fig. 1: Example graph of the Reporter Model.

algorithm calculates two scores: a hub score and an authority score. The two are defined recursively in terms of each other. In our setting, the authority score is replaced by the spam score and the hub score is replaced by the reporter score. The spam score $S(m)$ for a given message node m in the graph is the sum of all hubs (i.e., reporters) that are connected to it:

$$S(m) = \sum_{r \in R_m} H(r), \tag{1}$$

where R_m represents the set of all reporter nodes that are connected to message node m. The report score H of a node r is the sum over all authorities (messages) connected to this node:

$$H(r) = \sum_{m \in M_r} S(m), \tag{2}$$

where M_r represents all messages that are connected to reporter node r. An intuitive interpretation for this model is that the hub score $H(r)$ of a reporter r represents her trustworthiness. The authority (spam) score can thus be seen as a weighted version of a spam score based on raw report counts.

The final hub and authority scores are found using an iterative procedure. First, the reporter scores are initialized uniformly. The new message scores are calculated using Eq. 1 and normalized. Based on the new message scores, we invoke Eq. 2 to calculate new reporter scores. These steps are repeated until convergence occurs, which is defined as the total change compared the previous iteration being less than 0.0001.

3.2 Author-Reporter Model

We extend the graph of the Reporter Model by introducing a third node type: the author of a message. Each message has exactly one author, which is encoded using a directed edge from the author to the message. The reporter nodes and author nodes do not overlap, so even if the same user is the author of one message and reporter of another messages, it is still modeled using two different reporter and author nodes. Figure 2 depicts the new Author-Reporter Model graph.

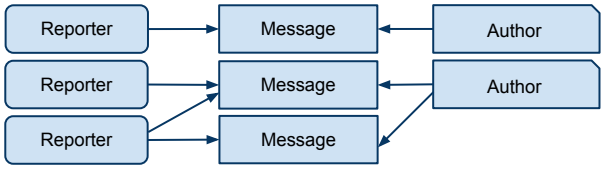

Fig. 2: Graph of the Author-Reporter Model.

Adding authors to the model leads to a change of Eq. 1, which now includes an author score:

$$S(m) = A(a_m) + \sum_{r \in R_m} H(r), \tag{3}$$

in which a_m denotes the author of message m. The author score A is similar to the reporter score H in Eq. 2, except that we now sum over all messages written by author a (i.e., M_a).

$$A(a) = \sum_{m \in M_a} S(m), \tag{4}$$

in which M_a indicates all messages authored by author a. The intuition behind this score is that an author, who posted lots of spam messages, is more likely to post another spam message than a user who posted no spam messages at all.

3.3 Similarity-Author-Reporter Model

Our final model also includes links between messages, that is, links between messages with similar content. We opt to model content similarity using cosine similarity and we include edges between a message and the n most similar messages, given that the similarity score is higher than zero. For now we set $n = 10$, but we revisit this setting

in Section 7. Note that this relation is not necessarily symmetric. Figure 3 shows the graph of the Similarity-Author-Reporter Model.

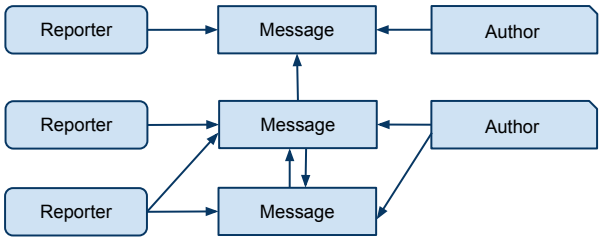

Fig. 3: Graph of the Similarity-Author-Reporter Model.

In this case, we calculate spam scores using

$$S(m) = (1-\gamma)[A(a_m) + \sum_{r \in R_m} H(r)] + \gamma \sum_{n \in I_m} S'(n), \tag{5}$$

where S' denotes the spam score from the previous iteration, I_m is the set of messages similar to message m, and γ is a free parameter. For now we set $\gamma = 0.35$, resulting in most of the spam score for message m coming from a message's own characteristics and the rest from it's "neighbors." We revisit the influence of this parameter in Section 7. The intuition behind adding similar messages is that messages with similar content should also have a similar spam score, and therefore, spam scores should be propagated between those messages.

4 Semi-supervised Variants

So far we have introduced three spam detection models that are unsupervised. We observe, however, that each of the three models can be extended by making them semi-supervised. The semi-supervised scores can be seen as a way to introduce value judgements by the proprietor of the social networking site so that comments concerning controversial issues, for example, can be assigned a low value spam value. This could avoid a "tyranny of the minority" situation, in which a small group reports messages that talk about a specific topic.

We implement semi-supervised variants of our three models by fixing the spam score for a message node for which we know whether or not it is spam. We then calculate the spam scores for all other nodes and apply the maximum calculated spam score to the fixed node(s). A message node m for which we know for certain that it is not a spam message, is assigned a value of 0, i.e., $S(m) = 0$.

5 Experimental Setup

Our dataset consists of messages, spam reports, and users from the largest Dutch social networking site, Hyves.[2] Each spam report concerns one single message on the profile page of a public figure or on a publicly accessible group page. The dataset consists of 28,998 spam reports, collected during the period from January 2010 to January 2011. The spam reports cover 13,188 unique messages and are generated by 9,491 unique reporters/users.

We find that the dataset is quite sparse. For the messages with at least one report, we find that by far most messages have just one user spam report (11,993 messages). About 750 messages have two user spam reports, 180 have three reports, 90 messages have four reports, etc. Only 38 messages in our dataset have 10 or more user spam reports. In a real-world scenario, the data is likely to be denser, which could result in an increase in performance. We touch on this issue in Section 7.

All messages that have been reported twice or more have been manually annotated as "spam" or "not spam." It is reasonable to assume one only considers messages that have been reported twice or more in spam detection, while still using other messages as a background collection. A message is considered spam if it is unsolicited and promotional, as described in the user policy of Hyves.[3] Spam messages can be commercial, i.e., trying to sell things, but the majority of messages are non-commercial spam. Non-commercial spam messages are, for example, friend and group invitations, and requests to follow a person on Twitter.

The annotators—professional moderators working at Hyves—annotated 1,195 messages in total. Of these, 698 messages are marked as spam and 497 are marked as not spam. For our semi-supervised variants we require a training set: we split the full set of messages in a training set (33%) and test set (66%). Messages for the training set are selected based on their posting date (i.e., the oldest messages are used for training). Note that we use the same test set for both the supervised and semi-supervised variants of our models. In the end we use 374 messages for training (232 spam, 142 not spam), and 821 messages for testing (466 spam, 355 not spam).[4]

5.1 Baselines

We compare our spam detection models to two baselines. The first baseline is independent of the user spam reports and only uses textual evidence, the second baseline only uses raw numbers of user spam reports.

Content Baseline: This baseline uses a Naive Bayes classifier on the textual content of messages, i.e., bag-of-words as features. Naive Bayes is widely used as in spam detection systems. We use the training set for training this baseline, to make results comparable to those obtained using our framework.

[2] http://www.hyves.nl

[3] http://hyves.nl/useragreement/

[4] The data is available at http://ilps.science.uva.nl/resources/hyvesspam.

Report Baseline: This baseline involves counting the number of user spam reports. The underlying assumption is that a report generally indicates that a message is spam. The more reports a message receives, the more likely it is the message is actually spam. In cases when two messages have an equal number of reports, we order them by reporting date.

5.2 Metrics

We compare our models on their ability to rank messages by spam score, that is, push spam messages to the top of a spam ranking, and the messages that are not spam to the bottom. For presenting the results we use receiver operating characteristic (ROC) curves and the area under the curve (AUC). The AUC metric is equivalent to the probability that a random positive (spam) instance is ranked higher then a random negative (not spam) instance [6]. We choose AUC over the more common average precision (AP) metric, since AP puts a strong emphasis to the top of the ranking. The challenge in this particular tasks, however, lies in ranking "the middle field."

6 Results

In this section we present the results of our models. We first compare our unsupervised models to each other and to the two baselines, and continue by comparing these results to the semi-supervised variants of our models.

Table 1 lists the AUC results of our models. The first row shows the performance of our unsupervised spam detection models and compares them to our two baselines. The related ROC curves for each of the models are depicted in Figure 4 (Left). From the results we observe that each of our models improves over the two baselines in terms of AUC and that adding evidence to our Reporter Model (authors, similarity) leads to better performance. The ROC curves show that the baselines perform quite well in the initial part of the curve, but underperform in the "middle field." The reason for this lies in the distribution of the number of reports: as long as there are a large number of reports, the Report Baseline is able to rank these properly. However, as the number of reports drops to three and less, which is by far the majority of messages, the ranking of these becomes more or less arbitrary. This leads to a drop in performance compared our models, that use more evidence for ranking and do not solely depend on the number of reports. In the end, the Similarity-Author-Reporter Model is the best performing model of our three spam detection models.

Table 1: Results of the unsupervised and semi-supervised models in terms of AUC.

	Baselines				
	Content	Report	Reporter	Author-Reporter	Similarity-Author-Reporter
Unsupervised	0.5282	0.5482	0.6590	0.7300	0.7671
Semi-supervised			0.6538	0.7251	**0.8007**

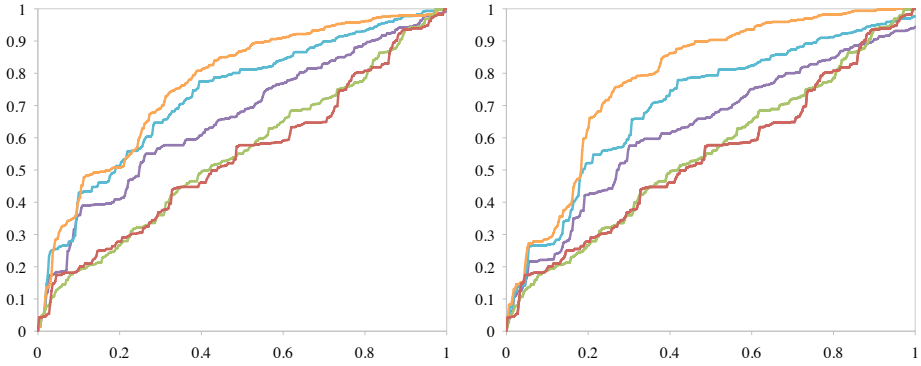

Fig. 4: ROC curves for the two baselines, Content Baseline (red) and Report Baseline (green) and our (Left) unsupervised and (Right) semi-supervised models: Reporter Model (purple), Author-Reporter Model (blue), and Similarity-Author-Reporter Model (orange).

The results of the semi-supervised variants of our models are presented in the second row of Table 1 and the ROC curves are plotted in Figure 4 (Right). We find that the semi-supervised variants of the Reporter and Author-Reporter Models are comparable to their unsupervised variants. However, once we add the message similarity into the model, we find that the semi-supervised scheme improves over the unsupervised one. A possible explanation for this lies in the increased connectedness of the graph: for the Reporter and Author-Reporter Models, most nodes of the graph have no more than three connections. For the Similarity-Author-Reporter Model, however, each node usually has 10 connections. We expect that this denser graph allows for the fixed node to have a greater impact. We come back to this point in the next section, where we analyze the results in more detail and explore the impact of the choices we have made.

7 Discussion

The results show that our three models outperform the baselines and that adding more evidence to our graph leads to an increase in performance. In this section we analyze the results of our unsupervised models in three ways: first, we explore the impact of the number of similar messages in the Similarity-Author-Reporter Model. Second, we look at the settings of parameter γ (Eq. 5) in this model, which indicates the weight given to the spam scores of similar messages. Finally, we analyze the influence of the number of messages in our dataset on spam detection performance for the Author-Reporter Model (we picked this model since the addition of similar messages make the model even more dependent on the dataset size and would not result in reasonable performances).

For the Similarity-Author-Reporter Model, we have until now taken into account the 10 most similar messages (“neighbors”). Here, we explore how performance is in-

fluenced by the number of neighbors. Figure 5 (Left) shows the performance in terms of AUC compared to the number of neighbors for a message node. We find that setting $n < 7$ hurts performance, but the adding more than these seven neighbors does not improve performance.

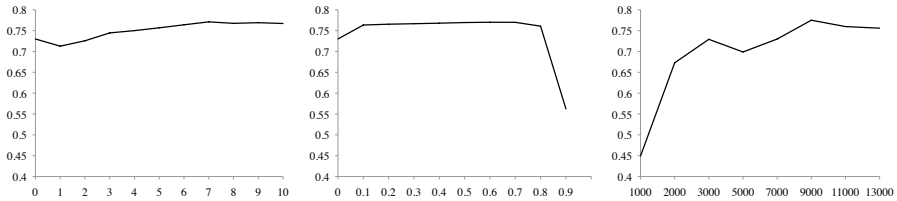

Fig. 5: Impact on AUC of (Left) number of similar messages, (Center) γ parameter, and (Right) size of the dataset on the Author-Reporter Model.

The γ parameter in Eq. 5 adjusts the influence of the spam score of the similar messages in the Similarity-Author-Reporter Model. Figure 5 (Center) shows the influence of this parameter on the performance of the unsupervised model. For $\gamma = 0$ the AUC is equal to the AUC of the Author-Reporter Model (0.7300) and we find that increasing γ leads to better performance, although differences between $\gamma = 0.1$ and $\gamma = 0.8$ are marginal. Giving almost all the weight to the similar messages results in a steep drop in performance.

Finally, we explore the impact of the size of the dataset on the performance. We hypothesize that, as the dataset grows larger, the graph becomes denser and performance should therefore go up. Figure 5 (Right) shows the AUC performance for dataset sizes between 1,000 messages and 13,000 messages for the unsupervised Author-Reporter Model. The results show an upward trend, with a bigger dataset improving the performance of this model. We expect the models to perform even better with a larger dataset.

8 Conclusion

In this paper we have introduced a framework for unsupervised spam detection in social networking sites, based on user spam reports. Being unsupervised, the framework benefits from a large dataset, without the need for costly annotations, and it should be able to respond to new types of spam quicker than content-based models.

We have instantiated our framework in three ways, each model building on the previous. The first model, the Reporter Model, uses only messages and reporters to build a graph and propagates spam scores through this graph. Our Author-Reporter Model adds the authors of the messages to the graph, and finally we add links between similar messages in our Similarity-Author-Reporter Model. We also introduce semi-supervised variant of our three models that can be used with explicitly labeled messages.

Results show that our models improve over two baselines (based on content and on raw report counts) and that adding evidence to the model's graph leads to improvements over the previous model. We also find that the semi-supervised variants or the Reporter and Author-Reporter Models do not improve over their unsupervised counterparts. For the Similarity-Author-Reporter Model, however, the semi-supervised variant does improve performance. Analyses of the results revealed that a larger dataset leads to better performance.

For future work we would like to apply our models to an even larger dataset, thereby also showing the scalability of our framework. Additional features—such as friendship connections between users—can be added to the framework, potentially improving performance even further. In our semi-supervised setting, we experimented with assigning negative values to message nodes that are not spam, to explicitly penalize "bad reporters." While this resulted in an improvement in performance in some situations, it also caused unstable, non-converging behavior in other situations. In future work we will investigate a more principled scheme for penalizing reporters.

Acknowledgments We would like to thank Hyves for providing us the data. Meij is supported by the European Community's Seventh Framework Programme (FP7/ 2007-2013) under grant agreement nr 288024 (LiMoSINe project). Weerkamp is supported by the Netherlands Organisation for Scientific Research (NWO) under project number 380.70.011 and by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430 (GALATEAS).

9 References

[1] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross. Identifying video spammers in online social networks. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 45–52. ACM, 2008.

[2] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on World wide web*, pages 51–60. ACM, 2009.

[3] C. Campbell, P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM, 2003.

[4] D. DeBarr and H. Wechsler. Using social network analysis for spam detection. *Advances in Social Computing*, pages 62–69, 2010.

[5] 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*, pages 42–48. ACM, 2003.

[6] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[7] C. Ghiossi. The facebook blog: Explaining facebook's spam prevention systems. http://blog.facebook.com/blog.php?post=403200567130, 2010. Accessed May 12, 2011.

[8] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.

[9] D. Irani, S. Webb, and C. Pu. Study of static classification of social spam profiles in myspace. In *Proceedings of the 4th International Conference on Weblogs and Social Media*, 2010.

[10] 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*, pages 919–922. ACM, 2007.

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

[12] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[13] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on World wide web*, pages 691–700. ACM, 2010.

[14] B. Mehta, T. Hofmann, and P. Fankhauser. Lies and propaganda: detecting spam users in collaborative filtering. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 14–21. ACM, 2007.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Stanford InfoLab*. Citeseer, 1999.

[16] A. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.

[17] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. *Advances in Databases: Concepts, Systems and Applications*, pages 1066–1069, 2010.

[18] C. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, pages 83–97. IEEE, 2004.

[19] A. Zinman and J. Donath. Is britney spears spam. In *Fourth Conference on Email and Anti-Spam, Mountain View, CA*. Citeseer, 2007.