Using Semantic Features to Detect Spamming in Social Bookmarking Systems

Amgad Madkour[1] , Tarek Hefni[2], Ahmed Hefny[1] , Khaled S. Refaat[1]

Human Language Technologies Group
IBM Cairo Technology Development Center[1]
P.O.Box 166 El-Ahram, Giza, Egypt
{amadkour,ahefny,ksaeed}@eg.ibm.com[1] , t_hefny@aucegypt.edu[2]

Abstract. Collaborative software is gaining pace as a vital means of information sharing between users. This paper discusses one of the key challenges that affect such systems which is identifying spammers. We discuss potential features that describe the system's users and illustrate how we can use those features in order to determine potential spamming users through various machine learning models.

1 Introduction

Spamming is a crucial challenge that affects both users and services providers. Users are faced with spamming obstacles during activities such as web-based searching. This often occurs when spammers use techniques or keywords to increase the rank of their websites. This in turn overshadows more relevant pages that users might be actually interested in. A famous form of spamming is that a spammer might create a website with keywords most relevant to a common accessory that a user maybe interested in. Other victims of spamming are email service providers. Email providers use a lot of spam detection techniques in order to minimize spam emails sent to their users.

Spam detection is faced with great challenges and thus various techniques have been deployed. Some techniques are based on detecting the most common keywords or key phrases that are frequently used in most spamming emails. Other techniques are based on learning general patterns that spammers tend to use to advertise their material. Some of them rely mostly on manually constructed pattern matching rules that need to be tuned to each user. A greater challenge is that the characteristics of spam change over time which makes maintaining the rules a daunting task. Other systems employ machine learning techniques which allow the system to automatically learn to separate spam from other messages. Classification techniques, based on different features that describe a user and posts, are used in order to differentiate between spamming and non-spamming users.

[2] Tarek Hefni is an undergraduate student at the American University in Cairo. At the time of this work, he was an intern with IBM Cairo Technology Development Center R&D Team.

In this work we target a specific type of collaborative social software systems that suffers from spamming which is social bookmarking[12]. Its main focus is collecting the user online bookmarks which the user generates. Other similar systems are used to store user-generated scientific bibliographies. The main element in such systems is a post. It's composed of a user, a resource and a number of tags annotating it. Systems such as social bookmarking allow users to upload their resources, and label them with tags. The systems differ depending on the type of resource being shared[7]. On most systems, users are described by their user ID and tags may be arbitrary strings. Users are allowed to copy the resources tagged by other users. They are also allowed to view information such as who tagged what resource and so forth. The power of such a system is that users are able to share or browse other users' shared data.

The set of tag-resource relations is referred to as *folksonomy*[1]. It stands for conceptual structures created by people. The collection of all user assignments is called a user *personomy*. A collection of all personomies in turn constitutes the folksonomy. There is widespread use of these systems due to the presence of sharing mechanisms between users which enable breaking the knowledge acquisition problem users face.

In this paper, we discuss various features that can be used to identify spammers that target such systems. The main motivation of our work is to capture the necessary features that discriminate spammers from non-spammers. Our second motivation is to determine what an efficient classification model would be like.

2 Related Work

Li and Hsieh [11] proposed a group-based anti-spam framework investigating the clustering structures of spammers based on spam traffic collected at a domain mail server. Their study showed that the relationship among spammers demonstrates highly clustering structures based on URL grouping.

Krause and Schmitz [9] proposed a social bookmarking setting to identify spammers using features based on the topological, semantic and profile-based information. They used different classification techniques to evaluate their proposed features. Their best results were achieved using SVM scoring a roc area of 0.936.

Koutrica et al [8] introduced a framework for modeling tagging systems and user tagging behavior. They proposed a tagging system where malicious tags and malicious user behaviors are well-defined, and described and studied a variety of query schemes and moderator strategies for tag spam detection.

Androutsopoulos et al [3] explored the idea that a Naive Bayesian classifier could be used to filter spam mail. They also investigated the effect of some parameters on the performance of the filter such as attribute-set size, training-corpus size, lemmatization, and stop-lists. They discovered that the filter has a stable significant positive contribution with additional safety nets like resending private messages but is not viable when blocked messages are deleted.

Beil et al [5] introduced an approach which uses frequent item (term) sets for text clustering. They used algorithms for association rule mining to discover such frequent sets. They presented two algorithms for frequent term-based text clustering: FTC and HFTC where the first creates flat clustering and the second allows hierarchical clustering respectively.

Wetzker et al [12] analyzed 150 million bookmarks. They showed how bookmarks are vulnerable to spamming and how to limit such vulnerability to avoid affecting the analysis process.

Hotho et al [2] specified a formal model for folksonomies and described their system: BibSonomy, used for sharing both bookmarks and publication references in a personal library. They showed that BibSonomy is valuable for researchers because of the fact that it combines both bookmarks and publication entries.

Gomes and Cazita [10] provided a characterization of spam traffic using workload variation, density, inter-arrival time distribution, email size distribution, temporal locality, etc., compared with non-spam emails. They showed that non-spam email transmissions are typically driven by bilateral social relationship while spam transmissions are usually unilateral actions based on the spammers' will to reach a large number of recipients.

3 Semantic Features

Semantic features refer to annotations that are derived from the content of the resources. We are motivated to show the value of using semantic features as a means of detecting spammers.

For the first set of features, we used those mentioned by Krause et al. [9]. They proposed a set of features that were extracted from a dataset with comparable characteristics to the one under investigation. Our contribution is the usage and creation of semantic features that could contribute to the classification accuracy.

The first feature [9] deals with counting the number of tags of the user which contain 'group=public'. This feature measures the amount of tags that are publicly shared with the social bookmarking community. This is used by spammers in order to increase the exposure of spammed material to the public.

For the second and third features [9], we counted the number of resources that are common between the current user and non-spammers in the training set and the number of resources that are common between the user and spammers in the training set, respectively. Common resources could be shared bookmarks or shared BibSonomy items. Those two features allow measuring a ratio between resources that the current user shares with the spammers and non-spammers community.

Following the same concept for the fourth and fifth features [9], we counted the number of tags that are common between the users and non-spammers in the training set as well as the number of tags that are common between the users and spammers in the training set. It is important to note that the difference between those two features and the previous two is that a resource could be assigned to more than one tag.

Similarly for the sixth and seventh features [9], we counted the number of resource-tag couples that are common between the user and non-spammers in the training set and the number of resource-tag couples that are common between the user and spammers in the training set. This gives an indication about the ratio of resources-tags for spammers and non spammers.

For the eighth, ninth and tenth features [9], we calculated the co-occurrences of some of the previously mentioned features. The eighth feature is a calculation of the co-occurrence generated when we compute the ratio between the second and third features. The ninth feature is a calculation of the ratio between the fourth and fifth features. The tenth feature is a calculation of the ratio between the sixth and seventh features. All the co-occurrences features are based on the assumption that spammers share the same vocabulary as non-spammers [9].

We are motivated to capture the keywords that always co-occur with spammed material [9]. Those keywords are referred to as black-listed words. The black-listed words are a list of words that generally occur in spammed material, such as emails or websites, with high frequency. We developed a weighted version of that list in order to use it for our proposed features. Using the training set, we gave each word a score which represents the frequency between word repetitions by spammers divided by frequency between word repetition by non-spammer. We used the same technique for both tags and descriptions.

For the eleventh feature, we calculated the total score of each word in the description of every bookmark for each user divided by the number of bookmarks for the user. For the twelfth feature, we calculated the total score of each word in the tags of every bookmark for each user. Those two features allow us to capture the frequency of black-listed keywords within resources such as bookmarks and tag names.

For our last feature, we counted how many times a tag was repeated with other tags for the same resource by non-spammers. We created what we call a tag-pair which consists of the two tokens inside the tag. The total score of the feature is calculated as follows: for each tag-pair of each resource, if the user used a defined tag-pair, we add one divided by the tag-pair score, otherwise, we add three.

4 Dataset

We use the dataset provided for the ECML PKDD Discovery Challenge 2008. The dataset consists of users and their posts. The information includes all public information such as the URL, the description and all tags of the post. The training data was composed of 22,200 patterns and the testing set was composed of 9,959 which included 741 non-spamming users. In this paper, we report the results obtained by using the testing set provided at training time.

5 Evaluation

The problem will be evaluated using AUC (Area Under ROC Curve). AUC estimates the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative instance [7]. It shows the relative trade-offs between benefits (true positives rates) and costs (false positives rates)[13].

6 Experimental Setup

We experimented with five models: K-Nearest Neighbor Regression, Gaussian Processes [6], Support Vector Machine (SVM), Neural Networks and Ensemble Learning using both SVM and Neural Networks. We used Rapid Miner[1] as our machine learning toolkit.

6.1 K-Nearest Neighbor Regression

The following table shows the AUC values obtained using different values of k. A low value of k results in an input-sensitive model while a high value of k results in a smoothing model.

Length Scale	AUC
3	0.863233176
5	0.87998662
7	0.886212826
13	0.901224341
15	0.900489829
17	0.902808017
19	0.904737342
30	0.908758165
35	0.90963878
40	0.909391328
50	0.909341884

Table 1. K-Nearest Neighbor Results

The best AUC value (0. 90963878) is achieved at k = 35. It is worth noting that both Gaussian Processes and KNN regression performed best at a point of notably high smoothing (low variance).

[1] http://rapid-i.com

6.2 Gaussian Processes

We used Gaussian Processes[6] with Radial Basis Function (RBF) kernel while restricting the maximum number of basis functions to 100. This restriction significantly reduced training time without any notable change to the AUC value (compared to 1000 basis functions). The following table shows the obtained AUC values on the validation set for different values of the Length-Scale of the RBF kernel. A low value of L results in an input-sensitive (high variance) model while a high value of L results in a smoothing model.

Length Scale	AUC
3	0.770784148624904
10	0.781429115194297
20	0.84661283935238
40	0.921302082732583
80	0.932317150535772
81	0.929417857595996
82	0.929390174977221
83	0.929390174977221
85	0.926760095505033
90	0.923650491558724
100	0.915941266710499
150	0.886343702780949

Table 2. Gaussian Processes Results

The best AUC value (0.932317150535772) is achieved at length scale of 80, a relatively high value.

6.3 Support Vector Machines

As for the SVM, we used cost-based learning [14][15][4]. Assigning a cost of 7 for labeling a user as non-spammer produced highest results. Accordingly, we reached the following figures:

Kernel	Gamma/Pol.Degree	AUC
RBF	1	0.9501623
RBF	1.2	0.9521352
RBF	2	0.9509834
RBF	7	0.9487790
Polynomial	9	0.9519381

Table 3. Support Vector Machine Results

To improve results, we employed a cascading scheme in which an SVM model (of configuration 2) is used to classify data patterns and another SVM model is used to reclassify difficult patterns. A difficult pattern is a pattern that produced an output in the range between -1 and 1 in the first SVM model, meaning that there is too little confidence to classify it. The second model is trained using the difficult points in the training set; we classified our training set using the first model and extracted the difficult points which we used to train the second model. Adding the second model with parameters different than the first model would assure that the difficult points that are misclassified by the first model are classified correctly by the second model. The table below shows the parameter configurations we tried for the second model (cost value is the same) and the AUC achieved by each.

Kernel	**Gamma/Pol.Degree**	**AUC**
RBF	1.22	0.9526184
RBF	1	0.9525310
RBF	0.9	0.9487790
Polynomial	9	0.9518192

Table 4. Support Vector Machine Results with Cascading Schema

It was observed that using one classifier misclassified 350 users of the 9950 users of the test set.

6.4 Neural Networks

We used a Neural Network Model with learning rate 0.6 and momentum 0.3 for 9000 epoch. The network contained one hidden layer of thirty neurons with a sigmoid activation function. The model resulted in an AUC of 0.9394533. Decreasing the learning rate to 0.4 decreased the AUC to 0.9238232.

6.5 Combined NN and SVM

Finally, we attempted an ensemble learning scheme where we averaged the outputs of the best two SVM models and the best Neural Network model. This produced an AUC of 0.9425323421.

Using the proposed features of [9] to train our models with the first ten features, we achieved an AUC of 0.941243347. Adding the eleventh feature alone (weighted black list feature) resulted in an AUC of 0.950537472. By adding the final feature, we reach our best model giving an AUC value of 0.9526184.

7 Conclusion

In this paper we discussed the semantic features that can be used to detect spammers in a social bookmarking system. Our proposed features demonstrate

improved results compared to the ones proposed by [9] on the competition test set taking into consideration the comparable dataset. The paper also discussed the results obtained by training various classifiers. In addition, this paper demonstrated how the cascading scheme model provided better results and partially tackled the border-line of classification that [9] mentioned. We used the Area Under ROC Curve method to evaluate our results and the best result obtained was 0.9526.

References

1. A Capocci, G Caldarelli. Folksonomies and clustering in the collaborative system CiteULike.
2. A Hotho, R Jaschke, C Schmitz, G Stumme. BibSonomy: A Social Bookmark and Publication Sharing System.
3. Androutsopoulos. An Evaluation of Naive Bayesian Anti-Spam Filtering. Proceedings of the workshop on Machine Learning in the New Information Age, G. Potamias, V. Moustakis and M. van Someren (eds.), 11th European Conference on Machine Learning, Barcelona, Spain, pp. 9-17, 2000.
4. B. Scholkopf, A. J. Smola. Learning with Kernels. The MIT Press Cambridge, Massachusetts London, England 2002.
5. Beil et al. Frequent Term-Based Text Clustering. SIGKDD 02 Edmonton, Alberta, Canada
6. Carl Edward Rasmussen, Chris Williams. Gaussian Processes for Machine Learning. the MIT Press, 2006
7. E Santos-Neto, M Ripeanu, A Iamnitchi. Tracking Usage in Collaborative Tagging Communities.
8. G. Koutrika, F. A. Effendi, Z. Gyongyi, P. Heymann, and H. Garcia-Molina.: Combating spam in tagging systems. In Proc. AIRWeb , pages 57 New York, NY, USA, 2007 ACM.
9. Krause, Beate. Schmitz, Christoph. Hotho, Andreas. Stumme, Gerd. The Anti-Social Tagger - Detecting Spam in Social Bookmarking Systems. AIRWeb '08, April 22, 2008 Beijing, China.
10. L. H. Gomes, C. Cazita. Characterizing a Spam Traffic. In the proceeding of IMC 04, Oct. 2004.
11. Li, Hsieh. An Empirical Study of Clustering Behavior of Spammers and Group-based Anti-Spam Strategies. CEAS 2006 Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA.
12. R Wetzker, C Zimmermann, C Bauckhage3. Analyzing Social Bookmarking Systems: del.icio.us Cookbook. July 10th, 2008.
13. T. Fawcett. An Introduction to ROC Analysis. Pattern Recogn. Lett., 27(8)861 2006.
14. Yoav Freund, Robert E. Schapire. Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference, pages 148156, 1996.
15. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119139, 1997.