Using Language Modeling for Spam Detection in Social Reference Manager Websites

Toine Bogers
ILK / Tilburg centre for Creative Computing
Tilburg University
P.O. Box 90153, 5000 LE
Tilburg, The Netherlands
A.M.Bogers@uvt.nl

Antal van den Bosch
ILK / Tilburg centre for Creative Computing
Tilburg University
P.O. Box 90153, 5000 LE
Tilburg, The Netherlands
Antal.vdnBosch@uvt.nl

ABSTRACT

We present an adversarial information retrieval approach to the automatic detection of spam content in social bookmarking websites. Our approach focuses on the use of language modeling, and is based on the intuitive notion that similar users and posts tend to use the same language. We compare using language modeling at two different levels of granularity: at the level of individual posts, and at an aggregated user level, where all posts of one user are merged into a single profile. We evaluate our approach on two spam-annotated data sets based on snapshots of the social bookmarking websites CiteULike and BibSonomy, and achieve promising results.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.4 [**Information Systems Applications**]: H.4.2 Types of Systems; H.4.m Miscellaneous

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Adversarial IR, spam detection, social bookmarking, language modeling

1. INTRODUCTION

The term 'spam' was originally used to refer to the abuse of electronic messaging systems that started in the mid-1990s on Usenet newsgroups, and quickly crossed over to e-mail messaging. According to conservative estimates, in the first half of 2007 around 85% of all e-mail sent in the world was spam[1]. The notion of spam is subjective by nature, but we define it here as content designed to mislead, and that the legitimate users of a system, site, or service therefore do not wish to receive. Motivation for spamming can range from advertising and self-promotion to disruption and disparagement of competitors [5]. Spamming is economically viable because the barrier for entry into the abused systems is generally low, and because it requires virtually no operating costs beyond the management of the automatic spamming software. In addition, it is often difficult to hold spammers accountable for their behavior.

Any system that relies on user-generated content is vulnerable to spam in one form or another. Search engines, for instance, suffer increasingly from so-called *spamdexing* attempts with content especially created to trick search engines into giving certain pages a higher ranking than they deserve [4]. Spam comments are also becoming an increasing problem for websites that allow users to post reactions to content, such as blogs and video and photo sharing websites [12].

By analogy, the relatively recent phenomenon of social websites and social bookmarking services have become an increasingly popular part of the Web, but their focus on user-generated content also makes them vulnerable to spam, threatening their openness, interactivity, and usefulness [5]. In this paper, we focus on how we can detect spam in social bookmarking systems. Our approach to spam detection is based on the intuitive notion that spam users are likely to use different language than 'legitimate' users when posting content to a social bookmarking system. We detect new spam users in the system by first ranking all the known users in the system by the KL-divergence of the language models of their posts—separately per post as well as merged into user profiles—and the language model of the new user or post. We then look at the spam labels assigned to the most similar users in the system to predict a spam label for the new user. We test our approach on two spam-annotated data sets, based on BibSonomy[2] and CiteULike[3], two so-called *social reference managers* that allow users to store and manage their reference list of scientific articles online.

The paper is structured as follows. We start by reviewing the related work in the next section, followed by a description of the task and the data sets, our pre-processing steps,

DIR 2009 February 2-3, 2009, Enschede, the Netherlands.

[1] http://www.maawg.org/about/MAAWG20072Q_Metrics_Report.pdf

[2] http://www.bibsonomy.org

[3] http://www.citeulike.org

and our evaluation setup in Section 3. In Section 4 we describe our spam detection approach; in Section 5 we report on our results. We conclude our paper by discussing our findings in Section 6 and listing possible future work in Section 7.

2. RELATED WORK

The issue of spam in social bookmarking services has received relatively little attention so far. Heymann et al. (2007) were the first to examine the relationship between spam and social bookmarking in detail [5], classifying the anti-spam strategies commonly adopted in practice into three different categories: *prevention*, *demotion*, and *detection*. *Prevention-based* approaches are aimed at making it difficult to contribute spam content to the social bookmarking system by restricting certain types of access through the submission interface (such as CAPTCHAs) or through usage limits (such as post or tagging quota). The `nofollow` HTML attribute of hyperlinks can also serve as a spam deterrent, since it instructs search engines that a hyperlink should not influence the link target's ranking in the search engine's index, thereby removing the main motivation of spammers.

Demotion-based strategies focus on reducing the prominence and visibility of content likely to be spam. Rank-based methods, for instance, try to produce orderings of the system's content that are more accurate and more resistant to spam [5]. A demotion-based strategy for combating spam is described by [5] and described in more detail in [8]. They construct a simplified model of tagging behavior in a social bookmarking system, and compare different ranking methods for tag-based browsing. They investigate the influence of various factors on these rankings, such as the proportion and behavior of spam users and tagging quota [8].

Spam detection methods, finally, are used to identify likely spam either manually or automatically, and then act upon this identification by either deleting the spam content or visibly flagging it as such to the user [5]. To our knowledge, the only published effort of automatic spam detection in the social reference manager context comes from Krause et al. (2008), who investigate the usefulness of different machine learning algorithms and features to automatically identify spam users and their posts[9]. They test their algorithms on a data dump of the BibSonomy system.

Later in 2008, this work on spam detection for BibSonomy was extended by means of the 2008 ECML/PKDD Discovery Challenge workshop[4], which focused on two data mining tasks related to social bookmarking. One of these tasks was detecting spam users in a social bookmarking system. So far, this has been the only TREC-like initiative focusing on the task of spam detection. With a total of 13 submissions, the majority of the participants' approaches used machine learning for the prediction task. Six out of the top eight approaches used a variety of content-based and co-occurrence-based features combined with machine learning algorithms to separate the spammers from the genuine users [6]. One of the top eight submission used a graph-based algorithm for the detection task [10]. We participated in the challenge with a preliminary version of our approach, described in [1], and finished in fourth position. In this paper, we extend our approach and test it more extensively using other data representations. Furthermore, we use an additional data set based on CiteULike to confirm the general applicability of our method.

Broadening the scope beyond social websites, we can also find a wealth of other anti-spam approaching in related fields such as blogs. Mishne et al. (2005) were among the first to address the problem of spam comments in blogs and used language model disagreement between the blog post itself, the comments, and any pages linked to from the comments to identify possible spam comments [12]. Their work inspired our approach to spam detection in social bookmarking. In 2006, the TREC Blog Track also paid attention the problem of blog spam [13].

3. METHODOLOGY

3.1 Task description

One of the two tasks in the 2008 Discovery Challenge was spam detection in a social bookmarking system [6]. We use their definition of the spam detection task to guide our experiments in this paper. The goal of the spam detection task is to learn a model that predicts whether a user is a spammer or not. An added requirement is that the model should be able to accurately classify initial posts made by new users, in order to detect spammers as early as possible. This decision to identify spam in BibSonomy at the user level instead of at the post level means that all of a spam user's posts are automatically labelled as spam. This decision was justified earlier in Krause et al. (2008) by the observation that users with malicious intent often attempt to hide their motivations with non-spam posts [9]. In addition, Krause et al. also cite workload reduction as a reason for the decision to classify at the user level. In the experiments described in this paper, we use the setup of the Discovery Challenge for our spam detection task and classify spam at the user level in both our BibSonomy and our CiteULike data set, to make for a fair comparison of our results.

3.2 Data Collection

Automatic spam classification approaches typically demand a training or seed set to learn to predict spam characteristics [5], so for us to be able to test our spam detection approach, we needed access to data sets with manually identified spam objects. We were able to obtain such spam labels for data sets based on two social bookmarking websites: BibSonomy and CiteULike. The BibSonomy collection came pre-labeled for spam as part of the aforementioned 2008 Discovery Challenge. For CiteULike we annotated a sizable part of the collection ourselves. Table 1 provides statistics for the presence of spam in the CiteULike and BibSonomy collections. One thing is clear from both data sets: spammers tend to add twice as few posts, but two to three times as many tags to their posts on average than genuine users. Tag count therefore seems to be an informative feature for spam prediction; a fact already signaled in [9]. In the next two subsections we go into more detail about how we obtained our spam annotations and about specific characteristics of the two data sets.

3.2.1 BibSonomy

BibSonomy is a system for sharing bookmarks and reference lists of scientific articles. It allows its users to add their academic reference library as well as their favorite book-

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

Table 1: Spam statistics of the BibSonomy and CiteULike data sets. All CiteULike items were treated as scientific articles, since there is no clear-cut distinction between bookmarks and scientific articles on CiteULike. For BibSonomy, these are the counts of the training material combined with the official test set.

	BibSonomy	CiteULike
posts	2,102,509	224,987
bookmarks, spam	1,766,334	
bookmarks, clean	177,546	
articles, spam	292	70,168
articles, clean	158,335	154,819
users	38,920	5,200
spam	36,282	1,475
clean	2,638	3,725
average posts/user	54.0	43.3
spam	48.7	47.6
clean	127.3	41.6
tags	352,542	82,121
spam	310,812	43,751
clean	64,334	45,401
average tags/post	7.9	4.6
spam	8.9	7.7
clean	2.7	3.2

marks to their online profile on the BibSonomy website. Articles are stored as their BibTeX representation, including abstracts, and links to the papers at the publishers' websites. User can also describe their references using tags and use these to browse and discover new and related references.

BibSonomy is used as a testbed for research into various knowledge organizational aspects of social bookmarking by the Knowledge and Data Engineering group of the University of Kassel, Germany. As part of their research efforts, they organized the 2008 ECML/PKDD Discovery Challenge and made a snapshot of their BibSonomy system available in the form of a MySQL dump. This dump consisted of all resources posted to BibSonomy between its inception in 2006 and March 31, 2008. The distinction between bookmarks and BibTeX records is also made in this snapshot. The data set contained flags that identify users as spammers or non-spammers, and these labels were included in the data set for training and tuning parameters. The Discovery Challenge organizers were able to collect data of more than 2,600 active users and more than 36,000 spammers by manually labeling users. This reveals that the BibSonomy data set is strongly skewed towards spam users with almost 14 spam users for each genuine user. Table 1 also shows that spam users in BibSonomy clearly prefer to post bookmarks, whereas legitimate users tend to post more scientific articles.

3.2.2 CiteULike

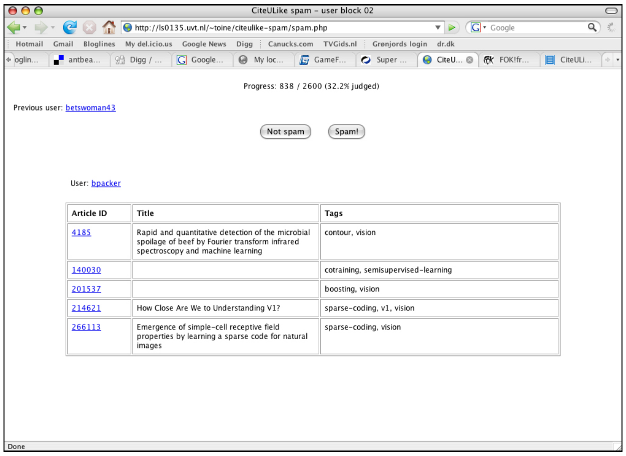
CiteULike is a website that offers a "a free service to help you to store, organise, and share the scholarly papers you are reading". It allows its users to add their academic reference library to their online profile on the CiteULike website. At the time of writing, CiteULike contains around 1,166,891 unique items, annotated by 35,019 users with 245,649 unique tags. Articles can be stored with their metadata (in various formats), abstracts, and links to the papers at the publishers' websites. CiteULike offers daily dumps of their core database[5]. We used the dump of November 2, 2007 as the basis for our experiments. A dump contains all information on which articles were posted by whom, the tags that were used to annotate the reference, and a time stamp of the post. It does not, however, contain any of the other metadata available in the online service, so we crawled this metadata ourselves from the CiteULike website using the article IDs. After crawling and data clean-up, our collection contained a total of 1,012,898 different posts, where we define a post as a user-item pair in the database, i.e. an item that was added to a CiteULike user profile. These posts comprised 803,521 unique articles posted by 25,375 unique users using 232,937 unique tags.

This self-crawled CiteULike data set did not come with pre-labelled spam users or posts as the BibSonomy data set did. We therefore set out to collect our own spam labels for this data set. In this we faced the same choice as the team behind the Discovery Challenge: at which level of the folksonomy should we identify spam usage—users, items, tags, or individual posts? Our CiteULike collection contains over 1 million posts and over 800,000 items, and going through all of these was not practical. Judging all of the more than 232,000 tags was also infeasible, in part because it is simply not possible for many tags to unequivocally classify them as spam or non-spam. For instance, while many spam entries are tagged with the tag `sex`, there are also over 200 valid scientific articles on CiteULike that are tagged with `sex`. We therefore aimed to obtain an estimate of the pervasiveness of spam on CiteULike by identifying spam users. Judging all 25,375 users in the CiteULike data set would still be impractical, so we randomly selected 5,200 users (∼20%) from the data set and asked two annotators to judge these users on whether they were spammers or not. Each user was judged by only a single annotator to save time.

Figure 1 illustrates the straightforward interface we created for the spam annotation process. For each user it randomly selects a maximum of five articles and displays the article title (if available) and the associated tags. It also shows a link to the CiteULike page of the article. Preliminary analysis showed that articles that were clearly spam were usually already removed by CiteULike and returned a *404 Not Found* error. We therefore instructed our judges to check the CiteULike links if a user's spam status was not obvious from the displayed articles. Missing article pages meant users should be marked as spam. In this process, we assumed that although spam users might add real articles to their profile in an attempt to evade detection, real dedicated CiteULike users would never willingly add spam articles to their profile. Finally, we noticed that spam content was injected into CiteULike in many different languages. From the experience of the annotators, most spam was in English, but considerable portions were in Spanish, Swedish, and German. Other languages in which spam content was found were, among others, Dutch, Finnish, Chinese, and Italian.

Of the 5,200 users in our subset, 3,725 (or 28.1%) were spam users, which is a smaller proportion than in BibSonomy. The numbers in Table 1 are reported for this 20% sample of CiteULike users. An extrapolation of these proportions to

[5] See `http://www.citeulike.org/faq/data.adp`.



Progress: 838 / 2600 (32.2% judged)

Previous user: betswoman43

User: bpacker

Article ID	Title	Tags
4185	Rapid and quantitative detection of the microbial spoilage of beef by Fourier transform infrared spectroscopy and machine learning	contour, vision
140030		cotraining, semisupervised-learning
201537		boosting, vision
214621	How Close Are We to Understanding V1?	sparse-coding, v1, vision
266113	Emergence of simple-cell receptive field properties by learning a sparse code for natural images	sparse-coding, vision

Figure 1: A screenshot of the interface used to annotate a subset of CiteULike users as possible spam users

the full CiteULike data set results in an estimated 7,198 spam users who posted articles to CiteULike. To assess the accuracy of this estimation we may look at the problem from a different angle. As already remarked, certain spam articles are removed quickly from the database by the CiteULike administrators, resulting in *404 Not Found* errors when crawling their article pages. During metadata crawling of all 803,521 articles in our November 7, 2007 data dump, about 26.5% of the articles returned *404 Not Found* errors. A second round of re-crawling the metadata of these 213,129 missing articles did not change this proportion. While spam removal is not necessarily the only reason for a *404 Not Found* error, we found that 18.7% of the 7,671 users that posted these 213,129 missing articles were spam users identified in our annotation process, which is commensurate with the 20% sample we took. Furthermore, we found that 60,796 of the missing articles (or 28.5%) belonged to the positively identified spam users. These estimates of 7,671 spam users (or 30.2%) and 213,129 spam articles (or 26.5%) strongly suggest that our extrapolation of spam presence on CiteULike is reliable.

3.3 Data Representation

After collecting the data we created a single representation format for all posts, capturing all the relevant metadata in separate fields. As mentioned before, two types of resources can be posted to BibSonomy: bookmarks and BibTeX records, the latter with a magnitude more metadata available. Because there is no such clear distinction in our CiteULike data set, we decided to treat BibTeX records and bookmarks the same and thus use the same format to represent both. We represented all resource metadata in an TREC-style SGML format using 4 fields: `<TITLE>`, `<DESCRIPTION>`, `<TAGS>`, and `<URL>`. URLs were pre-processed before they were used: punctuation was replaced by whitespace and common prefixes and suffixes like `www`, `http://`, and `.com` were removed. Figure 2 shows examples of clean and spam posts in our SGML representation.

A wide variety of metadata fields are available for the posts in the BibSonomy data set. For the bookmarks, the title information is taken from the `book_description` field in the MySQL dump, whereas the `title` field is used for the BibTeX records. The `<DESCRIPTION>` field is filled with the `book_extended` field for bookmarks, whereas the following fields are used for the BibTeX records: `journal`, `booktitle`, `howPublished`, `publisher`, `organization`, `description`, `annote`, `author`, `editor`, `bibtexAbstract`, `address`, `school`, `series`, and `institution`. For both resource types all tags are added to the `<TAGS>` field. URLs, finally, are extracted from the `book_url` and `url` fields, and pre-processed as described above.

Unfortunately, our post representations are significantly poorer for the CiteULike data set: since spam articles are removed from the CiteULike website, we could not crawl the associated metadata of these spam articles (cf. Section 3.2.2). Full metadata is available for the clean articles, but using all metadata of the clean posts and and only the tags of the spam posts would yield an unrealistic comparison. Any classifier would simply learn to predict a post to be spam if it was missing metadata, which is unlikely to be very useful in a real-world situation. We therefore used only the tags for all CiteULike posts, clean and spam alike.

3.4 Evaluation

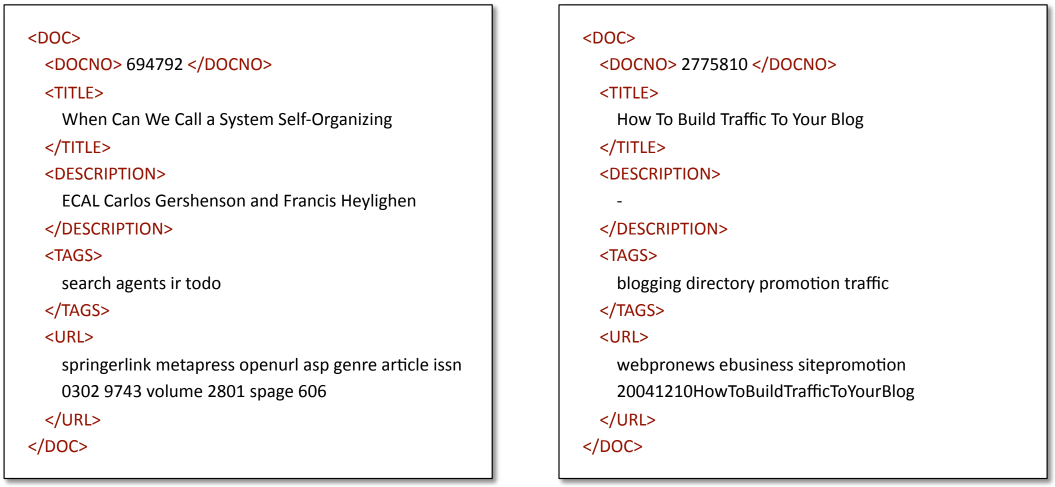
To evaluate our different approaches and optimized parameters, we divide each data set into a training set, a validation set, and a test set. Our models are trained on the training set, while parameters are optimized on the validation set to prevent overfitting [11]. For the BibSonomy data set, an official test set is supplied as part of the Discovery Challenge as well as training material, so we used this partitioning. We randomly select 80% of the users from the training material for our training set, and assign the remaining 20% to our validation set. This yields a training set of 25,372 users, a validation set of 6,343 users, and a test set of 7,205 users. For the CiteULike data set, we randomly select 60% of all users for our training set, 20% for our validation set, and assign the remaining 20% to our test set. This corresponds to 4,160 training users, 520 validation set users, and 520 users in the CiteULike test set. For the final predictions on the test sets we used only the training sets we created to train our algorithm and generate the spam labeling predictions.

We evaluate our approaches on the validation and test sets using the standard measure of AUC (area under the ROC curve). We optimize k using AUC rather than on measures like the F-score, as AUC is less sensitive to class skew than F-score [3], knowing that indeed the data is rather skewed, especially in the case of BibSonomy, with 12 spam users to every clean one.

4. SPAM DETECTION

4.1 Language Modeling for Spam Detection

Our approach to spam detection is based on the intuitive notion that spam users will use different language than legitimate users when posting content to a social bookmarking system. By comparing the language models of posts made by spammers and posts made by legitimate users, we can use the divergence between the models as a measure of (dis)similarity. After we have identified the k most similar posts or users using language modeling, we classify new users as spam users or genuine users by scoring these new users by how many spam posts and how many clean posts were found to be similar to it.



```
<DOC>
  <DOCNO> 694792 </DOCNO>
  <TITLE>
    When Can We Call a System Self-Organizing
  </TITLE>
  <DESCRIPTION>
    ECAL Carlos Gershenson and Francis Heylighen
  </DESCRIPTION>
  <TAGS>
    search agents ir todo
  </TAGS>
  <URL>
    springerlink metapress openurl asp genre article issn
    0302 9743 volume 2801 page 606
  </URL>
</DOC>
```

(a) Clean post

```
<DOC>
  <DOCNO> 2775810 </DOCNO>
  <TITLE>
    How To Build Traffic To Your Blog
  </TITLE>
  <DESCRIPTION>
    -
  </DESCRIPTION>
  <TAGS>
    blogging directory promotion traffic
  </TAGS>
  <URL>
    webpronews ebusiness sitepromotion
    20041210HowToBuildTrafficToYourBlog
  </URL>
</DOC>
```

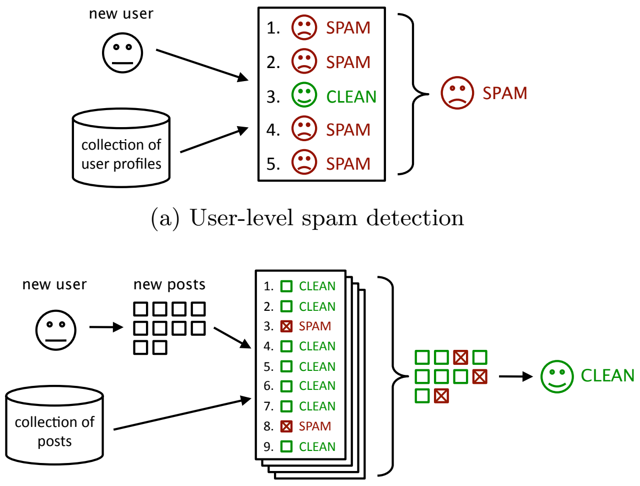
(b) Spam post

Figure 2: Examples of clean and spam posts in our SGML representation

Language models are a class of stochastic n-gram models, generally used to measure a degree of surprise in encountering a certain new span of text, given a training set of text [11]. The core of most language models is a simple n-gram word prediction kernel that, based on a context of two or three previous words, generates a probability distribution of the next words to come. Strong agreement between the expected probabilities and actually occurring words (expressed in perplexity scores or divergence metrics) can be taken as indications that the new text comes from the same source as the original training text. Language models are an essential component in speech recognition [7] and statistical machine translation [2], and are also an important model in information retrieval [14]. In the latter context, separate language models are built for each document, and finding related documents to queries is transformed into ranking documents by the likelihood, estimated through their language model, that each of them generated the query.

In generating our document language models, we have a range of options on the granularity level of what span of text to consider a document. At the most detailed level, we can construct a language model for each individual post, match these to the incoming posts, and use the known spam status of the k best-matching posts already in the system to generate a prediction for the incoming posts. We can also take a higher-level perspective and collate all of a user's posts together to form merged documents that could be considered "user profiles", and generate language models of these individual user profiles. Incoming posts or users can then be matched against the language models of spammers and clean users to classify them as being more similar to one or the other category.

Figure 3 illustrates these two approaches. In the user-level approach depicted in Figure 3(a), the new user's profile—the merged collection of posts made by this user to the system—are matched against all existing profiles. The most similar users then determine the spam label. In the post-level approach in Figure 3(b), each of the new user's posts is matched against all the posts in the collection. The best matching posts help determine the final spam label of the new user.



(a) User-level spam detection

(b) Post-level spam detection

Figure 3: Two levels of spam detection approaches

A third option—at an even higher level of granularity—would be to only consider two language models: one of all spam posts and one of all clean posts. We believe this to be too coarse-grained for accurate prediction, so we did not pursue this further. Another extension to our approach could have been to use language models for the target Web pages or documents such as proposed by[12]. However, it is far from trivial to obtain the full text of all the source documents linked to by the BibSonomy and CiteULike posts. Furthermore, we suspect that incorporating language models from all externally linked Web pages and documents would slow down a real-time spam filtering system to an undesirable degree.

We used the Kullback-Leibler divergence metric to measure the similarity between the language models. The KL-divergence measures the difference between two probability distributions Θ_1, Θ_2 is

$$KL(\Theta_1||\Theta_2) = \sum_w p(w|\Theta_1) \log \frac{p(w|\Theta_1)}{p(w|\Theta_2)} \qquad (1)$$

where $p(w|\Theta_1)$ is the probability of observing the word w according to the model Θ_1 [11, 12].

The Indri toolkit[6] implements different retrieval methods based on language modeling. We used this toolkit to perform our experiments and construct and compare the language models of the posts and user profiles. The language models we used are maximum-likelihood estimates of the unigram occurrence probabilities. We used Jelinek-Mercer smoothing to smooth our language models, which interpolates the language model of a post or user profile with the language model of background corpus, which in our case is the training collection of posts or user profiles. We chose Jelinek-Mercer smoothing as it has been shown to work better for verbose queries than other smoothing methods such as Dirichlet smoothing [15]. Preliminary experiments with Dirichlet smoothing also showed this to be true for our approach, as it was consistently outperformed by Jelinek-Mercer smoothing.

We experimented with both the user-level approach and the post-level approach as illustrated in Figure 3. At the user level, we compared the language models of the user profiles in our validation and test sets with the language models of the profiles in our training set. We then obtained a ranked list of the best-matching training users for each test user. We did the same at the post level by comparing the test post language models with the language models of the training posts. Here, ranked lists of best-matching posts were obtained for each test post. These similarity rankings were normalized, and used as input for the spam classification step.

In our BibSonomy data set we have four different metadata fields available to generate the language models of the posts and user profiles in our training collection: title, description, tags, and tokenized URL. In addition to these 'complete' runs with all fields, we also ran experiments where we only used the information from the four fields separately. An example would be to use only the tags from the training users and the test users. This resulted in five different runs for BibSonomy. For CiteULike we only had the tags available, so we performed only one run here. Finally, another option we tried was using all of the available metadata fields in the training set, but restricting the information used of the users and posts in the validation and test sets. This resulted in four extra runs on the BibSonomy data set, one for each metadata field.

4.2 Spam Classification

After we generated the language models for all posts and user profiles, we obtained the normalized rankings of all training documents relative to each test post or user profile. For each of the best-matching training documents, we used the manually assigned spam labels to generate a single spam score for the new user. The simplest method of calculating such a score would be to output the spam label of the top-matching document. A more elegant option would be to take the most common spam label among the top k hits. We settled on calculating a weighted average of the similarity scores multiplied by the spam labels, as preliminary experiments showed this to outperform the other options.

For post-level classification, this meant we obtained these weighted average spam scores on a per-incoming-post basis. To arrive at user-level spam scores, we then matched each incoming post to a user and calculated the average per-post score for each user. In the rare case that no matching documents could be retrieved, we resorted to assigning a default label of no spam ('0'). Our default classification was to predict a clean user, as for BibSonomy, for instance, these 0.7% of test users for which no matching documents could be retrieved were legitimate users in 84.2% of the cases.

One question remains: how many of the top matching results should be used to predict the spam score? In this, our approach is similar to a k-nearest neighbor classifier, where the number of best-matching neighbors k determines the prediction quality. Using too many neighbors might smooth the pool from which to draw the predictions too much in the direction of the majority class, while not considering enough neighbors might result in basing too many decisions on accidental similarities. We optimized the optimal value for k for all of the variants separately on the AUC scores on the validation set. These optimal values of k were then used to calculate the final scores on the test sets.

5. RESULTS

Table 2 lists the outcomes of our different spam detection approaches on the two collections. Since we optimized on the validation sets, we mainly focus on the test set scores to draw our conclusions. The best performing approach on BibSonomy, at an AUC score of 0.9661, is spam detection at the user level, using all available metadata fields for both the query and collection posts. The best post-level run on BibSonomy also used all of the data for all of the posts, and achieves a score of 0.9536. On the CiteULike data set, the best performance at the user level and post level yields AUC scores of 0.9240 and 0.9079, respectively. This seems to suggest that our approach generalizes well to other data sets and social bookmarking systems. We observe that in general, using the language models constructed at the user level outperforms using the post-level language models. This is also visible in Figure 4, which shows the ROC curves for the best user-level and post-level runs for each collection.

An interesting difference between the validation set and the test set is that using only the tags to construct the language models yields the best performance on the validation set, whereas performance using only tags drops markedly on the test set. Using all available metadata fields results in considerably more stable performance across both BibSonomy evaluation sets, and should therefore be considered the preferred variant.

Another interesting observation is the difference in the optimal size of the neighborhood k used to predict the spam labels. In almost all cases, the post-level approaches require a smaller k than at the user level. The optimal neighborhood size for CiteULike is the same for both the user-level and the post-level approach, and is surprisingly smaller than for BibSonomy.

[6] Available at http://www.lemurproject.org

Table 2: Results of our approaches on the BibSonomy and CiteULike data sets. Scores reported are AUC, with the best scores for each set of collection runs printed in bold. The two "all fields" rows are one and the same run, but they are repeated here for comparison purposes. The optimal neighborhood size k is listed for each user-level and post-level runs. For the same set of runs, the same value of k was used in both the validation and the test set.

		User level			Post level		
Collection	**Fields**	**Validation**	**Test**	**k**	**Validation**	**Test**	**k**
BibSonomy (matching fields)	all fields	0.9682	**0.9661**	235	0.9571	**0.9536**	50
	title	0.9290	0.9450	150	0.9055	0.9287	45
	description	0.9055	0.9452	100	0.8802	0.9371	100
	tags	**0.9724**	0.9073	110	**0.9614**	0.9088	60
	URL	0.8785	0.8523	35	0.8489	0.8301	8
BibSonomy (single fields in evaluation sets)	all fields	0.9682	**0.9661**	235	0.9571	**0.9536**	50
	title	0.9300	0.9531	140	0.9147	0.9296	50
	description	0.9113	0.9497	90	0.8874	0.9430	75
	tags	**0.9690**	0.9381	65	**0.9686**	0.9251	95
	URL	0.8830	0.8628	15	0.8727	0.8369	15
CiteULike	tags	**0.9329**	**0.9240**	5	**0.9262**	**0.9079**	5

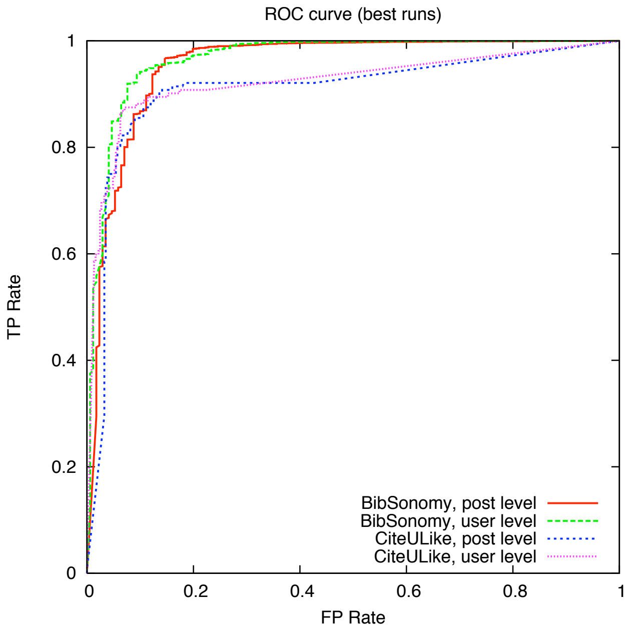


Figure 4: ROC curves of the best-performing user-level and post-level approaches for both collections.

Finally, comparing the two different sets of BibSonomy runs, using only the matching fields from both the collection and the incoming test posts results in slightly lower scores than when using the full data available in the collection and only restricting the fields of the incoming posts.

6. DISCUSSION & CONCLUSIONS

In this paper we presented a adversarial information retrieval approach employing language modeling to detect spam in social reference management websites. We start by using language models to identify the best-matching posts or user profiles for incoming users and posts. We then look at the spam status of those best-matching neighbors and use them to guide our spam classification. The results indicate that our language modeling approach to spam detection in social bookmarking systems is promising, yielding 0.953 and 0.966% AUC scores on spam user detection. This confirms the findings of [12], who applied a similar two-stage process using language modeling to detecting blog spam, albeit on a much smaller scale. One particular advantage of our approach is that it could be implemented with limited effort on top of an existing social bookmarking search engine. After any standard retrieval runs, the top k matching results can then be used to generate the spam classification, requiring only a lookup of predetermined spam labels.

We experimented with using language models at two different levels of granularity and found that matching at the user level and using all of the available metadata gave the best results. In general, matching at the user level resulted in better performance then matching at the post level for both BibSonomy and CiteULike. This difference can be partly explained by the fact that the spam labels for the users in both data sets were judged and assigned at the user level, as this is the desired level of the end application; even if a spam user posts 'genuine' posts, the entire content of the spam user should be deleted on grounds of the adversarial intentions behind them. Yet, the 'genuine' posts of spam users were automatically flagged as spam, thereby introducing more noise for the post-level classification than for the user-level classification. Early classification of spam users at their earliest posts can therefore be expected to be less accurate than the reported 0.95–0.96 range; post-level AUC scores suggest this accuracy would be closer to 0.91–0.95.

Another likely explanation for the better performance of the user-level approach is sparseness at the post level. A post-level approach is more likely to suffer from incoming posts with sparse or missing metadata. For instance, although 99.95% of all posts in the BibSonomy data set have valid tags[7], this also means that it is possible for incom-

[7] Valid meaning with a tag other than system:unfiled, the default tag that is assigned by the system when no tags were added by the user.

ing posts to have no tags. Without any tags as metadata or sparse metadata in the other fields, our approach cannot find any matching posts in the system. At the user level, this is much less likely to happen: only 0.009% of all users never assign any tags. Aggregating all metadata of a user's posts can yield enough metadata to base reliable predictions on, whereas the post-level approach can be affected by this to a greater extent. Missing tags might also be a reason for the fact that performance on CiteULike is slightly lower than performance on BibSonomy.

In the previous section, we observed that, comparing the two different sets of BibSonomy runs, using only the matching fields from both the collection and the incoming test posts resulted in slightly lower scores than when using the full data available from the collection and only restricting the fields of the incoming posts. This is probably also a matter of how much data is used: using only matching fields reduces the amount of available metadata for generating the language models, which could make the matching process slightly less effective. We can offer no explanation for the big drop in performance of the tag-based approaches on BibSonomy when comparing the validation set and the test set, other than overfitting on the validation set, as was to be expected.

Finally, when looking at the optimal neighborhood sizes k for BibSonomy, we see that in almost all cases the post-level approaches require a smaller k than at the user level. We believe this is because the presence of multiple topics in user profiles. Individual posts are usually about a single topic, whereas a user profile is composed of all of that user's posts, which are likely to be about multiple topics of interest. This makes finding the related posts to an individual post easier, in the sense that it requires less nearest neighbors to arrive at a prediction. At the user level, however, different parts of a user's profile might match up with different users already in the system, thus requiring more nearest neighbors to arrive at a reliable prediction.

7. FUTURE WORK

No spam detection approach can be expected to remain successful without adapting to the changing behavior of the spammers. One way spammers could circumvent our method of spam detection would be by generating metadata with a similar language model to the clean posts in the system. This way, spammers can make it more complicated for our approach to distinguish between themselves and genuine users. However, this also makes it more difficult for the spammers themselves: it is very hard for a spammer to post resources to a social bookmarking system that will be both similar to existing posts and to the language of the spam entry [12]. In addition, such behavior could easily be countered by extending our method to include the language models of the pages and documents behind the bookmarks. In the case of sparse metadata, this might be able to boost performance of the spam detection algorithm. Extending our approach in such a way is one of the possible avenues for future work. Another option would be to include extra features such as the PageRank scores of the bookmarked pages, and see if pages with low PageRank are more predictive of spam status than others.

Acknowledgments

We would like to thank Kirstine Wilfred Christensen for her help in annotating the CiteULike data set for spam. The work described in this paper was funded by SenterNovem / the Dutch Ministry of Economics Affairs as part of the IOP-MMI À Propos project, and by the Netherlands Organization for Scientific Research as part of the NWO Vernieuwingsimpuls program.

8. REFERENCES

[1] T. Bogers and A. Van den Bosch. Using Language Models for Spam Detection in Social Bookmarking. In *Proceedings of 2008 ECML/PKDD Discovery Challenge Workshop*, pages 1–12, September 2008.

[2] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. Lafferty, R. L. Mercer, and P. S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[3] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. *Machine Learning*, 31, 2004.

[4] Z. Gyöngyi and H. Garcia-Molina. Web Spam Taxonomy. In *AIRWeb '05: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, pages 39–47, Chiba, Japan, May 2005.

[5] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[6] A. Hotho, D. Benz, R. Jäschke, and B. Krause. Introduction to the 2008 ECML/PKDD Discovery Challenge Workshop. In *Proceedings of 2008 ECML/PKDD Discovery Challenge Workshop*, September 2008.

[7] F. Jelinek. Self-organized Language Modeling for Speech Recognition. *Readings in Speech Recognition*, pages 450–506, 1990.

[8] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating Spam in Tagging Systems. In *AIRWeb '07: Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, pages 57–64, New York, NY, USA, 2007. ACM.

[9] B. Krause, A. Hotho, and G. Stumme. The Anti-Social Tagger - Detecting Spam in Social Bookmarking Systems. In *AIRWeb '08: Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, 2008.

[10] R. Krestel and L. Chen. Using Co-occurrence of Tags and Resources to Identify Spammers. In *Proceedings of 2008 ECML/PKDD Discovery Challenge Workshop*, pages 38–46, September 2008.

[11] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[12] G. Mishne, D. Carmel, and R. Lempel. Blocking Blog Spam with Language Model Disagreement. In *AIRWeb '05: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, pages 1–6, New York, NY, USA, 2005. ACM.

[13] I. Ounis, M. de Rijke, C. McDonald, G. Mishne, and I. Soboroff. Overview of the TREC 2006 Blog Track. In *TREC 2006 Working Notes*, 2006.

[14] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, New York, NY, 1998. ACM Press.

[15] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.