Detecting Spam Blogs: A Machine Learning Approach*

Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, Anupam Joshi
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{kolari1, aks1, finin, oates, joshi}@cs.umbc.edu

Abstract

Weblogs or blogs are an important new way to publish information, engage in discussions, and form communities on the Internet. The *Blogosphere* has unfortunately been infected by several varieties of spam-like content. Blog search engines, for example, are inundated by posts from splogs – false blogs with machine generated or hijacked content whose sole purpose is to host ads or raise the PageRank of target sites. We discuss how SVM models based on local and link-based features can be used to detect splogs. We present an evaluation of learned models and their utility to blog search engines; systems that employ techniques differing from those of conventional web search engines.

Introduction

Weblogs or blogs are web sites consisting of dated entries typically listed in reverse chronological order on a single page. Based on the nature of these entries, blogs are considered to be one of personal journals, market or product commentaries, or just filters that discuss current affairs reported elsewhere, participating in an online dialogue. Independent of the content genre of blogs, they constitute such an influential subset on the Web, that they collectively create what is now known as the *Blogosphere*. While traditional search engines continue to discover and index blogs, the Blogosphere has produced custom blog search and analysis engines, systems that employ specialized information retrieval techniques. As the Blogosphere continues to grow, several capabilities have become critical for blog search engines. The first is the ability to recognize blog sites, understand their structure, identify constituent parts and extract relevant metadata. A second is to robustly detect and eliminate spam blogs (splogs).

Splogs are generated with two often overlapping motives. The first is the creation of fake blogs, containing gibberish or hijacked content from other blogs and news sources with the sole purpose of hosting profitable context based advertisements. The second, and better understood form, is to create false blogs, that realize a link farm (Wu & Davison 2005) intended to unjustifiably increase the ranking of affiliated sites. The urgency in culling out splogs has become all the more evident in the last year. The problem's significance is frequently discussed and reported on by blog search and analysis engines (Umbria 2005; Cuban 2005), popular bloggers (Rubel 2005), and more recently through a formal analysis by us (Kolari, Java, & Finin 2006). This analysis makes some disturbing conclusions on spam faced by ping servers, intermediate servers which relay pings from freshly updated blogs to blog search engines. Approximately 75% of such pings is received from splogs.

In general, the problem of spam is not new for Internet based applications. The ease of content creation (and plagiarism) and distribution has made the Internet a haven for spammers. If local machine learning models have been found effective for e-mail classification, on the Web this problem is studied from the PageRank and the link-based classification perspective. However, it is still not clear as to which of these techniques are useful in the Blogosphere, and in what form. This is, in fact, the primary motivation for our work on splog detection. In addition, we have a working system that performs far better than current techniques employed by blog search engines.

This paper contributes to solving the splog problem in three ways: (i) we formalize the problem of splog detection as it applies to blog search engines; (ii) we introduce new features, argue why they are important, and back our arguments with experimental validation using Support Vector Machines; and (iii) we introduce the notion of local and global models from the blog search engine perspective, and detail how local models can be coupled with global ones.

The Splog Detection Problem

In the classical web graph model $G(X, E)$, the set of nodes X represent web pages, and the set of edges E stand for hyper-links between these pages. In contrast, blog search engines treat the Web using a slightly more intricate and tuned model, $G([B, N, W], E)$, where $X = B \cup N \cup W$. The membership of nodes in this web-graph is in either of B, N or W, where B is the set of all blogs, N is the set of all news-sources (edited content), and W is the set representing the rest of the Web. The edges in this graph are highly localized to their respective subsets, but also feature many

*This work is supported by NSF Awards NSF-ITR-IIS-0326460 and NSF-ITR-IDM-0219649

crossover points[1].

Splog detection is a classification problem within the blogosphere subset, B. Practically, the result of such a classification leads to disjoint subsets B_A, B_S, B_U where B_A represents all authentic blogs, B_S represents splogs and B_U represents those blogs for which a judgment of authenticity or spam has not yet been made. The splog detection problem for any node $x \in B$, in the generic logistic regression setting can be considered as:

$$P(x \in B_S/O(x)); P(x \in B_S/L(x))$$

where $x \in B$, $O(x)$ represents local features, and $L(x)$ represents the link features.

On preliminary observation, this appears to be a classical web classification problem or perhaps a variation on the web spam problem (Gyöngyi & Garcia-Molina 2005) addressed by TrustRank (Gyöngyi, Garcia-Molina, & Pedersen 2004). However, the methodologies used by blog search engines[2] and the nature of the Blogosphere make this an interesting special case of web classification.

- **Nature of Blog Search.** Blog search engines rank results primarily by recency, rather than using popular social ranking techniques (Page *et al.* 1998). This is less of a technology related choice, and driven more by an audience that demands tracking "buzz" rather than authority. Increasing authority in the web link graph is less important for spam blogs, at least for those that display context based advertisements. This renders using link-only spam detection much less effective.
- **Quicker Assessment.** The availability of blogging services that are free, easy to use, remotely hosted and that have convenient APIs have led to the popularity of blogging. However, this has also made them much easier to use by spammers (Pirillo 2005), creating large spikes of splogs on the blogosphere, similar to those seen in e-mails. Unlike web spam detection which mainly applies detection algorithms days after the activity of web spam creation, splogs demand a much quicker assessment and action, similar to e-mails.
- **Genre of Blog Content.** Most of automated web spam detection techniques have ignored the role of local features in identifying spam pages. This is largely due to the nature of the web, which hosts content across almost all topics. The blogosphere, however, is a communication medium for specific genres like personal opinions and journals; and for unedited content words appearing on a page can provide interesting cues useful for classification.
- **Search Engine Coverage.** Blog search engines work on a much smaller subset of the Web than do comprehensive search engines. They employ preferential crawling and indexing towards the set B and N, and hence do not have knowledge of the entire web graph. Splogs typically link to non-splog Web pages, attempting to influence their PageRank, and it is infeasible for blog search engines to fetch all such pages before making a splog judgment. With such a state of the art, any feasible solution should be based on a combination of local and relatively simple link based models.

Some recent work addresses the problem of spam in different contexts. This includes the use of local content-based features (Drost & Scheffer 2005) and statistical outliers (Fetterly, Manasse, & Najork 2004) for web spam detection, and utilizing language models (Mishne, Carmel, & Lempel 2005) for blog comment spam detection. Though these approaches could be adapted, the problem of spam blogs, and the effectiveness of certain new features in this specialized domain have not yet been formally studied.

Background

Our work is based on a seed data set (Kolari, Finin, & Joshi 2006) of 700 positive (splogs) and 700 negative (authentic blog) examples. Each one is the entire HTML content of the blog home-page. To confirm the labels by the first author in prior work, the second author randomly sampled for around 500 samples, independently labeling them. Each labeling required two to three minutes, depending on whether the judgment could be made based on just local features or required following the page's in-links and out-links. Discriminating features included content, out-links, in-links, post time-stamps, blog URL, post comments and blogrolls. We verified the aggregate of this labeling with existing labels of the seed set. Labeling matched more than 90% of the time.

To simulate the index of a typical blog search engine, we used an API[3] provided by a popular blog search engine to fetch in-links for the seed set, almost all of which were from blogs. We also fetched the out-links from this seed set making sure we simulated the preferential indexing of blog search engines, and categorized each as either a news source or another blog[4].

The final dataset comprised approximately 20K unique pages with 8K outgoing links from the seed set and 16K incoming links to the seed set. There were fewer out-links because they were collected from a current snapshot of the blog home-page. While no formal study exists, we believe that out-links from the blog home-page which lists the most recent posts provides a sufficiently accurate summary of out-links from the entire blog, at least for the purpose of spam detection.

All of the models are based on SVMs (Boser, Guyon, & Vapnik 1992), which are known to perform well in classification tasks (Joachims 1998). Our evaluation uses the standard *area under ROC curve* metric over a 10-fold cross-validation and makes use of tools provided by libsvm (Chang & Lin 2001). In addition, the link-based model

[1]The nature of these crossover links is a subject of recent studies towards understanding the influence of the blogosphere on the Web and vice-versa.

[2]A part of this analysis is based on discussions with a blog search engine provider.

[3]http://developers.technorati.com/wiki

[4]The categorization was done with a module we developed whose accuracy is about 98%.

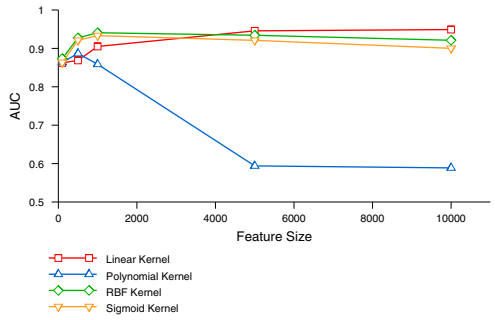

Figure 1: *Results from using Bag-of-words as Binary Features*

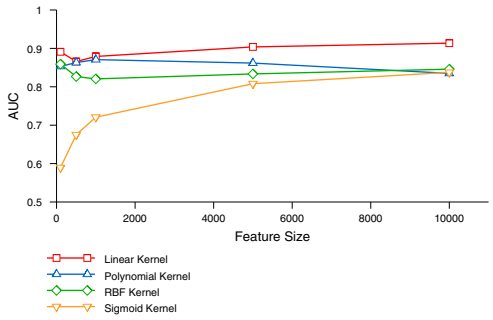

Figure 2: *Results from using Bag-of-words as TFIDF Features*

makes use of a logistic regression model using local features, also supported by libsvm. Kernel evaluations use default parameters, which were the most effective during our trials. Feature selection to identify the top features was made using frequency count over the entire sample set. All charts are plotted only for feature sizes of 100, 500, 1000, 5000, 10000. Larger feature sizes did not significantly improve results.

Local Models

In this section we describe how a blog's local features can be effective and back these claims with experimental results. A *local feature* is one that is completely determined by the contents of a single web page, i.e. it does not require following links or consulting other data sources. A local model is one built only using local features. Local models can provide a quick assessment of the authenticity of blogs.

Words and Word N-Grams

The most common type of feature used for topic and e-mail spam classification tasks is the bag-of-words, in which words occurring on page are treated as features. To verify its utility, we created bag-of-words for the samples based on their textual content. We did not use stop lists or perform stemming. Results from using the binary feature encoding is depicted in Figure 1 and TFIDF encoding in Figure 2.

Clearly, bag-of-words is quite effective using binary features and a linear kernel. To analyze the discriminating features we used a simple approach of ordering features by weights assigned to them in the SVM model. It turns out that the model was built around features which the human eye would have typically overlooked. Blogs often contain content that expresses personal opinions, so words like "I", "We", "my", "what" appear commonly on authentic blog posts. To this effect, the bag-of-words model is built on an interesting "blog content genre" as a whole. In general, such a content genre is not seen on the Web, which partly explains why spam detection using local textual content is not all that effective there.

In all of the evaluations that follow binary features were the most effective. In addition, polynomial kernels did not perform all that well in our domain of interest for binary

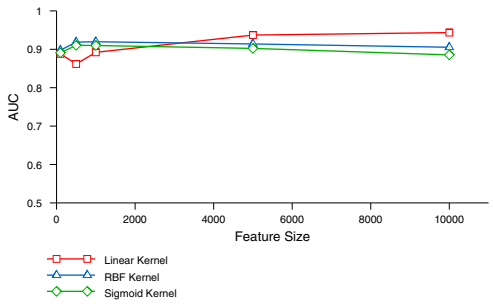

Figure 3: *Results from using 2gram-words as Binary Features*

features. Charts that follow are hence limited to linear, RBF and Sigmoid kernel based on binary feature encoding.

An alternative methodology on using textual content for classification is the bag-of-word-N-Grams, where N adjacent words are used as a feature. We evaluated both bag-of-word-2-Grams and bag-of-word-3-Grams. Results for word 2-grams are depicted in Figure 3 and those for word 3-grams are shown in Figure 4.

Interesting discriminative features were observed in this experiment. For instance, text like "comments-off" (comments are usually turned-off in splogs), "new-york" (a high paying advertising term), "in-uncategorized" (spammers do not bother to specify categories for blog posts) are features common to splogs, whereas text like "2-comments", "1-comment", "i-have", "to-my" were some features common to authentic blogs. Similar kind of features ranked highly in the 3-word gram model. Though these models were not as effective as bag-of-words, they came a very close second, providing significant merit to any splog detection system that uses them.

Tokenized Anchors

Our previous work revealed that a page's *anchor text* provided features that were useful for discriminating splogs from blogs. Anchor text is the text that appears in an HTML link (i.e., between the `<a...>` and `</a>` tags.) We used a bag-of-anchors feature, where anchor text on a page, with

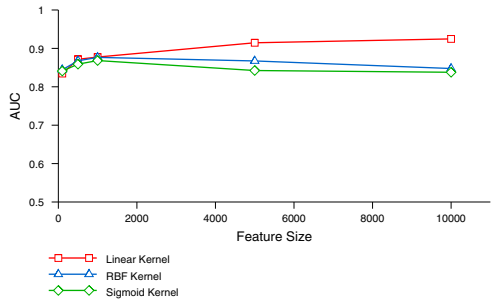

Figure 4: *Results from using 3gram-words as Binary Features*

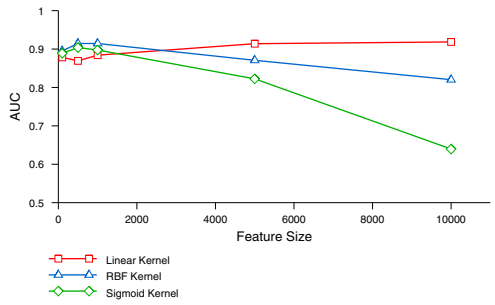

Figure 5: *Results from using Bag-of-anchors as Binary Features*

multiple word anchors split into individual words, is used. Anchor text is frequently used for web page classification, but typically to classifying the target page rather than the one hosting the link. Figure 5 shows them to be effective though not as good as bag-of-words. We observed that "comment" and "flickr" were among the highly ranked features for authentic blogs.

Tokenized URLs

The intuition behind this feature choice was to see if URLs on a page, both local and outgoing can be used as effective attributes for splog detection. This is motivated by the fact that many splogs point to the ".info" domain, whereas many authentic blogs point to well known websites like "flickr", "technorati" and "feedster" and strongly associated with blogs. We term these features as bag-of-urls, arrived at by tokenizing URLs using "/" and ".". Figure 6 confirms that bag-of-urls features are effective.

From these charts, it is clear that bag-of-anchors or bag-of-urls taken alone are quite effective for splog detection. Also the degradation of performance of non-linear kernels for higher feature counts can be attributed to the inclusion of many non-relevant features in binary feature encoding. Linear kernels resulted in AUC values of as high as 0.95 for bag-of-words.

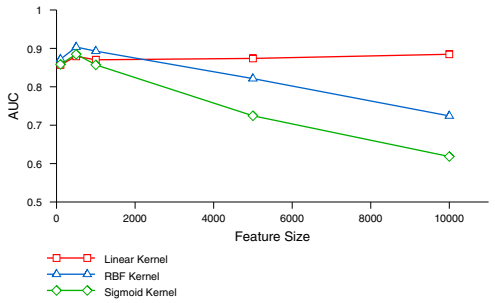

Figure 6: *Results from using Bag-of-URLs as Binary Features*

Specialized features

In the final set of evaluation using local models, we explored the utility of language models and other heuristics as shown in Table 1. Unlike binary features used in previous experiments, all these features were encoded using numeric floating point values between 0 and 1, except for feature number 11 whose value could potentially be greater than 1. Values for the first five features were based on extracted named entities using the ling-pipe[5] toolkit, and the ratios were computed using the count of all words on the page. Splogs usually promote products or websites (usually named entities) by repeating such named-entities many times within a page. This is a standard exploit on TFIDF indexing employed by search engines.

We also experimented with other heuristic based features. The repeatability of text, URLs and anchors on splogs prompted us to use compression ratio as a feature. The intuition being that such pages have low compression ratio. Compression ratio was computed as the ratio of the size of deflated to inflated size for each of the feature types. To capture the notion of splogs containing a higher amount of anchor text and URLs as compared to the size of the page, we computed the ratio of their character size to the character size of the blog home-page as a whole.

In addition we also used the ratio of distinct URLs (and Anchors) on a page divided by all URLs (and anchors) to check for repeating URL's and anchor text, which is quite common in splogs. To evaluate the hypothesis that hyperlinks on splogs feature many URLs with hyphens, we employed the ratio of number of hyphens to number of URLs as a feature.

Contrary to expectations, results from using these features together were significantly less effective than the standard features. The best performance was observed when using linear kernels with a value of 0.75 for AUC.

Global Models

A global model is one that uses some non-local features, i.e., features requiring data beyond the content of Web page under test. Most blog related global features capture relations among web resources. In particular, we have investigated

[5]http://www.alias-i.com/lingpipe/

No.	Feature Description
1	Location Entity Ratio
2	Person Entity Ratio
3	Organization Entity Ratio
4	Female Pronoun Entity Ratio
5	Male Pronoun Entity Ratio
6	Text Compression Ratio
7	URL Compression Ratio
8	Anchor Compression Ratio
9	All URLs character size Ratio
10	All Anchors character size Ratio
11	Hyphens compared with number of URLs
12	Unique URLs by All URLs
13	Unique Anchors by All Anchors

Table 1: *We evaluated the utility of a number of specialized features for splog identification in a local attribute model.*

the use of link analysis to see if splogs can be identified once they place themselves on the web hyper-link graph. We want to capture the intuition that authentic blogs are very unlikely to link to splogs and that splogs frequently do link to other splogs. We tackle this problem by using splogs that could be identified using local attribute models. These splogs now become part of link distributions over which link-based models are constructed.

Labeling Nodes using Local Models

In the spirit of simulating the index of a typical blog search engine we employed the following technique. We started with the seed set, and all of its in-link and out-link pages. We then used two fairly robust classifiers (with accuracy 95%, and part of our other projects) on blog and news-media detection to identify members of the set B, N and W for the in-link and out-link pages.

Next, using this B set created from in-link and out-link pages, we experimented using different cut-off thresholds on a logistic regression based splog detection model built using local features. Using these cut-offs we labeled members of the set B. For any node x identified as a blog (and not part of the seed set), these thresholds th_1 and th_2 are used as:

$$x \in B_S,\ if\ P(x \in B_S/O(x)) > th_1$$
$$x \in B_A,\ if\ P(x \in B_A/O(x)) > th_2$$
$$x \in W,\ otherwise$$

The interpretation of these thresholds is listed in Table 2. The first and last values completely ignore the use of a local feature model to feed into the link-model.

Link Features

The link features of our choice are similar to those used in other link-based classification tasks (Lu & Getoor 2003). Referring to our graph model, for all nodes in $X = \{x_1, x_2, .., x_N\}$, if $e_{i \to j} \in E$ is a hyper-link from node x_i to node x_j, then:

Threshold	Comment
$th_1 = 0, th_2 = 1$	All Blogs are Splogs
$th_1 = 0.25, th_2 = 0.25$	Aggressive Splog Threshold
$th_1 = 0.5, th_2 = 0.5$	Intermediate Splog Threshold
$th_1 = 0.75, th_2 = 0.75$	Conservative Splog Threshold
$th_1 = 1, th_2 = 0$	All Blogs are Authentic

Table 2: *Interpretation of Probability Thresholds.*

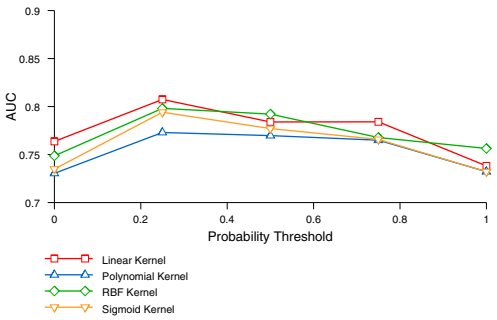

Figure 7: *Link features with binary encoding*

$L_I(x_i)$ - the set of all incoming links of node X_i, $\{X_j | E_{j \to i} \in E\}$

$L_O(x_i)$ - the set of all outgoing links of node x_i, $\{x_j | e_{i \to j} \in E\}$

$L_C(x_i)$ - the set of objects co-cited with node x_i, $\{x_j | x_j \neq x_i$, and there exists another object x_k, where $x_k \neq x_j, x_k \neq x_i$ and x_k links to both x_i and x_j $\}$

The nodes in each of these link distribution sets were assigned to their respective web graph sub-sets. Our features were finally based on using these assignments and computing set membership cardinality as ($|B_U|$, $|B_S|$, $|B_A|$, $|N|$, $|W|$) for each of the link-distributions. This created a total of fifteen features. We experimented with both binary and count based feature representation. Results are shown in Figure 7 and Figure 8, and probability threshold th_1 is represented on the x axis. These charts show how local models that can be used to pre-classify out-links and in-links of the seed set is effective, and renders an otherwise inconsequential link-features only model useful. Augmenting these link features of the seed set with their bag-of-words did not improve accuracy beyond bag-of-words taken alone. This suggests that given the current nature of splogs, local textual content is arguably the most important discriminating feature.

Discussion and Future Work

The state of the art in splog detection is currently set by blog search engines. To the best of our knowledge, all of the major blog search engines have splog filters that use one or

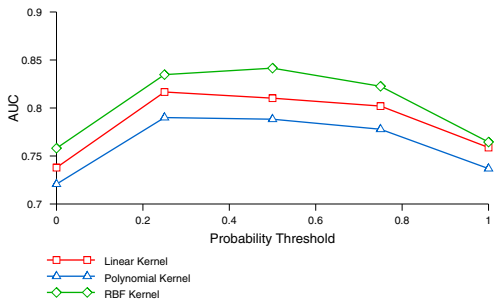

Figure 8: *Link features with frequency encoding*

more techniques including manual checking, URL patterns, URL blacklists, IP blacklists, and/or hand coded heuristics. To our knowledge, none use a machine learning based approach like the one we have developed. Our implemented system has two significant advantages: it appears to be significantly better at splog detection and it can be easily retrained as the character and tactics of splogs change.

To informally verify our claim of improved performance, we sampled for 10,000 blogs from a popular blog search engine[6], blogs which had passed its native splog filters. This sample was generated by querying the blog search engine for fresh posts, using words chosen randomly from a dictionary. Our system detected 1500 splogs in this set, demonstrating both the efficacy of our approach and the high frequency of splogs in the Blogosphere.

Our machine learning approach has allowed us to experiment with different features and evaluate their utility for recognizing splogs. Even though many plausible sounding features were found not to be useful, we have decided to keep a catalog of them for possible future use. As the strategies and tactics of sploggers evolves in response to attempts to eradicate them, some of these features may become useful. We are also assembling a list of potential new features, and exploring the changes that need to be made by blog search engines to better adopt relational splog detection techniques. Finally, as sploggers increasingly obfuscate discriminative features, splog detection can be considered in the adversarial classification (Dalvi *et al.* 2004) setting, which is another direction we are pursuing.

References

Boser, B. E.; Guyon, I. M.; and Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, 144–152. New York: ACM Press.

Chang, C.-C., and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cuban, M. 2005. A splog here, a splog there, pretty soon it ads up and we all lose. [Online; accessed 22-December-2005; `http://www.blogmaverick.com/entry/1234000870054492/`].

Dalvi, N. N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *KDD*, 99–108.

Drost, I., and Scheffer, T. 2005. Thwarting the nigritude ultramarine: Learning to identify link spam. In *ECML*, 96–107.

Fetterly, D.; Manasse, M.; and Najork, M. 2004. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases*, 1–6. New York, NY, USA: ACM Press.

Gyöngyi, Z., and Garcia-Molina, H. 2005. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*.

Gyöngyi, Z.; Garcia-Molina, H.; and Pedersen, J. 2004. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, 576–587. Morgan Kaufmann.

Joachims, T. 1998. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, 137–142. London, UK: Springer-Verlag.

Kolari, P.; Finin, T.; and Joshi, A. 2006. SVMs for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*.

Kolari, P.; Java, A.; and Finin, T. 2006. Characterizing the splogosphere. In *WWW 2006, 3rd Annual Workshop on the Weblogggging Ecosystem: Aggregation, Analysis and Dynamics*.

Lu, Q., and Getoor, L. 2003. Link-based classification. In *ICML*, 496–503.

Mishne, G.; Carmel, D.; and Lempel, R. 2005. Blocking blog spam with language model disagreement. In *AIRWeb '05 - 1st International Workshop on Adversarial Information Retrieval on the Web, at WWW 2005*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

Pirillo, C. 2005. Google: Kill blogspot already!!! [Online; `http://chris.pirillo.com/blog/_archives/2005/10/16/1302867.html`].

Rubel, S. 2005. Blog content theft. [Online; `http://www.micropersuasion.com/2005/12/blog_content_th.html`].

Umbria. 2005. Spam in the blogosphere. [Online; `http://www.umbrialistens.com/consumer/showWhitePaper`].

Wu, B., and Davison, B. D. 2005. Identifying link farm spam pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, 820–829. New York: ACM Press.

[6] http://technorati.com