Detecting Splogs via Temporal Dynamics Using Self-Similarity Analysis

YU-RU LIN and HARI SUNDARAM
Arizona State University
and
YUN CHI, JUNICHI TATEMURA, and BELLE L. TSENG
NEC Laboratories America

This article addresses the problem of spam blog (splog) detection using temporal and structural regularity of content, post time and links. Splogs are undesirable blogs meant to attract search engine traffic, used solely for promoting affiliate sites. Blogs represent popular online media, and splogs not only degrade the quality of search engine results, but also waste network resources. The splog detection problem is made difficult due to the lack of stable content descriptors.

We have developed a new technique for detecting splogs, based on the observation that a blog is a dynamic, growing sequence of entries (or posts) rather than a collection of individual pages. In our approach, splogs are recognized by their temporal characteristics and content. There are three key ideas in our splog detection framework. (a) We represent the blog temporal dynamics using self-similarity matrices defined on the histogram intersection similarity measure of the time, content, and link attributes of posts, to investigate the temporal changes of the post sequence. (b) We study the blog temporal characteristics using a visual representation derived from the self-similarity measures. The visual signature reveals correlation between attributes and posts, depending on the type of blogs (normal blogs and splogs). (c) We propose two types of novel temporal features to capture the splog temporal characteristics. In our splog detector, these novel features are combined with content based features. We extract a content based feature vector from blog home pages as well as from different parts of the blog. The dimensionality of the feature vector is reduced by Fisher linear discriminant analysis. We have tested an SVM-based splog detector using proposed features on real world datasets, with appreciable results (90% accuracy).

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; H.4.3 [**Information Systems Applications**]—*Communications Applications*; H.5.1 [**Information Interfaces and Representation**]— Multimedia Information Systems—*Evaluation/methodology*; H.5.4 [**Information Interfaces and Representation**]—*Hypertext/Hypermedia*

This work was supported by funds from NEC Laboratories America, Cupertino, CA.
Authors' address: Y.-R. Lin and H. Sundaram, Arts Media and Engineering Program, Arizona State University, AZ 85281; Y. Chi, J. Tatemura, B. L. Tseng, NEC Laboratories America, Cupertino, CA 95014.

General Terms: Experimentation, Measurement, Algorithms, Human Factors

Additional Key Words and Phrases: Blogs, temporal dynamics, regularity, spam, splog detection, topology, self-similarity

ACM Reference Format:

Lin, Y.-R., Sundaram, H., Chi, Y., Tatemura, J., and Tseng, B. L. 2008. Detecting splogs via temporal dynamics using self-similarity analysis. ACM Trans. Web, 2, 1, Article 4 (February 2008), 35 pages. DOI = 10.1145/1326561.1326565 http://doi.acm.org/10.1145/1326561.1326565

1. INTRODUCTION

This article addresses the problem of spam blog (splog) detection using temporal and structural regularity of content, post time and links. Splogs are undesirable blogs meant to attract search engine traffic, used solely for promoting affiliate sites. The splog detection problem is important—blogs represent a highly popular new media for communication, and the presence of splogs degrades blog search results as well as wastes network resources like crawling and indexing cost. The splog detection problem is made difficult by the lack of stable content features in a splog. Additionally, spammers' tricks used to create splogs are constantly evolving to avoid detection by blog search engines. We have developed a new technique for detecting splogs, based on the key observation that a blog is a dynamic, growing sequence of entries (or posts) rather than a static collection of individual pages. In our approach, splogs are recognized by their temporal characteristics as well as their content.

The growth in the numbers of blogs has led to an alarming increase in the number of splogs. The splogs have had a detrimental effect on the blogosphere. The authors of Umbria [2006] reported that for the week of Oct. 24, 2005, 2.7 million blogs out of 20.3 million (10–20%) were splogs, and that an average of 44 of the top 100 blogs search results in the three popular blog search engines came from splogs. It has been estimated in 2006 that 75% of new pings[1] came from splogs; more than 50% of claimed blogs pinging the Web site www.weblogs.com are splogs [Kolari 2005]. Statistics reveal that splogs can cause problems including: (1) the degradation of information retrieval quality and (2) the significant waste of network and storage resources.

The main motive for creating a splog is to drive visitors to affiliated sites that have some profit-making mechanisms, such as Google AdSense or other pay-per-click (ppc) affiliate programs [Wikipedia]. Spammers increase splog visibility by getting indexed with high rank with respect to the topics they are interested (e.g., mortgage) on popular search engines, which usually involves schemes (or spam tricks) such as keyword stuffing or content duplication.

1.1 Related Work

In this section we discuss related work. We shall start with a review of Web spam detection research, followed by splog-related research. Subsequently, we

[1]Pings are messages sent from blog and publishing tools to a centralized network service (ping server) providing notification of newly published posts or content [*Wikipedia*]. In 2007, the percentage of pings that are sent from spam blogs is estimated to be 53%.

will distinguish our work from existing approaches. We also discuss research in other domains that share similar analytical frameworks.

Splogs are relatively new phenomena; however there has been a lot of work on Web spam detection. While there are critical differences between the two, a review of Web spam research provides useful insights. Gyöngyi and Garcia-Molina [2005] present a Web spam taxonomy and provide a comprehensive discussion on the phenomenon of *spamming* in the Web. Following their taxonomy, we categorize prior works on detecting Web spam into link analysis and content analysis.

With the significant use of link-based ranking techniques such as PageRank, *link spamming* appears to be a popular form of Web spam. There has been an increasing interest in the research community in detecting link spam based on their hyperlink structure. Gyöngyi et al. [2006, 2004] identify link spam by propagating the trust score from a relatively small set of reputable seed pages to the rest of the Web pages by following the citation links. Without manually identifying a seed set, Wu and Davison [2005] present an automatic method of determining and expanding a seed set of spam pages based on the overlap between incoming and outgoing links on webpages. The similarity information and global graph properties in the link structure is investigated for detecting spam pages in Fogaras and Racz [2005] and Benczur et al. [2005]. Temporal correlation between Web snapshot graphs has been exploited in Shen et al. [2006]. They detect link spam using temporal link information extracted from two snapshots of the link graphs.

Content analysis has mainly been used to detect *term spamming*—another type of Web spam meant to make spam pages relevant to some queries or popular keywords. As observed in Fetterly et al. [2004] and Ntoulas et al. [2006], the spamming pages automatically generated by spammers' scripts differ in their statistical content and linkage properties from those pages authored by a human, and it is these distinct properties that serve as good indicators for certain classes of spam pages. Fetterly et al. [2005] investigate a class of spam pages that are automatically assembled by stitching together popular phrases. The phrase-level replication in spam pages is detected by a *shingling* technique. In addition to statistical content analysis, Urvoy et al. [2006] propose a structural content analysis by tracking spam Web sites based on their style similarity in HTML source code.

Splogs have been considered to be special case of Web spam [Kolari et al. 2006a]. The characteristics of splogs are investigated in Kolari et al. [2006b]; Kolari et al. [2006c]; and Lin et al. [2006]. In order to combat spam blogs, the Kolari et al. [2006a, 2006d] suggest using a set of content-based features such as *bag-of-word* and *bag-of-anchors*, as well as link-based information. In their work, each blog is treated as a single and static web page. In [Salvetti and Nicolov] a URL tokenization approach is suggested, based on an observation that sometimes spammers embed short phrases to form descriptive splog URLs. Regardless of the low recall from their experimental results, their lightweight technique can be used as a preliminary splog filtering step without the need to fetch the blog content. Han et al. [2006] propose a collaborative spam filtering method to block link spams on blogs. This technique relies on manual

identification of splogs and a trust-based information sharing scheme. Manually created URL/IP blacklists or update ping servers elimination are also used to block spam [Surbl].

There are other forms of Web spam. Comment spam is an unsolicited message added to editable Web pages such as blogs, wikis, and guestbooks. Content-based techniques such as language model [Mishne et al. 2005] and vocabulary size distribution [Narisawa et al. 2006] have been employed in detecting comment spam. Technical solutions such as adding *nofollow* tag [2005] and *captcha* mechanism [Von Ahn et al. 2004] are used to decrease the effect of comment spam.

Blogs have unique features. Unlike Web spam, where the content is usually static, a splog needs to have fresh content in order to continuously be relevant to blog search engines.[2] Additionally, the content is observed to have been generated by an automated framework. Therefore, extracting and using blog temporal dynamics is critical to detecting splogs. Relying only on content features is not sufficient because spammers typically copy content from normal blogs to appear legitimate. Trust propagation will work poorly due to the editable nature of the blog—a spammer can easily create links in the editable area of legitimate blogs to point to the splog. In addition, different crawling strategies and ranking criteria used in blog search engines raise new challenges to the link-based solutions. For example, a blog search engine collecting only a portion of blogs might be difficult to apply link-based solution that requires Web graph information. Finally due to the blog temporal dynamics, we cannot rely on snapshots alone—the content and link creation mechanisms used by blog spammers are different from Web spam. The changing behavior of splogs is more evasive than that of Web spam and cannot be easily captured by a set of snapshots.

Our splog detection approach investigates the correlation among blog post sequence and is a generalization of frameworks found in other domains. The recurrence plot was introduced by Eckmann et al. [1987] to visualize how a dynamic system comes back to a state similar to a former state after some time. In multimedia, there has been prior work on temporal analysis of audio and/or visual signals. Foote et al. [2002] propose self-similarities as a way to visualize musical structures. Multimedia objects are usually uniformly sampled time-series data. In this work we generalize this approach to arbitrary *nonnumeric* time-series data (blog posts) as opposed to analyzing continuous data.

1.2 Our Approach

We have developed a new technique for detecting splogs, based on the observation that a blog is a dynamic, growing sequence of entries (or posts) rather than

[2]Most search engine companies do not officially announce their ranking criteria. Our observation about the recency/frequency factor comes from the online discussion and our interaction with people from the search engine companies. Some online document that supports our argument can be found in http://blog.blogdimension.com/2007/08/01/how-does-the-ranking-work-on-blogdimensioncom/en/, http://prplanet.typepad.com/ceobloggers/2005/10/which_are_the_b.html, etc. A recent patent application, US20070061297 by Google could also support our hypothesis about spammers' strategy on manipulating the posting recency/frequency.

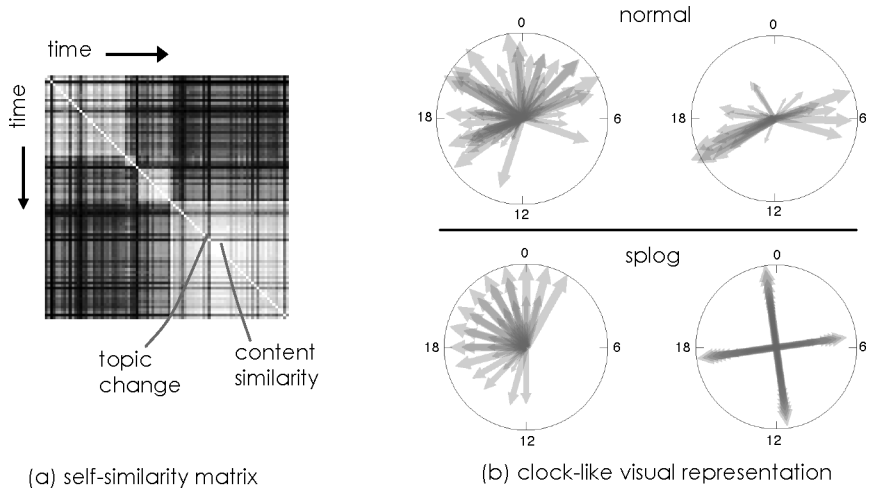


Fig. 1. In our approach, we represent the blog temporal dynamics using self-similarity matrices. We present a visualization framework to demonstrate distinct splog temporal dynamics compared to normal blogs. The distinct temporal characteristics then are captured by temporal features and used in splog detection. (a) Self-similarity matrix on the content attribute. (b) A clock-like visualization representing post content with respect to post time allows distinguishing the temporal dynamics of different types of blogs.

a collection of individual pages. In our approach, splogs are recognized by their temporal characteristics and content. There are three key ideas in our splog detection framework.

(1) We represent the blog temporal dynamics using self-similarity matrices defined on the histogram intersection similarity measure [Swain and Ballard 1991] of the time, content, and link attributes of posts. The self-similarity matrices function as a generalized spectral analysis tool. It allows investigation of the temporal changes within the post sequence. Figure 1(a) shows an example of a self-similarity matrix on the content attribute of posts, where the intensity of each entry (i,j) represents the content similarity between two posts p_i and p_j.
(2) We study the blog temporal characteristics based on a visual transformation derived from the self-similarity measures. We show that the blog temporal characteristics reveal correlation between attributes, depending on type of the blog (normal blogs and splogs). See Figure 1(b) for examples of our clock-like visual representation that distinguish normal blogs from splogs.
(3) We propose two types of novel temporal features to capture the splog temporal characteristics: (1) Regularity features are computed along the off-diagonals and from the coherent blocks of the self-similarity matrices on a single attribute. (2) Joint features are computed from self-similarity matrices across different attributes.

We conduct extensive experiments on real-world blog dataset (TREC-Blog). We randomly select 800 splogs and 800 normal blogs from 9167 labeled blogs to

create two evaluation sets—a balanced set and 1:9 imbalanced set. Our splog detector combined temporal features (regularity and joint features) with content based features into a large feature vector. The content based features are extracted from blog homepages as well as different parts of the blog—URLs, post content, etc. The dimensionality of the feature vector is reduced by Fisher linear discriminant analysis. We use a standard SVM classifier to identify splogs and use well-known metrics: precision, recall, and F1, to compare the relative performance of using different features, with fivefold cross validation. Experimental results from the evaluation sets indicate a significant improvement of incorporating temporal features, especially when the overall feature dimensionality is low.

The experimental results suggest that the temporal features have discriminatory value in this problem. These temporal features can augment content-based state-of-the-art methods for identifying splogs from blogs. Visualization tools that highlight the temporal patterns of splogs can be useful. However, we acknowledge that our method depends crucially on temporal features that may be susceptible to evolving spamming techniques. Thus part of our ongoing research focuses on improving rubustness by combining structural properties, including temporal and link structures, with content-based solutions in a splog identification framework [Lin et al. 2007].

The rest of the article is organized as follows. In the next section, we provide a working definition of splogs. We examine the self-similar aspects in Section 3. In Section 4, we present a visualization framework to demonstrate distinct splog temporal dynamics compared to normal blogs. In Section 5 we propose temporal features computed based on self-similarity analysis. We present experimental results in Section 6. Finally we present our conclusion and discuss future work in Section 7.

2. WHAT ARE SPLOGS?

In this section we provide a working definition of a splog and discuss the typical splog characteristics. We also highlight the distinctions between splogs and generic Web spam and motivate the need of a new detection technique for combating splogs.

A working definition for a splog. We extend the general definition of Web spam given in Gyöngyi and Garcia-Molina [2005]. There, Web spam is defined as a Web page created for *any deliberate action that is meant to trigger an unjustifiably favorable relevance or importance*, *considering the page's true value*. This general definition has been adopted in assessing a Web spam collection [Castillo et al. 2006]. Equivalently, we define splog as a blog created for *any deliberate action that is meant to trigger an unjustifiably favorable relevance or importance*, *considering the blog's true value*. In acknowledging the evolutionary nature of spam tricks, we shall not provide a more restricted definition for a splog. We shall instead discuss typical splog characteristics. Note that a splog is evaluated at the blog level, not at individual pages (single homepage or permalink pages). Additionally, a blog that contains spam in the form of comment spam or trackback spam (unsolicited comments or trackbacks sent by

spammers to a blog) might be innocent—we evaluate a blog by examining only the parts edited by the blog author or authors.

Typical splog characteristics. We now list some typical characteristics of splogs as well as generic Web spam, followed by the properties unique to splogs.

(1) *Machine-Generated Content*. Splog posts are generated algorithmically. Some might contain nonsense, gibberish, and repetitive text while others might copy or weave with text from other blogs or websites.
(2) *No Value Addition*. Splogs provide useless or no unique information to their readers. Note that there are blogs that use automatic content aggregation techniques to provide useful service, for example, daily tips, product reviews, etc.—although the content is gathered from other blogs or news sources, we consider these blogs as legitimate blogs because of their value addition.
(3) *Hidden Agenda, Usually an Economic Goal*. Splogs have commercial intent—they display affiliate ads or outgoing links to affiliate sites.

Uniqueness of splogs. Splogs are different from Web spam in the following aspects.

(1) *Highly Dnamic Content*. blog readers are mostly interested in recent posts. Unlike Web spam where the content changes relatively slowly, a splog has to continuously generate fresh content in order to attract traffic. In the blogosphere, rather than simply stuffing keywords, stitching phrases or duplicating content drawn from a limited corpus, spammers might constantly "steal" fresh content from legitimate blogs via delivery formats like RSS and Atom, so that their blog sites seem to update regularly as normal blogs. This is a mechanism for the splog to appear legitimate to search engines.
(2) *Nonendorsement Link*. Web pages are created to provide information; hence in Web pages, an incoming hyperlink is often interpreted as an endorsement by other pages. However, due to the conversational nature of blog, most blogs have editable areas welcome to contribute by their readers. This opens a hole for spammers to create hyperlinks (comment links or trackbacks) in normal blogs. Such links in blogs cannot be simply treated as endorsements.
(3) *Blog Search Engine Optimization*. Due to blog readers' interest in recent posts and the posts by authorities or celebrities, most blog search engines fetch data based on a predetermined lists or use ping services and emphasize more the content relevancy and *recency*. Web search engines on the other hand rely on crawlers to collect Web data and heavily exploit link analysis to determine the importance of Web pages. As a consequence, spammers in the blogosphere have tailored the form of link and term spam and have developed new spamming tricks in order to meet the blog ranking criteria.

Because of the property of nonendorsement link, link analysis based on trust is not directly applicable to splog detections. Given that most blog search engines have only restricted linking information and use different ranking criteria, it raises new challenges to the link-based solutions. And because the splog

content is algorithmically generated and highly dynamic, traditional detection techniques that focus only on content are not sufficient. Our proposed technique detects splogs by characterizing the splog content temporal dynamics, based on the following observation.

Temporal and link structures in splogs. Splogs tend to have repetitive patterns in the post sequence, due to algorithmically generated content. Figure 2 shows three example splogs that have identical posting times (during the day), post content, or links to affiliated websites appearing in their posts.

In comparison, normal (human) bloggers show variation in blog content and post time, and have a diverse set of outgoing links. We shall develop an approach that captures the different structural properties between normal blogs and splogs. In the following sections we show how self-similarity analysis of a blog is useful for distinguishing splogs from normal blogs. We shall use content features in addition to the features focused on splog temporal dynamics to detect splogs.

3. SELF-SIMILARITY ANALYSIS

We propose our approach for examining the temporal dynamics of blogs based on self-similarity analysis. In this section we start from constructing self-similarity matrices. Then in Sections 4 and 5 we shall introduce self-similarity based visualization and temporal features useful for capturing the temporal characteristics of splogs.

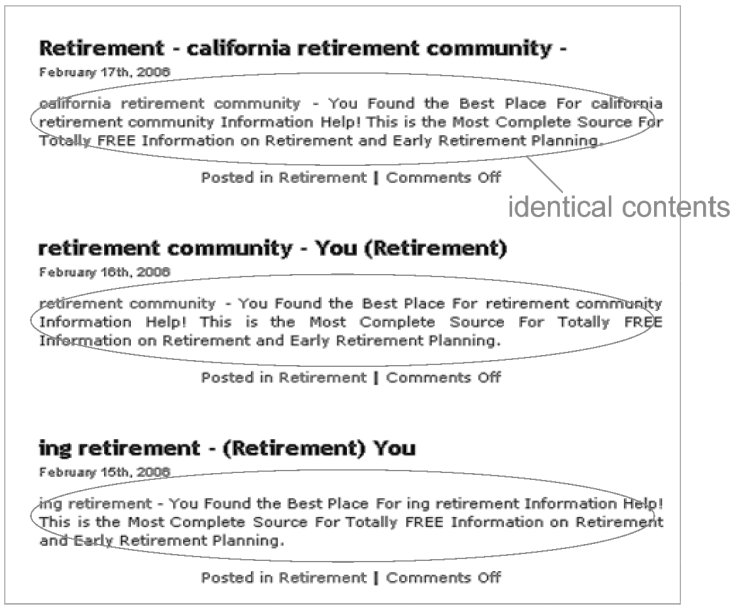
3.1 Constructing Self-Similarity Matrices

We analyze the temporal dynamics of a blog by examining the temporal self-similarity of its prime attributes, such as the post content, citation links, tags, etc. The intuition behind using self-similarity lies in its ability to reveal latent temporal structures.

We represent a blog as a sequence of media objects (e.g. blog posts) x_i for $i = 1 \ldots N$ in chronological order—that is, $t(x_i) \leq t(x_j)$ iff $i \leq j$, where $t(x_i)$ denotes the timestamp associated with x_i. Assume that a media object can be described using a set of attributes. Given an attribute α, we define an attribute-dependent similarity measure $s(i,j;\alpha)$. This measures the similarity between any pair of objects x_i and x_j, using attribute α. We further create an attribute-dependent topological matrix S_α using the topology induced by the similarity measure $s(i,j;\alpha)$ on the media objects, where the elements of the matrix $S_\alpha(i,j)$ are defined as follows: $S_\alpha(i,j) = s(i,j;\alpha)$, for $i,j \in \{1,\ldots,N\}$. Since the media objects are ordered chronologically, the topological matrix reveals the temporal self-similarity of the sequence x_i with respect to attribute α. In the following discussion, we shall refer S_α as a self-similarity matrix on a particular attribute α. Figure 3 shows an example topological graph.

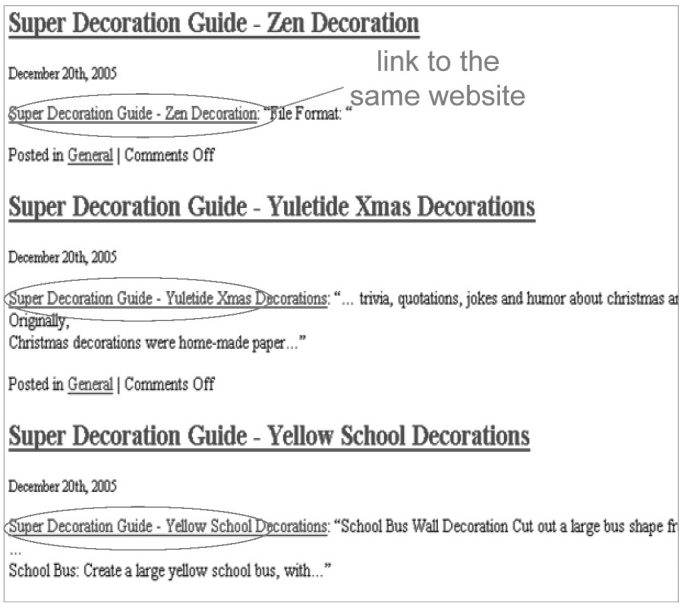
The self-similarity matrix S_α functions as a generalized autocorrelation over any time series of media objects—if we take the average over all the values along each diagonal in the upper triangular matrix, this would be equivalent to computing the autocorrelation of the nonnumeric sequence. Note that the nodes in Figure 3 refer to *posts from the same blog* and the edges refer to the similarity between posts on a specific attribute. We now examine the self-similarity

(a) The splog constantly generates posts at the same minute past the hour–11:30 am, 5:30 pm, 8:30 pm, etc.



(b) The splog constantly generates posts with almost identical content–advertising a retirement planning.

Fig. 2. Examples of repetitive patterns in posting times, post contents, and affiliated links in splogs (*Continues*).



(c) The splog generates posts with the same outgoing links embedded.

Fig. 2. (*Continued*).

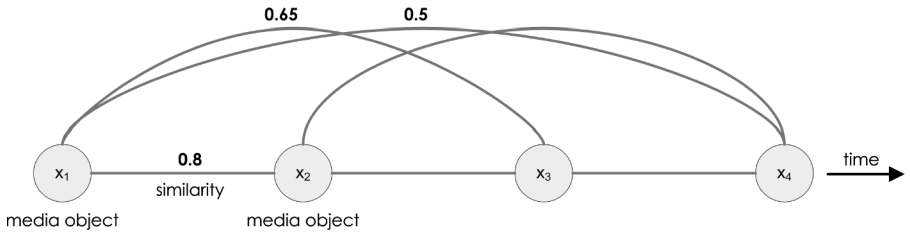


Fig. 3. A topological graph induced on time sequence of media objects (e.g. blog posts), due to a specific similarity measure $s(i, j; \alpha)$ on the attribute α.

matrices computed on the temporal sequences of blog posts. We focus on three prime attributes: post time, content and links.

3.1.1 *Post Time.* A blog post usually contains a time stamp indicating when the post is created. We first examine the post timing using self-similarity on the post time attribute. We use two different time scales—(a) at the micro-time scale (in daily time) and (b) at the macro-time scale (in absolute time). The similarity measures are as follows:

$$\begin{aligned} S_{micro}(i, j) &= |t_i - t_j| \bmod \tau_{day}, \\ S_{macro}(i, j) &= |t_i - t_j|, \end{aligned} \tag{1}$$

where t_i and t_j are the post time of post p_i and p_j, respectively, and τ_{day} is time of a day (e.g., if the post time is expressed in seconds, $\tau_{day} = 24 \times 60 \times 60 = 86400$). The micro time similarity reveals the relationships between posts that may be

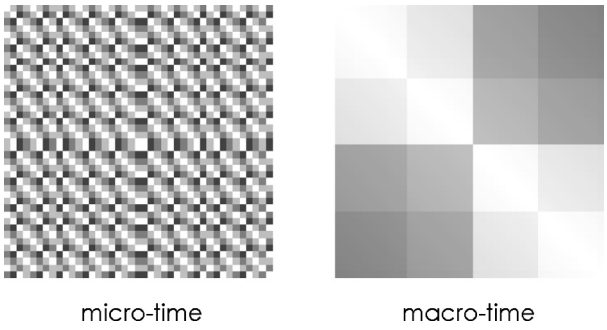


Fig. 4. The plots show the micro-(in sec. modulo days) and macro-time structure in the posts times of the blogs. The brightness value is proportional to similarity in both images—the highest similarity is scaled to have the largest brightness value.

days apart, but were posted at a similar time. It indicates regularity in posting time—perhaps some bloggers only post at specific times during the day. The macro time analysis is able to reveal post discontinuity at large time scales, which might due to vacations, etc. Figure 4 presents an example that demonstrates micro and macro time structures in blog post times. Note that along any row i, starting from the main diagonal, we have the similarity between post p_i and future posts. The white lines along the off-diagonals in the micro-time matrix suggest that the post creation time is similar in the micro-time scale at different posts, and the white blocks in the macro-time matrix suggest the creation time of successive posts is close in absolute time.

3.1.2 *Post Content.* We next examine post content using the self-similarity matrix. The similarity measure on post content is defined using histogram intersection similarity measure [Swain and Ballard 1991] on the tf-idf vectors. The histogram intersection similarity measure considers only nonzero corresponding elements (i.e., those "relevant" elements) in each pair of feature vectors and thus is less sensitive to the vector length. This property is appropriate in our case because most blog posts are short articles, especially for splog posts.

Let h_i and h_j be the tf-idf vectors (after stemming and stop-word removal) for posts p_i and p_j. Then the similarity between two posts p_i and p_j is defined as:

$$S_c(i, j) = \frac{\sum_{k=1}^{M} \min(h_i(k), h_j(k))}{\sum_{k=1}^{M} \max(h_i(k), h_j(k))}, \tag{2}$$

where c refers to the similarity based on the content attribute and M is the size of the vector. Note that the posts are ordered in time. The corresponding element in self-similarity matrix is then the content similarity between the two posts.

Figure 5(a) shows an example plot of the content-based temporal self-similarity matrix. It reveals that there is significant temporal correlation between posts. It also suggests that the users mostly tend to stay on a topic (large

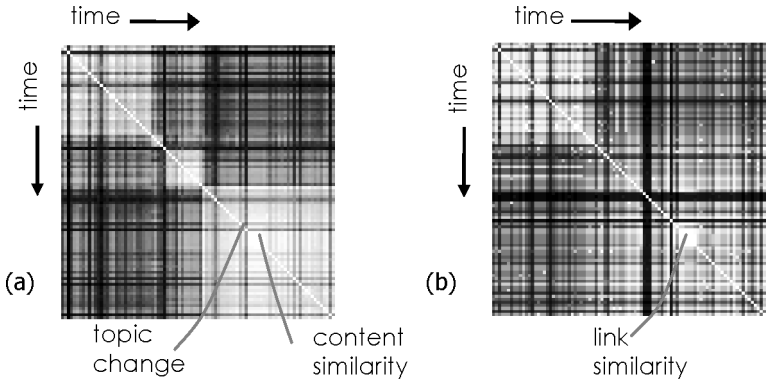


Fig. 5. (a) Self-similarity matrix on the content attribute. (b) Self-similarity matrix on the link attribute.

white blocks in the figure), and may occasionally post on different topics (causing black horizontal lines) before returning to the topics.

3.1.3 *Post Links.* The similarity measure on the links is defined in a similar manner to Equation (2) except that the tf-idf vectors are now calculated on the target links (collapsed by the host domain), rather than on the words. Hence, the similarity between two posts p_i and p_j is defined as:

$$S_l(i, j) = \frac{\sum_{k=1}^{M} \min\left(h_i(k), h_j(k)\right)}{\sum_{k=1}^{M} \max\left(h_i(k), h_j(k)\right)} \tag{3}$$

where l refers to the similarity on the link attribute, h_i and h_j are the link based tf-idf vectors and M is the size of the vector.

In Figure 5(b), we can see the link based self-similarity matrix. It reveals a similar block-based structure and changes as in Figure 5(a), and we can see that changes to the content and link patterns are usually coincident.

The self-similarity matrices exhibit several advantages: First, the media objects are represented by their relationship with the other objects. Hence the size of the matrix used to represent a dataset of N objects is always $N \times N$, regardless of the content complexity of the objects. Second, because the self-similarity matrices are constructed according to the time order of objects, they allow investigation of the temporal relationship among objects. We next discuss a visualization that allows us to observe the blog temporal dynamics in greater detail.

4. EXAMINING BLOG TEMPORAL DYNAMICS

In this section we present a visualization framework to demonstrate distinct splog temporal dynamics compared to normal blogs. We use self-similarity analysis in the visualization to distinguish amongst different blog types.

In order to examine the temporal dynamics of blogs, we represent the similarity relationship in the blog temporal sequence using a "clock" metaphor. The

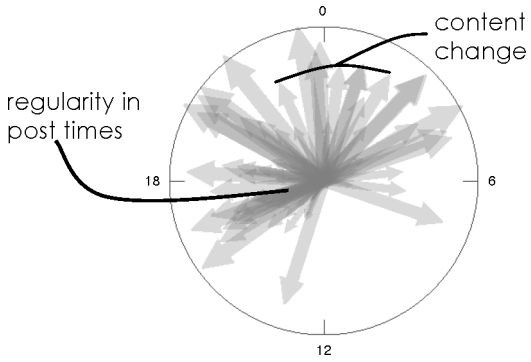


Fig. 6. This plot shows the change of post content over its daily post time.

idea is to show how the non-time attributes like content or links change with respect to the time attribute. Here we show how the attribute values of content and link change over the micro-time attribute (see Section 3.1.1). This approach can be applied to macro-time attribute to examine the long-term change of other attributes. As shown in Figure 6, we render each post as an arrow stemmed from the center of a circle. The length of an arrow indicates how the post is similar to its previous post in terms of the corresponding non-time attribute value, and the orientation of an arrow indicates when the post is created in a day.

We use a log-linear transformation to determine the length of arrows. Let $\rho_{i,\alpha}$ be the length of the ith post-arrow corresponding to the attribute α (e.g., content or link attribute), we compute $\rho_{i\alpha}$ for $i \geq 2$, as:

$$\rho_{i,\alpha} = \rho_{i-1,\alpha} + 1 - \log_2 \left(1 + S_\alpha \left(i, i-1\right)\right), \tag{4}$$

Note that for $i = 1$, $\rho_{i,\alpha} \equiv 1$ since p_0 is not defined. In Equation (4), the length of arrow ($\rho_{i,\alpha}$) corresponding to the ith post is calculated based on the length of previous post-arrow ($\rho_{i-1,\alpha}$), increased by their difference. The lengths of post-arrows grow if the two consecutive posts differ in terms of the similarity measure $S_\alpha(i, i–1)$, and remain unchanged if the two posts are identical. Note that the post-arrows can grow arbitrary as the number of posts increase, and thus we select to show up to r recent posts and scale the latest post-arrow on each clock-like plot to the same length.

These transforms reveal the blog temporal dynamics by showing the rate of change of specific attribute with respect to time. In Figure 6, the increase in arrow density suggests regularity in the post time. Spiral like growth in the length of the arrow suggests a slow change in the content.

We can see distinct temporal characteristics amongst normal blogs and splogs. A comparison of different blog types, including two normal blogs, personal and power blog, and one splog, is shown in Figure 7. Personal blogs are used as an online personal journal, to share with others the daily experience and also allow others to contribute, such as blogs hosted at LiveJournal. Power blogs are those that focus on a specific niche. They are often used to constantly update readers on the status of an ongoing project, to provide a review on a

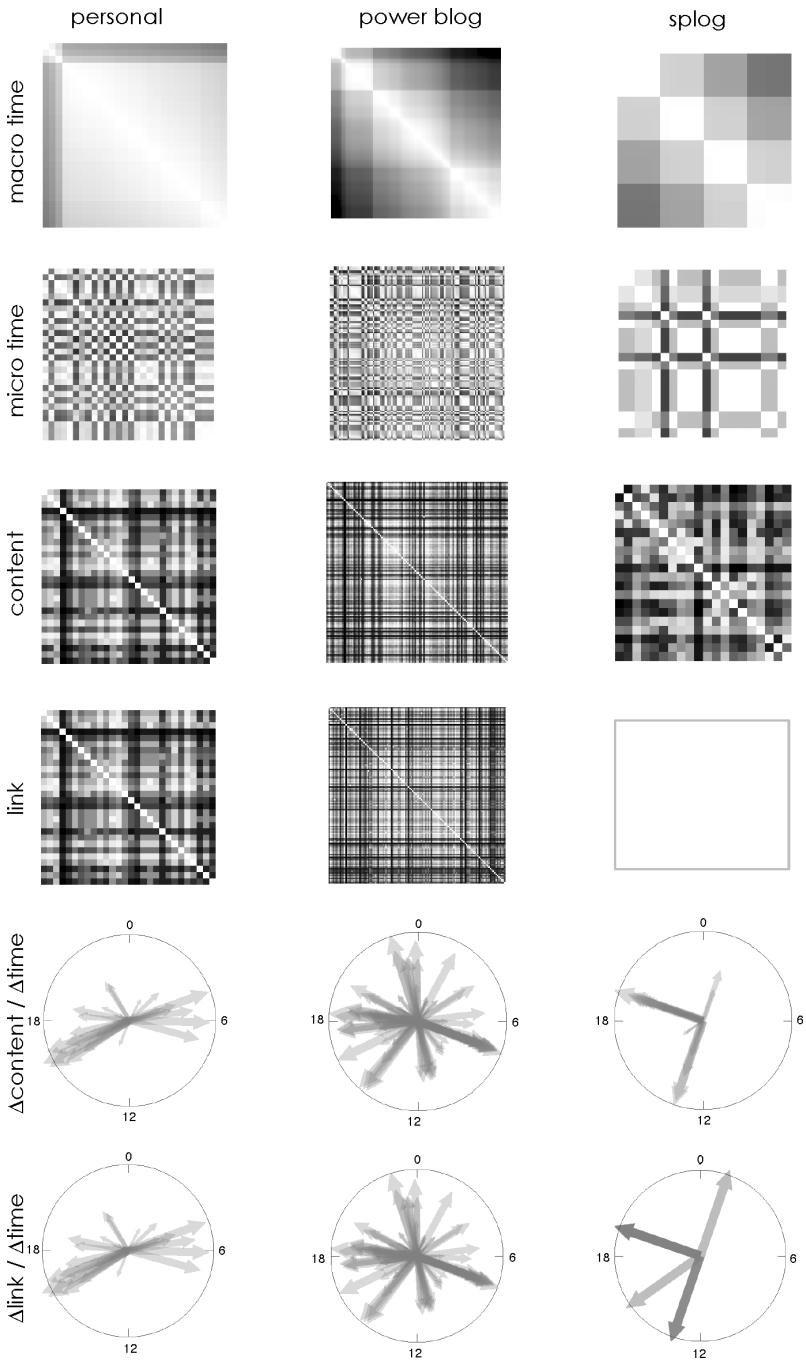


Fig. 7. The figure shows the self-similarity of the content, link and post-times, as well as self-similar clocks on content and link with respect to post time. Three blogs—a normal personal journal, a power blogger, and a splog—are shown as examples to demonstrate the proposed visualization framework allows distinguishing the temporal dynamics of different types of blogs.

new service or product, or to promote businesses of a company. The content of power blogs is contributed by entrepreneurs, domain experts, or amateurs, or comes from other blogs or new sources.

We use the three instances to illustrate interesting different temporal characteristics and the following observations have been empirically observed repeatedly across other samples. Normal blogs and splogs differ in their temporal characteristics in all three facets—post time, content, and link.

Post Time. Normal bloggers seem to prefer a regular posting time (e.g., morning/night) and the times show a spread, but are largely in a few time periods—they do not post at all times of the day. A splog will show machine-like regularity—this can be either posting at fixed stable times, or a uniform distribution of times (throughout the day). A normal personal blogger as well as a power blog exhibit macro time breaks (e.g., due to vacation), this characteristic is absent in a splog. A power blogger can have *both* highly diverse posting time and machinelike regularity as in Figure 7—in this particular case, the blogger, actually has a script that regularly aggregates content from the Web/other blogs and posts them on the blog at regular times.

Post Content. A normal blogger typically stays on topic for a while, before showing a topic drift. A splog often copies content (to escape detection, as well as to appear as a search result related to a query) from different parts of the Web/blogosphere and hence may show very high topic diversity. Some splogs on the other hand exhibit significant topic stability. We observe that a power blogger has a higher rate of change of content with time than a normal personal blogger. Interestingly, for both normal personal and power bloggers, changes to content appear to coincide with macro time breaks.

Post Links. For normal (personal and power) bloggers, changes to content affect the change to the linked sites in the same manner—this is to be expected as the links essentially function as supporting arguments in a blog post. It turns out, after log-linear transformation as described in Equation (4), the self-similar clocks on content and link are almost indistinguishable in these two normal blog cases, due to the highly correlated rate of changes on content and link. However, a splog is driven by a strong commercial interest to drive traffic to affiliate sites. In the example in Figure 7, the splog always points to the same site (exhibiting strong link regularity), thus the temporal self-similarity in the figure does not change. Other splogs show a similar characteristic—that is, they only show limited link temporal diversity.

We show more examples of "splog clock" in Figure 8. It can be seen that these splogs are highly regular in terms of post time, as the post arrows overlap only at certain time or evenly spread out. Also their post content or links change slowly, as the length of the post arrows slightly increase or even remain constant all the time.

5. FEATURES TO EXTRACT BLOG TEMPORAL DYNAMICS

We now discuss our approach using the self-similarity analysis to derive features useful for splog detection. We propose using two novel temporal features: the regularity features (Section 5.1) and the joint features (Section 5.2).

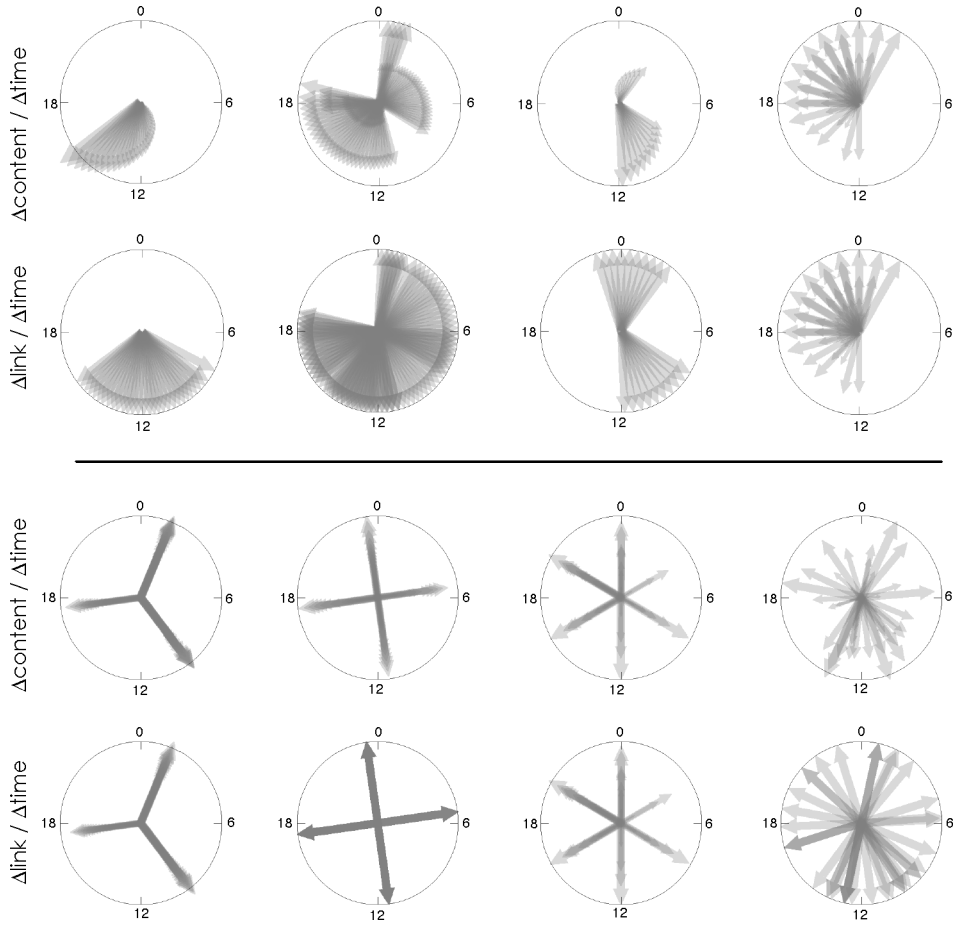


Fig. 8. Examples of "splog clock." These splogs are highly regular in terms of post time, as the post arrows overlap only at certain time or evenly spread out. Also their post content or links change slowly, as the length of the post arrows slightly increase or even remain constant all the time.

The proposed temporal features characterize blog temporal dynamics, based on the self-similarity matrix representation introduced in Section 3.1. We shall discuss the temporal features derived from the following self-similarity matrices: (1) S_{macro}: macro-scale time, (2) S_{micro}: micro-scale time, (3) S_c: content and (4) S_l: link self-similarity matrix.

5.1 Regularity Features

In this section we first provide an insight of capturing the regular patterns exhibited on certain attribute, for example, repetitive post time, links, etc. From the self-similarity matrices, we observe two types of patterns (see Figure 4) and the illustration in Figure 9): (1) high intensity off-diagonal lines appear when the attribute values are similar at different posts, and (2) high intensity blocks appear when the attribute values remain highly constant for some

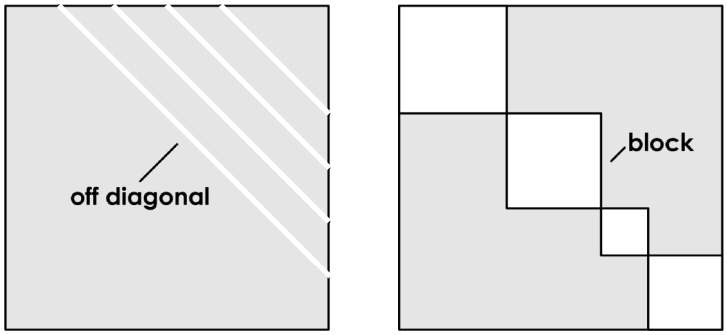


Fig. 9. The figure are computed using the off diagonals and blocks within the self-similarity matrix.

period. Both patterns reveal complementary temporal regularity characteristics. In next two sections, we present our methods to extract features to characterize each pattern.

5.1.1 *Features Along the Off-Diagonals.* First we extract features to capture those off-diagonal patterns. We use three measures—mean, standard deviation and entropy—to quantify the regularity patterns along the off-diagonals. The expectation along the diagonals of the topological matrix is equivalent to the generalized autocorrelation of the time series data under the specific similarity measure. Specifically the expectation along the kth off-diagonal, is a measure of average similarity of a post to another post, with $k-1$ posts in between.

Intuitively, the autocorrelation function of a numeric time series data is an estimate of how a future sample is dependent on a current sample. A noise-like signal will have a sharp autocorrelation function, while a highly coherent signal will fall off gradually. Here, we use the expectation of the off-diagonal values as a generalized autocorrelation on nonnumeric blog post data. This captures how the attribute value of post sequence changes, and we use the standard deviation to describe how the data deviates from the expectation. Additionally we compute the entropy of the off-diagonal elements to measure the amount of disorder among these elements.

We compute the statistical measures along the off-diagonals as features. Given a self-similarity matrix $M_\alpha \in \{S_{macro}, S_{micro}, S_c, S_l\}$, we compute the mean ($\mu_k$), standard deviation ($\sigma_k$) and entropy ($H_k$) along the kth off-diagonal, $0 < k \leq k_o$ for certain $k_o < N$, where N is the size of M_α (i.e. number of data points). This is formally computed as follows:

$$\begin{aligned} \mu_k(M_\alpha) &= E[z_k], \\ \sigma_k(M_\alpha) &= \sqrt{\text{var}[z_k]}, \\ H_k(M_\alpha) &= -\sum_{i=1}^{D} p_i \log_D p_i, \end{aligned} \tag{5}$$

where $z_k = \text{diag}(M_\alpha, k)$ is the kth off-diagonal of matrix M_α, and the probabilities $p_i = d_i/D$ are computed after quantizing z_k into D bins, and d_i is the

number of elements of z_k fall in the ith bin. We typically use $k_o = 4$ diagonals to make sure that all samples that have more than k posts have accurately estimated variables μ_k, σ_k and H_k.

5.1.2 *Features from Coherent Blocks.* The block-wise features measure the similarity of a post to other posts within a coherent group of data continuous in time. As blog posts usually exhibit short-term temporal coherence, coherent groups can be easily observed as white blocks along the main diagonal on the self-similarity matrices. To extract such blocks, we only need to segment the main diagonal such that each segment associates with a block. We discuss our method for extracting the coherent blocks in the following.

We use an agglomerative hierarchical clustering method, with single link merge criteria on the pairwise similarity values embedded in the self-similarity matrix. The original dataset is initialized into N clusters for N data points. Two clusters are merged into one if the distance (linkage) between the two is the smallest amongst all pair wise cluster distances. We use a clustering quality criterion: the "modularity function" Q, to select the number of clusters [Newman and Girvan 2004]. We found this simple measure works well in our case. Other heuristics to determine number of clusters such as minimum description length (MDL) can also be applied here. Once the clusters are determined, we further split a cluster if its data points (on the main diagonal) are not connected.

We compute the statistical measures from coherent blocks as features. Let $B_k = \{\mathrm{M_{u,v}}\}_{i \le u,v \le i+n-1}$ be a block that contains $n \times n$ connected elements (which are induced from n data points) on the matrix. Similar to the diagonalwise features, we now compute block-wise features—mean ($\mu_{b,k}$), standard deviation ($\sigma_{b,k}$), and entropy ($H_{b,k}$) for the kth block, as follows:

$$\begin{aligned} \mu_{b,k}(M_\alpha) &= E[B_k], \\ \sigma_{b,k}(M_\alpha) &= \sqrt{\mathrm{var}[B_k]}, \\ H_{b,k}(M_\alpha) &= -\sum_{i=1}^{D} p_i \log_D p_i, \end{aligned} \tag{6}$$

where p_i is the probability of values in the block that are quantized to D bins. Since the number of blocks on a matrix can be different, we simply take an average over all the blocks. That is, we use the overall mean $\mu_b = E[\mu_{b,k}]$, standard deviation $\sigma_b = E[\sigma_{b,k}]$ and entropy $H_b = E[H_{b,k}]$ as blockwise features of a self-similarity martix. We expect that blogs with highly short-term coherency are likely to have high $\mu_b(M_\alpha)$, low $\sigma_b(M_\alpha)$ or low $H_b(M_\alpha)$ for some attribute α.

5.2 Joint Features

We now discuss the joint features derived from a pair of self-similarity matrices on different attributes. From the matrices shown in previous section (see, e.g., Figure 5), we observe that changes in different attributes are usually coincident. This effect is observed to be much stronger in splogs. For example, a splog might have a links to a "sports gear" Web site whenever posting about "river sport," and links to a "sailboat" Web site whenever posting about "sail center."

We conjecture that because the splogs are financially motivated, we expect correlations between attributes.

We compute the joint features to measure attribute correlation using joint entropy. The joint entropy measures how the distributions of two variables are related. It is computed as follows:

$$H(X, Y) = -\sum_{i=1}^{D_x} \sum_{j=1}^{D_y} p(x_i, y_j) \log p(x_i, y_j), \tag{7}$$

Let $H_k(M_\alpha, M_\beta)$ be the joint entropy from the same kth off-diagonals of a pair of matrices M_α and M_β, where α and β are two different attributes. Let $H_b(M_\alpha, M_\beta)$ be the joint entropy over the blocks of M_α and M_β.

$H_k(M_\alpha, M_\beta)$ is computed from the joint probability $p_k^{(\mathrm{d})}(x_i, y_i)$, where $p_k^{(\mathrm{d})}(x_i, y_i)$ indicates, after quantizing the kth off-diagonal of M_α and the kth off-diagonal of M_β into D_x and D_y bins respectively, the probability of an element on the kth off-diagonal being contained in the bin x_i of M_α and in the bin y_i of M_β.

The joint entropy $H_b(M_\alpha, M_\beta)$ is computed from the joint probability $p^{(b)}(x_i, y_i)$, where M_α and M_β are segmented into D_x and D_y blocks respectively, and $p^{(\mathrm{b})}(x_i, y_i)$ indicates the probability of an element on the matrix being contained in the block x_i of M_α and in the block y_i of M_β. This analysis captures the block-structural similarity across the two attributes.

In this section, we proposed specific features computed on self-similarity matrices to reveal blog temporal dynamics. We next show how these features impact splog detection.

6. EXPERIMENTS

We now present experimental results on the splog detection. We first provide detailed discussion on the evaluation dataset in Section 6.1. In Section 6.2 we discuss the content-based features used in this work. These content features serve as the baseline features as they are derived from traditional content analysis and prior work. We demonstrate the utility of the proposed temporal features by showing their discriminability in Section 6.3. In Section 6.4, we show the usefulness of proposed features by comparing their performance in splog detection against the content-based features.

6.1 Dataset Description

In this work, we use the TREC (the Text Retrieval Conference) Blog Track 2006 collection for analysis. This dataset is a crawl of 100,649 feeds collected over 11 weeks, from Dec. 6, 2005, to Feb. 21, 2006, totaling 77 days. According to the Macdonald and Ounis [2006], the blogs included in the collection were predetermined before fetching their content, and no new blogs were added to the collection after the first day of the crawl. In order to create a realistic scenario, up to 17,969 known splogs were originally inserted into the collection, which is corresponding to 17.8% of the feeds.[3]

[3]We were not informed of the list of inserted splogs.

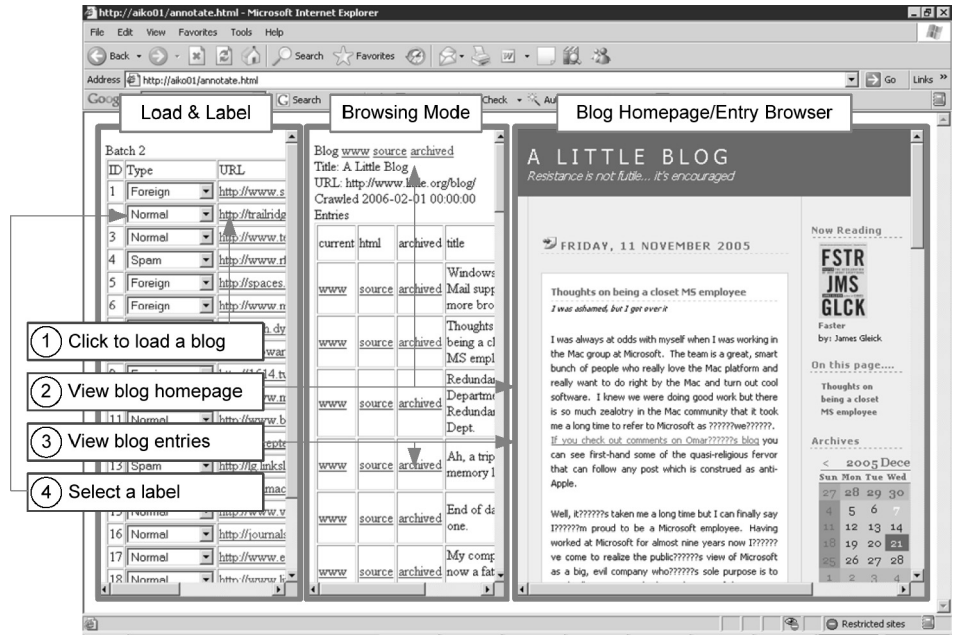


Fig. 10. Splog Annotation Tool for viewing and labeling blogs.

The collection contains blogs that are difficult to analyze. There are blogs whose contents are missing from the collection. That is, either their homepage or permalinks are absent. This makes it difficult to judge if a blog is spam (i.e., a splog). Hence we decided not to assess blogs with missing content. We exclude non-English blogs for the same reason, as the annotators might not understand non-English content. After removing duplicate feeds, feeds without homepage or permalinks, and non-English blogs (by using TextCat language guesser [Cavnar and Trenkle 1994]), we end up with a subset of 22,300 unique blogs. We shall examine this subset in this paper.

6.1.1 *Annotation.* We have developed an annotation tool (see Figure 10) for annotators[4] to label the TREC-Blog dataset. By using this annotation tool, the annotators can either browse the blog homepage and posts that have been downloaded in the TREC-Blog dataset, view the HTML page source, or visit the blog site directly online.[5] For each blog, the annotators examine its content, out-going links, appearance of affiliate ads, etc. and assign one of the five labels listed in Table I to the blog. These labels are defined similar to the assessment task initiated in the Web spam detection community.[6] Annotators recognize spam by recognizing commonly used spam tricks such as keyword stuffing, content weaving or duplication, hidden text, link farm, etc. However,

[4]These annotators are researchers and graduate students working at NEC Labs America, Cupertino, CA.

[5]At the time of annotation, some blogs, especially splogs, were not present online and can only be browsed from the dataset.

[6]http://www.yr-bcn.es/webspam/datasets/uk2006-info/.

Table I. Annotation Labels
For each blog being annotated, the annotators examine its content, out-going links, appearance of affiliate ads, etc. and assign one of the five labels.

Label	Definition
(N) *Normal*	Blogs that do not use spam tricks.
(S) *Splog*	Blogs that use spam tricks.
(B) *Borderline*	Blogs that are heavily optimized for (blog) search engines or for selling advertising, but that also provide some original content or useful services.
(U) *Undecided*	Blogs where content is not accessible or requires a password, or the annotator cannot decide which label to assign to.
(F) *Foreign*	Blogs that are writing in language other than English.

Table II. A Pilot Study for Understanding Challenges in Annotating Blogs
A group of seven annotators are asked to annotate the same set of 60 blogs. Each blog is annotated independently by the seven annotators, with one of the five labels: (N) Normal, (S) Splog, (B) Borderline, (U) Undecided and (F) Foreign. Each row shows the distribution of labels from the respective annotator's judgment.

Annotator	N	S	B	U	F
X_1	45	3	4	1	7
X_2	37	1	0	16	6
X_3	36	4	3	7	10
X_4	47	4	1	0	8
X_5	44	8	0	1	7
X_6	33	6	11	5	5
X_7	48	4	1	0	7

we want to point out that in the splog context, more sophisticated spam tricks are observed. For example, as discussed in Section 2, spammers might "steal" content from legitimate blogs through for example, RSS. Sometimes such plagiarism can be confused with content syndication services (often seen on power blogger pages). Such sophisticated spam tricks raise additional challenges for annotators.

To examine the challenges in the annotation task, we conduct a pilot study on a group of seven annotators. These seven annotators are asked to independently annotate the same set of 60 blogs and their results are compared. Table II shows the result comparison. From the result we observe a very interesting phenomenon: the annotators tend to agree on most of the normal blogs but they have varying opinions about assigning the *Splog/Borderline/Undecided* labels.[7] This suggests that splog detection is not trivial even for humans. Based on the observation obtained from this pilot study, we prepare the ground truth in our data analysis in the following way.

Ground Truth Definition. We have manually labeled 9167 blogs randomly sampled from the above mentioned set of 22,300 unique blogs. In order to disambiguate between splogs and non-splogs, and because most normal blogs are less ambiguous, we decided that those that are labeled as *Splog* need to be confirmed by a different human annotator. Thus, we ended up with 7380 normal

[7]Some blog posts are written in a mixture of English and foreign. The language judgment for the same blog might differ if two annotators evaluate different sets of posts. This problem, however, is minor because any blogs labeled as foreign will then be removed from the evaluation sets.

blogs and 897 splogs. We then randomly select 800 splogs and 800 normal blogs to create the evaluation set.

6.2 Baseline Content Features

In this section we discuss the content-based features used in this work. We create two types of content feature sets: (1) homepage content features (HPC): features suggested in [Kolari et al. 2006a; Kolari et al. 2006c; Kolari et al. 2006d], and (2) blog contextual content features (BCC): features derived from traditional content analysis and tailored in the blog context. We shall describe HPC and BCC feature sets in Section 6.2.1 and 6.2.2, respectively. Then in Section 6.2.3 we discuss a feature selection method used in our work.

6.2.1 *Homepage Content Features (HPC).* Kolari et al. [2006a, 2006c, 2006d] suggest using local features extracted from a single homepage of each blog for splog detection. The rationale of extracting local content-based features from a single homepage is that they consider blogs to be a different genre of web pages. Blogs often contain content that express personal opinions, so words like "I," "We," "my," "what" appear commonly on legitimate blogs.

We experiment with three prime types of local features mentioned in their work: (1) *bag-of-words*, (2) *bag-of-anchors*, and (3) *bag-of-urls*. In *bag-of-words*, the page content is represented as a feature vector whose elements are frequency-based weights, such as binary, term frequency (tf) or term frequency/inverse document frequency (tf-idf), of the words in the page. In *bag-of-anchors*, features are extracted from the anchor text of all URLs on a page. In *bag-of-urls*, URLs are tokenized and each token is used as a feature. Based on their evaluation, the three types of features are more effective than other features such as *bag-of-word-N-Grams* or link-based features. As suggested by the authors, we create these three feature sets without stemming and stop word removal. We also create feature sets that merge the three prime types of features.

6.2.2 *Blog Contextual Content Features (BCC).* The effectiveness of content analysis in detecting spam web pages has been studied in Fetterly et al. [2004, 2005] and Ntoulas et al. [2006]. In Ntoulas et al. [2006] different parts of a page, for example, URLs, anchor text, page content, etc. are analyzed separately, and the authors show that features extracted from different parts are complementary in detecting spam pages.

Blogs are special cases of Web pages—the content of a blog are not just a collection of Web pages, but are organized as a home page with a sequence of posts (permalinks). Thus we apply content analysis to the blog context as follows.

We extract features from five different parts of a blog: (1) tokenized URLs, (2) blog and post titles, (3) anchor text, (4) blog homepage content, and (5) post content. For each category we extract the following features: number of words (w_c), average word length (w_l) and a tf-idf vector representing the weighted word frequency distribution (w_f) in that part. These three types of features are constructed independently for each content category. w_c and w_l capture the

statistical properties of the category, while w_f is similar to the *bag-of-word* term frequency based feature vector discussed in Section 6.2.1.

6.2.3 *Feature Selection Using Fisher Linear Discriminant Analysis (LDA).* The original HPC and BCC feature sets are very large because the total number of unique terms (excluding words containing digits) is greater than 100,000 (this varies per category, and includes nontraditional usage such as "helloooo"). Some particular terms might appear only in one or few blogs. This can easily lead to overfitting the data. Secondly, the distribution of the words is long-tailed; that is, most of the words are rarely used.

To avoid data over-fitting, we start by removing words whose occurrences are less than a threshold. After that, we select feature words using a well-known feature selection method[8]—Fisher discriminant analysis (Fisher LDA) [Duda et al. 2001]. We give a brief overview of this method in the following.

We expect good features to be highly correlated with the class (in our case, normal blog vs. splog), but uncorrelated with each other. The goal of Fisher LDA is to find a linear transformation that performs feature dimensionality reduction while preserving as much of the class discrimination as possible. The so-called Fisher criterion minimizes the within-class scatter and maximizes the between-class scatter simultaneously. The optimum transformation can be found by maximizing the following trace criteria:

$$J = tr\left(S_w^{-1} S_b\right), \tag{8}$$

where S_w denotes the within-class scatter and S_b denotes the between-class scatter matrix. The scatter matrices are defined in multivariate feature space x, as follows:

$$\begin{aligned} S_b &= \sum_{i=1}^{c} (m_i - m)(m_i - m)^T, \\ S_w &= \sum_{i=1}^{c} S_i, \quad \text{and} \quad S_i = \sum_{x \in C_i} (x - m_i)(x - m_i), \end{aligned} \tag{9}$$

where m_i is the mean of each class i and m is the overall mean. S_i is the covariance matrix of class i and C_i is the set of elements belonging to class i. In our case, we have $c = 2$, that is, splog and normal blog.

This criterion computes the ratio of between-class variance to the within-class variance in terms of the trace of the product, where the trace is the sum of eigenvalues of $S_w^{-1} S_b$. It can also be used as a measure for feature selection—the classes will be more separable if we select features corresponding to the larger eigenvalues of $S_w^{-1} S_b$. Table III lists some examples selected from top 20 discriminant terms extracted from different parts of blog and determined by Fisher LDA. Note that these terms are in stemmed forms (by using the Porter stemming algorithm). It can be seen that many of these terms are informative

[8]We also investigated other feature selection methods such as mutual information as used in Kolari et al. [2006a] and find that Fisher LDA is more effective. This is because Fisher LDA considers the correlation between features with respect to classes.

Table III. Examples Selected from Top 20 Discriminant Terms Determined by Fisher LDA
The terms are in stemmed forms. It can be seen that many of these terms are informative and also indicative of commercial interests.

URLs	Title	anchor text	homepage	post
pain	*win*	*content*	*credit*	*adapt*
board	*quot*	*girl*	*creat*	*life*
tip	*washington*	*gratuitement*	*foundat*	*busi*
ball	*album*	*car*	*phone*	*decor*
antiag	*train*	*ads*	*internet*	*love*
thanksgiv	*movi*	*thumb*	*guid*	*info*
footbal	*food*	*unofficial*	*offer*	*valu*
abus	*watch*	*cigar*	*rate*	*googlesynd*
coin	*gift*	*security*	*law*	*wireless*
christm	*cash*	*lawyers*	*improv*	*need*

and also indicative of commercial interests. Those implying commercial interests, such as "pain," "antiag" (anti-aging), etc, appear more frequently in splog posts than normal blog posts.

Because the content of splogs is highly dynamic, that is, the spam terms might change quickly, a large content-based feature vector tends to lose its generalizability. It is preferable to select a small k instead of using a large dimensional term feature vector.

6.3 Temporal Feature Discriminability

In this section we discuss the impact of the proposed temporal features derived from the self-similarity analysis.

We examine the relationship between a feature value distribution and the probability of finding a splog with that feature value in the ground truth annotated data. In Figure 11 we show two histograms using bars and a probability curve. The two histograms represent the distribution of nonsplogs (solid bars) and splogs (open bars) respectively at each feature value. In addition, in the figure we plot a curve to show the probability of splogs (i.e., the number of splogs divided by the total number of blogs in the annotated ground truth) at each feature value. Note that the probability curves might fluctuate at certain values, for example, close to 0 or 1, since there are only few samples in these areas.

Spammers artificially generate splog content, and hence the temporal features of splogs have very different feature distributions, compared to the feature distributions of nonsplogs. For example, Figure 11(a) shows the feature values of $H_b(S_{macro}, S_{micro})$—the blockwise joint entropy between the macro and micro-scale time self-similarity matrices. The mass of the splog distribution is concentrated on the left to the nonsplog distribution (solid bars). It indicates splogs tend to have higher correlation between these two attributes (S_{macro} and S_{micro}) than non-splogs, and the splog distribution and probability curve both peak at lower joint entropy values. In some cases the splog probability curve doesn't follow a similar trend to the splog distribution. Figure 11(b) shows the feature values of $H_2(S_c)$—the 2nd off-diagonal entropy of the content similarity matrix. Splogs tend to have lower values for this feature, but compared with nonsplogs,

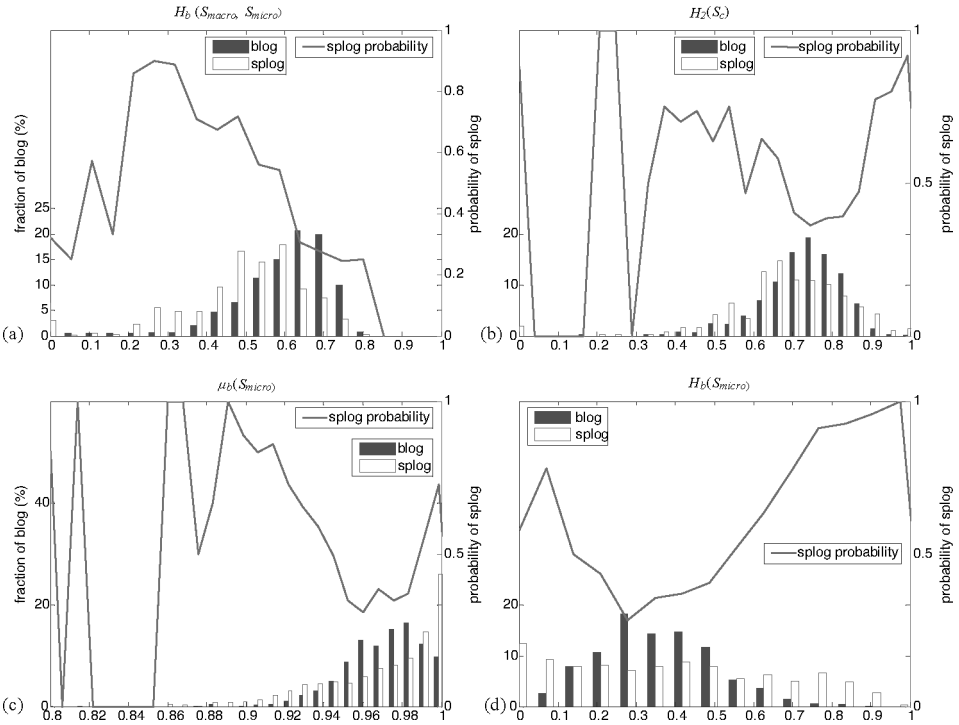


Fig. 11. Each figure shows two histograms as bars and a probability curve. The two histograms represent the fraction of nonsplogs (solid bars) and splags (open bars) respectively for each feature vale. The curve shows splog: nonsplog ratio for each feature value, computed from the annotated ground truth. It indicates the probability of a blog being a splog for each feature value. (a) $H_b(S_{macro}S_{micro})$—blockwise joint entropy between the macro and micro-time self-similarity matrices. (b) $H_2(S_2)$—the 2nd off-diagonal entropy of the content similarity matrix. (c) $\mu_b(S_{\text{micro}})$—block mean of the micro-time self-similarity matrix. (d) $H_b(S_{micro})$—block entropy of the micro-time self-similarity matrix.

the feature values of splogs have larger variance. This results in a bimodal splog probability curve. Figure 11(c) shows the feature value of $\mu_b(S_{micro})$—the blockwise mean of micro time self-similarity matrix. The splog distribution is extremely left-skewed, but due to its long left tails, the splog probability curve peaks mostly at the left of the figure. Figure 11(d) shows the feature values of $H_b(S_{micro})$—the block-wise entropy of micro time self-similarity. In this figure, splogs almost are evenly distributed, while the mass of the non-splog distribution concentrates at 0.3–0.4, which also results in a bimodal splog probability curve.

The splog probability curves for each feature are indicative of the utility of the feature in splog detection, as the splog to nonsplog ratio becomes either very high or very low for certain values of the features. In order to easily compare across different features, we only show the splog probability curve over the nonsplog distribution in Figure 12.

In Figure 12 we show examples of the utility of the temporal features. The diagonalwise features are shown on the left and the blockwise features are

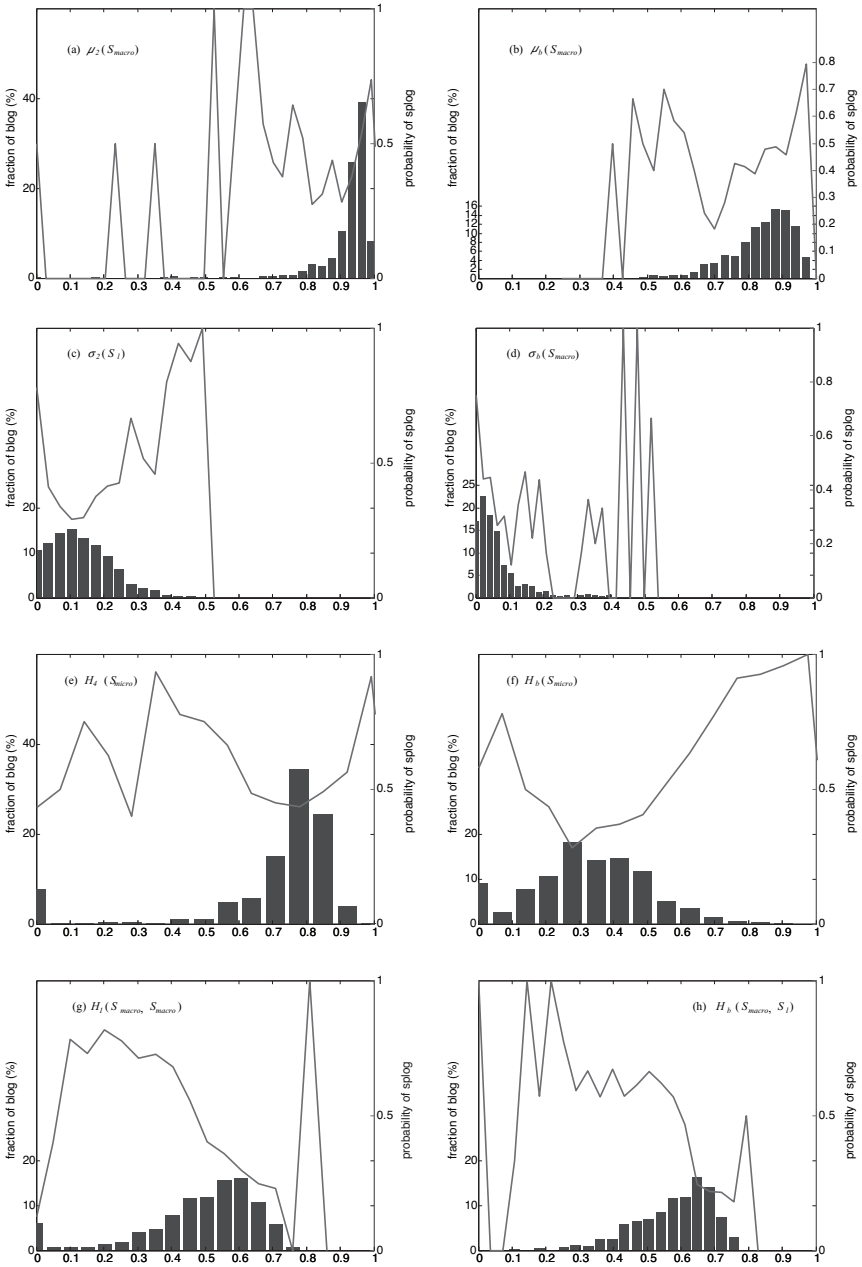


Fig. 12. Each figure shows a histogram as solid bars and a probability curve. The histogram represents the fraction of nonsplogs for each feature value. The curve shows splog: nonsplog ratio for each feature value, computed from the annotated ground truth. (a)–(f): regularity features for signal attributes. (g) and (h): joint features for two attributes.

shown on the right column. Figures 12(a)–(d) are regularity features for single attributes. In general the diagonalwise features are more discriminative than the blockwise features. This is particularly evident in the diagonal entropy, as in Figure 12(e) $H_4(S_{micro})$, while the block entropy has relatively poor discriminabiliy, as in Figure 12(f) $H_b(S_{micro})$. This might because our clustering algorithm doesn't effectively segment splog posts into coherent groups—unlike normal blogs that tend to have short-term coherence, splog posts could be similar either all the time, or very constantly. Figures 12(g) and (h) show joint features for two attributes. Except for few extreme splog cases, attributes of splogs are mostly likely to correlate.

We select the top 32 temporal features using Fisher LDA (see Section 6.2.3). These features are then combined with the baseline content features in splog detector. In the next section we shall discuss the splog detection performance with respect to these features.

6.4 Detection Performance

In this section we compare our results against content-based spam detection approaches. Our approach combines the temporal features with the BCC content features—features extracted from different parts of a blog (see Section 6.2.2). These combined feature sets are compared with the two types of baseline content features. We first compare (in Section 6.4.1) these combined feature sets with the BCC features alone. Secondly, we compare (in Section 6.4.2) these combined feature sets with the HPC features—features extracted from a single blog homepage, as suggested in prior work (see Section 6.2.1). Finally we present the detection performance of different feature sets tested on symmetric and imbalanced datasets (in Section 6.4.3).

Our splog detector combines the new features (regularity and joint features) with traditional content based features into a large feature vector. We then use standard machine learning techniques—SVM classifier implemented using libsvm package [Chang and Lin 2001]—to classify each blog into two classes: splog or normal blog. We tested different types of SVM kernels and found a radial basis function (RBF) kernel work the best in our case. The parameters of RBF kernel, C and γ, are determined using a grid search method.

We use well-known metrics—precision, recall, and F1—to compare the relative performance of using different features. They are defined as follows:

$$\begin{aligned} precision &= \frac{\text{\#splogs detected as splog}}{\text{\#blogs detected as splog}}, \\ recall &= \frac{\text{\#splogs detected as splog}}{\text{\# splogs}}, \\ F1 &= \frac{2 \cdot precision \cdot recall}{precision + recall}, \end{aligned} \tag{10}$$

where the F1 is the harmonic mean between precision and recall, which gives a single measurement of the detection performance.

A 5-fold cross-validation technique is used to evaluate the performance of the feature sets. Given N examples, a standard k-fold cross-validation uses

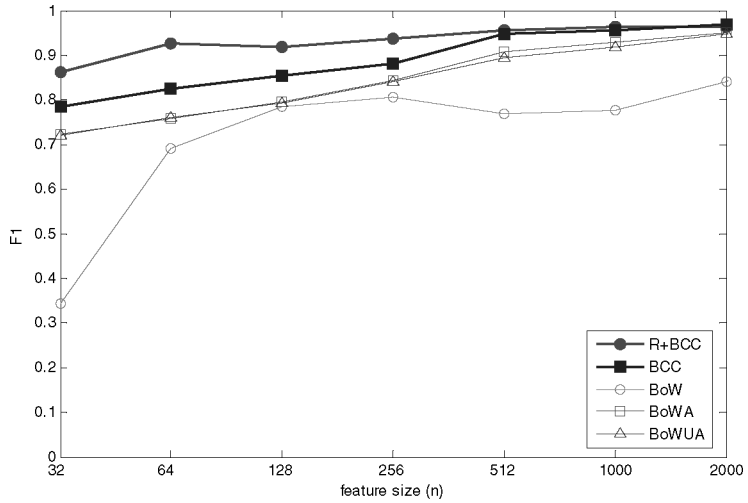


Fig. 13. F1 measures of different feature sets tested on symmetric testing set. We merge the temporal features with content features, designated as $R + BCC$–n, where n is the total feature size of the merged feature set. The figure shows our proposed $R + BCC$–n feature sets significantly improve using pure content-based features—the BCC features (BCC–n) and the HPC features (BoW–n, $BoWA$–n and $BoWUA$–n).

$N(k–1)/k$ examples for training and the rest N/k for testing. Each feature set is tested against two types of testing sets: symmetric and imbalanced. For the symmetric setting, all the 800 splogs and 800 normal blogs are used. The imbalanced dataset was created to mimic real world splog distribution statistics. We create an imbalanced dataset as follows. Let us assume that p is the observed splog fraction. Then we down-sample the splogs and blogs at the ratio p:(1–p) from the testing examples to create a p:(1–p) imbalanced testing set.

6.4.1 *Merging and Comparing with Blog Content (BCC) Features.* The baseline BCC features, designated as *BCC-n*, are n-dimensional feature vectors constructed only using the content-based analysis. The top n features are selected using Fisher LDA (see Section 6.2.3). The temporal features, designated as R, are constructed as a 32-dimensional feature vector. We merge the temporal features with content features, designated as $R + BCC$–n, where n is the total feature size of the merged feature set. Note that the notation of $R + BCC$–32 and R are exchangeable—both denotes the feature set containing 32 dimensional temporal features and no content features.

The top two curves of Figure 13 indicate the performance of temporal features and content baseline BCC features in terms of the F1 measures on symmetric testing set. Detailed results are provided in Table IV. The comparison suggests that, for small feature spaces, by *using the temporal characteristics* we get significant improvement over only using content features. It shows the temporal structural features alone out perform the best 32 content features. It also suggests the baseline and temporal features jointly work very well. In each case the performance of $R + BCC$–n improves over using

Table IV.
The table shows a comparison of the temporal features (R) and the combination of baseline BCC features with the temporal features ($R + BCC$–n) against the BCC features (BCC–n) alone and the HPC features (*BoW*, *BoWA*, *BoWUA*, etc.) over two types of testing sets In the table, the performance measures for BCC–n and all the HPC features are subtracted from $R + BCC$–n with the same dimensionality, for example, the precision of BCC-32 is $0.862 - 0.056 = 0.806$. The table indicates a significant improvement due to a small set of temporal features. It also shows that the temporal features alone (R) performs better than the BCC-32 feature set.

		Symmetric testing set			Imbalanced testing set		
	Feature	Precision	Recall	F1	Precision	Recall	F1
	R	0.862	0.861	0.862	0.634	0.578	0.605
	BCC-32	−0.056	−0.093	−0.076	−0.270	−0.489	−0.462
HPC	BoW−32	0.138	−0.653	−0.518	0.366	−0.345	−0.227
	BoA−32	−0.283	0.139	−0.129	−0.634	−0.578	NaN
	BoU−32	−0.328	0.139	−0.165	0.366	−0.545	−0.540
	BoWU−32	−0.298	0.139	−0.141	0.366	−0.478	−0.423
	BoWA−32	−0.297	0.139	−0.140	0.366	−0.534	−0.520
	BoUA−32	−0.328	0.139	−0.165	0.366	−0.545	−0.540
	BoWUA−32	−0.298	0.139	−0.141	0.366	−0.478	−0.423
	R+BCC−64	0.923	0.930	0.927	0.741	0.700	0.720
	BCC−64	−0.108	−0.097	−0.103	−0.176	−0.411	−0.338
HPC	BoW−64	−0.341	−0.082	−0.237	0.259	−0.444	−0.313
	BoA−64	−0.311	0.070	−0.168	0.259	−0.611	−0.557
	BoU−64	−0.326	0.070	−0.179	0.259	−0.656	−0.635
	BoWU−64	−0.310	0.070	−0.167	0.259	−0.578	−0.502
	BoWA−64	−0.312	0.070	−0.169	0.259	−0.600	−0.538
	BoUA−64	−0.326	0.070	−0.179	0.259	−0.656	−0.635
	BoWUA−64	−0.310	0.070	−0.167	0.259	−0.578	−0.502
	R+BCC−128	0.918	0.919	0.918	0.688	0.711	0.700
	BCC−128	−0.057	−0.073	−0.064	−0.153	−0.367	−0.281
HPC	BoW−128	−0.272	0.081	−0.133	0.312	−0.444	−0.279
	BoA−128	−0.273	0.081	−0.134	0.312	−0.589	−0.482
	BoU−128	−0.297	0.081	−0.152	0.312	−0.622	−0.537
	BoWU−128	−0.262	0.081	−0.125	0.312	−0.567	−0.448
	BoWA−128	−0.256	0.081	−0.122	0.312	−0.555	−0.431
	BoUA−128	−0.297	0.081	−0.152	0.312	−0.622	−0.537
	BoWUA−128	−0.262	0.081	−0.125	0.312	−0.567	−0.448
	R+BCC−256	0.935	0.940	0.938	0.787	0.778	0.782
	BCC−256	−0.072	−0.042	−0.058	−0.125	−0.278	−0.212
HPC	BoW−256	−0.242	0.024	−0.132	−0.352	−0.667	−0.605
	BoA−256	−0.215	0.060	−0.101	0.213	−0.578	−0.449
	BoU−256	−0.252	0.060	−0.126	0.213	−0.578	−0.449
	BoWU−256	−0.208	0.060	−0.096	0.213	−0.467	−0.307
	BoWA−256	−0.205	0.060	−0.094	0.213	−0.522	−0.375
	BoUA−256	−0.252	0.060	−0.126	0.213	−0.578	−0.449
	BoWUA−256	−0.208	0.060	−0.096	0.213	−0.467	−0.307
	R+BCC−512	0.956	0.954	0.955	0.845	0.789	0.816
	BCC−512	0.006	−0.018	−0.006	0.016	−0.100	−0.051
HPC	BoW−512	−0.191	−0.184	−0.187	−0.578	−0.700	−0.683
	BoA−512	−0.203	0.046	−0.096	0.155	−0.522	−0.395
	BoU−512	−0.201	0.046	−0.095	0.155	−0.478	−0.341
	BoWU−512	−0.148	0.046	−0.061	0.155	−0.389	−0.245
	BoWA−512	−0.125	0.046	−0.047	0.155	−0.433	−0.291

(*Continues*)

Table IV. (*Continued*)

		Symmetric testing set			Imbalanced testing set		
	Feature	Precision	Recall	F1	Precision	Recall	F1
	BoUA−512	−0.201	0.046	−0.095	0.155	−0.478	−0.341
	BoWUA−512	−0.148	0.046	−0.061	0.155	−0.389	−0.245
	R+BCC−1000	0.964	0.961	0.963	0.879	0.889	0.884
	BCC−1000	−0.005	−0.005	−0.006	0.019	−0.011	0.004
HPC	BoW−1000	−0.175	−0.193	−0.185	−0.460	−0.600	−0.542
	BoA−1000	−0.176	0.039	−0.081	0.121	−0.556	−0.384
	BoU−1000	−0.174	0.039	−0.080	0.121	−0.511	−0.336
	BoWU−1000	−0.115	0.039	−0.044	0.121	−0.389	−0.217
	BoWA−1000	−0.095	0.039	−0.033	0.121	−0.478	−0.301
	BoUA−1000	−0.174	0.039	−0.080	0.121	−0.511	−0.336
	BoWUA−1000	−0.115	0.039	−0.044	0.121	−0.389	−0.217
	R+BCC−2000	0.970	0.960	0.965	0.916	0.844	0.879
	BCC−2000	0.005	0.004	0.004	−0.017	0.045	0.015
HPC	BoW−2000	−0.128	−0.119	−0.123	−0.381	−0.422	−0.407
	BoA−2000	−0.123	0.040	−0.048	0.084	−0.488	−0.354
	BoU−2000	−0.143	0.040	−0.059	0.084	−0.444	−0.308
	BoWU−2000	−0.067	0.040	−0.016	0.084	−0.411	−0.274
	BoWA−2000	−0.066	0.040	−0.015	0.084	−0.388	−0.253
	BoUA−2000	−0.143	0.040	−0.059	0.084	−0.444	−0.308
	BoWUA−2000	−0.067	0.040	−0.016	0.084	−0.411	−0.274
	R+BCC−5000	0.886	0.921	0.903	0.803	0.589	0.680
	BCC−5000	−0.014	−0.033	−0.023	−0.117	−0.056	−0.080
HPC	BoW−5000	0.062	0.010	0.037	−0.143	0.100	−0.006
	BoA−5000	−0.233	−0.172	−0.205	−0.584	−0.511	−0.565
	BoU−5000	−0.053	0.079	0.006	0.197	−0.267	−0.193
	BoWU−5000	0.018	0.079	0.047	0.197	−0.200	−0.120
	BoWA−5000	−0.115	−0.136	−0.125	−0.487	−0.456	−0.492
	BoUA−5000	−0.053	0.079	0.006	0.197	−0.267	−0.193
	BoWUA−5000	0.018	0.079	0.047	0.197	−0.200	−0.120
	R+BCC−10000	0.974	0.834	0.898	0.704	0.556	0.621
	BCC−10000	−0.003	−0.004	−0.003	0.096	0.022	0.050
HPC	BoW−10000	−0.059	0.124	0.038	0.021	0.177	0.108
	BoA−10000	−0.165	0.087	−0.036	−0.136	−0.089	−0.109
	BoU−10000	−0.174	0.060	−0.054	−0.140	−0.067	−0.097
	BoWU−10000	−0.066	0.070	0.008	−0.075	0.066	0.005
	BoWA−10000	−0.025	0.096	0.041	−0.092	0.111	0.017
	BoUA−10000	−0.196	0.091	−0.053	−0.340	−0.467	−0.478
	BoWUA−10000	−0.197	0.057	−0.068	−0.113	−0.412	−0.389

content features alone, indicating that they are complementary. The performance gain by using the temporal features is promising—the size of temporal features is relatively small, compared to the large size content features. The improvement is significant, especially in the low-dimensional feature vector cases. While the high-dimensional content features perform very well, there is a danger of overfitting and the results may not generalize well to new data.

These promising results suggest that the temporal features of a blog are key distinguishing characteristics and allow us to distinguish between splogs and normal blogs. We next compare against related work [Kolari et al. 2006a; Kolari et al. 2006c; Kolari et al. 2006d].

6.4.2 *Comparing with Homepage Content (HPC) Features.* We compare our splog detection technique with the prior work by Kolari et al. [2006a, 2006c, 2006d], using the HPC feature sets as described in Section 6.2.1.

The three types of HPC prime features, *bag-of-words*, *bag-of-anchors* and *bag-of-urls* are designated as *BoW-n*, *BoA-n* and *BoU-n* respectively, where n is corresponding feature size, and the letter W, A and U stand for *words*, *anchors* and *urls*. *BoWU-n*, *BoWA-n* and *BoUA-n* are the merged set of two prime features and *BoWUA-n* is the merged set of all three prime features.

We use SVMs [Chang and Lin 2001] with a 5-fold cross-validation to evaluate these HPC features. We have experimented with binary, tf and tf-idf feature values over different types of SVM kernel functions. The best performance of these HPC feature sets (binary feature values and a polynomial kernel with optimal parameters found by a grid method) are shown in Figure 13 and reported in Table IV to compare with *BCC-n* and $R + BCC$–n feature sets.

Figure 13 compares the detection performance over different feature sets with size n from 32 to 2000, based on the F1 measures on symmetric testing set. The overall results indicate that our $R + BCC$–n features with $n < 512$ are as effective as large content-based features with $n \geq 512$. We also found that BCC features (*BCC-n*), the content features extracted from different parts of a blog (URLs, titles, anchor text, homepage content and permalink content) slightly work better than HPC features (such as *BoWUA-n* and *BoWA-n*), the content features extracted from a single homepage of a blog. It can be checked in Table IV that the detection performance does not show much improvement when $n > 2000$ for *any* feature sets. Some larger feature sets like *BoW-1000* and *BoA-5000* perform even worse than smaller feature sets. This indicates a consequence of large feature sets over-fitting the data. We conclude that the temporal features can significantly improve detection performance over content based analysis. We now present results on an imbalanced testing data set.

6.4.3 *Testing on Imbalanced Data.* A symmetric testing set is often useful for comparing the utility against different types of features. However, a more realistic scenario has to deal with the class imbalanced distribution. In the context of splog detection, there are many more instances of normal blogs than the splogs. The splog ratio is around 10% in our annotated subset and has been reported during 10–20% in Umbria [2006]. Hence, we create an imbalanced testing set with splog ratio $p = 0.1$ and compare the testing results with those on symmetric testing set ($p = 0.5$).

Figure 14 shows the detection performance of different feature sets, based on the F1 measures tested on imbalanced testing set. The performance measures are not as high as those on symmetric testing set, mainly because of lower recall values (see Table IV). However, the temporal features appear to be relatively robust on imbalanced testing set, compared with using content features alone. This suggests that temporal structural analysis can contribute to a robust splog detector in a more realistic setting.

The table shows that the temporal features significantly outperform the content based features, including the BCC features and the HPC features

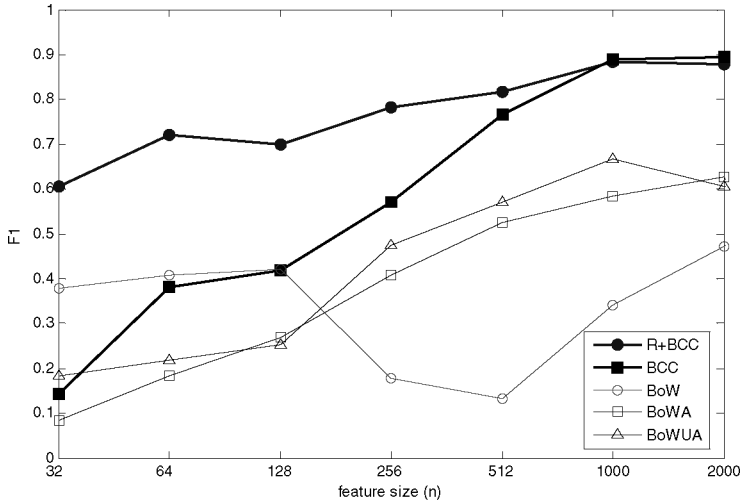


Fig. 14. F1 measures of different feature sets on imbalanced testing set. The figure shows our proposed $R + BCC$–n feature sets are relatively robust, compared to the BCC features (BCC-n) alone and the HPC features (BoW-n, $BoWA$-n and $BoWUA$-n).

suggested in prior work. In low dimensional feature sets (with feature size no more than 256), the temporal features improve over the best content based features 6.6%–17.6% on symmetric testing dataset, and 37.2%–76.9% on imbalanced testing dataset. The results indicate that the proposed temporal features well capture the splog characteristics.

7. CONCLUSION

In this article, we presented a new framework to detect splogs in the blogosphere. In our approach, splogs are recognized by their temporal dynamics in addition the traditional content descriptors. While blog content typically varies over time, the temporal structures captured by the regularity and joint features reveal a stable blog character. This stability makes time based structural features particularly attractive in splog detection. There were three main ideas in this paper.

First, we proposed the use of the topology of a time series of blog posts as an analysis framework. The topology is induced using a (histogram intersection) similarity measure on the posts. Topological analysis allows for a robust stable representation of a blog as it functions as a generalized spectral analysis tool. We showed how we can compute the self-similar characteristics of a blog, with respect to a specific attribute. We also showed how to extract statistical measures from the self-similar matrix representations.

Second, we examined the temporal dynamics of blogs through a novel "clock metaphor"–based visualization. The visualizations reveal very interesting differences—normal blogs and splogs differ in their temporal characteristics in all three facets—post time, content and link. Normal bloggers are used to posting at a regular but not precise time, while a splog will show machine-like regularity in an exact manner. A normal blogger typically stays on topic for a

while, before showing a topic drift. A splog often copies content from different parts of the Web/blogosphere and hence may show either very high or very low topic diversity. For normal bloggers, changes to content affect the change to the linked sites in the same manner. However, a splog is driven by a strong commercial interest to drive traffic to affiliate sites and shows only limited link temporal diversity.

Third, we proposed two types of temporal features, including (1) regularity features that are computed from the off-diagonals of the self-similar matrix as well as coherent blocks from self-similarity matrices, and (2) the joint features computed from self-similarity matrices across different attributes. We used three measures—mean, standard deviation and entropy—to quantify the regularity patterns along the off-diagonals. The expectation along the diagonals of the topological matrix is equivalent to the generalized autocorrelation of the time series data under the specific similarity measure. We computed the statistical measures from coherent blocks as features. The self-similarity matrix was segmented using blocks via an agglomerative hierarchical clustering method. The joint features were derived from a pair of self-similarity matrices on different attributes. We computed the joint entropy to measure attribute correlation.

We conducted extensive experiments on real-world, large blog dataset (TREC-Blog), with appreciable results. We began the process by manually labeling 9167 blogs selected using a random sampling strategy, ending up with 7380 normal blogs and 897 splogs. We then randomly selected 800 splogs and 800 normal blogs to create two evaluation sets—a balanced set and 1:9 imbalanced set. We created two types of content feature sets: (1) homepage content features (HPC): features suggested in Kolari et al. [2006a, 2006c, 2006d], and (2) blog contextual content features (BCC): features derived from traditional content analysis and tailored in the blog context. Both HPC and BCC features were reduced in dimensionality using Fisher LDA.

Our splog detector combined temporal features (regularity and joint features) with BCC content based features into a large feature vector. We then used a standard SVM classifier to classify each blog into two classes: splog or normal blog. We used well-known metrics—precision, recall, and F1—to compare the relative performance of using different features, with five fold cross validation.

Our results indicate that temporal features significantly outperform the content based features, including the BCC features and the HPC features suggested in prior work. In low dimensional feature sets (with feature size no more than 256), the temporal features improve over the best content based features 6.6%–17.6% on symmetric testing dataset, and 37.2%–76.9% on imbalanced testing dataset. Our results also show that the BCC and temporal features jointly work very well. The results generally show that adding temporal features can give improved performance when compared with content-based features alone, and that this difference is more significant when the number of content-based features is small and diminishes as more content-based features are added. While the high-dimensional content features perform very well, there is a danger of over fitting.

The basic idea of our method is to identify splog by machine-like patterns. This method can be used in the hosting or ping server side where the timestamps of posts are more reliable than those extracted by crawlers. However, we admit that this method can be defeated if spammers manage to avoid regular posting behavior. More sophisticated posting pattern analysis can be included. Though we have concentrated our discussion on the temporal properties of splogs and their effectiveness on splog detection, we recognize that link-based solutions can have significant impact. In Lin et al. [2007], we incorporate the linking structure in detecting splogs by using a HITS based hub score measure and the preliminary results are promising. There are additional challenges that we propose to address in future research, as we expect the splogs to evolve into more sophisticated forms to evade detection—(a) develop probabilistic representations of the topology and (b) short term topological analysis (similar to the short time Fourier Transform), (c) investigate blog signature dynamics—that is, characterize the temporal variations in the spectral signature.

REFERENCES

BENCZUR, A., CSALOGANY, K., SARLOS, T., AND UHER, M. 2005. Spamrank-fully automatic link spam detection. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*.

CASTILLO, C., DONATO, D., BECCHETTI, L., BOLDI, P., SANTINI, M., AND VIGNA, S. 2006. A reference collection for web spam. *SIGIR Forum* (Dec.) ACM Press, 11–24.

CAVNAR, W. B. AND TRENKLE, J. M. 1994. N-gram-based text categorization. In *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval*.

CHANG, C.-C. AND LIN, C.-J. 2001. Libsvm: A library for support vector machines. ntv.edu.twocjlin/papers (libsvm, ps.gz).

DUDA, R. O., HART, P. E., AND STORK, D. G. 2001. *Pattern Classification.* John Wiley & Sons, Inc. New York.

ECKMANN, J., KAMPHORST, S. O., AND RUELLE, D. 1987. Recurrence plots of dynamical systems. *Europhysics Lett.* 4, 973–977.

FETTERLY, D., MANASSE, M., AND NAJORK, M. 2004. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases*. Colocated with ACM SIGMOD/PODS. 1–6.

FETTERLY, D., MANASSE, M., AND NAJORK, M. 2005. Detecting phrase-level duplication on the world wide web. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 170–177.

FOGARAS, D. AND RACZ, B. 2005. Scaling link-based similarity search. In *Proceedings of the 14th International Conference on World Wide Web*. ACM Press. 641–650.

FOOTE, J., COOPER, M., AND NAM, U. 2002. Audio retrieval by rhythmic similarity. In *Proceedings of the International Conference on Music Information Retrieval*. 265–266.

GYÖNGYI, Z., GARCIA-MOLINA, H., AND PEDERSEN, J. 2004. Combating Web spam with trustrank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04)*. Toronto, Canada. Morgan Kaufmann. 576–587.

GYÖNGYI, Z. AND GARCIA-MOLINA, H. 2005. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*.

GYÖNGYI, Z., BERKHIN, P., GARCIA-MOLINA, H., AND PEDERSEN, J. 2006. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*. Seoul, Korea. 439–450.

HAN, S., AHN, Y., MOON, S., AND JEONG, H. 2006. Collaborative blog spam filtering using adaptive percolation search. *WWW2006 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*. Edinburgh.

KOLARI, P. 2005. Welcome to the splogosphere: 75% of new pings are spings (splogs). http://ebiquity.umbc.edu/blogger/2005/12/15/welcome-to-the-splogosphere-75-of-new-blog-posts-are-spam/.

KOLARI, P., FININ, T., AND JOSHI, A. 2006a. Svms for the blogosphere: Blog identification and splog detection. In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*.

KOLARI, P., JAVA, A., AND FININ, T. 2006b. Characterizing the splogosphere. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wide Web Conference*.

KOLARI, P., JAVA, A., FININ, T., MAYFIELD, J., JOSHI, A., AND MARTINEAU, J. 2006c. Blog track open task: Spam blog classification. *TREC Blog Track Notebook*.

KOLARI, P., JAVA, A., FININ, T., OATES, T., AND JOSHI, A. 2006d. Detecting spam blogs: A machine learning approach. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*. Boston, MA.

LIN, Y., SUNDARAM, H., CHI, Y., TATEMURA, J., AND TSENG, B. 2007. Splog detection using content, time and link structures. *IEEE International Conference on Multimedia and Expo 2007*: 2030–2033.

LIN, Y.-R., CHEN, W.-Y., SHI, X., SIA, R., SONG, X., CHI, Y., HINO, K., SUNDARAM, H., TATEMURA, J., AND TSENG, B. 2006. The splog detection task and a solution based on temporal and link properties. In *Poceedings of the 15th Text REtrieval Conference (TREC'06)*.

MACDONALD, C. AND OUNIS, I. 2006. *The trec blogs06 collection: Creating and analyzing a blog test collection*. TR-2006-224. Department of Computer Science, University of Glasgow.

MISHNE, G., CARMEL, D., AND LEMPEL, R. 2005. Blocking blog spam with language model disagreement. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*.

NARISAWA, K., YAMADA, Y., IKEDA, D., AND TAKEDA, M. 2006. Detecting blog spams using the vocabulary size of all substrings in their copies. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem*.

NEWMAN, M. AND GIRVAN, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E 69*, 2, 26113.

NTOULAS, A., NAJORK, M., MANASSE, M., AND FETTERLY, D. 2006. Detecting spam Web pages through content analysis. In *Proceedings of the 15th International Conference on World Wide Web*. Edinburgh, Scotland. ACM Press, 83–92.

SALVETTI, F. AND NICOLOV, N. Weblog classification for fast splog filtering: A url language model segmentation approach. In *Proceedings of the Human Language Technology Conference of the NAACL*. Companion Volume: Short Papers, 137–140.

SHEN, G., GAO, B., LIU, T.-Y., FENG, G., SONG, S., AND LI, H. 2006. Detecting link spam using temporal information. In *Proceedings of the 6th International Conference on Data Mining*. IEEE Computer Society. 1049–1053.

SURBL *Surbl—spam uri realtime blocklists*. http://www.surbl.org/.

SWAIN, M. AND BALLARD, D. 1991. Color indexing. *Int. J. Comput. Vision 7*, 1, 11–32.

UMBRIA. 2006. *Spam in the blogosphere*. http://www.umbrialistens.com/files/uploads/umbria_splog.pdf.

URVOY, T., LAVERGNE, T., AND FILOCHE, P. 2006. Tracking web spam with hidden style similarity. *AIRWEB*, Seattle, WA.

VON AHN, L., BLUM, M., AND LANGFORD, J. 2004. Telling humans and computers apart automatically. *Comm. ACM 47*, 2, 56–60.

WIKIPEDIA.h ttp://en.wikipedia.org/wiki/.

WU, B. AND DAVISON, B. 2005. Identifying link farm spam pages. In *Proceedings of the International World Wide Web Conference*. ACM Press. 820–829.

ZAWODNY, J. 2005 Yahoo! Search blog: A defense against comment spam. http://www.ysearchblog.com/archives/000069.html.

Received April 2007; revised September 2007; accepted October 2007