The Splog Detection Task and A Solution Based on Temporal and Link Properties

Yu-Ru Lin, Wen-Yen Chen, Xiaolin Shi, Richard Sia, Xiaodan Song,
Yun Chi, Koji Hino, Hari Sundaram, Jun Tatemura and Belle Tseng

NEC Laboratories America

10080 N. Wolfe Road – Suite SW3-350, Cupertino, CA 95014

ABSTRACT

Spam blogs (splogs) have become a major problem in the increasingly popular blogosphere. Splogs are detrimental in that they corrupt the quality of information retrieved and they waste tremendous network and storage resources. We study several research issues in splog detection. First, in comparison to web spam and email spam, we identify some unique characteristics of splog. Second, we propose a new online task that captures the unique characteristics of splog, in addition to tasks based on the traditional IR evaluation framework. The new task introduces a novel time-sensitive detection evaluation to indicate how quickly a detector can identify splogs. Third, we propose a splog detection algorithm that combines traditional content features with temporal and link regularity features that are unique to blogs. Finally, we develop an annotation tool to generate ground truth on a sampled subset of the TREC-Blog dataset. We conducted experiments on both offline (traditional splog detection) and our proposed online splog detection task. Experiments based on the annotated ground truth set show excellent results on both offline and online splog detection tasks.

1. INTRODUCTION

The blogosphere is growing extremely fast and provides new business opportunities in areas such as advertisement, opinion extraction, and marketing. However, spam blogs (splogs) have become a major problem in the blogosphere—they reduce the quality of information retrieval results and waste network and storage resources [5,8]. Therefore, detecting splogs in the blogosphere has great importance.

In this paper, we propose our solution to detect splogs in the blogosphere. The main contributions of our work are as follows:

1. **Modeling the splog problem:** Unlike web or email spam, a splog is dynamic since it continuously generates fresh content to drive traffic. To solve the splog problem, we need to take advantage of the unique splog properties.
2. **Evaluation:** Splogs are characterized by temporal content dynamics and hence need to be identified as quickly as possible before they waste network and storage resources. We propose a time-sensitive evaluation framework to measure splog detection performance based on how fast the detection is made.
3. **Regularity based Detection:** Our detection algorithm identifies unique features such as temporal and link properties useful for detecting splogs. We identify temporal content regularity (self-similarity of content) and temporal structural regularity (regular post times) as well as regularity in the linking structure (frequent links to non-authoritative websites).

We have evaluated our approach using the traditional offline task, as well as our proposed online metrics. The results are excellent indicating the combined feature set works well in both offline and online splog detection tasks. The online task reveals the sensitivity of the detection to the amount of evidence (posts) as well as the complimentary roles played by content and splog regularity features.

Most of previous work in spam detection comes from web spam detection. Prior work to detect web spams can be further categorized into content analysis [6,7] and link analysis [2,3]. Our work combines traditional features with temporal and link features that are unique to blogs.

The rest of this paper is organized as follows. In the next section we provide a high-level definition of splogs; we shall also discuss splog characteristics and how splog differ from web sites. In section 3, we provide the online task definition and also provide a baseline offline splog detection task. In section 4, we present out splog detection framework and we discuss our proposed regularity (temporal and link) based features. In section 5, we discuss data pre-processing and our annotation tool to label data. In section 6, we present out experimental results and we present our conclusions in section 7.

2. WHAT ARE SPLOGS?

In this section, we provide a high-level definition of *splogs* and the splog problem we face today (Section 2.1), the typical splog characteristics (Section 2.2), and the differences between splog and other types of spam (Section 2.3).

2.1 Working definition of splogs

Spam blogs, which are called *splogs*, are undesirable weblogs that the creators use solely for promoting affiliated sites [1]. As blogs became increasingly mainstream, the presence of splogs has a detrimental effect in the blogosphere. According to multiple reports, the following are alarming statistics.

- 10-20% of blogs are splogs. For the week of Oct. 24, 2005, 2.7 million blogs out of 20.3 million are splogs [8].
- An average of 44 of the top 100 blogs search results in the three popular blog search engines came from splogs [8].
- 75% of new pings came from splogs; more than 50% of claimed blogs pinging *weblogs.com* are splogs [5].

The statistics exhibit serious problems caused by splogs, including (1) the degradation of information retrieval quality and (2) the tremendous waste of network and storage resources.

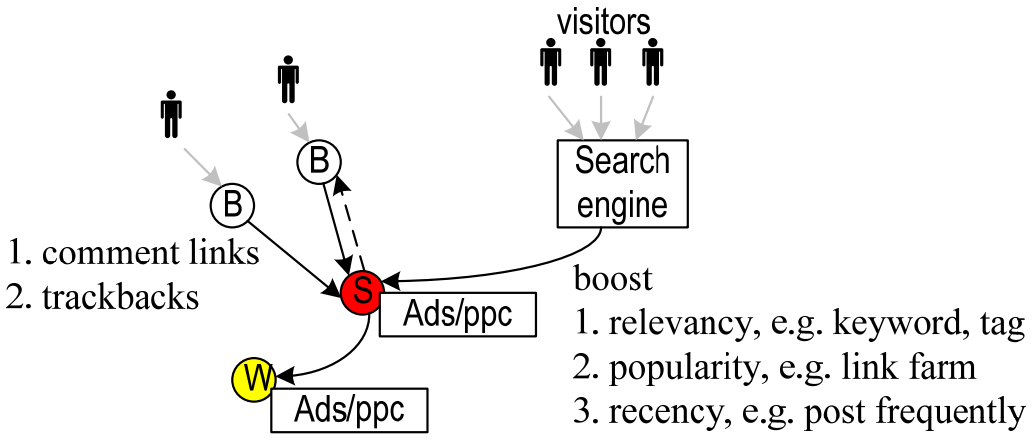

Figure 1: Splogs use different schemes to achieve spamming. "B" represents a blog, "S" represents a splog, and "W" refers to an affiliate site. There is usually a profitable mechanism (Ads/ppc) in the splog or affiliated site(s).

Figure 1 illustrates the overall scheme taken by splog creators. Their motive is to drive visitors to affiliated sites (including the splog itself) that have some *profitable mechanisms*. By profitable mechanism, we refer to web-based business methods, such as search engine advertising programs (e.g. *Google AdSense*) or pay-per-click (ppc) affiliate programs. There are several schemes used by spammers to increase the visibility of splogs by getting indexed with high ranks on popular search engines. To deceive the search engine, the spammer may boost (1) relevancy (e.g. via keyword stuffing), (2) popularity (e.g. via link farm), or (3) recency (e.g. via frequent posts), based on some ranking criteria used by search engines. The increased visibility is unjustifiable since the content in splogs is often nonsense or stolen from other sites [1]. The spammer also attacks regular blogs through comments and trackbacks to boost the splog ranking.

2.2 Typical splog characteristics

In a typical splog, content is usually generated by machines in order to attract visitors through their appearance in either search engines or individual blogs. By splog, we refer to a blog created by an author who has the intention of spamming. Note that a blog that may contain spam in the form of comment spam or trackback spam is not considered a splog.

There are typical characteristics observed in splogs:

1. **Machine-generated content:** splog entries are generated automatically, usually nonsense, gibberish, repetitive or copied from other blogs or websites.
2. **No value-addition:** splogs provide useless or no unique information to their readers. There are blogs using automatic content aggregating techniques to provide useful service such as podcasting—these are legitimate blogs because of their value addition.
3. **Hidden agenda, usually an economic goal:** splogs have commercial intention that can be revealed if we observe any affiliate ads or out-going links to affiliate sites.

Some of these characteristics, such as no value-addition or hidden agenda, can also be found in other types of spams (e.g. web spam). However, splogs have unique properties that will be highlighted in the next section.

2.3 Uniqueness of splogs

Splogs are different from web spams in the following aspects.

1. **Dynamic content:** blog readers are mostly interested in recent entries. Unlike web spams where the content is static, a splog continuously generates fresh content to drive traffic.
2. **Non-endorsement link:** A hyperlink is often interpreted as an endorsement of other pages. It is less likely that a web spam gets endorsements from normal sites. However, since spammers can create hyperlinks (comment links or trackbacks) in normal blogs, links in blogs cannot be simply treated as endorsements.

Because of these two significant differences, the splog problem is different from that of traditional web spam as discussed next.

3. TASK DEFINITION

In this section we propose our evaluation methodology for comparing splog detection techniques on TREC blog dataset. We first describe the detection task framework in Section 3.1. Next, two detection tasks used in traditional information retrieval are given in Section 3.2. In Section 3.3 we propose an online detection task with novel assessment method.

3.1 Framework for detection task

The objective of a splog detector is to remove unwanted blogs. Blog search engines need splog detectors to improve the quality of their search results. Blog search engines differ from general web search engines in their growing contents – namely feeds. The detection decision is performed on a blog that consists of a growing list of entries. Because entries become available gradually, there can be time delay to gather enough evidences (i.e.,

entries) for detection. Since a splog will persist in the index until it is detected, earlier detection with few evidences is crucial for the overall search quality. We refer a detector that can make a decision with less evidence *fast*.

An illustration of how early splog detection is beneficial is shown in Figure 2. The grid represents how the amount of entries (x-axis) increases over time for each blog (y-axis). For a specific time, the gray area denoted by "downloaded in the storage" shows the number of blogs discovered with the corresponding amount of entries. As time passes, more blogs are indexed as well as growing amounts of entries, as shown by the dashed border and arrows.

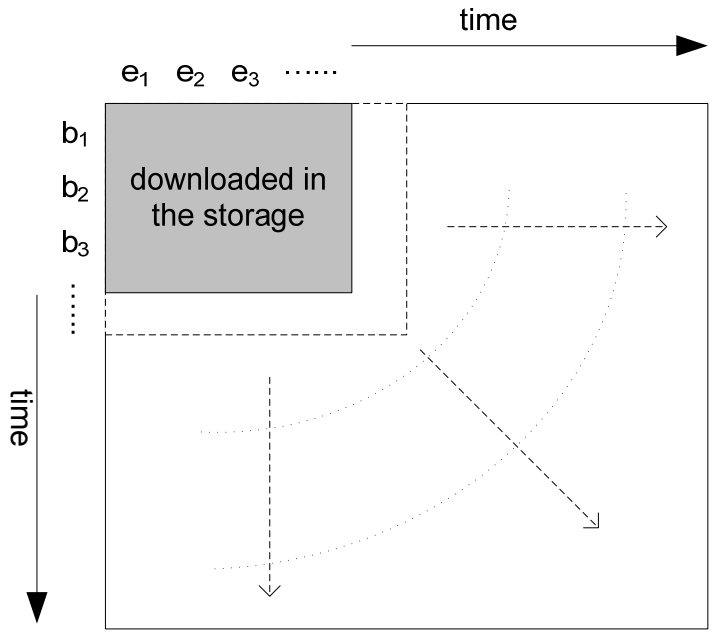

Figure 2: Blogs are discovered and downloaded over time. Similarly, the amount of entries downloaded grows over time. The gray area represents blog data that have been downloaded, and the dashed border and arrows show the downloading process continuing over time.

The target objective for blog search engines is to detect splogs as early as possible. As a result, we need to measure the speed of splog detection. Traditional detectors are evaluated *offline*, where a batch of data is inputted into the detector and some performance metrics are calculated on the detection results. Because we want to also evaluate the speed of splog detection, we propose an *online* detection evaluation. Another aspect of evaluation depends on the availability of ground truth information. Both offline and online detection can be evaluated with or without ground truth. Accordingly, there are four tasks as identified in Table 1.

Table 1: Four detection tasks are identified based on Offline and Online detections.

Dataset \ Task Type	Offline (Traditional)	Online (Time-Sensitive)
With Ground Truth	TASK 1	TASK 3
Without Ground Truth	TASK 2	TASK 4

3.2 Traditional IR-based detection

To compare different detection methods, there are two evaluation frameworks used in traditional information retrieval research and also widely applied in many TREC tracks.

3.2.1 Evaluation with ground truth

Evaluation is designed to compare detectors for an input set of blogs. Given a set of input blogs B with labels, the detectors can be evaluated by k-fold cross-validation, where the performance can be measured by metrics such as precision/recall, AUC, or ROC plot.

3.2.2 Evaluation without ground truth

To evaluate detector performances on a large dataset, there will be limited amount of labeled ground truth. Each detector makes its decision on the large dataset, and returns the detection results as a ranked list. The detector performance is evaluated by measuring the precision at top N (precision@N) of the ranked list based on pooling of multiple detection lists.

Based on the availability of ground truth, splog detectors can be compared using one of the above offline evaluations. However to measure the speed of detection efficiency, we propose an online detection framework.

3.3 Online detection

As discussed above, the benefit of early splog detection is to quickly remove entries by splogs from the search index. Hence, we propose a new framework to evaluate time-sensitive detection performance.

We want to measure the detection performance on newly discovered blogs and observe how the decisions on these blogs can improve as more entries are available. First, blogs in the dataset are partitioned based on the time of discovery (i.e., the first appearance in the dataset). We assume the splog detector evaluates the blog contents at uniform frequency, i.e. $t_0 = t$, $t_1 = t+\Delta t$, …, $t_k=t+k\Delta t$. $B(t_i)$ is defined as a partition that consists of blogs discovered after time t_{i-1} and before t_i. $B(t_0)$ is the initial training set, usually given with labels (splog or non-splog).

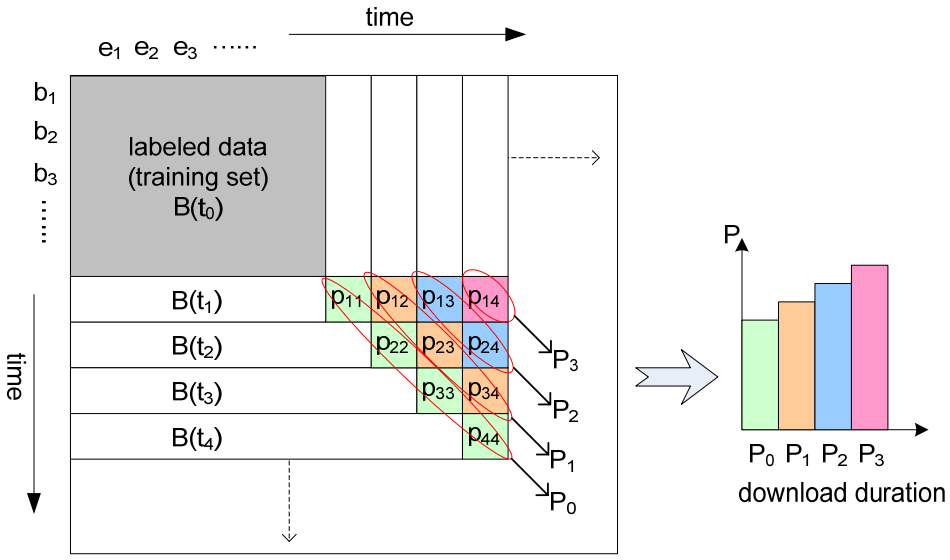

Figure 3: Online time-sensitive evaluation performance.

For each partition $B(t_k)$ ($k > 0$), the detector gives a decision at time t_j ($j >= k$), for which the performance p_{jk} is measured. p_{jk} is the detection performance at time t_j on the partition at t_k ($B(t_k)$). In order to measure the speed of detection, we are interested in how the detector performance p_{jk} improves as j increases. Then, we introduce an overall performance measure of all decisions made with a specific delay. More specifically, for each delay $i = j - k$, the overall performance P_i is given as an average, where $P_i=E[p_{jk} \mid i=j-k]$. The performance is plotted with i on the x-axis as shown in Figure 3, to demonstrate how quickly the detector can make a good decision. Note that

each performance p_{jk} is measured based on the same evaluation metrics as the traditional offline evaluations. We expect the proposed online evaluation to provide significant insights on early detection of splogs.

4. OUR DETECTION METHOD

We have developed new techniques for splog detection based on temporal and linking patterns, which are unique features that distinguish blogs from regular web pages.

Due to the special characteristics of splogs, traditional content-based or link-based spam detection techniques are not sufficient. It is difficult to detect spams for individual pages (i.e., entries) by content-based techniques, since a splog can steal (copy) content from normal blogs. Link-based techniques based on propagation of trust from legitimate sites will work poorly for blogs since spammers can create links (comments and trackbacks) from normal blogs to splogs.

Our observation is that a blog is a growing sequence of entries rather than individual pages. We expect that splogs can be recognized by their abnormal temporal and link patterns observed in entry sequences, since their motivation is different from normal, human-generated blogs. In a splog, the content and link structures are typically machine-generated (possibly copied from other blogs / websites). The link structure is focused on driving traffic to a specific set of affiliate websites. To capture such differences, we introduce new features, namely, temporal regularity and link regularity, which are described in the following subsection.

4.1 Baseline features

We shall now discuss the content based features used in this work – these will serve as the baseline feature set as they are widely used in splog detection. We use a subset of the content features presented in [7]. These features are used to distinguish between two classes of blogs – normal and splogs, based on the statistical properties of the content.

In this work 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: word count (w_c), average word length (w_l) and a vector containing the word frequency distribution (w_f). In this work, each content category is analyzed separately from the rest for computational efficiency.

4.1.1 feature selection using Fisher linear discriminant analysis (LDA)

We need to reduce the length of the vector w_f as the total number of unique terms (excluding words containing digits) is greater than 100,000 (this varies per category, and includes non-traditional usage such as "helloooo"). This can easily lead to over fitting the data. Secondly, the distribution of the words is long-tailed – i.e. most of the words are rarely used.

We expect good feature subsets contain features highly correlated with (predictive of) the class, but uncorrelated with each other. The objective of Fisher LDA is to enable us to determine discriminative features while preserving as much of the class discrimination as possible. The solution is to compute the optimal transformation of the feature space based on a criterion that minimizes the within-class scatter (of the data set) and maximizes the between-class scatter simultaneously. This criterion can also be used as a separability measure for feature selection. We use the trace criteria, $J = tr(S_w^{-1}S_b)$ where S_w denotes the within-class scatter and S_b denotes the between-class scatter matrix. This criterion computes the ratio of between-class variance to the within-class variance in terms of the trace of the product (the trace is just the sum of eigenvalues of $S_w^{-1}S_b$). We select the top k eigenvalues to determine the key dimensions of the w_f vector.

4.2 Temporal regularity

Temporal regularity captures consistency in timing of content creation (structural regularity), and similarity between contents (content regularity). *Content regularity* is given by the autocorrelation of the content, derived from computing a similarity measure on the baseline content feature vectors. We define a similarity measure based on the histogram intersection distance. *Structural regularity* is given by the entropy of the post time difference distribution. A splog will have low entropy, indicating machine generated content.

4.2.1 Temporal Content Regularity (TCR):

We use the autocorrelation of the content to estimate the TCR value. Intuitively, the autocorrelation function (conventionally depicted as R(τ)) of a time series is an estimate of how a future sample is dependent on a current sample. A noise like signal will have a sharp auto-correlation function, while a highly coherent signal's autocorrelation function will fall off gradually. Since splogs are usually finally motivated, we conjecture that their content will be highly similar over time. However human bloggers will tend to post over a diverse set of topics, leading to a sharper autocorrelation function. We define TCR as a self-similarity measure of content and compute it by auto-correlation.

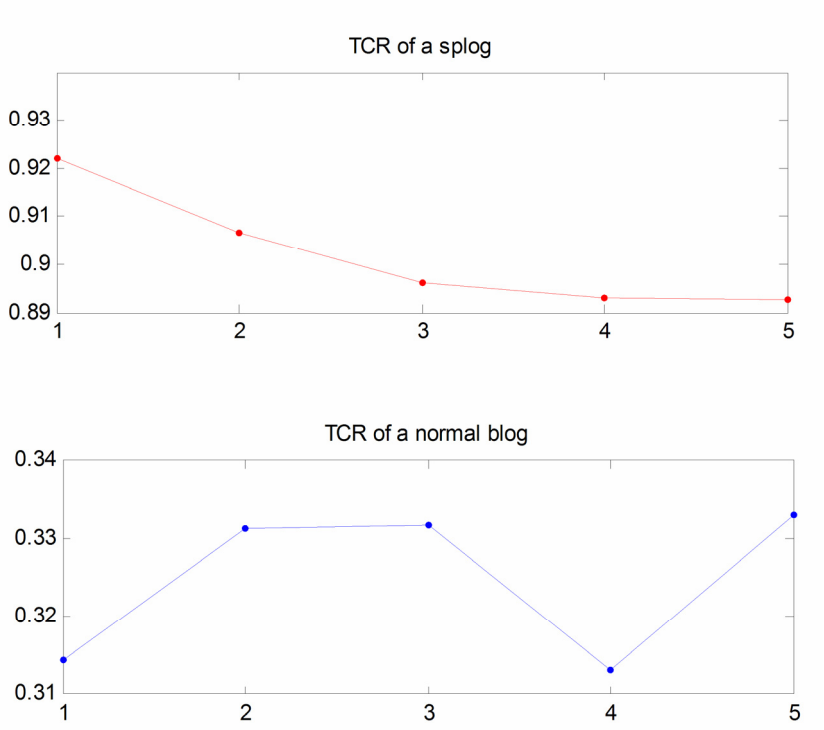

Figure 4: The figure shows the difference in the autocorrelation function between a splog and a normal blog. Notice that auto-correlation function for a splog is very high and nearly constant, while the values for a normal blog are relatively low and fluctuate. These graphs have been derived from a splog and a normal blog from the TREC dataset.

We compute the discrete time autocorrelation function R(k) for the posts. The posts are time difference normalized – i.e. we are only interested in the similarity between the current post and a future post in terms of the number of posts in between (e.g. will the post after next be related to the current post), ignoring time. This is a simplifying assumption, but is useful when many posts do not have the time meta data associated with them. The autocorrelation function is defined as follows:

$$R(k) = 1 - d(p(l), p(l+k)),$$
$$d(p(l), p(l+k)) \triangleq 1 - E\left(\frac{\left|w_f(l) \cap w_f(l+k)\right|}{\left|w_f(l) \cup w_f(l+k)\right|}\right), \quad <1>$$

Where E is the expectation operator, R(k) is the expected value of the autocorrelation between the current l^{th} post and the $(l+k)^{th}$ post; d is the dissimilarity measure, | | is the cardinality operator, and $\cup$, $\cap$ refer to the familiar union and intersection operators. We use the autocorrelation vector R(k) as a feature to discriminate between splogs and normal blogs.

4.2.2 Temporal Structural Regularity (TSR):

We estimate TSR of a blog by computing the entropy of the post time difference distribution. In order to estimate the distribution, we use hierarchical clustering on the post time difference values from a blog.

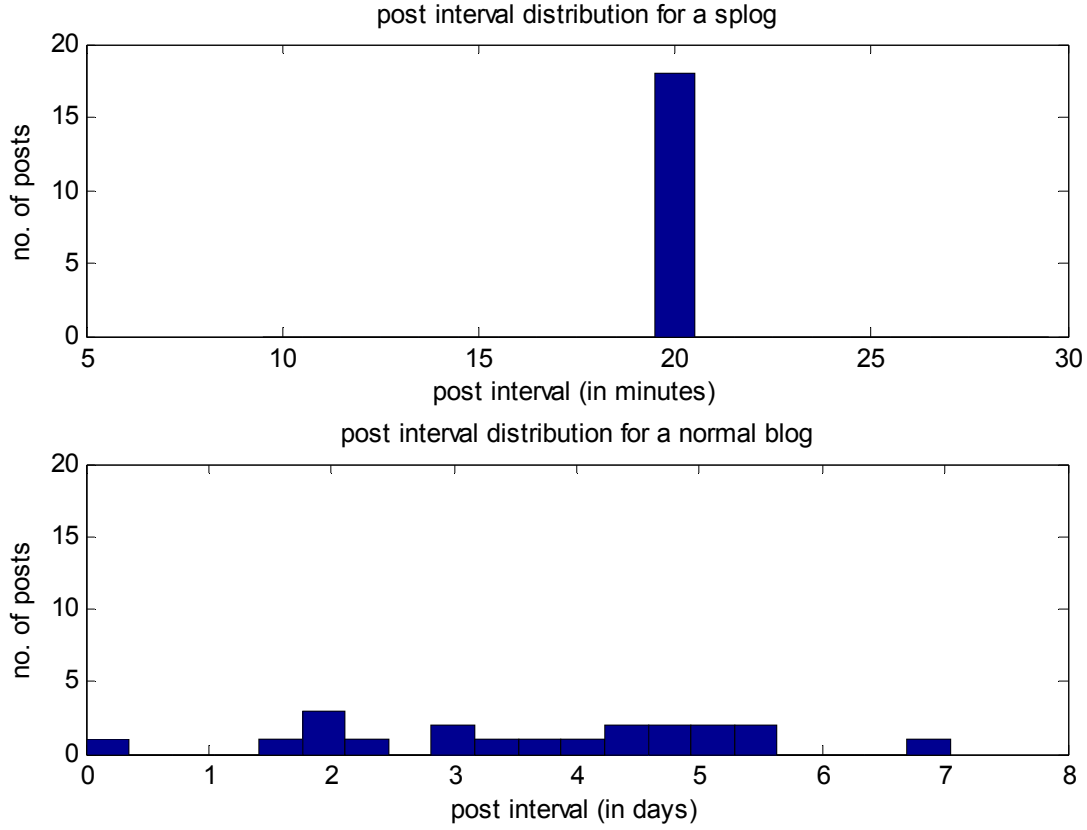

Figure 5: The figure shows the differences in the temporal regularity between a splog and a normal blog. A splog posts with a post time of 20 minutes, while a normal blog has post time interval that is highly varied and ranges to several hours to nearly a week.

The temporal structure of post time interval can be discovered by clustering close post intervals. We use hierarchical clustering method with single link merge criteria on the post interval difference values. 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 the average variance stopping criteria for clustering. Once the clusters are determined, we compute cluster entropy as a measure of TSR:

$$B_e = -\sum_{i=1}^{M} p_i \log p_i, \quad p_i = \frac{n_i}{N},$$
$$TSR = 1 - \frac{B_e}{B_{\max}}, \qquad \text{<2>}$$

where B_e is the blog entropy, B_{max} is the maximum observed entropy, N is the total number of posts, n_i and p_i are the number of posts and the probability of the i^{th} cluster respectively, and M is the number of clusters. Note that for some blogs including normal or splogs, post time is not available as part of the post metadata. We treat such cases as missing data. And if a blog does not have post time information, we do not use TSR as a feature.

4.3 Link regularity estimation

Link regularity (LR) measures consistency in target websites pointed by a blog. We expect that a splog will exhibit more consistent behavior since the main intent of such splogs is to drive traffic to affiliate websites. Secondly we conjecture that there will be a significant portion of links that will be targeted to affiliated websites rather than normal blogs / websites. Importantly these affiliate websites will *not* be authoritative and we do not expect normal bloggers to link to such websites.

We analyze the splog linking behavior using the HITS algorithm [4]. The intuition is that splogs target focused set of websites, while normal blogs usually have more diverse targeting websites. We use HITS with out-link normalization to compute hub scores. The normalized hub score for a blog is a useful indicator of the blog being a splog.

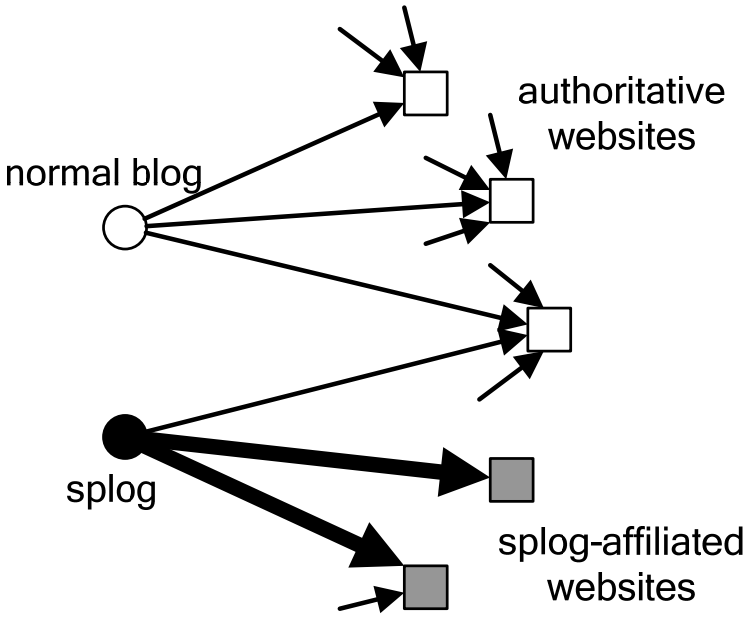

Figure 6: Normal blogs tend to link to authoritative websites while splogs frequently link to affiliate websites that are not authorities. The thickness of the arrow from the splog indicates the frequency with which the splog links to an non-authoritative, affiliate website.

We put blogs and their linking websites on two sides of a bi-partite graph to construct an adjacency matrix A, where A_{ij}=1 indicates there is a hyperlink from blog b_i to website w_j. In the original HITS algorithm, good hubs and good authorities are identified by the mutually reinforcing relationship: $a=A^T h$, $h=Aa$, where a is the authority score, h is the hub score and A is the adjacency matrix. A blog with divergent out-links to authoritative websites will obtain a higher hub score. To suppress this effect and on the other side reinforce the influence of blogs with focused targets, we normalize A by out-degrees of blogs and then compute hub scores for blogs as LR.

Our splog detector combines these new features (TCR, TSR, LR) with traditional content features into a large feature vector. We then use standard machine learning techniques (SVM classifier with a radial basis function kernel) to classify each blog into two classes: splog or normal blog.

5. DATA-PREPROCESSING AND GROUND TRUTH DEFINITION

We have made significant efforts to pre-process the TREC-Blog dataset and to establish ground truth for training and testing. Our major contributions are summarized as follows:

1. **Pre-processing:** The TREC-Blog 2006 dataset is a crawl of 100,649 feeds collected over 11 weeks, from Dec. 6, 2005 to Feb. 21, 2006, totaling 77 days. After removing duplicate feeds and feeds without homepage or permalinks, we have about 43.6K unique blogs. We focus our analysis on this subset of blogs having homepage and at least one entry.

2. **Annotation tool:** We have developed a user-friendly interface (Figure 7) for annotators to label the TREC-Blog dataset. The detailed description of the tool is available at our webpage[1]. Essentially, in the interface, the content of the blogs and their contents are fetched from the database and presented to the annotator.

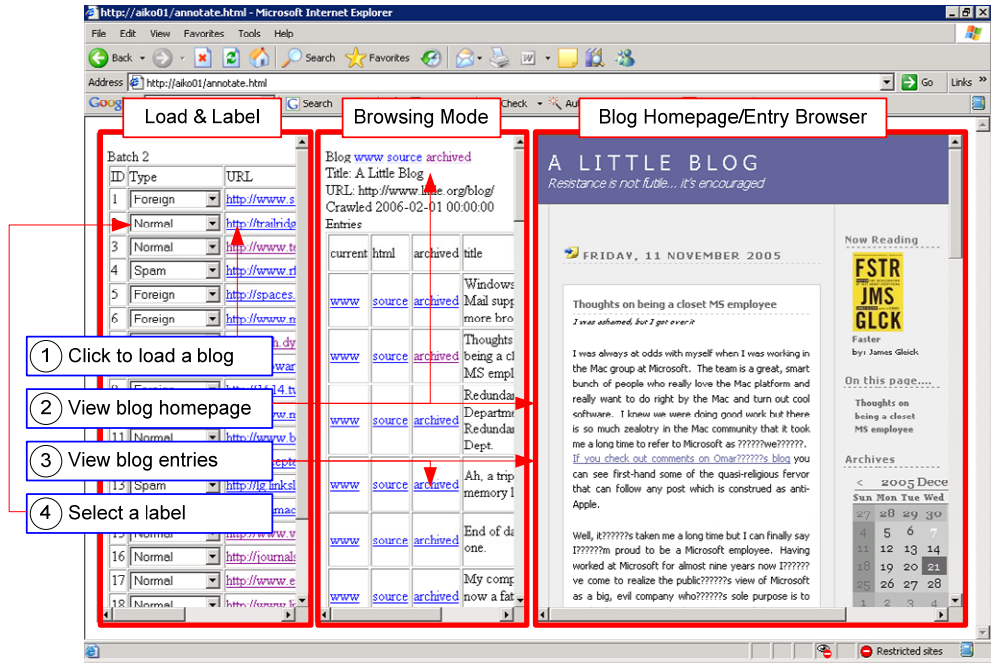

Figure 7: Splog Annotation Tool to view and label blogs.

Through the interface as shown in Figure 7, an annotator can browse the blog homepage and entries that have been downloaded in the TREC-Blog dataset, or visit the blog site directly online, in order to assign one of the following five labels: (N) Normal, (S) Splog, (B) Borderline, (U) Undecided, and (F) Foreign Language.

3. **Disagreement among annotators:** We performed a pilot study to investigate how different annotators identify splogs. We presented a set of 60 blogs to a group of 6 annotators, asking them to assign each blog to one of the five labels. One interesting result is that the annotators have agreement on normal blogs but have varying opinions on splogs (S/B/U), which suggests that splog detection is not trivial even for humans. We plan to conduct further intensive user studies.

4. **Ground truth:** As of August 24, 2006, we have labeled 9240 blogs by using our annotation tool. The 9240 blogs are selected using random sampling as well as stratified sampling methods. Among these 9240 blogs, 7905 are labeled as normal blogs, 525 are labeled as splogs, and the rest are borderline/undecided/foreign. The annotated splog percentage is lower than what has been reported because (1) some known splogs are pre-filtered from the TREC dataset, and (2) we have selected to examine the 43.6K subset of blogs that have both homepages and entries downloaded.

Using the annotation tool to generate ground truth, we built a baseline splog detector and our detector.

[1] http://www.public.asu.edu/~ylin56/project/splog_detection/

6. EXPERIMENTAL RESULTS

In this section, we present the annotation results. We manually labeled 7905 normal blogs and 525 splogs. We decided to create a symmetric set for evaluation containing 525 splogs and 525 normal blogs.

6.1 Offline (Traditional)

In the traditional offline task, we used all the 1050 samples for evaluation. We used a five fold cross-validation technique to evaluate the performance of the splog detector. The results show that the proposed classifier and the new temporal and structural features work well together. In Table 2, we summarize the results of the offline detection. The table shows the comparison between content features of different dimensionality against the combination of the same feature with the regularity features (TCR, TSR, LR). We use four measures – AUC (area under the ROC curve, also ref. Figure 8), accuracy, precision and recall. The results indicate that the proposed feature set combines well with the traditional content based features, however the largest gains occur when the dimensionality of the content features is low.

Table 2: The table shows a comparison of the baseline content scheme against the combination of baseline (designated as base-n, where n is the dimension of the baseline feature) with temporal and link-structure features (designated as R). The table indicates that the improvement due to the non-baseline features is smaller with increase in the number of dimensions to the baseline features.

Feature	AUC	accuracy	precision	recall
base-253	0.966	0.915	0.923	0.907
R+base-253	0.974	0.919	0.918	0.920
base-127	0.957	0.893	0.899	0.886
R+base-127	0.968	0.925	0.931	0.918
base-64	0.938	0.874	0.885	0.861
R+base-64	0.948	0.908	0.918	0.895
base-32	0.895	0.834	0.837	0.831
R+base-32	0.921	0.870	0.883	0.851
R	0.814	0.696	0.860	0.469

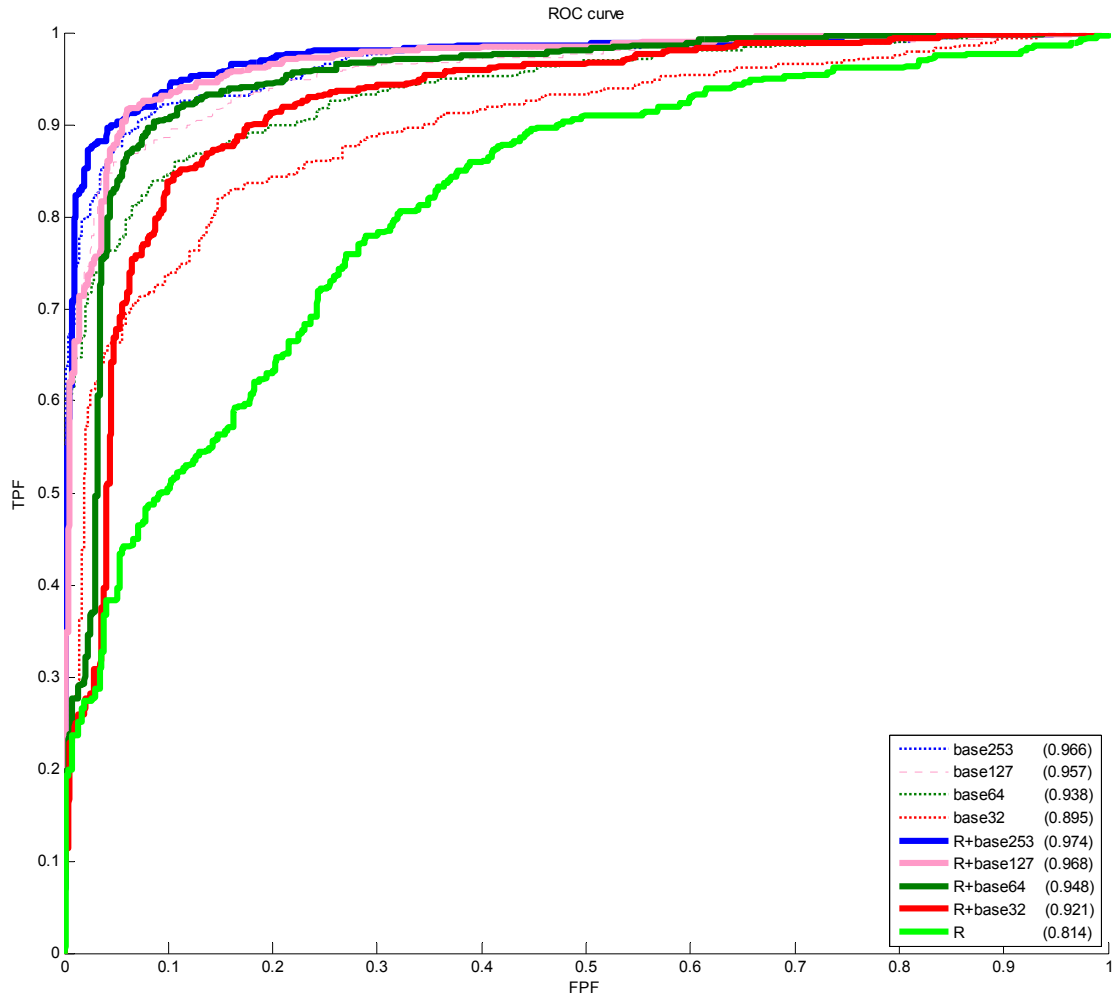

Figure 8: The plot of the ROC curves for different values of the baseline feature size, base-n + R as well as the ROC curve due to the new features alone.

6.2 Online (Proposed)

In the online evaluation framework, we are interested in the rate understanding the temporal effects in the splog classifier performance. In order to do this we create a training set B_0 and testing sets $B_1 - B_7$. In the TREC dataset the crawler discovers new blogs every day, but refreshes the feed only every week. Due to this refresh anomaly, we test the data "weekly" as there is no intermediate download available for the blogs. The training sets $B_1 - B_7$ refer to the blogs that were discovered on day i where $1 \leq i \leq 7$. We use 400 blogs for training (B_0), and 360 blogs for testing (B_1-B_7). The remaining 290 blogs were discovered after the 1st week and are not part of the testing set. We now introduce some notation for clarity:

- $t_1, t_2, \ldots, t_n$: testing period, week 1, 2, …, 7.
- $T_r(t)$ denotes the training set blog B_0 data downloaded until time t. $T_r(1)$ would denote the data for the blogs corresponding the end of week 1.
- $T_e(t)$: denotes the training set blog (B_1, B_2, …, B_7 which are discovered in week 1) data downloaded until time t. Note that $T_r \cap T_e = \phi$.
- $C(T_r(t))$ or C_t: a classifier trained on $T_r(t)$. $C_{t,F}$ denotes the classifier is trained using feature set F. For example $C_{1,\text{base32+R}}$ denotes the classifier C_1 using feature set base32+R and where R = <TCR, TSR, LR>.

- $P(T_r(t_1), T_e(t_2))$ or $P(t_1, t_2)$: the performance of classifier trained on $T_r(t_1)$, i.e. C_1 and tested on $T_e(t_2)$, or $P(t_1, t_2)$. For example P(1, 3) denotes the performance of classifier C_1 trained on the 1st week training set and tested on the 3rd week testing set.

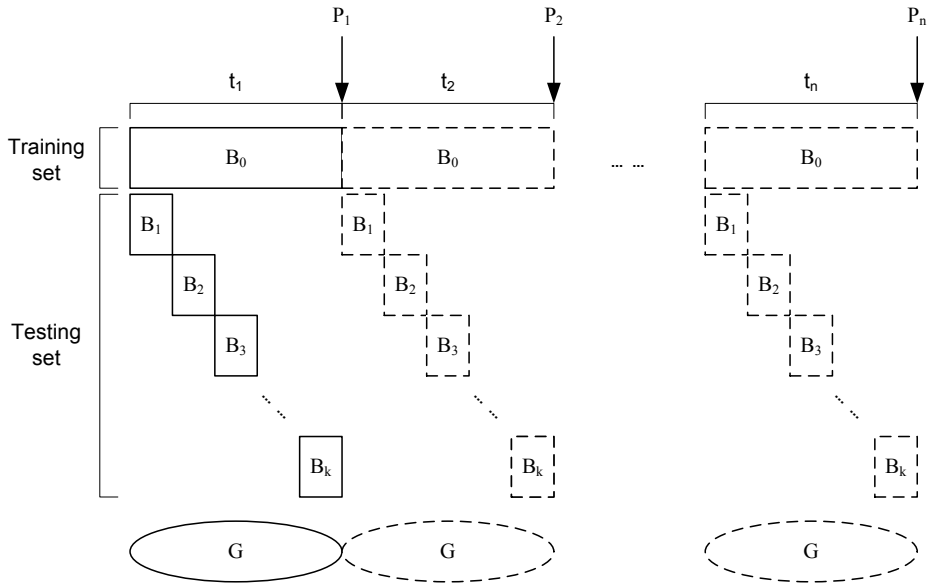

Figure 9: The figure illustrates the online testing framework. The training set B_0 is partitioned weekly, as are the testing sets B_1-B_7. At the end of each week, the splog detector is re-trained to include the latest week's labeled data and is then tested on blogs B_1-B_7. The test blogs have additional downloaded entries for the latest week as well. The symbols P_i refer to testing at the end of the i^{th} week.

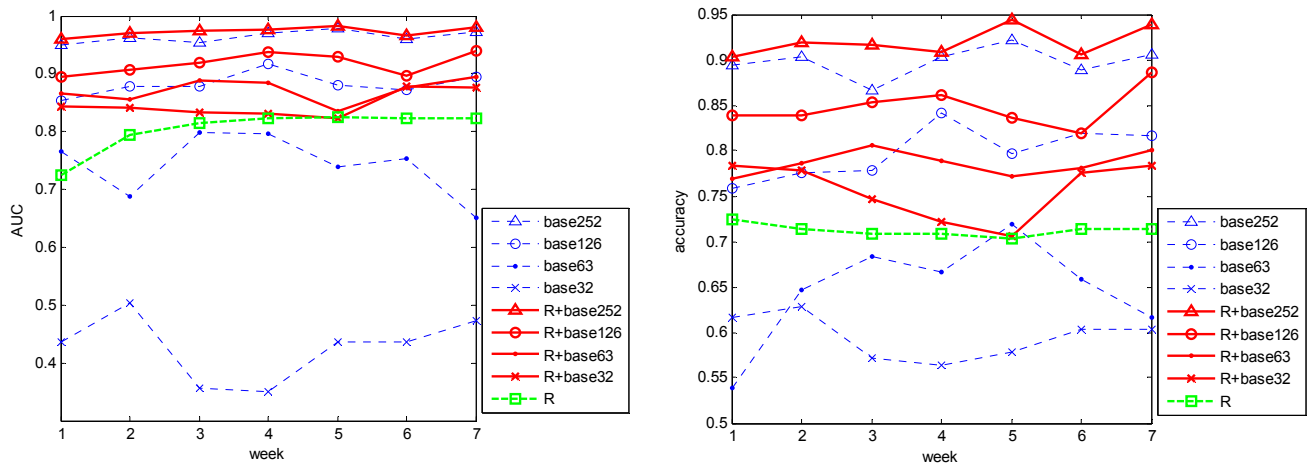

Figure 10: The figures for the online experiments show impact of adding the structural features in the online evaluation. In the absence of content data, the regularity features provide a significant boost to the both the AUC and the accuracy results.

The results for the online evaluation scheme are shown in Figure 10. They show the plot of $P(i,i)$ against the weekly index i, with the metrics AUC (area under the ROC curve) and the accuracy metrics. The plot shows the result of testing the classifier on the data (testing sets B_1-B_7) collected after the i^{th} week, with the classifier being

retrained on B_0, with additional data for the set B_0. They indicate that the utility of adding the structural regularity features in on-line detection. In the absence of enough content, these features play a critical role in discriminating between splogs and normal blogs.

It is striking to compare the results as shown in Figure 10 against the results for offline testing as shown in Table 2 and Figure 8. They indicate that the information provided by the structural information is complementary to the information in the content – this is evidenced by the jump in the AUC and the accuracy values for the corresponding week. Importantly the contribution to the results in the online case due to the regularity features is significant. Not only is there a clear improvement for classifiers $C_{i,\ base\text{-}n+R}$ over the corresponding classifiers $C_{i,\ base\text{-}n}$ for all weeks, but also the difference is high for the low dimensional features.

The explanation for the significant difference is as follows. In online tests when in the first week there is not enough content to train the classifier based on content analysis alone, and hence the regularity features become very important. Over time, the training data available to the classifier increases (i.e. the blogs in the training set B_0 have more entries over time) hence making the decision surface more stable. Additionally, we note that the content features that we have extracted are statistical in nature. Hence the feature vector corresponding to the content of *both* the training data and test data begin to stabilize only after a sufficient number of posts in each blog. Note also that over time the combination of both baseline and regularity features shows less fluctuation than the baseline features alone (ref. Figure 10).

7. CONCLUDING REMARKS

In this paper we analyze the splog detection problem as an important open task for TREC-Blog track. A splog is significantly different from web spam, and thus new online detection tasks are identified. The new task measures how quickly a detector can identify splogs – this is important as temporal dynamics are key distinguishing features of a blog from traditional web pages. We identify new features based on temporal content and structural regularity as well as link regularity. We also provide a set of ground truth labeled through our annotation tool. Our experimental results on both the offline and the online tasks are excellent and validate the importance of the both the proposed online task as well as the addition of regularity based features to traditional content features in such tasks.

8. REFERENCES

[1] *Wikipedia, Spam blog* http://en.wikipedia.org/wiki/Splog.

[2] Z. GYÖNGYI, P. BERKHIN, HECTOR GARCIA-MOLINA and J. PEDERSEN (2006). *Link Spam Detection Based on Mass Estimation*, 32nd International Conference on Very Large Data Bases (VLDB), Seoul, Korea.

[3] Z. GYÖNGYI, H. GARCIA-MOLINA and J. PEDERSEN (2004). *Combating web spam with TrustRank*, Proceedings of the 30th International Conference on Very Large Data Bases (VLDB) 2004, Toronto, Canada.

[4] J. M. KLEINBERG (1999). *Authoritative sources in a hyperlinked environment.* J. ACM **46**(5): 604-632.

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

[6] P. KOLARI, A. JAVA, T. FININ, T. OATES and A. JOSHI (2006). *Detecting Spam Blogs: A Machine Learning Approach*, Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006), July 2006, Boston, MA.

[7] A. NTOULAS, M. NAJORK, M. MANASSE and D. FETTERLY (2006). *Detecting spam web pages through content analysis*, Proceedings of the 15th International Conference on World Wide Web, May 2006, Edinburgh, Scotland.

[8] UMBRIA (2006) *SPAM in the blogosphere* http://www.umbrialistens.com/files/uploads/umbria_splog.pdf.