Weblog Classification for Fast Splog Filtering: A URL Language Model Segmentation Approach

Franco Salvetti[†⋆]
franco.salvetti@colorado.edu

Nicolas Nicolov[⋆]
nicolas@umbrialistens.com

[†]Dept. of Computer Science, Univ. of Colorado at Boulder, 430 UCB, Boulder, CO 80309-0430
[⋆]Umbria, Inc., 1655 Walnut Str, Boulder, CO 80302

Abstract

This paper shows that in the context of statistical weblog classification for splog filtering based on n-grams of tokens in the URL, further segmenting the URLs beyond the standard punctuation is helpful. Many splog URLs contain phrases in which the words are glued together in order to avoid splog filtering techniques based on punctuation segmentation and unigrams. A technique which segments long tokens into the words forming the phrase is proposed and evaluated. The resulting tokens are used as features for a weblog classifier whose accuracy is similar to that of humans (78% vs. 76%) and reaches 93.3% of precision in identifying splogs with recall of 50.9%.

1 Introduction

The blogosphere, which is a subset of the web and is comprised of personal electronic journals (weblogs) currently encompasses 27.2 million pages and doubles in size every 5.5 months (Technorati, 2006). The information contained in the blogosphere has been proven valuable for applications such as marketing intelligence, trend discovery, and opinion tracking (Hurst, 2005). Unfortunately in the last year the blogosphere has been heavily polluted with spam weblogs (called *splogs*) which are weblogs used for different purposes, including promoting affiliated websites (Wikipedia, 2006). Splogs can skew the results of applications meant to quantitatively analyze the blogosphere. Sophisticated content-based methods or methods based on link analysis (Gyöngyi et al., 2004), while providing effective splog filtering, require extra web crawling and can be slow. While a combination of approaches is necessary to provide adequate splog filtering, similar to (Kan & Thi, 2005), we propose, as a preliminary step in the overall splog filtering, a fast, lightweight and accurate method merely based on the analysis of the URL of the weblog without considering its content.

For quantitative and qualitative analysis of the content of the blogosphere, it is acceptable to eliminate a small fraction of good data from analysis as long as the remainder of the data is splog-free. This elimination should be kept to a minimum to preserve counts needed for reliable analysis. When using an ensemble of methods for comprehensive splog filtering it is acceptable for pre-filtering approaches to lower recall in order to improve precision allowing more expensive techniques to be applied on a smaller set of weblogs. The proposed method reaches 93.3% of precision in classifying a weblog in terms of `spam` or `good` if 49.1% of the data are left aside (labeled as `unknown`). If all data needs to be classified our method achieves 78% accuracy which is comparable to the average accuracy of humans (76%) on the same classification task.

Sploggers, in creating splogs, aim to increase the traffic to specific websites. To do so, they frequently communicate a concept (e.g., a service or a product) through a short, sometimes non-grammatical phrase embedded in the URL of the weblog (e.g., `http://adult-video-mpegs.blogspot.com`). We want to build a statistical classifier which leverages the language used in these descriptive URLs in order to classify weblogs as `spam` or `good`. We built an initial language model-based classifier on the tokens of the URLs after tokenizing on punctuation (`.`, `-`,

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 137–140,
New York, June 2006.

_, /, ?, =, etc.). We ran the system and got an accuracy of 72.2% which is close to the accuracy of humans—76% (the baseline is 50% as the training data is balanced). When we did error analysis on the misclassified examples we observed that many of the mistakes were on URLs that contain words glued together as one token (e.g., `dailyfreeipod`). Had the words in these tokens been segmented the initial system would have classified the URL correctly. We, thus, turned our attention to additional segmenting of the URLs beyond just punctuation and using this intra-token segmentation in the classification.

Training a segmenter on standard available text collections (e.g., PTB or BNC) did not seem the way to procede because the lexical items used and the sequence in which they appear differ from the usage in the URLs. Given that we are interested in unsupervised lightweight approaches for URL segmentation, one possibility is to use the URLs themselves after segmenting on punctuation and to try to learn the segmenting (the majority of URLs are naturally segmented using punctuation as we shall see later). We trained a segmenter on the tokens in the URLs, unfortunately this method did not provide sufficient improvement over the system which uses tokenization on punctuation. We hypothesized that the content of the splog pages corresponding to the splog URLs could be used as a corpus to learn the segmentation. We crawled 20K weblogs corresponding to the 20K URLs labeled as `spam` and `good` in the training set, converted them to text, tokenized and used the token sequences as training data for the segmenter. This led to a statistically significant improvement of 5.8% of the accuracy of the splog filter.

2 Engineering of splogs

Frequently sploggers indicate the semantic content of the weblogs using descriptive phrases—often noun groups (non-recursive noun phrases) like `adult-video-mpegs`. There are different varieties of splogs: commercial products (especially electronics), vacations, mortgages, and adult-related.

Users don't want to see splogs in their results and marketing intelligence applications are affected when data contains splogs. Existing approaches to splog filtering employ statistical classifiers (e.g., SVMs) trained on the tokens in a URL after tokenization on punctuation (Kolari et al., 2006). To avoid being identified as a splog by such systems one of the creative techniques that sploggers use is to glue words together into longer tokens for which there will not be statistical information (e.g., `businessopportunitymoneyworkathome` is unlikely to appear in the training data while `business`, `opportunity`, `money`, `work`, `at` and `home` are likely to have been seen in training). Another approach to dealing with splogs is having a list of splog websites (SURBL, 2006). Such an approach based on blacklists is now less effective because bloghosts provide tools which can be used for the automatic creation of a large quantity of splogs.

3 Splog filtering

The weblog classifier uses a segmenter which splits the URL in tokens and then the token sequence is used for supervised learning and classification.

3.1 URL segmentation

The segmenter first tokenizes the URLs on punctuation symbols. Then the current URL tokens are examined for further possible segmentation. The segmenter uses a sliding window of n (e.g., 6) characters. Going from left to right in a greedy fashion the segmenter decides whether to split after the current third character. Figure 1 illustrates the processing of `www.dietthatworks.com` when considering the token `dietthatworks`. The character '∘' indicates that the left and right tri-grams are kept together while '•' indicates a point where the segmenter decides a break should occur. The segmentation decisions are

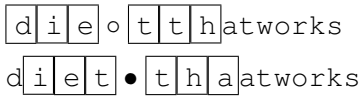

Figure 1: Workings of the segmenter

based on counts collected during training. For example, during the segmentation of `dietthatworks` in the case of iet•tha we essentially consider how many times we have seen in the training data the 6-gram '`iettha`' vs. '`iet_tha`'. Certain characters (e.g., digits) are generalized both during training and segmentation.

3.2 Classification

For the weblog classification a simple Naïve Bayes classifier is used. Given a token sequence $T = \langle t_1, \ldots, t_n \rangle$, representing the segmented URL, the class $\hat{c} \in C = \{\texttt{spam}, \texttt{good}\}$ is decided as:

$$\begin{aligned}\hat{c} &= \arg\max_{c \in C} P(c|T) = \arg\max_{c \in C} \frac{P(c) \cdot P(T|c)}{P(T)} \\ &= \arg\max_{c \in C} P(c) \cdot P(T|c) \\ &= \arg\max_{c \in C} P(c) \cdot \prod_{i=1}^{n} P(t_i|c)\end{aligned}$$

In the last step we made the conditional independence assumption. For calculating $P(t_i|c)$ we use Laplace (add one) smoothing (Jurafsky & Martin, 2000). We have also explored classification via simple voting techniques such as:

$$a = sgn \sum_{i=1}^{n} sgn\left(P(t_i|\texttt{spam}) - P(t_i|\texttt{good})\right)$$

$$\hat{c} = \begin{cases} \texttt{spam}, & if \quad a = 1 \\ \texttt{good}, & otherwise \end{cases}$$

Because we are interested in having control over the precision/recall of the classifier we introduce a score meant to be used for deciding whether to label a URL as `unknown`.

$$score(T) = \left| \frac{P(\texttt{spam}|T) - P(\texttt{good}|T)}{P(\texttt{spam}|T) + P(\texttt{good}|T)} \right|$$

If $score(T)$ exceeds a certain threshold τ we label T as `spam` or `good` using the greater probability of $P(\texttt{spam}|T)$ or $P(\texttt{good}|T)$. To control the presicion of the classifier we can tune τ. For instance, when we set $\tau = 0.75$ we achieve 93.3% of precision which implied a recall of 50.9%. An alternate commonly used technique to compute a score is to look at the log likelihood ratio.

4 Experiments and results

First we discuss the segmenter. 10,000 `spam` and 10,000 `good` weblog URLs and their corresponding HTML pages were used for the experiments. The 20,000 weblog HTML pages are used to induce the segmenter. The first experiment was aimed at finding how common extra segmentation beyond punctuation is as a phenomenon. The segmenter was run on the actual training URLs. The number of URLs that are additionally segmented besides the segmentation on punctuation are reported in Table 1.

# of splits	# `spam` URLs	# `good` URLs
1	2,235	2,274
2	868	459
3	223	46
4	77	7
5	2	1
6	4	1
8	3	–
Total	3,412	2,788

Table 1: Number of extra segmentations in a URL

The multiple segmentations need not all occur on the same token in the URL after initial segmentation on punctuations.

The segmenter was then evaluated on a separate test set of 1,000 URLs for which the ground truth for the segmentation was marked. The results are in Table 2. The evaluation is only on segmentation events and does not include tokenization decisions around punctuation.

Precision	Recall	F-measure
84.31	48.84	61.85

Table 2: Performance of the segmenter

Figure 2 shows long tokens which are correctly split. The weblog classifier was then run on the test set. The results are shown in Table 3.

```
cash • for • your • house
unlimitted • pet • supllies
jim • and • body • fat
weight • loss • product • info
kick • the • boy • and • run
bringing • back • the • past
food • for • your • speakers
```

Figure 2: Correct segmentations

accuracy	78%
prec. spam	82%
rec. spam	71%
f-meas spam	76%
prec. good	74%
rec. good	84%
f-meas good	79%

Table 3: Classification results

The performance of humans on this task was also evaluated. Eight individuals performed the splog identification just looking at the unsegmented URLs. The results for the human annotators are given in Table 4. The average accuracy of the humans (76%) is similar to that of the system (78%).

	Mean	σ
accuracy	76%	6.71
prec. spam	83%	7.57
rec. spam	65%	6.35
f-meas spam	73%	7.57
prec. good	71%	6.35
rec. good	87%	6.39
f-meas good	78%	6.08

Table 4: Results for the human annotators

From an information retrieval perspective if only 50.9% of the URLs are retrieved (labelled as either `spam` or `good` and the rest are labelled as `unknown`) then of the `spam`/`good` decisions 93.3% are correct. This is relevant for cases where a URL splog filter is in cascade followed by, for example, a content-based one.

5 Discussion

The system performs better with the intra-token segmentation because the system is forced to guess unseen events on fewer occasions. For instance given the input URL `www.ipodipodipod.com` in the system which segments solely on punctuation both the `spam` and the `good` model will have to guess the probability of `ipodipodipod` and the results depend merely on the smoothing technique.

Even if we reached the average accuracy of humans we expect to be able to improve the system further as the maximum accuracy among the human annotators is 90%. Among the errors of the segmenter the most common are related to plural nouns (‘`girl•s`’ vs. ‘`girls`’) and past tense of verbs (‘`dedicate•d`’ vs. ‘`dedicated`’) .
The proposed approach has ramifications for splog filtering systems that want to consider the outward links from a weblog.

6 Conclusions

We have presented a technique for determining whether a weblog is splog based merely on alalyzing its URL. We proposed an approach where we initially segment the URL in words and then do the classification. The technique is simple, yet very effective—our system reaches an accuracy of 78% (while humans perform at 76%) and 93.3% of precision in classifying a weblog with recall of 50.9%.

Acknowledgements. We wish to thank Ted Kremer, Howard Kaushansky, Ash Beits, Allen Bennett, Susanne Costello, Hillary Gustave, Glenn Meuth, Micahel Sevilla and Ron Woodward for help with the experiments and comments on an earlier draft.

References

Gyöngyi, Zoltan, Hector Garcia-Molina & Jan Pedersen. 2004. “Combating Web Spam with TrustRank”. *Proceedings of the 30th International Conference on Very Large Data Bases* (*VLDB*).

Matthew Hurst. 2005. “Deriving Marketing Intelligence from Online Discussion”. *11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining* (*KDD05*), 419-428. Chicago, Illinois, USA.

Jurafsky, D. & J.H. Martin. 2000. *Speech and Language Processing*. Upper Saddle River, NJ: Prentice Hall.

Min-Yen Kan & Hoang Oanh Nguyen Thi. 2005. “Fast Webpage Classification Using URL Features”. *14th ACM international conference on Information and Knowledge Management*, 325-326.

Kolari, Pranam, Tim Finin & Anupam Joshi. 2006. “SVMs for the Blogosphere: Blog Identification and Splog Detection”. *AAAI Symposium on Computational Approaches to Analyzing Weblogs*, 92-99. Stanford.

SURBL. 2006. *SURBL — Spam URI Realtime Blocklists*, `http://www.surbl.org`

Technorati. 2006. *State of the Blogosphere, February 2006 Part 1: On Blogosphere Growth*, `technorati.com/weblog/2006/02/81.html`

Wikipedia. 2006. *Splog (Spam blog)*, `http://en.wikipedia.org/wiki/Splog`