BlogVox: Separating Blog Wheat from Blog Chaff*

Akshay Java, Pranam Kolari, Tim Finin, Anupam Joshi and Justin Martineau
University of Maryland, Baltimore County
{aks1, kolari1, finin, joshi, jm1}@cs.umbc.edu

James Mayfield
Johns Hopkins University Applied Physics Laboratory
james.mayfield@jhuapl.edu

Abstract

Blog posts are often informally written, poorly structured, rife with spelling and grammatical errors, and feature non-traditional content. These characteristics make them difficult to process with standard language analysis tools. Performing linguistic analysis on blogs is plagued by two additional problems: (i) the presence of spam blogs and spam comments and (ii) extraneous non-content including blog-rolls, link-rolls, advertisements and sidebars. We describe techniques designed to eliminate noisy blog data developed as part of the BlogVox system - a blog analytics engine we developed for the 2006 TREC Blog Track. The findings in this paper underscore the importance of removing spurious content from blog collections.

1 Introduction

Traditional natural language text processing systems are usually applied to tasks with high quality text. In practical environments, including online chat, SMS message, email messages, wiki pages and blog posts, NLP systems are less effective. Blog posts contain noisy, ungrammatical and poorly structured text. In addition blog processing system must address two key issues: (i) detecting and eliminating spam blogs and spam comments and (ii) eliminating noisy text from link-rolls and blog-rolls.

Recently, Spam blogs, or splogs have received significant attention, and techniques are being developed to detect them. Kolari, et al. [Kolari *et al.*, 2006a] have recently discussed the use of machine learning techniques to identify blog pages (as opposed to other online resources) and to categorize them as authentic blogs or spam blogs (splogs). [Kolari *et al.*, 2006b] extends this study by analyzing a special collection of blog posts released for the Third Annual Workshop on the Weblogging Ecosystem held at the 2006 World Wide Web Conference. Their findings on spam blogs confirms the seriousness of the problem.

The very nature of blogging platforms poses an important challenge. Blog owners promote friends, products, services and often their own posts by featuring them on blog-rolls and link-rolls that are often replicated across the entire blog. Spam blogs, spam comments and extraneous content indexed by a blog processing system put an unnecessary strain on the computational infrastructure, and ultimately skew results of blog analysis.

BlogVox [Java *et al.*, 2006] is a prototype system built to to perform "opinion extraction" from blog posts as part of the 2006 TREC blog track [1], a yearly information retrieval competition. The goal of this competition is to find opinionated posts about a topic specified by a query string (e.g., "march of the penguins"). Retrieval is to be done over a special dataset exceeding three million posts collected from about 80 thousands blogs. We have learned that removing splogs, and eliminating spurious content from the posts, e.g., removing blogrolls, advertisements, sidebars, headers and footers, navigation panels, etc improve results significantly.

In the remainder of this paper we describe the TREC Blog Track in more detail and the BlogVox system we implemented to perform the task. Section 2 summarizes how we detect and eliminate splogs, and section 3 describes some new techniques we developed for recognizing and differentiating important post content from non-content. Section 5 evaluates the importance of data cleaning for the TREC task. Section 6 concludes the paper and describes ongoing work to extend BlogVox.

2 Identifying and Removing Spam

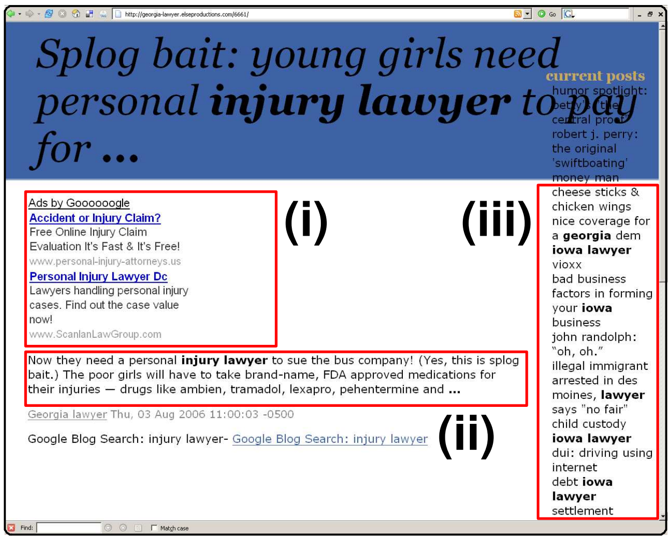
Two kinds of spam are common in the blogosphere (i) spam blogs or splogs, and (ii) spam comments. We first discuss spam blogs, approaches on detecting them, and how they were employed for BlogVox.

2.1 Problem of Spam Blogs

Splogs are blogs created for the sole purpose of hosting ads, promoting affiliate sites (including themselves) and getting new pages indexed. Content in splogs is often auto-generated and/or plagiarized, such software sells for less than 100 dollars and now inundates the blogosphere both at ping servers (around 75% [Kolari, 2005]) that monitor blog updates, and at blog search engines (around 20%, [Kolari *et al.*, 2006d]) that

*Partial support was provided by an IBM Fellowship and by NSF awards ITR-IIS-0326460 and ITR-IDM-0219649.

[1]http://trec.nist.gov/tracks.html


http://georgia-lawyer.elseproductions.com/6661/

Splog bait: young girls need personal **injury lawyer** to pay for ...

current posts
humor spotlight: betty's 'the central proof' robert j. perry: the original 'swiftboating' money man cheese sticks & chicken wings nice coverage for a **georgia** dem **iowa lawyer** vioxx bad business factors in forming your **iowa** business john randolph: "oh, oh." illegal immigrant arrested in des moines, **lawyer** says "no fair" child custody **iowa lawyer** dui: driving using internet debt **iowa lawyer** settlement

Ads by Gooooooogle
Accident or Injury Claim?
Free Online Injury Claim Evaluation It's Fast & It's Free!
www.personal-injury-attorneys.us
Personal Injury Lawyer Dc
Lawyers handling personal injury cases. Find out the case value now!
www.ScanlanLawGroup.com

Now they need a personal **injury lawyer** to sue the bus company! (Yes, this is splog bait.) The poor girls will have to take brand-name, FDA approved medications for their injuries — drugs like ambien, tramadol, lexapro, pehentermine and **...**

Georgia lawyer Thu, 03 Aug 2006 11:00:03 -0500

Google Blog Search: injury lawyer- Google Blog Search: injury lawyer

Figure 1: A typical splog, plagiarizes content (ii), promotes other spam pages (iii), and (i) hosts high paying contextual advertisements

index them. Spam comments pose an equally serious problem, where authentic blog posts feature auto-generated comments that target ranking algorithms of popular search engines. A popular spam comment filter [2] estimates the amount of spam detected to be around 93%.

Figure 1 shows a splog post indexed by a popular blog search engine. As depicted, it features content plagiarized from other blogs (ii), displays ads in high paying contexts (i), and hosts hyperlinks (iii) that create link farms. Scores of such pages now pollute the blogosphere, with new ones springing up every moment. Splogs continue to be a problem for web search engines, however they present a new set of challenges for blog analytics. This paper stresses the latter.

2.2 Detecting Splogs

Splogs are well understood to be a specific instance of the more general spam web-pages [Gyöngyi and Garcia-Molina, 2005]. Though offline graph based mechanisms like TrustRank [Gyöngyi *et al.*, 2004] are sufficiently effective for the Web, the blogosphere demands new techniques. The quality of blog analytics engines is judged not just by content coverage, but also by their ability to index and analyze recent (non-spam) posts. This requires that fast online splog detection/filtering [Kolari *et al.*, 2006a][Salvetti and Nicolov, 2006] be used prior to indexing new content.

We employ statistical models to detecting splogs as described by [Kolari *et al.*, 2006d], based on supervised machine learning techniques, using content local to a page, enabling fast splog detection. These models are based solely on blog home-pages, and are based on a training set of 700 blogs and 700 splogs. Statistical models based on local blog features perform well on spam blog detection. See Table 1. The bag-of-words based features slightly outperforms bag-of-outgoingurls (URL's tokenized on '/') and bag-of-outgoinganchors. Additional results using link based features are slightly lower that local features, but effective nonetheless. Interested readers are referred to [Kolari *et al.*, 2006d] for further details. Therefore, BlogVox used only local features to detect splogs.

[2]http://akismet.com

Feature	*Precision*	*Recall*	*F1*
words	.887	.864	.875
urls	.804	.827	.815
anchors	.854	.807	.830

Table 1: *SVMs with 19000 word features and 10000 each of URL and anchor text features ranked using Mutual Information.*

2.3 Comment spam

Comment spam occurs when a user posts spam inside a blog comment. Comment spam is typically managed by individual bloggers, through moderating comments and/or using comment spam detection tools (e.g. Akismet) on blogging platforms. Comment spam and splogs share a common purpose. They enable indexing new web pages, and promoting their page rank, with each such page selling online merchandise or hosting context specific advertisements. Detecting and eliminating comment spam [Mishne *et al.*, 2005] depends largely on the quality of identifying comments on a blog post, part of which is addressed in the next section.

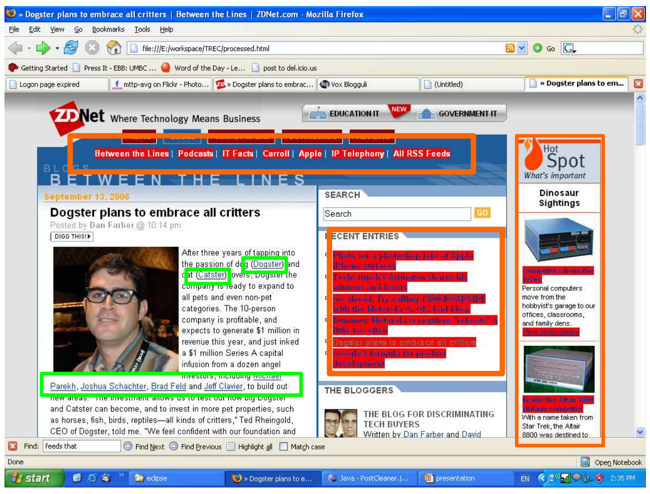
3 Identifying Post Content

Most extraneous features in blog post are links. We describe two techniques to automatically classify the links into content-links and extra-links. Content links are part of either the title or the text of the post. Extra links are not directly related to the post, but provide additional information such as: navigational links, recent entries, advertisements, and blog rolls. Differentiating the blog content from its chaff is further complicated by blog hosting services using different templates and formats. Additionally, users host their own blogs and sometimes customize existing templates to suit their needs.

Web page cleaning techniques work by detecting common structural elements from the HTML Document Object Model (DOM) [Yi and Liu, 2003; Yi *et al.*, 2003]. By mining for both frequently repeated presentational components and content in web pages, a site style tree is constructed. This tree structure can be used for data cleaning and improved feature weighting. Finding repeated structural components requires sampling many web pages from a domain. Although blogs from the same domain can share similar structural components, they can differ due to blogger customization. Our proposed technique does not require sampling and works independently on each blog permalink.

Instead of mining, we used a simple general heuristic. Intuitively extraneous links tend to be tightly grouped containing relatively small amounts of text. Note that a typical blog post has a complex DOM tree with many parts, only one of which is the content of interest in most applications.

After creating the DOM tree we traverse it attempting to eliminate any extraneous links and their corresponding an-



September 13, 2006

Dogster plans to embrace all critters

Posted by Dan Farber @ 10:14 pm

After three years of tapping into the passion of dog (Dogster) and cat (Catster) lovers, Dogster the company is ready to expand to all pets and even non-pet categories. The 10-person company is profitable, and expects to generate $1 million in revenue this year, and just inked a $1 million Series A capital infusion from a dozen angel investors, including Michael Parekh, Joshua Schachter, Brad Feld and Jeff Clavier, to build out new areas. "The investment allows us to test out how big Dogster and Catster can become, and to invest in more pet properties, such as horses, fish, birds, reptiles—all kinds of critters," Ted Rheingold, CEO of Dogster, told me. "We feel confident with our foundation and

- Photo for a photoshop job of Apple iPhone surfaces
- TechCrunch's Arrington shares his winners and losers
- Go ahead. Try calling 1-800-ROADSIDE with the Motorola Q. OK, bad idea.
- Bummer: Motorola Q requires "reboots" a little too often
- Dogster plans to embrace all critters
- Google's formula for product development

Computers from the 1970s

Personal computers move from the hobbyist's garage to our offices, classrooms, and family dens.

View photo gallery

Inside the Altair 8800 vintage computer

With a name taken from Star Trek, the Altair 8800 was destined to

THE BLOG FOR DISCRIMINATING TECH BUYERS

Written by Dan Farber and David

Figure 2: A typical blog post containing navigational links, recent posts, advertisements, and post content with additional links in it. Highlighted links are eliminated by the blog post cleaning heuristic.

Algorithm 1 Blog post cleaning heuristic

```
Nodes[] $tags$ = tags in the order of the depth first traversal
of the DOM tree
**for all** $i$ such that $0 \leq i \leq |tags|$ **do**
    $dist$ = nearestLinkTag(tags, $i$);
    **if** $dist \leq \theta_{dist}$ **then**
        eliminate tags[$i$]
    **end if**
**end for**
```

chor text, based upon the preceding and following tags. A link **a** is eliminated if another link **b** within a θ_{dist} tag distance exists such that:

- No title tags (H1, H2...) exist in a θ_{dist} tag window of **a**.
- Average length of the text bearing nodes between **a** and **b** is less than some threshold.
- **b** is the nearest link node to **a**.

The average text ratio between the links, $\alpha_{avgText}$ was heuristically set to 120 characters and a window size, θ_{dist} of 10 tags was chosen. The Algorithm 1 provides a detailed description of this heuristic.

Next we present a machine learning approach to the link classification problem. From a large collection of blog posts, a random sample of 125 posts was selected. A human evaluator judged a subset of links (approximately 400) from these posts. The links were manually tagged as either content-links or extra-links. Each link was associated with a set of features. Table 2 summarizes the main features used. Using this feature set an SVM model was trained [3] to recognize links to be eliminated. The first set of features (1-7) was based on the tag information. The next set of features (8-9) was based on position information and the final set of features (10-13)

[3]http://svmlight.joachims.org/

Procedure 2 int nearestLinkTag(Nodes[] tags, int pos)

```
$minDist$ = $|tags|$
$textNodes$ = 0
$textLength$ = 0
$title$ = false;
**for all** $j$ such that $pos - \theta_{dist} \leq j \leq pos + \theta_{dist}$ **do**
    $node$ = $tags[j]$
    **if** $j = 0 || j = pos || j > (|tags| - 1)$ **then**
        continue
    **end if**
    **if** $node$ instanceOf $TextNode$ **then**
        textNodes++;
        textLength += node.getTextLength();
    **end if**
    $dist$ = $|pos - j|$
    **if** $node$ instanceOf $LinkNode$ && $dist < minDist$
    **then**
        $minDist$ = $dist$
    **end if**
    **if** $node$ instanceOf $TitleNode$ **then**
        $title$ = true
    **end if**
**end for**
$ratio$ = $textLength$ / $textCount$
**if** $ratio > \alpha_{avgText} || title ==$ true **then**
    return $tags.size()$
**end if**
return $minDist$
```

consisted of word-based features. Using features (1-7) yields a precision of 79.4% and recall of 78.39%, using all our features (1-13) yields a precision of 86.25% and recall of 94.31% under 10-fold cross validation.

We compared the original baseline heuristic against human evaluators. The average accuracy for the baseline heuristic is about 83% with a recall of 87%.

4 The TREC Blog Track

In this section we describe a prototype system, BlogVox that was built for the TREC Blog track. We evaluate the data cleaning methods presented in Section 2 and Section 3 in the context of this “opinion extraction” task.

UMBC and JHU/APL collaborated as a team for the 2006 TREC Blog track sponsored by NIST. This track asked participants to implement and evaluate a system to do ”opinion retrieval” from blog posts. Specifically, the task was defined as follows: build a system that will take a query string describing a topic, e.g., “March of the Penguins”, and return a ranked list of blog posts that express an opinion, positive or negative, about the topic. For evaluation, NIST provided a dataset of over three million blogs drawn from about 80 thousand blogs. Participants built and trained their systems to work on this dataset. Contestants do an automatic evaluation by downloading and running, without further modification to their systems, a set of fifty test queries.

Opinion extraction has been studied for mining sentiments and reviews in specific domains such as consumer

ID	Features
1	Previous Node
2	Next Node
3	Parent Node
4	Previous N Tags
5	Next N Tags
6	Sibling Nodes
7	Child Nodes
8	Depth in DOM Tree
9	Char offset from page start
10	links outside the blog?
11	Anchor text words
12	Previous N words
13	Next N words

Table 2: Features used for training an SVM for classifying links as content links and extra links.

products [Dave *et al.*, 2003] or movies [Pang *et al.*, 2002; Gilad Mishne, 2006]. More recently, blogs have become a new medium though which users express sentiments. Opinion extraction has thus become important for understanding consumer biases and is being used as a new tool for market intelligence [Glance *et al.*, 2005] [Nigam and Hurst, 2004][Liu *et al.*, 2005].

For TREC our team developed a novel system based upon the Lucene information retrieval system for the basic retrieval task. Compared to domain-specific opinion extraction, identifying opinionated documents about a randomly chosen topic from a pool of documents that are potentially unrelated to the topic is a much more difficult task. Our goal for this project was to create a system that could dynamically learn topic sensitive sentiment words to better find blog posts expressing an opinion about a specified topic. We use a meta-learning approach and designed an architecture where a set of scorers would each evaluate every relevant document and produce a score representing how opinionated it is. These scores would then be used as a feature vector for an SVM to classify our documents. Following is a description of the BlogVox system which utilized machine learning techniques for both pre-indexing data preparation and post-retrieval ranking for opinionatedness.

4.1 Pre-indexing Processing

The TREC dataset consisted of a set of XML formatted files, each containing blog posts crawled on a given date. The entire collection consisted of over 3.2M posts from 100K feeds [Macdonald and Ounis, 2006]. These posts were parsed and stored separately for convenient indexing, using the HTML parser tool [4]. Non-English blogs were ignored in addition to any page that failed to parse due to encoding issues.

In order to make the challenge realistic NIST explicitly included 17,969 feeds from splogs, contributing to 15.8% of the documents. There were 83,307 distinct homepage URLs present in the collection, of which 81,014 could be processed. The collection contained a total of 3,214,727 permalinks from all these blogs. Our automated splog detection technique identified 13,542 blogs as splogs. This accounts for about 16% of the identified homepages. The total number of permalinks from these splogs is 543,086 or around 16% of the collection. While the actual list of splogs are not available for comparison, the current estimate seem to be close. To prevent the possibility of splogs skewing our results permalinks associated with splogs were not indexed.

We noticed that in order to improve the quality of opinion extraction results, we also need to narrow down on the title and content of the blog post because the scoring functions and Lucene indexing engine can not differentiate between text present in the links and sidebars of the blog post. Thus, a post which has a link to a recent post titled 'Why I love my iPod' would be retrieved as an opinionated post even if the actual post is actually about some other topic.

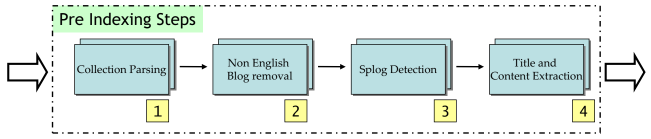


Figure 3: BlogVox text Preparation steps: (i) parsing the TREC corpus (ii) removing non English posts (iii) Eliminating splogs from the collection (iv) removing spurious material from the DOM tree.

4.2 Post-retrieval Processing

After pre-indexing, blog posts are indexed using Lucene, an open-source search engine. Lucene internally constructs an inverted index of the documents by representing each document as a vector of terms. Given a query term, Lucene uses standard Term Frequency (TF) and Inverse Document Frequency (IDF) normalization to compute similarity. In addition, the scoring formula can also be tuned to perform document length normalization and term specific boosting [5]. We used the default parameters while searching the index. However, in order to handle phrasal queries such as "United States of America" we reformulate the original query to boost the value of exact matches or proximity-based matches for the phrase.

Given a TREC query a set of relevant posts are retrieved from the Lucene index and sent to the scorers. As shown in figure 4, a number of heuristics are employed to score the results based on the likelihood that it contains an opinion about the query terms. These scorers work by using both document level and individual sentence level features. Some of the scoring heuristics were supported by a hand-crafted list of 915 generic postive and 2712 negative sentiment words.

The following is a brief description of each scoring function:

Query Word Proximity Scorer finds the average number of sentiment terms occurring in the *vicinity* of the query terms using a window size of 10, 15 or 20 words before and after the query terms. If the query is a phrasal query, the presence

[4]http://htmlparser.sourceforge.net/

[5]http://lucene.apache.org/java/docs/scoring.html

of sentiment terms around the query contributes to a boosted score (approximately twice).

Query Word Count Scorer counts the number of times the query term occurs in the document.

Title Word Scorer checks for the presence of the query terms in the title.

First Occurrence Scorer finds the distance in characters from the start of the blog post to the first query term match.

Context Word Scorer determines contextual terms that can be used to describe the topic or query. We used two external sources to help find the context for the query.

We also experimented with two external sources for deriving context words. The first approach is to use the Google API and obtain matching web documents by sprinkling the query terms with generic positive and negative sentiment words such as "hate", "love", "sux", "annoyed", "great" to slightly bias the documents retrieved towards the sentiment bearing pages. Using only the summary of the pages returned we can create a histogram of terms that are 'about' this topic. The second external context source were keywords in reviews from Amazon product categories.

Lucene Relevance Score was used to find how closely the post matches the query terms.

We also experimented with other scoring functions, such as adjective word count scorer. This scorer used an NLP tool to extract the adjectives around the query terms. However, this tool did not perform well mainly due to the noisy and ungrammatical sentences present in blogs. Once the results were

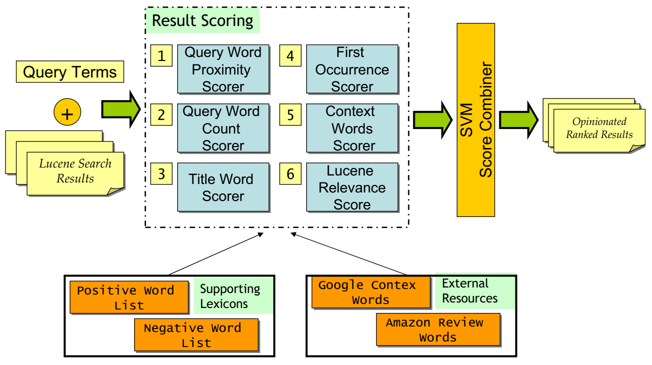


Figure 4: After relevant posts are retrieved, they are scored by various heuristics and an overall measure of opinionatedness computed by a SVM.

scored by these scoring modules, we used a meta-learning approach to combine the scores using SVMs, the details of which are beyond the scope of this paper. A description of the SVM score combiner is available in [Java *et al.*, 2006].

5 Evaluation

The opinion extraction system provides a testbed application for which we evaluate different data cleaning methods. There are three criteria for evaluation: i) improvements in opinion extraction task with and without data cleaning ii) performance evaluation for splog detection iii) performance of the post content identification.

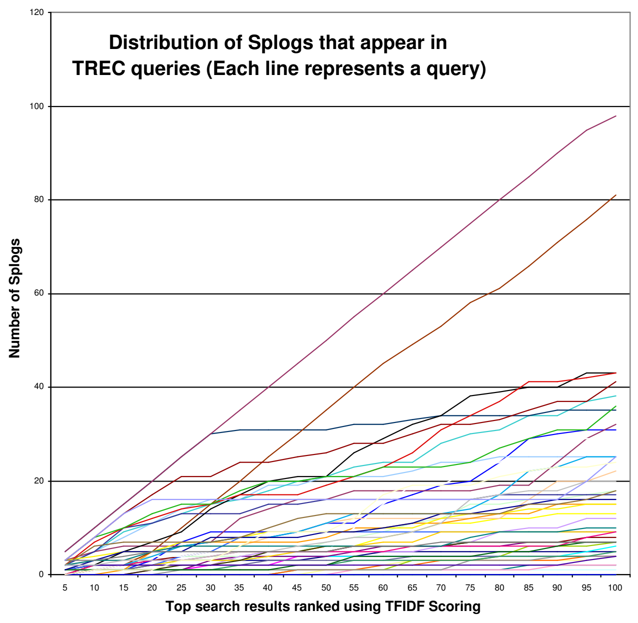


Figure 5: The number of splogs in the top x results for 50 TREC queries. Top splog queries include "cholesterol" and "hybrid cars"

5.1 Splog Detection Evaluation

For now, we evaluate the influence of splogs and post cleaning in the context of search engine retrieval. Given a search query, we would like to estimate the impact splogs have on search result precision. Figure 5 shows the distribution of splogs across the 50 TREC queries. The quantity of splogs present varies across the queries since splogs are query dependent. For example, the topmost spammed query terms were 'cholesterol' and 'hybrid cars'. Such queries attract a target market, which advertiser can exploit.

The description of the TREC data [Macdonald and Ounis, 2006] provides an analysis of the posts from splogs that were added to the collection. Top informative terms include 'insurance', 'weight', 'credit' and such. Figure 6 shows the distribution of splogs identified by our system across such spam terms. In stark contrast from Figure 5 there is a very high percentage of splogs in the top 100 results.

5.2 Post Cleaning Evaluation

In BlogVox data cleaning improved results for opinion extraction. Figure 7 highlights the significance of identifying and removing extraneous content from blog posts. For 50 TREC queries, we fetch the first 500 matches from a Lucene index and used the baseline data cleaning heuristic. Some documents were selected only due to the presence of query terms in sidebars. Sometimes these are links to recent posts containing the query terms, but can often be links to advertisements, reading lists or link rolls, etc. Reducing the impact of sidebar affects on opinion rank through link elimination or feature weighing can improve search results.

Table 3 shows the performance of the baseline heuristic and

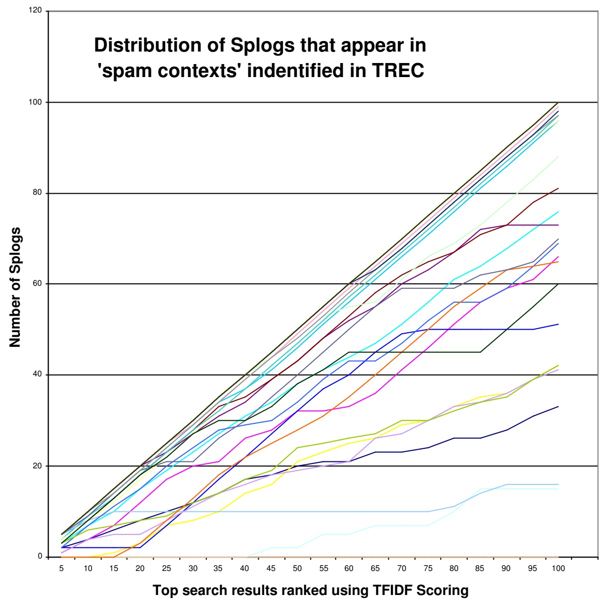


Figure 6: The number of splogs in the top x results of the TREC collection for 28 highly spammed query terms. Top splog queries include 'pregnancy', 'insurance', 'discount'

Method	*Precision*	*Recall*	*F1*
baseline heuristic	0.83	0.87	0.849
svm cleaner (tag features)	0.79	0.78	0.784
svm cleaner (all features)	0.86	0.94	0.898

Table 3: Data cleaning with DOM features on a training set of 400 HTML Links.

the SVM based data cleaner on a hand-tagged set of 400 links. The SVM model outperforms the baseline heuristic. The current data cleaning approach works by making a decision at the individual HTML tag level, we are currently working on automatically identify the DOM subtrees that correspond to the sidebar elements.

5.3 Trec Submissions

Figure 8 shows the results from the TREC submissions for opinion retrieval. Figure 9 shows the results for the topci relevance. The core BlogVox system produces results with two measures. The first is a relevance score ranging from 0.0 to 1.0, which is the value returned by the underlying Lucene query system. The second was a measure of opinionatedness returned by the SVM score combiner. We produced the final score from a weighted average of the two numbers after normalizing them using the standard Z-normalization technique. The Mean Average Precision (MAP) for opinion retrieval was 0.0764 and the R-Prec was around 0.1307. The MAP for topic relevance was about 0.1288 with an R-Prec of 0.1805. These scores were around the median scores across all submissions.

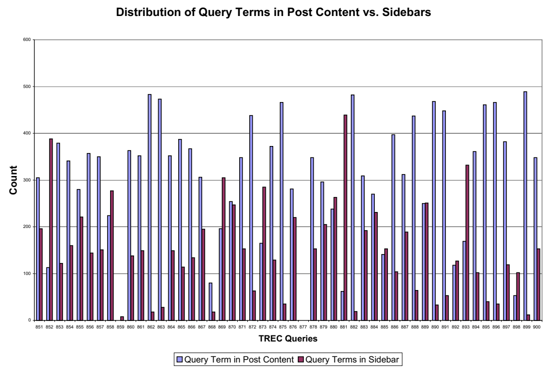


Figure 7: Documents containing query terms in the post title or content vs. exclusively in the sidebars, for 50 TREC queries, using 500 results fetched from the Lucene index.

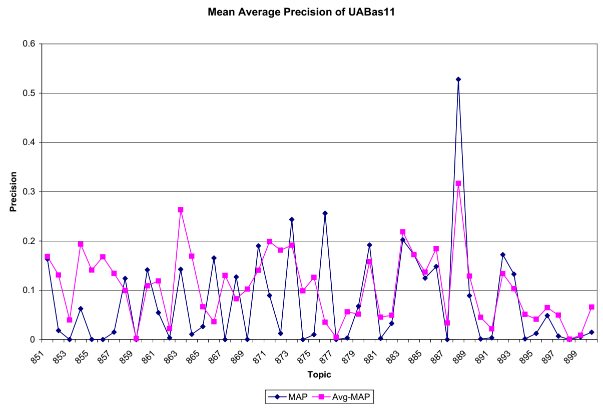


Figure 8: Mean average precision of submission UABas11

6 Conclusion

Much of the content in the Blogosphere is difficult to analyze linguistically because of its informal nature. This is especially true of the personal diary blogs typical to Myspace and LiveJournal but is also true of many other blogs. The analysis challenge is further exacerbated by the presence of spam and large amounts of extraneous material such as advertisements. Our system mitigates the affect of both kinds of "noise" from blog data and its effect on opinion retrieval tasks as specified by the 2006 TREC Blog track.

We are currently expanding the prototype BlogVox system implemented for TREC 2006 to use more sentence level scorers and fewer document level scorers. We are also working on ways to further mitigate both types of noise.

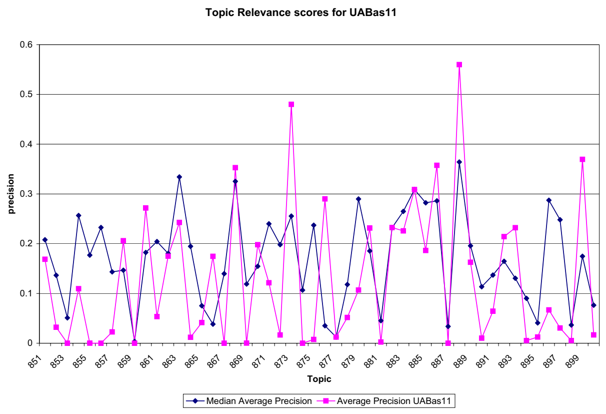


Figure 9: Topic relevance of submission UABas11

References

[Dave *et al.*, 2003] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.

[Gilad Mishne, 2006] Natalie Glance Gilad Mishne. Predicting movie sales from blogger sentiment. In *AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW 2006)*, 2006.

[Glance *et al.*, 2005] Natalie S. Glance, Matthew Hurst, Kamal Nigam, Matthew Siegler, Robert Stockton, and Takashi Tomokiyo. Deriving marketing intelligence from online discussion. In *KDD*, pages 419–428, 2005.

[Gyöngyi and Garcia-Molina, 2005] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[Gyöngyi *et al.*, 2004] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587. Morgan Kaufmann, 2004.

[Hatcher and Gospodnetić, 2004] E. Hatcher and O. Gospodnetić. *Lucene in Action*. Manning Publications Co., 2004.

[Java *et al.*, 2006] Akshay Java, Pranam Kolari, Tim Finin, James Mayfield, Anupam Joshi, and Justin Martineau. The UMBC/JHU blogvox system. In *Proceedings of the Fifteenth Text Retrieval Conference*, November 2006.

[Kolari *et al.*, 2006a] Pranam Kolari, Tim Finin, and Anupam Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*. AAAI Press, March 2006.

[Kolari *et al.*, 2006b] Pranam Kolari, Akshay Java, and Tim Finin. Characterizing the Splogosphere. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*, May 2006.

[Kolari *et al.*, 2006c] Pranam Kolari, Akshay Java, Tim Finin, James Mayfield, Anupam Joshi, and Justin Martineau. Blog Track Open Task: Spam Blog Classification. Technical report, September 2006. TREC 2006 Blog Track.

[Kolari *et al.*, 2006d] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting Spam Blogs: A Machine Learning Approach. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, July 2006.

[Kolari, 2005] Pranam Kolari. Welcome to the splogosphere: 75% of new pings are spings(splogs), 2005. [Online; accessed 22-December-2005; `http://ebiquity.umbc.edu/blogger/?p=429`].

[Liu *et al.*, 2005] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA, 2005. ACM Press.

[Macdonald and Ounis, 2006] Craig Macdonald and Iadh Ounis. The trec blogs06 collection: Creating and analyzing a blog test collection. Technical report, 2006. Department of Computer Science, University of Glasgow Tech Report TR-2006-224.

[Mishne *et al.*, 2005] Gilad Mishne, David Carmel, and Ronny Lempel. Blocking blog spam with language model disagreement. In *AIRWeb '05 - 1st International Workshop on Adversarial Information Retrieval on the Web, at WWW 2005*, 2005.

[Nigam and Hurst, 2004] Kamal Nigam and Matthew Hurst. Towards a robust metric of opinion. In *Exploring Attitude and Affect in Text: Theories and Applications, AAAI-EAAT 2004*, 2004.

[Pang *et al.*, 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP 2002*, 2002.

[Salvetti and Nicolov, 2006] Franco Salvetti and Nicolas Nicolov. 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*, pages 137–140, New York City, USA, June 2006. Association for Computational Linguistics.

[Yi and Liu, 2003] Lan Yi and Bing Liu. Web page cleaning for web mining through feature weighting. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence, IJCAI-03*, 2003.

[Yi *et al.*, 2003] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2003*, 2003.