A Novel Approach for Combating Spamdexing in Web using UCINET and SVM Light Tool

Ms. D. Saraswathi*,Dr. A. Vijaya Kathiravan #,Ms.S. Anita*

******Asst. Professor in Computer Science, K.S.R. College of Arts & Science, Tiruchengode-637209, Namakkal, TN, INDIA.*
# Professor, MCA Dept., Nandha Engineering College, Perundurai, Erode, TN, INDIA

ABSTRACT-Search Engine spam is a web page or a portion of a web page which has been created with the intention of increasing its ranking in search engines. Web spamming refers to actions intended to mislead search engines and give some pages higher ranking than they deserve. Anyone who uses a search engine frequently has most likely encountered a high ranking page that consists of nothing more than a bunch of query keywords. These pages detract both from the user experience and from the quality of the search engine. Search engine spam is a webpage that has been designed to artificially inflating its search engine ranking. Recently this search engine spam has been increased dramatically and creates problem to the search engine and the web surfer. It degrades the search engine's results, occupies more memory and consumes more time for creating indexes, and frustrates the user by giving irrelevant results. Search engines have tried many techniques to filter out these spam pages before they can appear on the query results page. In this paper, various ways of creating spam pages, a collectionof current methods that are being used to detect spam, and a new approach to build a tool for spam detection that uses machine learning as a means for detecting spam. This new approach uses UCINET software and a series of content combined with a Support Vector Machine (SVM) Binary classifier to determine if a given webpage is spam. The link farm can identify based on degree, betweenness and Eigen vector value of link. The spam classifier makes use of the Wordnet word database and SVMLight tool to classify web documents as either spam or not spam. These features are not only related to quantitative data extracted from the Web pages, but also to qualitative properties, mainly of the page links.

Keywords: Search engine, PageRank, Spam, Content Spam, Link Farm, Classification

I. NTRODUCTION

Search Engines consist of three major components: spider, index, and search engine program. The spider or crawler starts with an initial set of URLs called seed URLs, retrieves the Web pages of the seed URLs, and follows the links to other sites from those pages. Keywords found on a Web page are added to the index or catalog of the search engine. The search-engine program finds the relevant pages, from the millions of pages recorded in its index, which match a query and returns them to the user after ranking them in order of relevance. A PageRank is determined for all Web pages in the links database and this PageRank is used to evaluate the relevance of a result. Search Engines are entryways to the web. The objective of a search engine is to provide high quality results by correctly identifying all web pages that are relevant for a query, and presenting the user with the most important of those relevant pages. Relevancy is the search engine's measure of how well a particular Web page matches a search. It refers the textual similarity between the query and a page. Pages can be given a query specific, numeric relevance score; the higher the number, the more relevant the page is to the query. Relevancy is measured by using On the Page Criteria and Off the page Criteria factors. The former determines the keyword density by dividing the Keyword count and the total no of keywords in a page. The "off the page" criteria are Number of links, Relevance of links, Click through rates which refers how many people click on a particular link. This is often the quickest route to get a listing and can provide a boost to ranking also. Importance refers to the global popularity of a page, as often inferred from the link structure (e.g., pages with many in-links are more important), or perhaps other indicators. In practice, search engines usually combine relevance and importance, computing a combined rank score that is used to order query results presented to the user. The term spamming or spamdexing refers any deliberate human action that is meant to trigger an unsustainably favourable relevance or importance for some web page, considering the page's true value.

II. SPAMDEXING DETECTION STRATEGY

A. Link Analysis

More complex ranking methods are also vulnerable to spam. For example, the cosine similarity method used in latent semantic indexing will always rank a document that is an exact match to the query higher than any other document. The popular search engine Google uses a system called PageRank to determine the order in which it returns results [4]. This ranking method orders pages based on the inbound links to each page. Essentially, when one page links to another it is casting a vote that the target of the link is valuable. Although users have found Google resistant to spam, PageRank can be manipulated by artificially altering the link structure of the web. While Page et al. note that this could be done by web authors paying others for inbound links, though they thought it would be financially infeasible [6]. It seems that their predictions were incorrect as atleast one company is in the business of brokering these link sales [7]. Another problem with PageRank is that it only works on interlinked collections of documents. There are many valuable document repositories that do not have links such as newsgroup postings and archived emails. In addition, running PageRank on a small subset of the Web (e.g., the IBM.com website), will not produce as useful results since links from outside documents can not be considered. Although PageRank has been successful at keeping spammers from manipulating results, it is not impenetrable and link analysis is not applicable in certain instances.

B. HTML Analysis

All search engines do some analysis of the HTML elements on a web page in order to determine its ranking. In order to keep authors from simply filling the title and description elements with keywords, search engines will usually only look at a finite number of characters in these fields [12]. Additionally, search engines also look for attempts to hide keywords by putting them at the bottom of a page, in a small font, or in a font whose color closely matches the background color [8]. While these methods can detect many spam pages, others remain unnoticed by having spam text appear in the web page, masking itself as normal text. Search engines will also look for "doorway" pages that are setup to rank highly on common searches and then send the user to a different page which would not have ranked as highly [11]. However, many doorway pages are difficult to detect since they use complicated JavaScript code rather than a simple redirect tag.

Although both email and search engine spamdexing are attempts to gain the attention of Internet users, they do not have much in common. Search engine spamdexing is a largely technical task in that spammers are trying to get their results placed as highly as possible and it is called as Spamdexing. Thus, filters that foil an email spammer may not be sufficient to stop a more technical search engine spammer. Classifying search engine spam in this manner is more difficult since many non-spam web pages exist for commercial purposes and contain many of the same keywords as the spam pages.

C. Human Experts

About.com, Yahoo!, and the Open Directory Project all provide directories of pages on frequently requested topics. These directories have been edited and thus manually screened for spam. However, these listings will reflect the biases of their editors. While this may not bother some users, those searching for information on controversial topics may be more comfortable with search results that have not been filtered by a human. Finally, these directories may not be as up-to-date as other search engines since it is difficult for a human editor to keep up with the fast-changing web. A system designed by Bharat and Mihalia [3] uses existing directories of sites on a particular topic to rank search results for that topic. It scours the web for pages that link to a wide variety of sites that are on the same topic but from different sources and stores these pages as "expert pages." It then ranks pages based on the number of "experts" that link to them. While this system appeared very promising in tests, if it were deployed in an actual search engine, it would be subject to phony expert pages. Since the criteria for an "expert page" is based entirely on the content of the page itself, a spammer could create a number of pages that are designed to be recognized as "expert pages" and use them to manipulate the search engine.

D. Text Classification

Androutsopoulos et al. tested a Bayesian classifier (developed by Sahami et al.) that determines spam [1]. They note that "it seems that the language of current spam constitutes a distinctive genre" and using natural classification to distinguish between spam and legitimate page thus makes sense. This classification scheme treats each word as a token and analyzes pages based on the frequency of the words they contain. Another spam classifier is Spam Assassin [7]. This classifier tests the presence of key phrases (e.g., pornographic text) and other properties (e.g., an invalid date in the headers). It assigns each of these phrases and properties a numeric value and adds the values for a particular email together. If the sum of these values is above a certain threshold, it marks the message as spam. Quek developed a system to classify web pages into categories using a Bayesian classifier [6]. In addition to using only the textual components of pages, he tried using a couple web-specific classification schemes. One of these was to use only the text contained within header and title tags, as this text is assumed to be representative of the page's contents. The other was to use the hyperlink structure and the text in the hyperlinks to derive relationships between webpages.

III. PROPOSED ARCHITECTURE

In order to properly classify spam, we first have to define precisely what constitutes a link farm and spam document. This definition is complex because spam in one context may not be spam in another. A webpage is spam if it or a portion of it was created with the purpose of increasing its ranking through use of link and content that does not add to the user experience. Unfortunately, it is not always possible to detect spam by content analysis, as some spam pages only differ from normal pages because of their links, not because of their contents. This research attempt to derive intent based on the link structure and content of the document. Many of these pages are used to create link farms. A link farm is a densely connected set of pages, created explicitly with the purpose of deceiving a link-based ranking algorithm. A link farm may have a high in-degree and statically differ from non-spam pages. This is done by using UCINET Software[13], measure Eigen vector, centrality of degree and betweenness in network. A network that possesses just a few or perhaps even one node with high centrality is a centralized network. In this type of network, all nodes are directly connected to each other. Subordinate nodes direct information to the central node and the central node distributes it to all other nodes. Centralized networks are susceptible to disruption because they have few central nodes and damage to a central node could be devastating to the entire network. Decentralized networks are those that do not possess one central hub; but rather possess several important hubs. Each node is indirectly tied to all others and therefore the network has more elasticity. Consequently, candidate's profile networks choose this type of structure whenever possible. Social network analysts use the term degrees in reference to the number of direct connections that a node enjoys. The node that possesses the largest number of connections is the hub of the network. The term betweenness refers to the number of groups that a node is indirectly tied to through the direct links that it possesses. Therefore, nodes with high a degree of betweenness act as liaisons or bridges to other nodes in the structure. These nodes are known as "brokers" because of the power that they wield. However, these "brokers" represent a single point of failure because if their communication flow is disrupted than they will be cut off to the nodes that it connects. The sum of degree, betweenness and Eigen vector values will get a threshold value. The non links spam can identify based on threshold value and this parameter value pass to SVM Light tool to identify the content of the document.

The motivation behind the content analyzers lies in the fact that written English has certain consistent statistical properties. These include sentence length analyzer, stop word analyzer and part of speech analyzer. From TREC data [9], it has been found that the average sentence length is 18 words. Stop words are the set of the most frequently occurring words in a collection of documents. Collecting information about stop word frequency in a document can help detect spam pages because an author trying to create spam may not include stop words in their spam efforts. The third analyzer uses WordNet database [12] to ascertain part of speech information for all the words in a document with the intention of collecting frequencies of noun, verb, adjective, and adverb usage. This analyzer can be useful because spam pages often have more nouns than non-spam pages because most query terms involve nouns. While all three analyzers report parameters

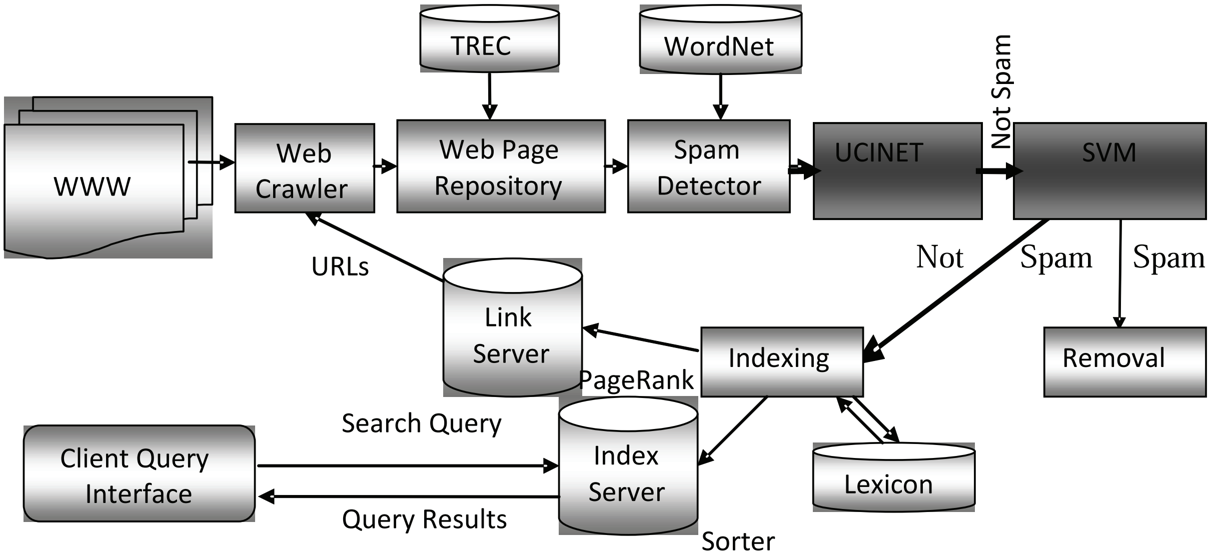


Fig 1. Proposed Architecture for Combating Spamdexing

on the documents, there is still the problem of using these collected parameters to determine if a given webpage is spam. This is solved by using Vipnik's SVMLight tool [10] to implement Support Vector Machine binary classifier, which fed parameters as feature vectors from the three analyzers to classify documents as either spam or not spam. Fig 1 gives an overview of this architecture.

IV. IMPLEMENTATION PLAN

Software has been developed to load documents into a local file repository, index, and query those documents.

Under normal operation, the indexer runs the classifier to determine if a file is spam and only indexes it if it is not. For purposes of training the classifier, the indexer is slightly modified to write statistics about each document to a file.

A. Development

In order to load TREC data files into our local repository, a Java program has been developed that separates each TREC data file into separate files each composed of one article. By separating these files into articles, the TREC data more closely matches the nature of data on the World Wide Web. To load web pages into the repository, a Java crawler has been developed to download all those documents to the local file system. The indexing engine is based on a flexible architecture that allows us to crawl a directory tree on the local file system and process each file encountered. There are three stages of processing a file:

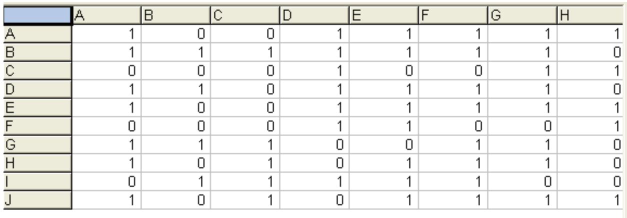
1. *Pre-process.* This includes reading the file from disk and extracting the natural language portions from any markup language.

2. *Combating Spamdexing* Determine the threshold value for link and classify the document in the form of feature vector. Use the UCINET software and SVM based binary classifier model to classify the link and document as either spam or non-spam.

3. *Indexing.* Add the document to an index file if it is classified as non-spam. In order to use a SVM classifier, it is necessary to first train the model on sample data. This is accomplished by running the indexing process in a mode where it wrote the parameters from the semantic analyzers for each document to a file. Each document is manually classified in the collection indicating whether or not it was spam. Finally, this data file is fed into SVMLight to create and train the Binary Support Vector Machine (SVM) classifier. A query interface is provided to allow users to test this system. The spam and non-spam pages were found by performing five queries on AltaVista and manually classifying the top one hundred results of each. We selected five queries that we thought were likely to result in a large number of spam documents: “MP3”, “breast”, “college girl”, “Apple IBM Dell Gateway”, and “ford chevy nissan toyota honda”. Of the five hundred AltaVista results, 337 were non-spam, eighty were spam, and the remaining eighty-three were not in English. Even using the most powerful open-source Support Vector Machine Binary Classifier implemented by SVMLight, the classifier could not split documents into spam and non-spam. More promising was the fact that many of the web pages classified as TREC contained proportionally large amounts of natural language data.

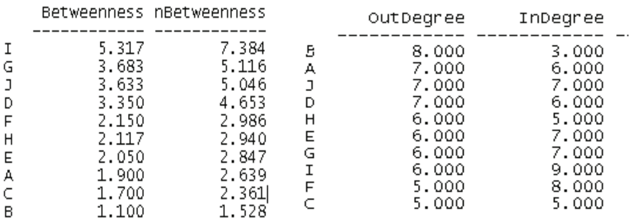
V. RESULTS AND DISCUSSION

Table 1: Sample Input values for UCINET



	A	B	C	D	E	F	G	H
A	1	0	0	1	1	1	1	1
B	1	1	1	1	1	1	1	0
C	0	0	0	1	0	0	1	1
D	1	1	0	1	1	1	1	0
E	1	0	0	1	1	1	1	1
F	0	0	0	1	1	0	0	1
G	1	1	1	0	0	1	1	0
H	1	0	1	0	1	1	1	0
I	0	1	1	1	1	1	0	0
J	1	0	1	0	1	1	1	1

Table 2: Degree Centrality & Betweenness Measures



```
    Betweenness nBetweenness              OutDegree      InDegree
   ------------ ------------           ------------  ------------  -
I         5.317        7.384      B           8.000         3.000
G         3.683        5.116      A           7.000         6.000
J         3.633        5.046      J           7.000         7.000
D         3.350        4.653      D           7.000         6.000
F         2.150        2.986      H           6.000         5.000
H         2.117        2.940      E           6.000         7.000
E         2.050        2.847      G           6.000         7.000
A         1.900        2.639      I           6.000         9.000
C         1.700        2.361|     F           5.000         8.000
B         1.100        1.528      C           5.000         5.000
```

Fig 2: Web Graph for Link Structure

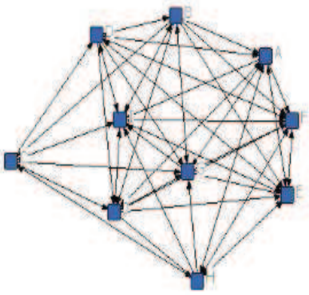


Table 3: Sample Test Results for Content

SEARCH KEYS	Total Pages	Spam Pages		Non-Spam Pages	Search Time Sec(s)
		Bef.	Aft.		
MP3	131	13	7	124	16
BREAST	129	8	4	125	14
SPAM	125	9	4	121	17
SPOON	122	11	5	117	13
PIRACY	127	11	6	121	18
'FILTER	112	6	4	108	16
WINDO	136	12	6	130	17
SCALING	126	5	3	123	18

An input values are given manually in Table 1, the centrality of the each node by performing Degree Centrality and Betweenness Centrality in Table 2 and corresponding Web Graph in Fig 2.A sample test result for combating spamdexing has been given in Table 3. A search engine downloads web pages one by one starting from the root node, using focused crawler. These documents are stored in web repository, then preceded by tokenization, HTML tags removal, stop words removal, stemming and lexicon formation. Then it is followed by forward indexing, inverse indexing with the help of an indexer, and dumping into file barrels for accessed by the client query interface. In this research, the most complicating factor is data collection. Even though several web pages have been collected by this search engine, all the web pages do not contain relevant information. Again the resulted relevant pages may be bounced with dead pointers sometimes. The hyperlinks are not properly bound during crawling process. Another aspect of this search engine is to use storage efficiently. Due to the dynamic storage of forward indexing, a huge amount of memory size is reduced comparing with the conventional search engines. Furthermore, most queries can be answered using just the inverted index. The current version of search engine with spamdexing filter answers most keys in between 1 and 10 seconds. Its accuracy and precision are found to be satisfactory. To improve this in future, plans are made to design a separate spamdexing tool for combating spasm in any search engine results. From the above results, it has been understood that the average relevancy, precision and recall values of this tool is also fine by combating spamdexing. From the following Fig 3, the processing time for initial query and repeated query has been identified evidently.

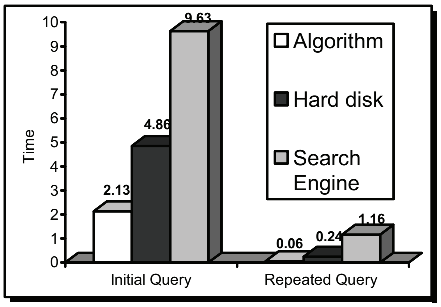


Fig 3: Processing Time for combating Spamdexing

Assuming the user is more interested in finding a quick answer to their query, a page with more textual information should have a higher rank. The analyzers could help to determine this rank. In order to better classify web documents it is a belief that it is necessary to take advantage of the meta information that is included in the html as well as the link structure. With this extra information at hand, a spam analyzer will have a better chance of being able to classify spam vs. one that only looks at plain text.

VI. CONCLUSIONS & FUTURE WORK

Due to the similarities between spam and non-spam the original semantic analyzers are not an effective method to classify spam content. Since spam and non-spam documents are so similar, it is sometimes very difficult for a human to differentiate between the two. Because of these similarities, it is unlikely that any natural language analysis method will be successful in differentiating between spam and non-spam. However, using semantic analyzers to determine the usefulness of information on a webpage had much more promising results.

REFERENCES

[1] Androutsopoulos*, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal Email Messages." In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (July 2000).*

[2] *Zolt´an Gy¨ongyi, Hector Garcia Molina, and Jan Pedersen. Combating web spam with TrustRank. Technical report, Stanford University, 2004.*

[3] *Krishna Bharat and George A. Mihala. "When Experts Agree: Using Non-Affiliated Experts to Rank Popular Topics." ACM Transactions on Information Systems (January 2002). 47-58.*

[4] *Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The PageRank Citation Ranking: Bringing Order to the Web." 1999. Technical Report, Stanford University. http:// dbpubs. stanford. edu/ pub/ 1999-66.*

[5] *Alan Perkins. The classification of search engine spam. http://www. ebrandmanagement.com/ whitepapers/ spamclassification/.*

[6] *Choon Yang Quek. "Classification of World Wide Web Documents." 1997. Senior Honors Thesis, Carnegie Mellon University. ttp://www2.cs.cmu.edu/ afs/cs.cmu.edu/ project/theo11/www/ wwkb/choon-thesis.html.*

[7] *Spam Assassin. (Software.) http:// www. spamassassin. org.*

[8] *Danny Sullivan, ed. "Search Engine Placement Tips." Last updated October 14, 2002. http:// searchenginewatch. com / webmasters/ tips.html.*

[9] *S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In WWW Conference, volume 7, 1998. http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm*

[10] *Zolt´an Gy¨ongyi, Hector Garcia Molina. Web Spam Taxonomy, March, 2004.*

[11] *http://www.searchenginewatch.com*

[12] *Andrew Westbrook, Russell Greene. "Using Semantic Analysis to classify Search Engine Spam" Stanford University, 2005.*

[13] *www.analytictech.com/ucinet/*