Volume 4, Issue 11, November 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Web Spam Recognition through Classification Algorithms

Ms. Meenal M. Shingare
Dept. of CSE
Marathwada Institute of Technology,
BAMU University Aurangabad, India

Prof. S. R. Chaudhary
Dept of Information Technology
Marathwada Institute of Technology,
BAMU University Aurangabad, India

Abstract — Search Engine is becoming highly importance for commercial sites which lead rise to "Web Spam" which makes fool or mislead to search engine. As search engine is the default gateway to the web. Web spam is a nuisance to users as well as to search engines: users have a harder time finding the information they need, and search engines have to cope with an inflated corpus, which in turn causes their cost per query to increase. Therefore, search engines have a strong incentive to weed out spam web pages from their index. Spam is a deliberate action solely in order to boost a web page's position in search engine results, incommensurate with page's real value. In this paper, an efficient spam detection system based on a classifier that combines new link-based features with language-model (LM)-based ones. These features are not only related to quantitative data extracted from the Web pages, but also to qualitative properties, mainly of the page links. The ability of a search engine to find, using information provided by the page for a given link, the page that the link actually points at. This can be regarded as indicative of the link reliability. To check the coherence between a page and another one pointed at by any of its links. Two pages linked by a hyperlink should be semantically related, by at least a weak contextual relation. An LM approach to different sources of information from a Web page that belongs to the context of a link, in order to provide high-quality indicators of Web spam. The Kullback–Leibler divergence is specifically applied on different combinations of these sources of information in order to characterize the relationship between two linked pages. The result is a system that significantly improves the detection of Web spam using fewer features, on Combining the two large and public datasets such as WEBSPAM-UK2006 and WEBSPAM-UK2007. For implementing this an SVMs algorithm is used which aim at searching for a hyper plane that separates two classes of data with the largest margin. As well as using boosting decision tree algorithm such as C5.0 on datasets some rules are derived and create the Decision tree, which helps in improving the accuracy.

Keywords — Web Mining, Language Model, Web Spam Detection, SVM, C5.0,

I. INTRODUCTION

A. PURPOSE

Web spam can significantly deteriorate the quality of search engine results. Thus there is a large incentive for commercial search engines to detect spam pages efficiently and accurately. In this paper a spam detection system that uses the topology of the Web graph by exploiting the link dependencies among the Web pages, and the content of the pages themselves. It is observed that linked hosts tend to belong to the same class: either both are spam or both are non-spam. It is demonstrated into three methods of incorporating the Web graph topology into the predictions obtained by our base classifier:

(i) Clustering the host graph, and assigning the label of all hosts in the cluster by majority vote
(ii) Propagating the predicted labels to neighboring hosts, and
(iii) The predicted labels of neighboring hosts as new features and retraining the classifier.

Web spam is one of the main current problems of search engines because it strongly degrades the quality of the results. Many people become frustrated by constantly finding spam sites when they look for legitimate content. In addition, Web spam has an economic impact since a high ranking provides large free advertising and so an increase in the Web traffic volume. During recent years, there have been many advances in the detection of these fraudulent pages but, in response new spam techniques have appeared. Research in this area has became an arms race to fight an adversary who constantly uses more and more sophisticated methods. For this reason, it is necessary to improve anti-spam techniques to get over these at-tacks. Web spam, or spamdexing, includes all techniques used for the purpose of getting an undeservedly high rank. In general terms, there are three types of Web spam: link spam, content spam, and cloaking, a technique in which the content presented to the search engine spider is different to that presented to the browser of the user. However, link and content spam are the most common types, and the ones considered in this work. Popularity and link analysis algorithms are based significantly on the links that point to a document. Since the number of such links contributes to the value of the document, it recognize nepotistic links so that their effect can be reduced or eliminated. While a typical definition of nepotism includes favouritism to relatives, the broader interpretation of "bestowal of patronage in consideration of relationship, rather than of merit." Thus links between pages that are related because of common

administrative control but also links that effect some reward (e.g. advertising or paid links). Nepotistic links are typically considered undesirable, and so it is useful to eliminate them before calculating the popularity or status of a page [2].

Finding link nepotism is similar to, but distinct from the problem of identifying duplicate pages or mirrored sites . While mirrored pages are an instance of the problem it is being addressed (that one entity gets to "vote" more often than it should), for considering the cases in which pages (mirrored or not) are pointing to a target for reasons other than merit. Instead of link popularity, some services which is used by many of the major engines including MSN Search (Microsoft Corporation 2000)) collect and analyse usage popularity. While calculations based on usage popularity may benefit from knowing about nepotistic links, we have not considered their effects. The issues surrounding the question of what links to keep, and report of preliminary results on the use of a machine learning tool operating on a hand-generated feature set to automatically recognize nepotistic links. Link spam can be defined as "links between pages that are present for reasons other than merit." An important voice in the web spam is that of search engine optimizers (SEOs), such as SEO Inc. (www.seoinc.com) or Bruce Clay (www.bruceclay.com). The activity of some SEOs benefits the whole web community, as they help authors create well-structured, high-quality pages. However, most SEOs engage in practices that we call spamming. For instance, there are SEOs who define spamming exclusively as increasing relevance for queries not related to the topic(s) of the page. These SEOs endorse and practice techniques that have an impact on importance scores, to achieve what they call ethical web page positioning or optimization. Please note that according to our definition, all types of actions intended to boost ranking (either relevance, or importance, or both), without improving the true value of a page, are considered spamming. There are two categories of techniques associated with web spam. The worst category includes the boosting techniques, i.e., methods through which one seeks to achieve high relevance and/or importance for some pages. The second category includes hiding techniques, methods that by themselves do not influence the search engine's ranking algorithms, but that are used to hide the adopted boosting techniques from the eyes of human web users.

Some of the highlights of the case of link spam are: Becchetti *et al.* [1], who used automatic classifiers to detect link-based spam and, who analyzed supporting sets and Page Rank contributions for building an algorithm to detect link spam and Benczúr *et al.*[2] , who analyzed supporting sets and Page Rank contributions for building an algorithm to detect link spam. Other works are focused on content spam: A spam page or host is a page or host that is used for spamming or receives a substantial amount of its score from other spam pages. There are many techniques for Web spam and they can be broadly classified into content (or keyword) spam and link spam. Content spam includes changes in the content of the pages, for instance by inserting a large number of keywords. In Measuring Qualified links, it is shown that 82-86% of spam pages of this type can be detected by an automatic classifier. The features used for the classification include, among others: the number of words in the text of the page, the number of hyperlinks, the number of words in the title of the pages, the compressibility (redundancy) of the content, etc. 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. Many of these pages are used to create link farms.

II. RELATED WORK

A. TOPOLOGICAL LINK SPAM

A link farm is a densely connected set of pages, created explicitly with the purpose of deceiving a link-based ranking algorithm. It is the manipulation of the link structure by a group of users with the intent of improving the rating of one or more users in the group. A page that participates in a link farm, have a high in-degree, but little relationship with the rest of the graph. Heuristically, we call spamming achieved by using link farms topological spamming. In particular, a topological spammer achieves its goal by means of a link farm that has topological and spectral properties that statistically differ from those exhibited by non spam pages [1]. This definition embraces the cases considered in, and their method based on shingles can be also applied in detecting some types of link farms (those that are dense graphs). Link-based and content-based analysis owner two orthogonal approaches. These approaches are not alternative and should probably be used together.

B. PAGE RANK DISTRIBUTION IN YOUR NEIGHBOURHOOD LOOKS HONEST OR SPAM

Our key assumption is that supporters of an honest page should not be overly dependent on one another, i.e. they should be spread across sources of different quality. Just as in the case of the entire Web, the Page Rank distribution of an honest set of supporters should be power law. The two key observations in detecting link farms, colluding pages or other means of Page Rank boosting in the neighbourhood of a page are the following: Portions of the Web are self-similar; an honest set of supporter pages arise by independent actions of individuals and organizations that build a structure with properties similar to the entire Web. Link spammers have a limited budget; when boosting the Page Rank of a target page, "unimportant" structures are not replicated. A perfect form of a link spam is certainly a full replica of the entire Web that automatic link based methods are unable to distinguish from the original, honest copy. Other works are focused on content spam: Ntoulas *et al.* [3] introduced new features based on checksums and word weighting techniques and Webb *et al.* [4], who proposed a real-time system for web spam classification by using HTTP response headers to extract several features.

C. CONTENT-BASED SPAM DETECTION

In some of the spam detection heuristics were completely independent of the content of the web pages (instead using features such as the hyperlink structure between pages and the DNS records of hosts), while others treated words as

uninterrupted tokens (for example, by clustering pages into sets of near-duplicates, and by measuring the rate of page evolution). In this paper, an additional set of heuristics, all of them based on the content of web pages. Some of these heuristics are independent of the language a page is written in, others use language-dependent statistical properties [5].

III. PROPOSED WORK

In this work , new features are proposed to characterise the web spam pages, earlier work was based on content and link-based features to detect spam. New qualitative features is used to improve Web spam detection

1) A group of link-based features which check the reliability of links

2) A group of content-based features extracted with the help of a language-model (LM) approach. An automatic classifier that combines both types of features, reaching a precision that improves the results of each type separately and those obtained by other proposals.

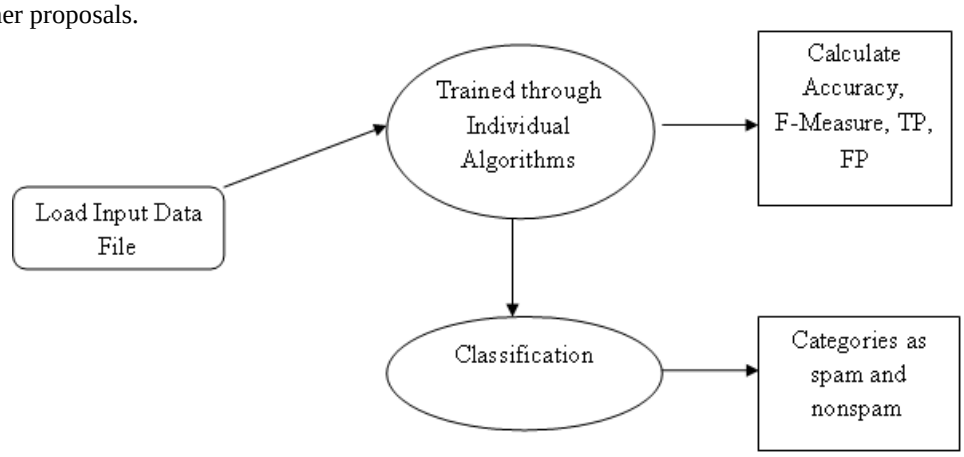

Figure 1 System Flow Diagram of Spam Detection

D. LANGUAGE MODEL DISAGREEMENT

1. *Content Filtering and Spam:* A different set of approaches for fighting comment spam works by analyzing the content of the spam comment, and possibly also the contents of pages linked by the comment. All these techniques are currently based on detecting a set of keywords or regular expressions within the comments. This approach suffers from the usual drawbacks associated with a manual set of rules, i.e. high maintenance load as spammers are getting more sophisticated. Typically, content-based methods require training with a large amount of spam and non-spam text, and correcting mistakes that are made; failure to continuously maintain the learner will decrease its accuracy, as it will create an inaccurate conception of what's spam and what's not. Having said that, regular expression based methods are fairly successful currently, partly due to the relatively young history of link spamming [7].
2. *Identifying Spam Sites:* An altogether different approach to spam filtering is not to classify individual links as spam links or legitimate links, but to classify pages or sites as spam; recent work in this area includes usage of various non-content features and link analysis methods. The drawback of these approaches is that spam is essentially not a feature of pages, but of links between pages; sites can have both legitimate and spam incoming links (this is true for many online shops). Additionally, usage of some of these methods requires full connectivity knowledge of the domain, which is beyond the abilities of most bloggers. In comparison to the existing methods presented, our approach requires no training, no maintenance, and no knowledge of additional information except that present on the commented web page.
3. *Language Models for Text Comparison:* A language model is a statistical model for text generation: a probability distribution over strings, indicating the likelihood of observing these strings in a language. Usually, the real model of a language is unknown, and is estimated using a sample of text representative of that language. Different texts can then be compared by estimating models for each of them, and comparing the models using well-known methods for comparing probability distributions. Indeed, the use of language models to compare texts in the Information Retrieval setting is empirically successful and becoming increasingly popular.

E. LANGUAGE MODEL FOR HYPERLINK

An algorithm that identifies hyperlinks where the language model of the target and the source disagree. A suspicious edges is feed into a weighted PageRank calculation to obtain NRank, the "nepotism rank" of the page are subtracted from the original PageRank values. As in our key ingredient is the Kullback-Leibler divergence (KL) between the unigram language model of the target and source pages. In fact it is infeasible to compute KL for all pairs of documents connected by hyperlinks [2]. Two computationally easier tasks are to compare each anchor text to (i) the document containing (ii) the document pointed by it. While the former task is simply performed by a linear scan, the latter task requires an external memory sorting of all anchor text found. The hyperlink is set aside if the corresponding language models differ. A typical anchor spam is generated by the owner of the page, we consider case (ii) above, complementary to the malicious anchors of reputable pages. It is observed that best performance when the anchor text by a few neighbouring words to properly handle very short anchor such as "here"; a segment boundaries defined by HTML and punctuation. By using Interpolated Aggregate Smoothing, the language model for document D has the form.

$$p(\omega| D) = \lambda \frac{tf(\omega, D)}{\sum_{v \in D} tf(v, D)} + (1 - \lambda) \frac{tf(\omega, C)}{\sum_{v \in C} tf(v, C)}$$

where C is the text of the entire corpus and w is a word. A language model similar for an anchor A. The value is set to λ = 0.8; an smooth anchor term frequencies by the corpus formed by all extended anchor text. Finally the Kullback-Leibler divergence is computed

$$KL(A \| D) = \sum_{\omega} p(\omega | A) \log \frac{p(\omega | A)}{p(\omega | D)}$$

a formula asymmetric in A and D. The current form weights words by their relevance within anchors; it is observed that the degradation in performance when computing penalties by exchanging the role of A and D in (2). KL will have normal distribution over the documents if all anchor text behave the same since the sum of random variables that correspond to words and the words themselves have sufficient independence to yield a normally distributed sum.

F. LM- BASED FEATURES

The relationship between two linked Web pages are characterize according to different values of divergence. These values are obtained by calculating the KL divergence between one or more sources of information from each page. The KL divergence applied to the anchor text of a link and the title of the page pointed by this link. In particular, the following three sources of information from the source page [11]:

Anchor Text: When a page links to another, this page has only a way to convince a user to visit this link, that is by showing relevant and summarized information of the target page. This is the function of the anchor text.

Surrounding Anchor Text: Sometimes anchor terms provide little or no descriptive value. Let us imagine a link whose anchor text is "click here." For this reason, text surrounding a link can provide contextual information about the pointed page. A better behaviour is observed when the anchor text is extended with neighbouring words.

URL Terms: Besides the anchor text, the only information available of a link is its URL. A URL is mainly composed of a protocol, a domain, a path, and a file. These elements are composed of terms that can provide rich information from the target page.

Title: Document titles bear a close resemblance to queries, and that they are produced by a similar mental process. The similarity of title and anchor text and they concluded that both titles and anchor text capture some notion of what a document is about, though these sources of information are linguistically dissimilar.

Page Content: The page content is the main source of information that is usually available. Although in many cases, the title and meta tags from the target page are not available, most Web pages have at least a certain amount of text [11].

Meta Tags: Meta tags provide structured meta data about a Web page and they are used in SEO. Although they have been the target of spammers for a long time and search engines consider these data less and less, there are pages still using them because of their clear usefulness. In particular the attributes "description" and "keywords" from meta tags to build a virtual document with their terms are considered.

G. QUALIFIED LINK ANALYSIS

Web Spam detection are improved on a group of link based features which checks reliability of links and a group of content based features extracted with the help of Language Model approach. Some of the considered features are related to the quality of the links in the page, behaviour of standard search Engines, applied to the queries thus increasing the spam detection rate. This qualified link analysis has been designed to study neither the network topology, nor link characteristics in a graph [6]. With this sort of analysis, it is found that mainly nepotistic links that are present for reasons other than merit. We have developed an information retrieval system that retrieves the URL, anchor Text, and a cached page version of the analyzed link that can be stored in a search engine.

Thus, the more negative the difference between the recovered and not recovered links, the greater the likelihood that this site is applying spam techniques [11].

Incoming–Outgoing: It is well-known that spam pages link to nonspam pages, but nonspam pages do not link to spam pages. Taking advantage of the possibilities of the system to submit queries to a search engine, a new query to request to the search engine is included, how many sites point to the analyzed page (incoming links). The difference in the amount of links of each type. It is observed that the difference shape of the graphic for spam and nonspam pages. In addition, the number of outgoing links as another feature is also considered.

External–Internal: Several theories exist about the impact of internal and external links in the PageRank of a site. Although there is no definitive evidence to prove it, it is observer that many websites apply these theories. For this reason, the number of external and internal links as features. This feature takes negative values for spam pages, and positive for nonspam pages.

Broken Links: Broken links are a common problem for both spam and nonspam pages, even when this sort of link has a negative impact in the PageRank. The number of spam pages is higher in almost the whole range of numbers of the broken links considered.

Anchor Text Typology: It is usual that spam pages contain text and links automatically generated. Moreover, the anchor text of many links are usually generated thinking in the context of the search engines instead of the users. Thus, a four features in order to measure the number of links that are formed only by 1) punctuation marks, 2) digits, 3) URL 4) an empty chain. Though there are areas where the values for overlap for spam and nonspam pages, is considered into the account that the classifier uses a whole set of features, by assigning different weights to the most appropriates in every case. Thus, in all there are 12 features for each Web page. One of the most successful methods based on term distribution

analysis uses the concept of Kullback-Leibler Divergence (KLD) to compute the divergence between the probability distributions of terms of two particular documents considered. An KLD to measure the divergence between two text units of the source and target pages.

H. COMBINATION OF SOURCES OF INFORMATION

In addition to using these sources of information individually, are combined from some of the source page with the goal of creating virtual documents which provide richer information. It is observed that an Anchor Text (A), Surrounding Anchor Text (S), and URL terms (U) as sources of information. To create two new sources of information: 1) combining Anchor Text and URL terms (AU) 2) combining Surrounding Anchor Text and URL terms (SU). In addition, to that other sources of information from the target page: Content Page (P), Title (T), and Meta Tags (M). It is being ruled out the use of any combination due to the limited relationship between these sources of information.

Table 1. Combination of Different Sources

COMBINATION OF DIFFERENT SOURCES OF INFORMATION
Page Content (P) Anchor Text(A ⟶ P), Surrounding Anchor Text (S ⟶ P), URL Terms (U ⟶ P) Anchor Text U URL Terms (AU ⟶ P), Title vs Page(T ⟶ P), Surrounding Anchor Text U URL Terms (SU ⟶ P) Meta Tags vs Page (M ⟶ P).
Title (T) Anchor Text (A ⟶ T), Surrounding Anchor Text (S ⟶ T), URL Terms (U ⟶ T) , Surrounding Anchor Text U URL Terms (SU ⟶ T)
Meta Tags (M) Anchor Text (A ⟶ M), Surrounding Anchor Text (S ⟶ M), Surrounding Anchor Text U URL Terms (SU ⟶ M)

I. CLASSIFICATION ALGORITHMS

1. *C5.0 Model:* A C5.0 model is based on the information theory. Decision trees are built by calculating the information gain ratio. The algorithm C5.0 works by separating the sample into subsamples based on the result of a test on the value of a single feature. The specific test is selected by an information theoretic heuristic. This procedure is iterated on each of the new subsample and keeps on until a subsample cannot be separated or the partitioning tree has reached the threshold. The information gain ratio is defined as [9]:

$$\text{Information Gain Ratio}(D,S) = \frac{\text{Gain}(D,S)}{H(\frac{|D_1|}{D},\ldots,\frac{|D_S|}{D})}$$

where in equation is a database state, finds the amount of order in that state, when the state is separated into C 5.0 builds decision trees from a set of training data, using the concept of information entropy. The training data is a set S = (s1. S2, …s n) of already classified samples. Each sample consists of a p-dimensional vector (x1 , x2,….xn) where the xn represent attributes or features of the sample, as well as the class in which si resides.

$$\text{Entropy (S)} = \sum_{i=1}^{\sigma} -p_i \log_2 p_i$$

where pi is the proportion of S belonging to class i. Note the logarithm is still base 2 because entropy is a measure of the expected encoding length measured in bits. The maximum possible entropy is log2c.

Information Gain Calculation

$$\text{Gain(S, A)} = \text{Entropy (s)} \sum_{v \in \text{Values (A)}} \frac{|Sv|}{|S|} \text{Entropy (Sv)}$$

where Values(A) is the set of all possible values for attribute A, and Sv is the subset of S for which attribute A has value *v* (i.e., Sv = {s ε S | A(s) = *v*}).

Step 1 - To build either decision tree or rule set the node uses the C5.0 algorithm.

Step 2 - C5.0 model splits the sample based on the field that provides the maximum information gain.

Step 3 - The defined each subsample is then split again, usually based on a different field, and the process repeats until the subsamples cannot be split any further.

Step 4 - Finally, the lowest-level splits are re-examined, and those that do not contribute significantly to the value of the model are removed. Normal procedure: top down in recursive divide-and-conquer fashion.

First: Attribute is selected for root node and branch is created for each possible attribute value.

Then: The instances are split into subsets (one for each branch extending from the node).

Finally: Procedure is repeated recursively for each branch, using only instances that reach the branch Process stops if all instances have the same class. Then it is compared a tree before the split and after the split using Information Gain = Info (before) – Info (after).Information Gain increases with the average purity of the subsets that an attribute produces Strategy: choose attribute that results in greatest information gain.

C5.0 can produce two kinds of models. A decision tree is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree. In other words, exactly one prediction is possible for any particular data record presented to a decision tree.

2. *SVM Model:* SVM (Support Vector Machines) are a useful technique for data classification. Steps to be considered for SVM Algorithm.

Step 1 - Transform data to the format of an SVM package.
Step 2 - Conduct simple scaling on the data.
Step 3 - Consider the RBF kernel K(x, y) = $e^{-\Upsilon||x-y||^2}$.
Step 4 - Use cross-validation to find the best parameter C.
Step 5 - Use the best parameter C and Υ to train the whole training set.
Step 6 – Test

The RBF kernel is a reasonable first choice. This kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. The second reason is the number of hyper parameters which influences the complexity of model selection. There are two parameters for an RBF kernel: C and Υ. It is not known beforehand which C and Υ are best for a given problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good (C, Υ) so that the classifier can accurately predict unknown data (i.e. testing data). Note that it may not be useful to achieve high training accuracy (i.e. a classifier which accurately predicts training data whose class labels are indeed known). As discussed above, a common strategy is to separate the data set into two parts, of which one is considered unknown. The prediction accuracy obtained from the "unknown" set more precisely reflects the performance on classifying an independent data set. Our purpose is to give SVM novices a recipe for rapidly obtaining acceptable results. Although users do not need to understand the underlying theory behind SVM, we briefly introduce the basics necessary for explaining our procedure. A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one "target value" (i.e. the class labels) and several "attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes. Given a training set of instance-label pairs (xi; yi); i = 1; l where xi ε R^n and y ε $\{1, -1\}^l$ the support vector machines (SVM) require the solution of the following optimization problem: Here training vectors xi are mapped into a higher (maybe infinite) dimensional space by the function ϕ. SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term. Furthermore, K(xi, xj) = $\phi(xi)^T \phi(xj)$ is called the kernel function. Though new kernels are being proposed by researchers, beginners may find in SVM books the following four basic kernels[10]:

- linear: K(xi, xj) = $x_i^T x_j$.
- polynomial: K(xi, xj) = $(\Upsilon x_i^T xj + r)^d$, $\Upsilon > 0$.
- radial basis function (RBF): K(x_i, x_j) = exp(-$\Upsilon||x_i - x_j||^2$), $\Upsilon > 0$.
- sigmoid: K(x_i; x_j) = tanh($_\Upsilon x_i^T x_j$ + r). Here Υ , r and d are kernel parameters.

In today's machine learning applications, support vector machines (SVM) are considered a must try—it offers one of the most robust methods among all classification algorithms. It has a sound theoretical foundation, requires only a dozen examples for training, and is insensitive to the number of dimensions. In addition, efficient methods for training SVM are also being developed at a fast pace. In a two-class learning task, the aim of SVM is to find the best classification function to distinguish between members of the two classes in the training data. The metric for the concept of the "best" classification function can be realized geometrically. For a linearly separable dataset, a linear classification function corresponds to a separating hyper planes f(x) that passes through the middle of the two classes, separating the two [10]. Once this function is determined, new data instance xn can be classified by simply testing the sign of the function f(xn), xn belongs to the positive class if f(xn) > 0. Because there are many such linear hyper planes, what SVM additionally guarantee is that the best such function is found by maximizing the margin between the two classes. Intuitively, the margin is defined as the amount of space, or separation between the two classes as defined by the hyper planes. Geometrically, the margin corresponds to the shortest distance between the closest data points to a point on the hyper planes. Having this geometric definition allows us to explore how to maximize the margin, so that even though there are an infinite number of hyper planes, only a few qualify as the solution to SVM.

IV. EXPERIMENT RESULTS

In this we are presenting the results of performance evaluation of our practical implementation on WEBSPAM-UK2006 & WEBSPAM-UK2007 [8]. Performance of different methodology like Language Model, and Qualified Link Analysis for Web Spam Detection which has been measured by different Classification Algorithm. All results explained below are measured on the proposed designed semantic similarity application.

A. PERFORMANCE MEASURE

1. *Result of C5.0 Algorithm:* Performance measure of system is calculated with different algorithm like C5.0, SVM, where it is observed that there is a increase in the F-Measure parameter by 4% as compared with result of SVM for the combination CULULMUQL.

Table 2. Experimental Result of C5.0

Feature Set	F	%	AUC	%

LModel.txt	0.7649	76.49	0.8153	81.53
Content.txt	0.8358	83.58	0.8562	85.62
CUL.txt	0.8837	88.37	0.9117	91.17
CULULMUQL.txt	0.8827	88.27	0.8843	88.43
CULUQL.txt	0.8335	83.35	0.8676	86.76
Link.txt	0.8002	80.02	0.9203	92.03

2. *Results of SVM Algorithm:*

Table 3. Experimental Result of SVM

Feature Set	F	%	AUC	%
LModel.txt	0.6711	67.11	0.7206	72.06
Content.txt	0.6841	68.41	0.7518	75.18
CUL.txt	0.8303	83.03	0.7859	78.59
CULULMUQL.txt	0.8401	84.01	0.8923	89.23
CULUQL.txt	0.7376	73.76	0.7431	74.31
Link.txt	0.805	80.5	0.7993	79.93

3. *Accuracy Graph of C5.0 & SVM Algorithm:*

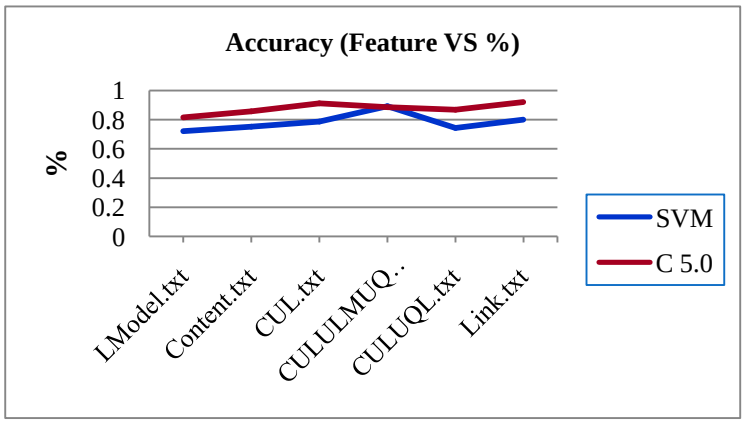

Figure 2. Graph-for Accuracy

4. *F- Measure Graph of C5.0 & SVM Algorithm:*

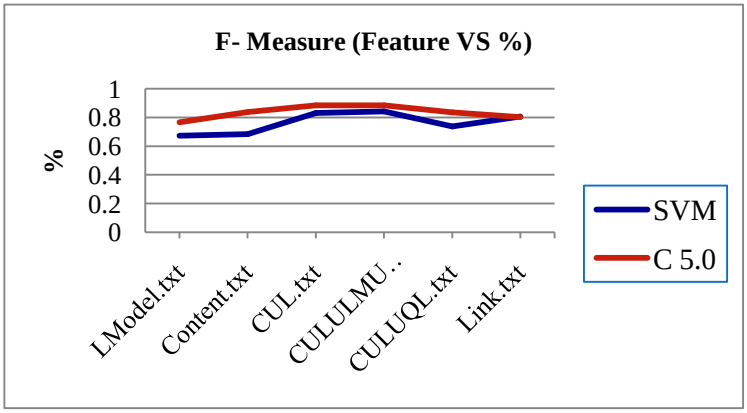

Figure 3. Graph for F- Measure

V. CONCLUSION

Classification is an important technique in data mining, and the decision tree is the most efficient approach to classification problems. The training cases were provided to the C5.0 classifier to generate decision trees or classification rules. Then, the decision trees or the rules are used to classify the test cases. The experiment was repeated multiple times, each time using different sets of training and test cases (dependent on number of packets used to create the case), different set of attributes used for classification (set A, or set A plus B), and different classification options (normal, rules generating, boosting, softening thresholds). Both error rates of provided classifiers C5.0 can produce two kinds of models.

A decision tree is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree. In other words, exactly one prediction is possible for any particular data record presented to a decision tree.The proposed method showed high Performance Measure on WEBSPAM- UK2006 and WEBSPAM-UK2007 datasets than all the previous methods, and also, the performance measurement result using precision, recall and F-Score are higher than previous methods.

ACKNOWLEDGEMENT

We would like to thank N. Eiron and K.S. Mc Curley for analysing the anchor text for web search and J. Abernethy, O. Chapelle, and C. Castillo, as well as X. Qi, L. Nie, and B. D. Davison for understanding us web spam identification and measuring the similarity. We would also like to express our gratitude to a collaborative effort by a team of volunteers who made available for us theWEBSPAM-UK2006 & WEBSPAM-UK2007 dataset to run experiments efficiently.

REFERENCES

[1] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Link-based characterization and detection of web spam," in Proc. 2nd Int. Workshop on Adversarial Information Retrieval on the Web (AIRWeb'06), Seattle, WA, 2006, pp. 1–8.

[2] A. A. Benczúr, I. Bíró, K. Csalogány, and M. Uher, "Detecting nepotistic links by language model disagreement," in Proc. 15th Int. Conf. World Wide Web (WWW'06), New York, 2006, pp. 939–940, ACM.

[3] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in Proc. 15th Int. Conf. World Wide Web (WWW'06), New York, 2006, pp. 83–92, ACM.

[4] S.Webb, J. Caverlee, and C. Pu, "Predicting web spam with http session information," in Proc. 17th ACM Conf. Information and Knowledge Management (CIKM'08), New York, 2008, pp. 339–348, ACM.

[5] J. Abernethy, O. Chapelle, and C. Castillo, "Webspam identification through content and hyperlinks," in Proc. Fourth Int. Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Beijing, China, 2008, pp. 41–44.

[6] X. Qi, L. Nie, and B. D. Davison, "Measuring similarity to detect qualified links," in Proc. 3rd Int. Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07), New York, 2007, pp. 49–56, ACM.

[7] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement," in Proc. First Int. Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Chiba, Japan, 2005, pp. 1–6.

[8] N. Eiron and K. S. McCurley, "Analysis of anchor text for web search," in Proc. 26th Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR'03), New York, 2003, pp. 459–460, ACM.

[9] Vahid Golmah , "An Efficient Hybrid Intrusion Detection System based on C5.0 and SVM " in Proc International Journal of Database Theory and Application Vol.7, No.2 (2014), pp.59-70,

[10] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin , "A Practical Guide to Support Vector Classification" Technical Report Department of Computer Science National Taiwan University, Taipei 106, Taiwan. URL http://www.csie.ntu.edu.tw/~cjlin.

[11] Lourdes Araujo and Juan Martinez-Romo ," Web Spam Detection: New Classification Features Based on Qualified Link Analysis and Language Models" , IEEE Transactions on Information Forensics and Security, Vol. 5, No. 3, September 2010.

