Machine Learning Techniques for Detecting Untrusted Pages on the Web

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Computer Science and Engineering

By

Vikash Kumar Singh

Under the Guidance of

Prof. S.K. JENA

Department of Computer Science Engineering

National Institute of Technolo gy

Rourkela

2009

2009

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled, " **Machine Learning Techniques for Detecting Untrusted Pages on the Web** " submitted by Vikash Kumar Singh in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Computer Science Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. S. K. Jena
Dept. of Computer Science Engineering
National Institute of Technology
Rourkela - 769008

ACKNOWLEDGEMENT

We would like to articulate our deep gratitude to our project guide Prof. Jena who has always been our motivation for carrying out the project. It is our pleasure to refer Microsoft Word exclusive of which the compilation of this report would have been impossible. An assemblage of this nature could never have been attempted without reference to and inspiration from the works of others whose details are mentioned in reference section. We acknowledge out indebtedness to all of them. Last but not the least our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.

VIKASH KUMAR SINGH

ROLL NO. : 10506003

CONTENTS

PAGE NO.

Chapter 5

List Of Figures

Abstract

The Web is both an excellent medium for sharing information, as well as an attractive platform for delivering products and services. This platform is, to some extent, mediated by search engines in order to meet the needs of users seeking information. Search engines are the "dragons" that keep a valuable treasure: information.

Many web pages are unscrupulous and try to fool search engines to get to the top of ranking. The goal of this project is to detect such spam pages. We will particularly consider content spam and link spam, where untrusted pages use link structure to increase their importance. We pose this as a machine learning problem and build a classifier to classify pages into two category - trustworthy and untrusted .We use different link features, in other words structural characteristics of the web graph and content based features, as input to the classifier.

We propose link-based techniques and context based techniques for automating the detection of Web spam, a term referring to pages which use deceptive techniques to obtain undeservedly high scores in search engines. We propose Naïve Bayesian Classifier to detect the content Spam and PageRank and TrustRank to detect the link spam.

CHAPTER 1

INTRODUCTION

1.Introduction

The Web is both an excellent medium for sharing information, as well as an attractive platform for delivering products and services. This platform is, to some extent, mediated by search engines in order to meet the needs of users seeking information. Search engines are the “dragons” that keep a valuable treasure: information [Gori and Witten, 2005].

Given the vast amount of information available on the Web, it is customary to answer queries with only a small set of results (typically 10 or 20pages at most). Search engines must then rank Web pages, in order to create a short list of high-quality results for users. A large fraction of the visits to a Web site originate from search engines, and most of the users click on the first few results in a search engine. Therefore, there is an economic incentive for manipulating search engine’s listings. For instance, the authors of report that “**among the top 20 URLs in our 100 million page PageRank calculation 11 were pornographic, and these high positions appear to have all been achieved using the same form of link manipulation**”[1].

The objective of a search engine is to provide high quality results by correctly identifying all web pages that are relevant for a specific query, and presenting the user with some of the most important of those relevant pages. Relevance is usually measured through 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. Importance refers to the global(query-independent) popularity of a page, as often inferred from the link structure (e.g., pages with many incoming links are more important), or perhaps other indicators[2].

The term “**spam”** has been commonly used in the Internet era to refer to unsolicited (and possibly commercial) bulk messages. The most common form of electronic spam is e-mail spam, but in practice each new communication medium has created a new opportunity for sending unsolicited messages. These days there are many types of electronic spam, including spam by instant messaging (spim), spam by internet telephony (spit), spam by mobile phone, by fax, etc. The Web is not absent from this list, but as the request-response paradigm of the HTTP protocol makes it impossible for spammers to actually “send” pages directly to the users, spammers try to deceive search engines and thus break the trust that search engines establish with their users.

We use the term spamming (also, spamdexing) to refer to any deliberate human action that is meant to trigger an unjustifiably favorable relevance or importance for some web page, considering the page's true value. We will use the adjective spam to mark all those web objects (page content items or links) that are the result of some form of spamming.

1.1 How Google’s search engine works[9]

Google runs on a distributed network of thousands of low-cost computers and can therefore carry out fast parallel processing. Parallel processing is a method of computation in which many calculations can be performed simultaneously, significantly speeding up data processing. Google has three distinct parts:

- **Googlebot**, a web crawler that finds and fetches web pages.
- **The indexer** that sorts every word on every page and stores the resulting index of words in a huge database.
- **The query processor**, which compares your search query to the index and recommends the documents that it considers most relevant.

1.1.1 Googlebot, Google’s Web Crawler

Googlebot is Google’s web crawling robot, which finds and retrieves pages on the web and hands them off to the Google indexer. It’s easy to imagine Googlebot as a little spider scurrying across the strands of cyberspace, but in reality Googlebot doesn’t traverse the web at all. It functions much like your web browser, by sending a request to a web server for a web page, downloading the entire page, and then handing it off to Google’s indexer. Googlebot finds pages in two ways: through an add URL form, **www.google.com/addurl.html**, and through finding links by crawling on the web.

When Googlebot fetches a page, it culls all the links appearing on the page and adds them to a queue for subsequent crawling. Googlebot tends to encounter little spam because most web authors link only to what they believe are high-quality pages. By harvesting links from every page it encounters, Googlebot can quickly build a list of links that can cover broad reaches of the web. This technique, known as deep crawling, also allows Googlebot to probe deep within individual sites. Because of their massive scale, deep crawls can reach almost every page in the web.

1.1.2 Google's Indexer

Googlebot gives the indexer the full text of the pages it finds. These pages are stored in Google's index database. This index is sorted alphabetically by search term, with each index entry storing a list of documents in which the term appears and the location within the text where it occurs. This data structure allows rapid access to documents that contain user query terms.

To improve search performance, Google ignores (doesn't index) common words called *stop words* (such as *the, is, on, or, of, how, why,* as well as certain single digits and single letters). Stop words are so common that they do little to narrow a search, and therefore they can safely be discarded. The indexer also ignores some punctuation and multiple spaces, as well as converting all letters to lowercase, to improve Google's performance.

1.1.3 Google's Query Processor

The query processor has several parts, including the user interface (search box), the "engine" that evaluates queries and matches them to relevant documents, and the results formatter.

PageRank is Google's system for ranking web pages. A page with a higher PageRank is deemed more important and is more likely to be listed above a page with a lower PageRank.

Google considers over a hundred factors in computing a PageRank and determining which documents are most relevant to a query, including the popularity of the page, the position and size of the search terms within the page, and the proximity of the search terms to one another on the page.

Google also applies machine-learning techniques to improve its performance automatically by learning relationships and associations within the stored data. For example, the **spelling-correcting system** uses such techniques to figure out likely alternative spellings. Google closely guards the formulas it uses to calculate relevance; they're tweaked to improve quality and performance, and to outwit the latest devious techniques used by spammers.

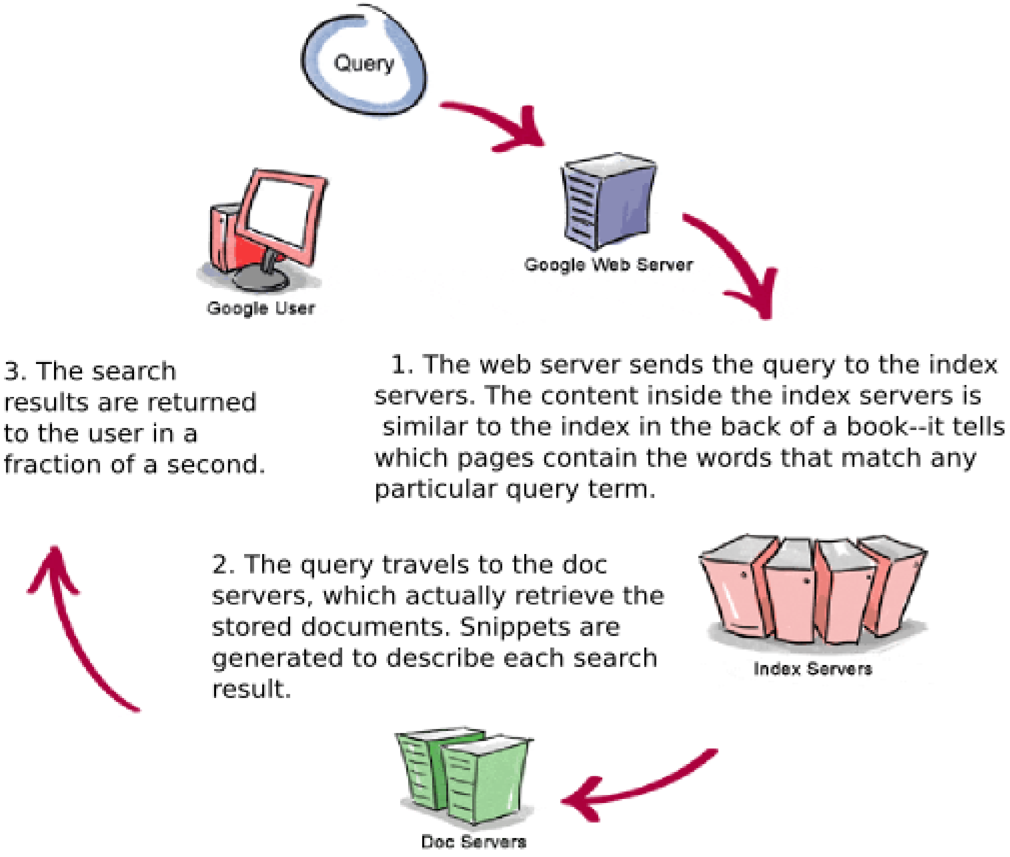

Query
Google User
Google Web Server
1. The web server sends the query to the index servers. The content inside the index servers is similar to the index in the back of a book--it tells which pages contain the words that match any particular query term.
3. The search results are returned to the user in a fraction of a second.
2. The query travels to the doc servers, which actually retrieve the stored documents. Snippets are generated to describe each search result.
Index Servers
Doc Servers

Figure 2 How the Google search engine works[9]

1.2 Google and Web Spam

Search has become the default gateway to the web. So every single website wants to come on the first page of the search results because it is the human tendency to click on the first few links on the first page of search result. So they manipulate the search engine algorithms to come on the first page.

All deceptive actions which try to increase the ranking of a page in search engines are generally referred to as Web spam or spamdexing (a portmanteau, or combination of “spam” and “index”). A spam page or host is a page or host that is either used for spamming or receives a substantial amount of its score from other spam pages. An alternative way of defining Web spam could be any attempt to get “an unjustifiably favorable relevance or importance score for some web page, considering the page’s true value”. A spam page is a page which is used for spamming or receives a substantial amount of its score from other spam pages. Another definition of spam, given in is “any attempt to deceive a search engine’s relevancy algorithm” or simply “anything that would not be done if search engines did not exist”. Seeing as there are many steps which content providers can take to improve the ranking of their Web sites, and given that is an important subjective element in the evaluation of the relevance of Web pages, to offer an exact definition of Web spam would be misleading.

Indeed, there is a large gray area between “ethical” **Search Engine Optimization (SEO)** services and “unethical” spam [1]. SEO services range from ensuring that Web pages are index able by Web crawlers, to the creation of thousands or millions of fake pages aimed at deceiving search engine ranking algorithms. Our main criteria for deciding in borderline cases is the perceived effort spent by Web authors on providing good content, against the effort spent on trying to score highly in search engines. There are three types of web spam on the web .They are:-

- **Content spam**: maliciously crafting the content of Web pages.
- **Link spam**: creation of a link structure aimed at affecting the outcome of a link-based ranking algorithm.
- **Cloaking**: sending different content to a search engine than to the regular visitors of a web site.

CHAPTER 2

The World of Web Spam: A Study

2.1 Types of Web Spam

2.2 Difference in Spam and Non-Spam Pages

The World of Web Spam: A Study

Spamdexing (also known as **search spam** or **search engine spam or Web Spam**) involves a number of methods, such as repeating unrelated phrases, to manipulate the relevancy or prominence of resources indexed by a search engine, in a manner inconsistent with the purpose of the indexing system. Some consider it to be a part of search engine optimization, though there are many search engine optimization methods that improve the quality and appearance of the content of web sites and serve content useful to many users. Search engines use a variety of algorithms to determine relevancy ranking. Some of these include determining whether the search term appears in the **META keywords** tag, others whether the search term appears in the body text or URL of a web page. Many search engines check for instances of spamdexing and will remove suspect pages from their indexes. Also, people working for a search-engine organization can quickly block the results-listing from entire websites that use spamdexing, perhaps alerted by user complaints of false matches. The rise of spamdexing in the mid-1990s made the leading search engines of the time less useful.

The success of **Google** at both producing better search results and combating keyword spamming, through its reputation-based **PageRank link analysis system**.

2.1 Types of Web Spam

There are many techniques for Web spam and they can be broadly classified in three groups: content (or keyword) spam, link spam and cloaking [1].

Content spam refers to changes in the content of the pages, for instance by inserting a large number of keywords; 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, amongst 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.

Link spam includes changes to the link structure of the sites, by creating link farms .A link farm is a densely connected set of pages, created explicitly with the purpose of deceiving a link based ranking algorithm. Define this form of collusion as 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".**

Cloaking includes sending different content to a search engine than to the regular visitors of a web site. If spammers can clearly identify web crawler clients, they can adopt the following strategy, called cloaking. Given a URL, spam web servers return one specific HTML document to a regular web browser, while they return a different document to a web crawler. This way, spammers can present the ultimately intended content to the web users (without traces of spam on the page),and, at the same time, send a spammed document to the search engine for indexing.

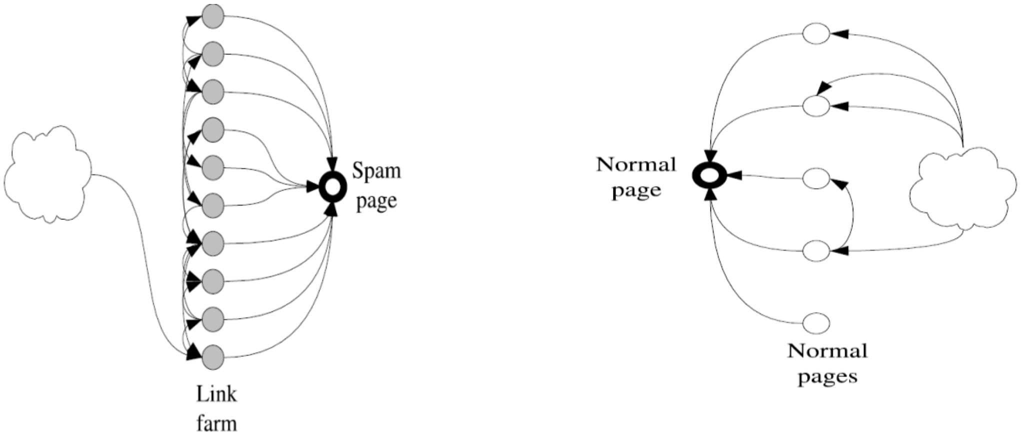
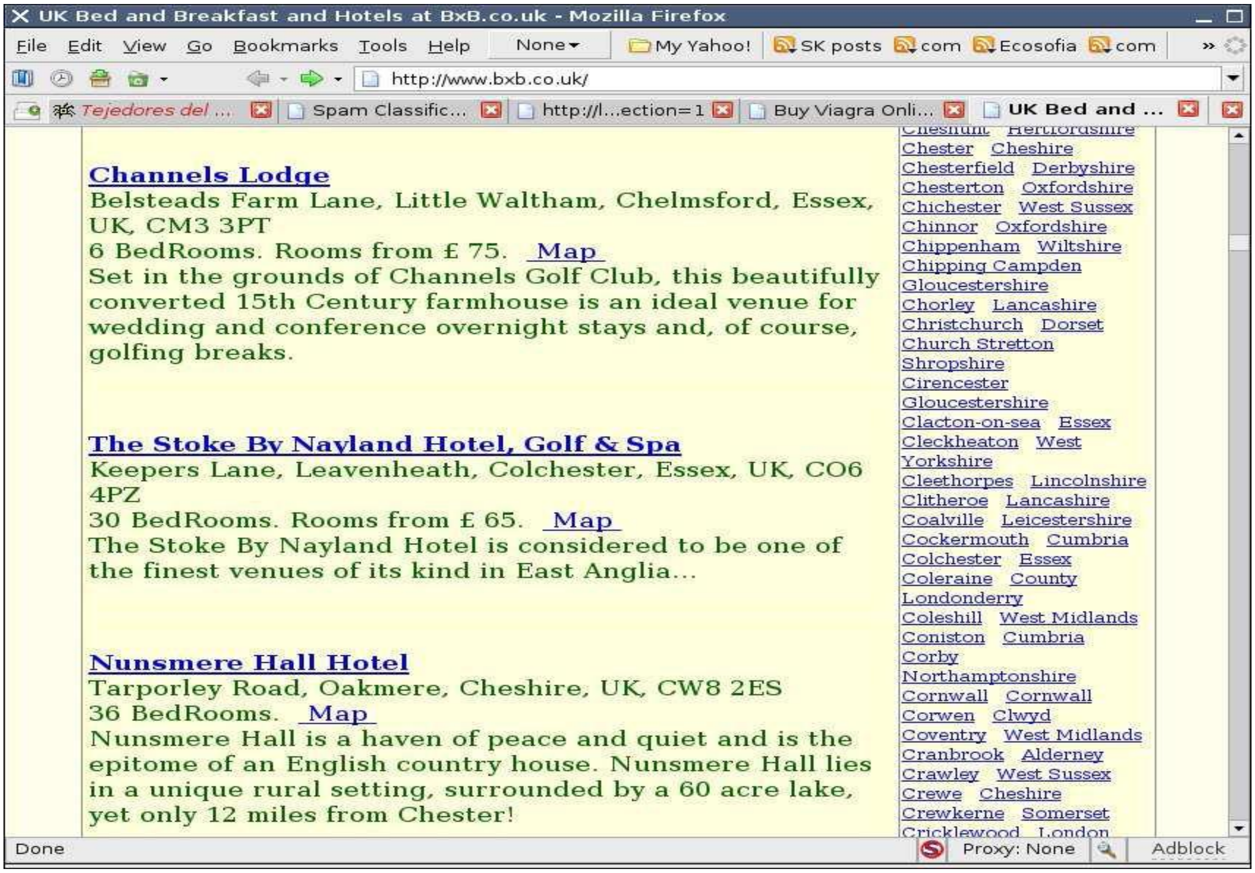


Figure 2 Link Based Web Spam[1]


UK Bed and Breakfast and Hotels at BxB.co.uk - Mozilla Firefox
File Edit View Go Bookmarks Tools Help None
My Yahoo! SK posts com Ecosofia com
http://www.bxb.co.uk/
Tejedores del ... Spam Classific... http://l...ection=1 Buy Viagra Onli... UK Bed and ...
Done
Proxy: None
Adblock

Channels Lodge
Belsteads Farm Lane, Little Waltham, Chelmsford, Essex, UK, CM3 3PT
6 BedRooms. Rooms from £ 75. Map
Set in the grounds of Channels Golf Club, this beautifully converted 15th Century farmhouse is an ideal venue for wedding and conference overnight stays and, of course, golfing breaks.

The Stoke By Nayland Hotel, Golf & Spa
Keepers Lane, Leavenheath, Colchester, Essex, UK, CO6 4PZ
30 BedRooms. Rooms from £ 65. Map
The Stoke By Nayland Hotel is considered to be one of the finest venues of its kind in East Anglia...

Nunsmere Hall Hotel
Tarporley Road, Oakmere, Cheshire, UK, CW8 2ES
36 BedRooms. Map
Nunsmere Hall is a haven of peace and quiet and is the epitome of an English country house. Nunsmere Hall lies in a unique rural setting, surrounded by a 60 acre lake, yet only 12 miles from Chester!

Cheshunt Hertfordshire
Chester Cheshire
Chesterfield Derbyshire
Chesterton Oxfordshire
Chichester West Sussex
Chinnor Oxfordshire
Chippenham Wiltshire
Chipping Campden
Gloucestershire
Chorley Lancashire
Christchurch Dorset
Church Stretton
Shropshire
Cirencester
Gloucestershire
Clacton-on-sea Essex
Cleckheaton West
Yorkshire
Cleethorpes Lincolnshire
Clitheroe Lancashire
Coalville Leicestershire
Cockermouth Cumbria
Colchester Essex
Coleraine County
Londonderry
Coleshill West Midlands
Coniston Cumbria
Corby
Northamptonshire
Cornwall Cornwall
Corwen Clwyd
Coventry West Midlands
Cranbrook Alderney
Crawley West Sussex
Crewe Cheshire
Crewkerne Somerset
Cricklewood London

Figure 3 : Content Based Spam Page [1]

Link-based analysis does not capture all possible cases of spamming, since some spam pages appear to have spectral and topological properties that are statistically close to those exhibited by non spam pages. In this case, content-based analysis can prove extremely useful.

On the other hand, content-based analysis seems less resilient to changes in spammers strategies, in much the same way that content-based techniques for detecting email spamming are. For instance, a spammer could copy an entire Web site (creating a set of pages that may be able to pass all tests for content spam detection) and change a few out-links in every page to point to the target page. This may be a relatively inexpensive task to perform in an automatic way, whereas creating, maintaining, reorganizing a link farm, possibly spanning more than one domain, is economically more expensive.

It has been proposed that when content and link based web spam detection algorithm are used together they give the best result in web spam detection [1].

2.2 Difference in Spam and Non-Spam Pages

The basic difference in spam and non-spam pages are based on the following factors:-

- **Average Word Length**

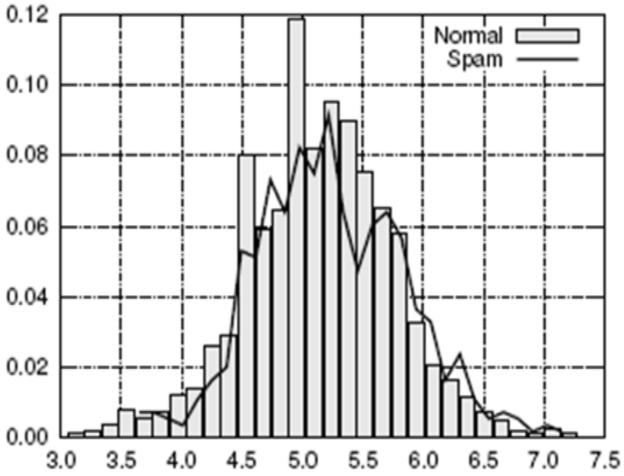


Figure 4 Histogram of the average word length of the spam and non-spam page[3]

- **Number Of Words**

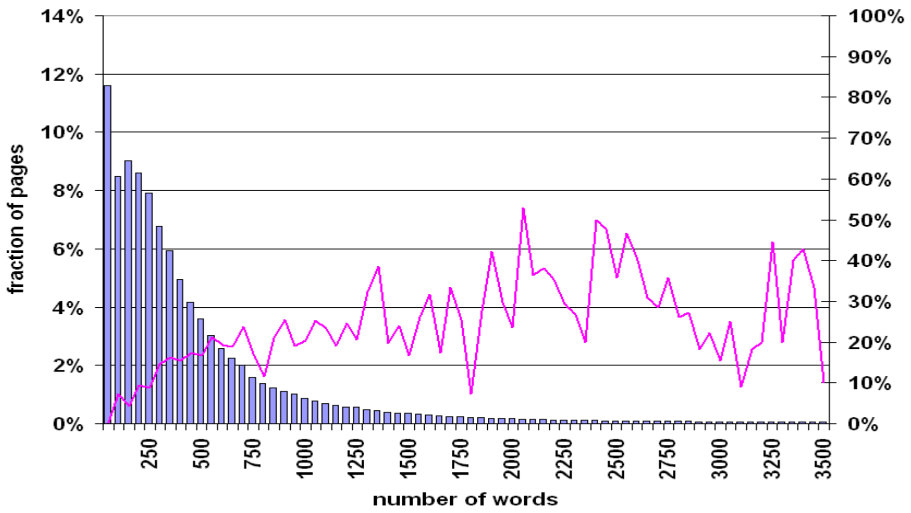
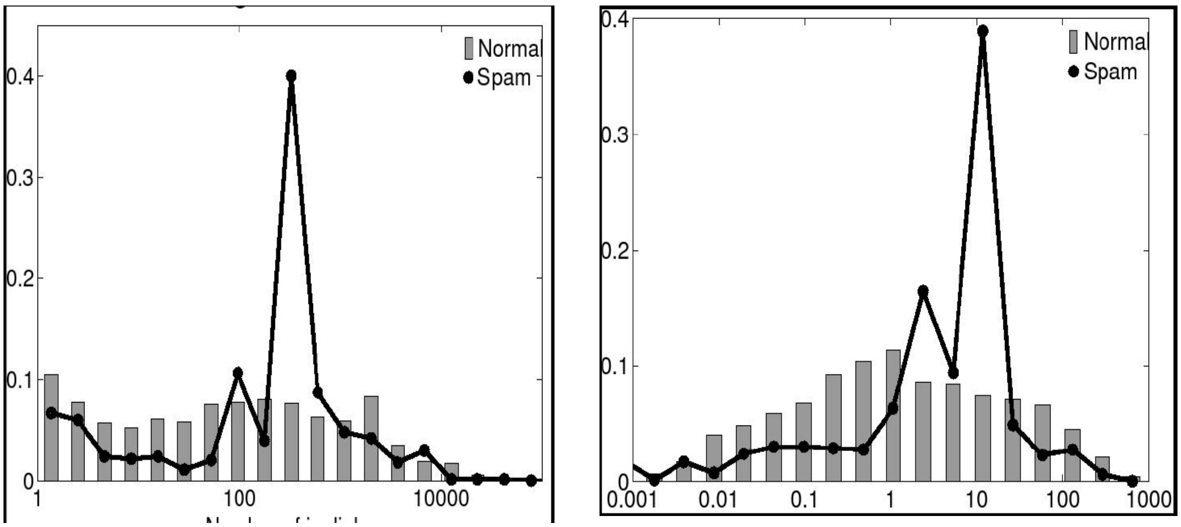


Figure 5 Histogram Representing number of words between spam vs non spam pages[3]

- **In-Degree and Edge Reciprocity**



In Degree

Edge Reciprocity

Figure 6 : In degree and Edge Reciprocity of Spam and Non-Spam Page [3]

CHAPTER 3

Web Spamming Techniques

Web Spamming Techniques

There are two categories of techniques associated with web spam. The first 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. The following two sections discuss each of these two categories in more detail.

3.1 Boosting Techniques

In this section we present spamming techniques that influence the ranking algorithms used by search engines. Boosting techniques are the methods used by the web spam to increase their PageRank by link farm manipulation or by term spamming. There are two types of boosting techniques they are **Term Spamming** and **Link Spamming [2].**

3.1.1 Term Spamming

In evaluating textual relevance, search engines consider where on a web page query terms occurs. Each type of location is called a field. The common text fields for a page p are the document body, the title, the meta tags in the HTML header, and page p's URL. In addition, the anchor texts associated with URLs that point to p are also considered belonging to page p (anchor text field), since they often describe very well the content of p. The terms in p's text fields are used to determine the relevance of p with respect to a specific query (a group of query terms), often with different weights given to different fields. Term spamming refers to techniques that tailor the contents of these text fields in order to make spam pages relevant for some queries. There are different techniques by which term spamming is done. Term spamming techniques can be grouped based on the text field in which the spamming occurs. Therefore, we distinguish it into:

- **Body spam:** In this case, the spam terms are included in the document body. This spamming technique is among the simplest and most popular ones, and it is almost as old as search engine.
- **Title spam:** Today's search engines usually give a higher weight to terms that appear in the title of a document. Hence, it makes sense to include the spam terms in the document title.
- **Meta tag spam:** The HTML meta tags that appear in the document header have always been the target of spamming. Because of the heavy spamming, search engines currently give low priority to these tags, or even ignore them completely. Here is a simple example of a spammed keywords meta tag:

  <meta name=\keywords" content=\buy, cheap, cameras, lens, accessories, nikon, canon">

- **Anchor text spam:** Just as with the document title, search engines assign higher weight to anchor text terms, as they are supposed to over a summary of the pointed document. Therefore, spam terms are sometimes included in the anchor text of the HTML hyperlinks to a page. This spamming technique is different from the previous ones, in the sense that the spam terms are added not to a target page itself, but the other pages that point to the target. As anchor text gets indexed for both pages, spamming it has impact on the ranking of both the source and target pages. A simple anchor text spam is:

  <a href=\target.html">free, great deals, cheap, inexpensive, cheap, free</a>

- **URL spam:** Some search engines also break down the URL of a page into a set of terms that are used to determine the relevance of the page. To exploit this, spammers sometimes create long URLs that include sequences of spam terms. For instance, one could encounter spam URLs like:

buy-canon-rebel-20d-lens-case.camerasx.com,
buy-nikon-d100-d70-lens-case.camerasx.com,

Another way of grouping term spamming techniques is based on the type of terms that are added to the text fields. Correspondingly, we have [2] :

- **Repetition of one or a few specific terms** :This way spammers achieve an increased relevance for a document with respect to a small number of query terms.
- **Dumping of a large number of unrelated terms**[often even entire dictionaries]: This way, spammers make a certain page relevant to many different queries. Dumping is effective against queries that include relatively rare, obscure terms: for such queries, it is probable that only a couple of pages are relevant, so even a spam page with a low relevance or importance would appear among the top results.
- **Weaving of spam terms into copied contents** : Sometimes spammers duplicate text corpora (e.g., news articles) available on the Web and insert spam terms into them at random positions. This technique is effective if the topic of the original real text was so rare that only a small number of relevant pages exist. Weaving is also used for dilution, i.e., to conceal some repeated spam terms within the text, so that search engine algorithms that filters out plain repetition would be misled. A short example of spam weaving is:

Remember not only airfare to say the right plane
tickets thing in the right place, but far cheap travel
more difficult still, to leave hotel rooms unsaid the
wrong thing at vacation the tempting moment.

3.1.2 Link Spamming Techniques

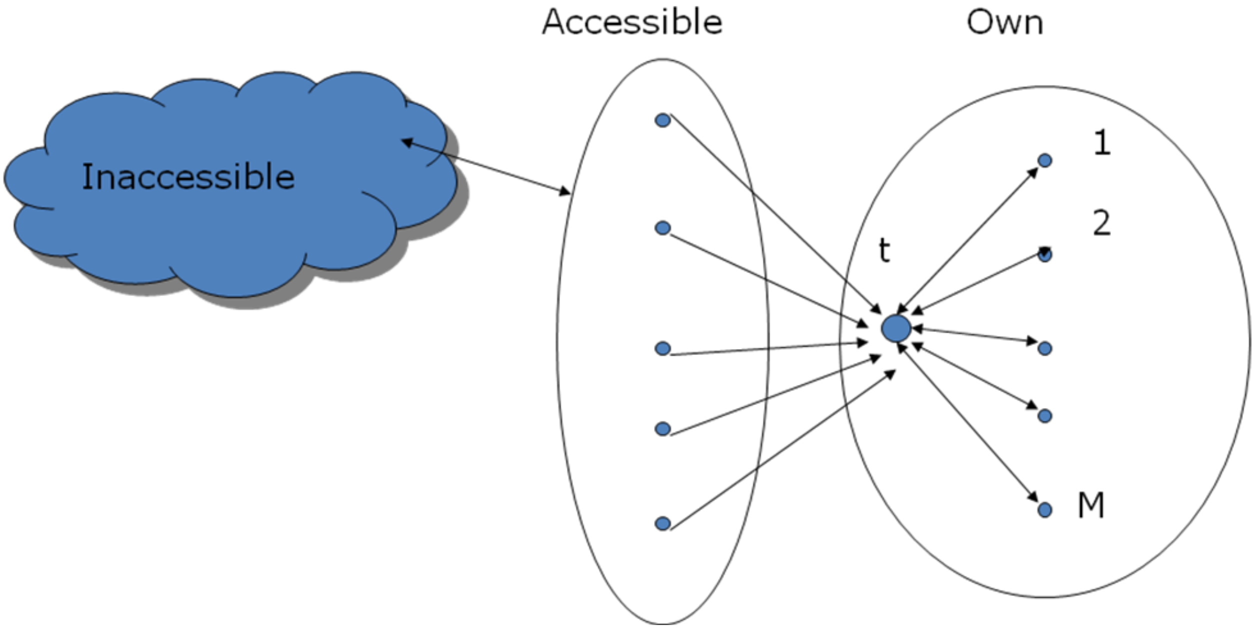
Beside term-based relevance metrics, search engines also rely on link information to determine the importance of web pages. Therefore, spammers often create link structures that they hope would increase the importance of one or more of their pages. PageRank Algorithm uses the link structure to calculate the page relevance.

For our discussion of the algorithms targeted by link spam, we will adopt the following model. For a spammer, there are three types of pages on the Web:

- **Inaccessible pages** are those that a spammer cannot modify. These are the pages out of reach, the spammer cannot influence their outgoing links.(Note that a spammer can still point to inaccessible pages.)
- **Accessible pages** are maintained by others (presumably not affiliated with the spammer), but can still be modified in a limited way by a spammer.For example, a spammer may be able to post a comment to a blog entry, and that comment may contain a link to a spam site. As infiltrating accessible pages is usually not straightforward, let us say that a spammer has a limited budget of m accessible pages. For simplicity, we assume that at most one outgoing link can be added to each accessible page.
- **Own pages** are maintained by the spammer, who thus has full control over their contents. We call the group of own pages a spam farm ∑. A spammer's goal is to boost the importance of one or more of his or her own pages. For simplicity, say there is a single target page t. There is a certain maintenance cost (domain registration, web hosting) associated with a spammer's own pages, so we can assume that a spammer has a limited budget of n such pages, not including the target page.

With this model in mind, we discuss the algorithm used to compute importance scores based on link information by Google and that is **PageRank** Algorithm.

PageRank uses incoming link information to assign global importance scores to all pages on the Web. It assumes that the number of incoming links to a page is related to that page's popularity among average web users (people would point to pages that they find important). The intuition behind the algorithm is that a web page is important if several other important web pages point to it. Correspondingly, PageRank is based on a mutual reinforcement between pages: the importance of certain page influences and is being influenced by the importance of some other pages.



.**Figure 7 Types of Pages on the Web [2]**

A recent analysis of the algorithm showed that the total PageRank score PR(t) of a group t of pages depends on four factors:

PR(t) = PRstatic(t)+PRin(t)-PRout(t)-PRsink(t)

where PRstatic is the score component due to the static score distribution (random jump) , PRin is the score received through the incoming links from external pages , PRout is the score leaving through the outgoing links to external pages , and PRsink is the score loss due to those pages within the group that have no outgoing links.

Suppose rank contributed by accessible pages = x
Let page rank of target page = y

Rank of each “farm” page = $by/M + (1-b)/N$

$y = x + \beta M[by/M + (1-b)/N] + (1-b)/N$

$\quad = x + b^2y + b(1-b)M/N + (1-b)/N$

$y = x/(1-b^2) + cM/N$ where $c = \beta/(1+\beta)$

$y = x/(1-b^2) + cM/N$ where $c = \beta/(1+\beta)$

For $b = 0.85$, $1/(1-b^2)= 3.6$

Thus getting Multiplier effect for “acquired” page rank thus we can see that by making M large, we can make y(PageRank) as large as we want.

3.2 Hiding Techniques

It is usual for spammers to conceal the telltale signs (e.g., repeated terms, long lists of links) of their activities. They use a number of techniques to hide their abuse from regular web users visiting spam pages, or from the editors at search engine companies who try to identify spam instances. This section offers an overview of the most common spam hiding techniques

3.2.1 Content Hiding

Spam terms or links on a page can be made invisible when the browser renders the page. One common technique is using appropriate color schemes: terms in the body of an HTML document are not visible if they are displayed in the same color as the background. Color schemes can be defined either in the HTML document or in an attached cascading style sheet (CSS). We show a simple HTML example next:

```
<body background=\white">
<font color=\white">hidden text</font>
: : :
</body>
```

In a similar fashion, spam links can be hidden by avoiding anchor text. Instead, spammers often create tiny, 1*1-pixel anchor images that are either transparent or background-colored:

```
<a href=\target.html"><img src=\tinyimg.gif"></a>
```

A spammer can also use scripts to hide some of the visual elements on the page, for instance, by setting the visible HTML style attribute to false.

3.3 Cloaking

If spammers can clearly identify web crawler clients, they can adopt the following strategy, called cloaking Given a URL, spam web servers return one specific HTML document to a regular web browser, while they return a different document to a web crawler. This way,spammers can present the ultimately intended content to the web users (without traces of spam on the page),and, at the same time, send a spammed document to the search engine for indexing.

The identification of web crawlers can be done in two ways. On one hand, some spammers maintain a list of IP addresses used by search engines, and identify the web crawlers based on their matching IPs. On the other hand, a web server can identify the application requesting a document based on the user-agent field in the HTTP request message. For instance, in the following simple HTTP request message the user-agent name is that one used by the Microsoft Internet Explorer 6 browser:

```
GET /db pages/members.html HTTP/1.0
Host: www-db.stanford.edu
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win-
dows NT 5.1)
```

The user-agent names are not strictly standardized, and it is really up to the requesting application what to include in the corresponding message field. Nevertheless, search engine crawlers identify themselves by a name distinct from the ones used by traditional web browser applications.

CHAPTER 4

Web Spam Detection and Results

Web Spam Detection and Results

The foundation of our spam detection system is a cost sensitive decision tree. The features used to learn the tree were derived from a combined approach based on link and content analysis to detect different types of Web spam pages. Most of features we are going to use are based on our attempt to combine both link-based and content-based features.

Evaluation: The evaluation of the overall process is based on a set of measures commonly used in Machine Learning and Information Retrieval and focused on the spam detection task. Given a classification algorithm C, we consider its confusion matrix:

True Label /Prediction	**Non Spam**	**Spam**
Non-Spam	a	b
Spam	c	d

Where a represents the number of non-spam examples that were correctly classified, b represents the number of non-spam examples that were falsely classified as spam, c represents the spam examples that were falsely classified as non-spam, and d represents the number of spam examples that were correctly classified. We consider the following measures:

- True positive rate, or recall: R = d/(c+d)
- False positive rate: b/(b+a)
- F-measure: F = 2PR/(P+R) where P is the precision P=d/(b+d)

.

For evaluating the classification algorithms, we focus on the F-measure F as it is a standard way of summarizing both precision and recall. We also report the true positive rate and false positive rate as they have a direct interpretation in practice. The true positive rate R is the amount of spam that is detected (and thus deleted or demoted) by the Search engine. The false positive rate is the fraction of non-spam objects that are mistakenly considered to be spam by the automatic classifier. We divide our Spam Detection Method on the basis of Content Based and Link Based attributes.

4.1 Based On Content Based Features

For each web page in our data set we extract a number of features based on the content of the pages. A more detailed discussion of each feature is presented here [3]:-

- **Number of words in the page:** Number of words in the title, average word length. For these features we count only the words in the visible text of a page, and we consider words consisting only of alphanumeric characters.
- **Fraction of anchor text**: Fraction of the number of words in the anchor text to the total number of words in the visible text.
- **Fraction of visible text**: Fraction of the number of words in the visible text to the total number of words in the page, include html tags and other invisible text.

A comparative study of the above mentioned content based features in Figure 4 and Figure 5 show following results:-

- Average word length in Spam vs Non-Spam Page- Average Word Length in Spam pages are much higher in spam pages(Figure 4)
- Number of words in Spam vs Non-Spam page : Number of words in spam page is much higher than non-spam page (Figure 5)

Thus based on the following features the content based spam pages can be detected by Naïve Bayesian Classifier which focuses on the no of times a word is repeated in the content of the page.

4.1.1 Naïve Bayesian Classifier

A **Naive Bayesian classifier** is a term in Bayesian statistics dealing with a simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even though these features depend on the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers often work much better in many complex real-world situations than one might expect. Recently, careful analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers. An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The Naïve Bayesian Probabilistic Model

Abstractly, the probability model for a classifier is a conditional model

$$\boldsymbol{p(C|F_1,\dots.,F_n)}$$

over a dependent class variable C with a small number of outcomes or *classes*, conditional on several feature variables F_1through F_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

$$p(C|F_{1,\dots.,}F_n)= \frac{\mathrm{p(C)\ p(F1,.\,..,Fn\mid C)}}{\mathrm{p(F1,\dots.,Fn)}}$$

In plain English the above equation can be written as

$$\boldsymbol{posterior} = \frac{prior \times likelihood}{evidence}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$\boldsymbol{p(C|F_1,\dots.,F_n)}$$

which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$
\begin{aligned}
&p(C, F_1, \dots, F_n) \\
&= p(C)\ p(F_1, \dots, F_n|C) \\
&= p(C)\ p(F_1|C)\ p(F_2, \dots, F_n|C, F_1) \\
&= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \dots, F_n|C, F_1, F_2) \\
&= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ p(F_4, \dots, F_n|C, F_1, F_2, F_3) \\
&= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ \dots p(F_n|C, F_1, F_2, F_3, \dots, F_{n-1})
\end{aligned}
$$

and so forth. Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$. This means that

$$p(F_i|C, F_j) = p(F_i|C)$$

and so the joint model can be expressed as

$$
\begin{aligned}
&p(C, F_1, \dots, F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots \\
&= p(C) \prod_{i=1}^{n} p(F_i|C).
\end{aligned}
$$

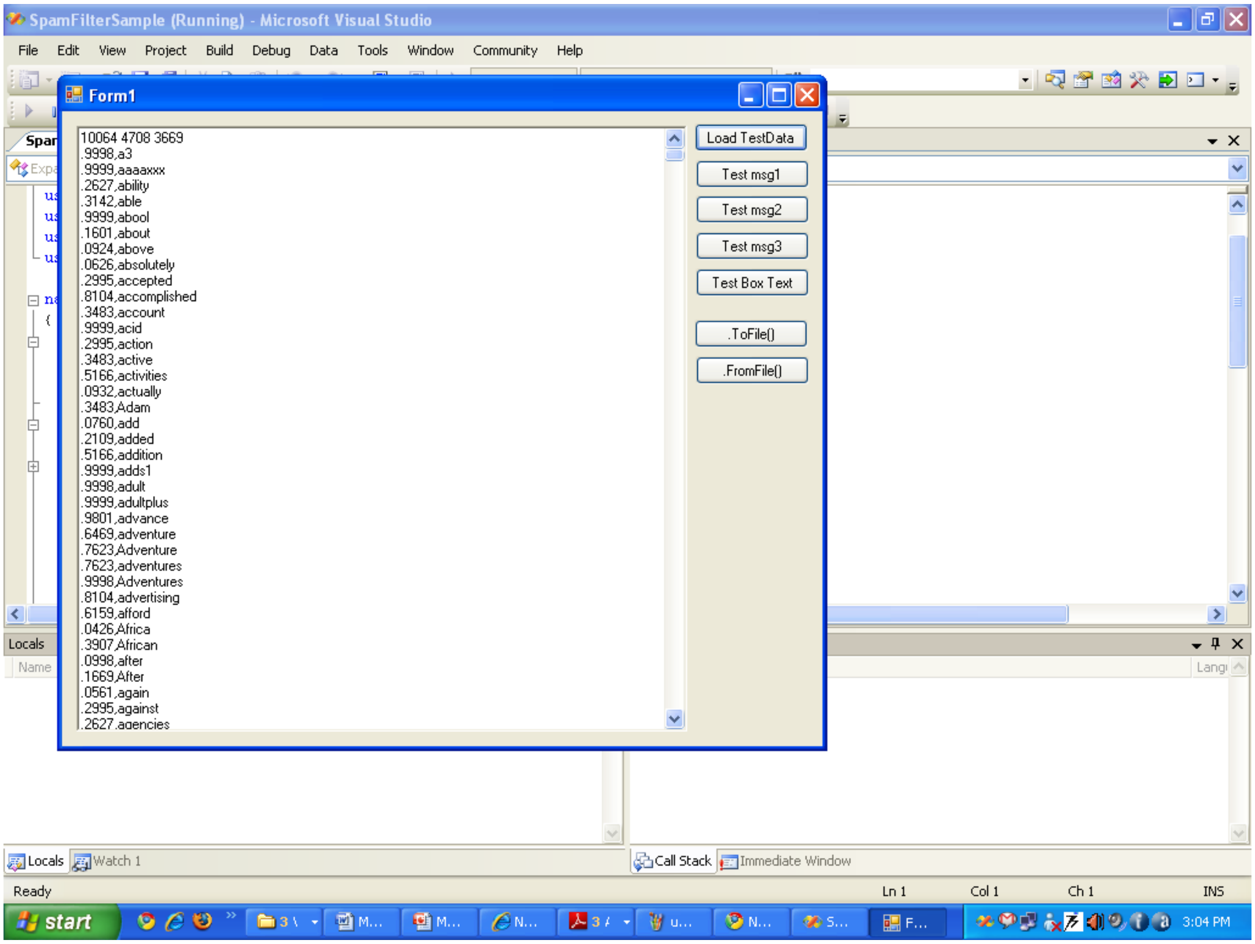
This means that under the above independence assumptions, the conditional distribution over the class variable C can be expressed like this:

$$p(C|F_1, \dots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

where Z is a scaling factor dependent only on $F_1, \dots, F_n$, i.e., a constant if the values of the feature variables are known.

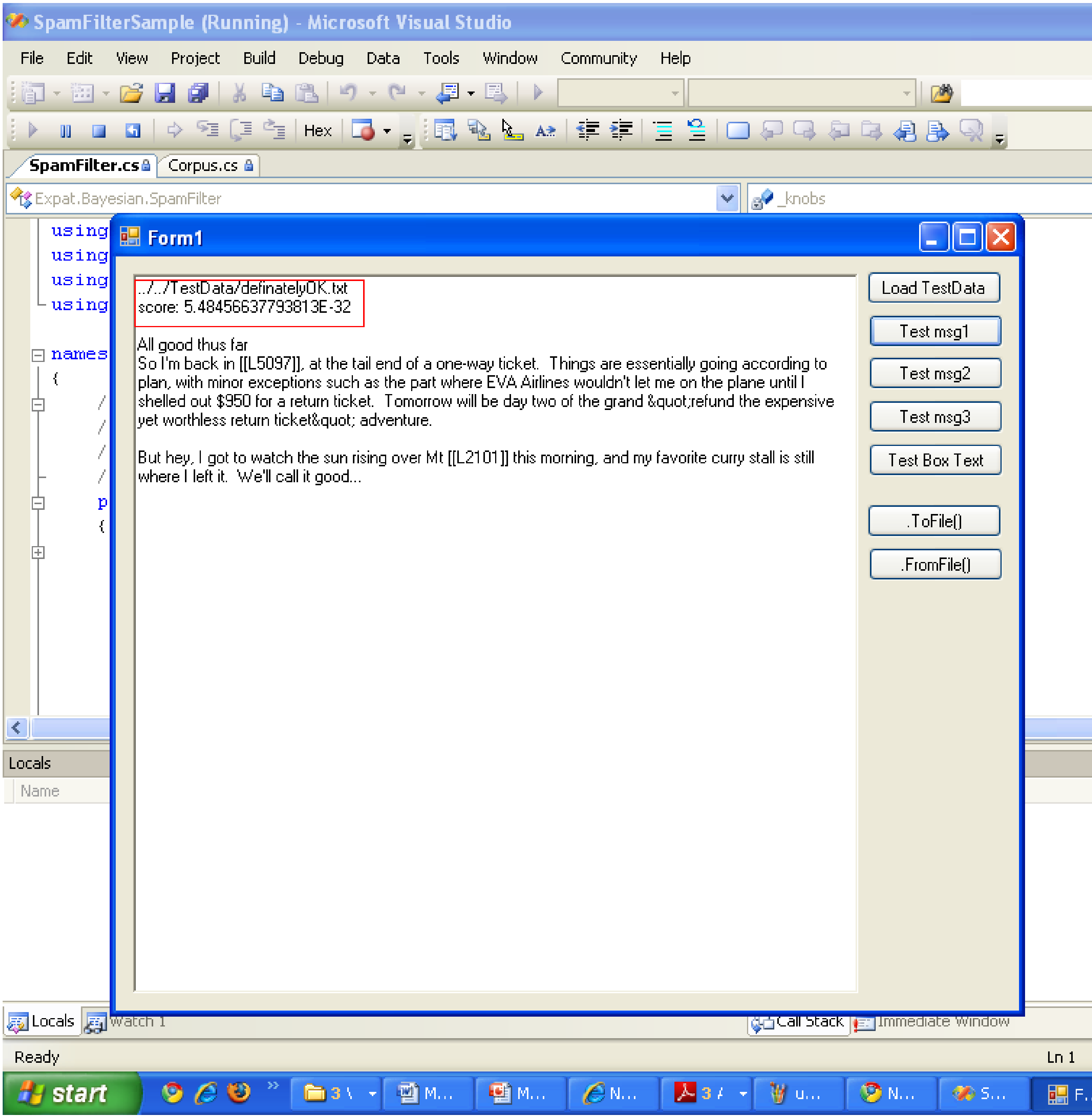
Models of this form are much more manageable, since they factor into a so-called *class prior* $p(C)$ and independent probability distributions $p(F_i|C)$. If there are k classes and if a model for $p(F_i)$ can be expressed in terms of r parameters, then the corresponding naive Bayes model has $(k - 1) + n\, r\, k$ parameters. In practice, often $k = 2$ (binary classification) and $r = 1$ (Bernoulli variables as features) are common, and so the total number of parameters of the naive Bayes model is $2n + 1$, where n is the number of binary features used for prediction.

4.1.2 Results Of the Naïve Bayesian Classifier



```
10064 4708 3669
.9998,a3
.9999,aaaaxxx
.2627,ability
.3142,able
.9999,abool
.1601,about
.0924,above
.0626,absolutely
.2995,accepted
.8104,accomplished
.3483,account
.9999,acid
.2995,action
.3483,active
.5166,activities
.0932,actually
.3483,Adam
.0760,add
.2109,added
.5166,addition
.9999,adds1
.9998,adult
.9999,adultplus
.9801,advance
.6469,adventure
.7623,Adventure
.7623,adventures
.9998,Adventures
.8104,advertising
.6159,afford
.0426,Africa
.3907,African
.0998,after
.1669,After
.0561,again
.2995,against
.2627,agencies
```

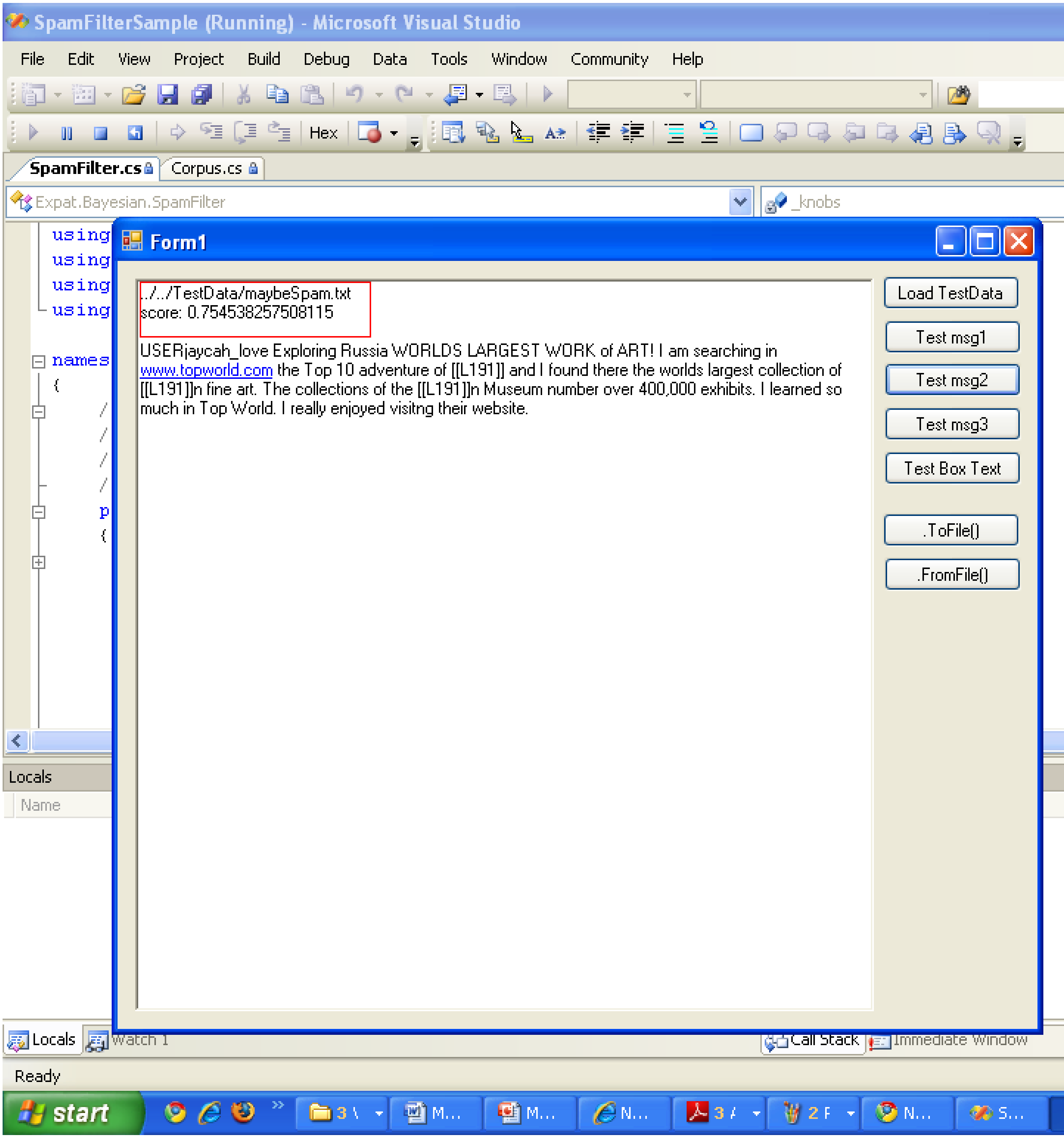
Figure 8 depicts first the test data is being inputted to the spam filter



```
using
using
using
using

names
{
```

```
../../TestData/definatelyOK.txt
score: 5.48456637793813E-32

All good thus far
So I'm back in [[L5097]], at the tail end of a one-way ticket.  Things are essentially going according to
plan, with minor exceptions such as the part where EVA Airlines wouldn't let me on the plane until I
shelled out $950 for a return ticket.  Tomorrow will be day two of the grand &quot;refund the expensive
yet worthless return ticket&quot; adventure.

But hey, I got to watch the sun rising over Mt [[L2101]] this morning, and my favorite curry stall is still
where I left it.  We'll call it good...
```

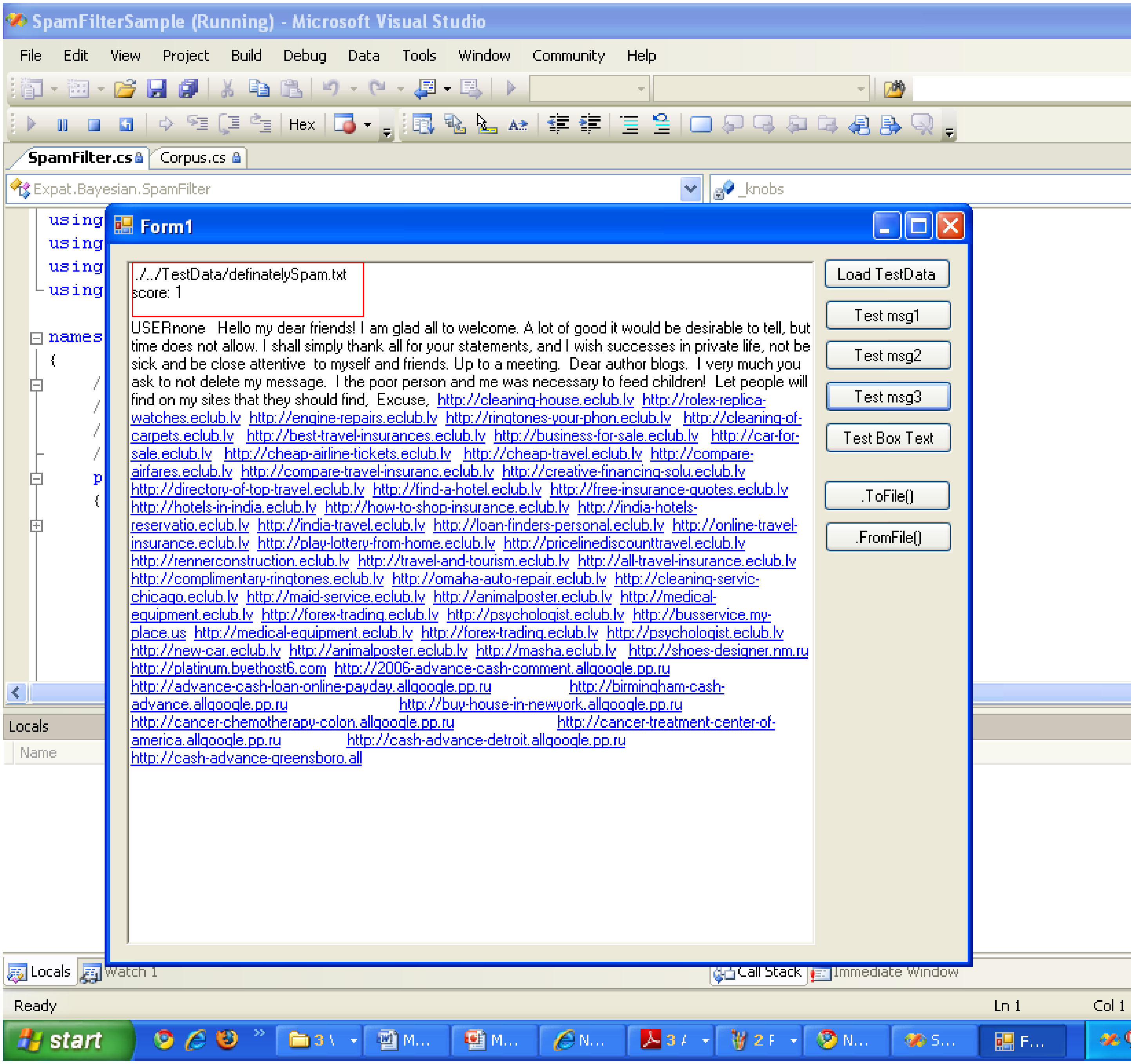
Figure 9 : This shows a normal page having the words normal, meaningful and relevant to its context. It is a Definitely OK page.



```
../../TestData/maybeSpam.txt
score: 0.754538257508115
```

USERjaycah_love Exploring Russia WORLDS LARGEST WORK of ART! I am searching in www.topworld.com the Top 10 adventure of [[L191]] and I found there the worlds largest collection of [[L191]]n fine art. The collections of the [[L191]]n Museum number over 400,000 exhibits. I learned so much in Top World. I really enjoyed visitng their website.

Figure 10 : This shows the a page which can be suspicious and it can be a spam having score 0.7.



../../TestData/definatelySpam.txt
score: 1

USERnone   Hello my dear friends! I am glad all to welcome. A lot of good it would be desirable to tell, but time does not allow. I shall simply thank all for your statements, and I wish successes in private life, not be sick and be close attentive to myself and friends. Up to a meeting. Dear author blogs. I very much you ask to not delete my message. I the poor person and me was necessary to feed children! Let people will find on my sites that they should find. Excuse, http://cleaning-house.eclub.lv http://rolex-replica-watches.eclub.lv http://engine-repairs.eclub.lv http://ringtones-your-phon.eclub.lv http://cleaning-of-carpets.eclub.lv http://best-travel-insurances.eclub.lv http://business-for-sale.eclub.lv http://car-for-sale.eclub.lv http://cheap-airline-tickets.eclub.lv http://cheap-travel.eclub.lv http://compare-airfares.eclub.lv http://compare-travel-insuranc.eclub.lv http://creative-financing-solu.eclub.lv http://directory-of-top-travel.eclub.lv http://find-a-hotel.eclub.lv http://free-insurance-quotes.eclub.lv http://hotels-in-india.eclub.lv http://how-to-shop-insurance.eclub.lv http://india-hotels-reservatio.eclub.lv http://india-travel.eclub.lv http://loan-finders-personal.eclub.lv http://online-travel-insurance.eclub.lv http://play-lottery-from-home.eclub.lv http://pricelinediscounttravel.eclub.lv http://rennerconstruction.eclub.lv http://travel-and-tourism.eclub.lv http://all-travel-insurance.eclub.lv http://complimentary-ringtones.eclub.lv http://omaha-auto-repair.eclub.lv http://cleaning-servic-chicago.eclub.lv http://maid-service.eclub.lv http://animalposter.eclub.lv http://medical-equipment.eclub.lv http://forex-trading.eclub.lv http://psychologist.eclub.lv http://busservice.my-place.us http://medical-equipment.eclub.lv http://forex-trading.eclub.lv http://psychologist.eclub.lv http://new-car.eclub.lv http://animalposter.eclub.lv http://masha.eclub.lv http://shoes-designer.nm.ru http://platinum.byethost6.com http://2006-advance-cash-comment.allgoogle.pp.ru http://advance-cash-loan-online-payday.allgoogle.pp.ru http://birmingham-cash-advance.allgoogle.pp.ru http://buy-house-in-newyork.allgoogle.pp.ru http://cancer-chemotherapy-colon.allgoogle.pp.ru http://cancer-treatment-center-of-america.allgoogle.pp.ru http://cash-advance-detroit.allgoogle.pp.ru http://cash-advance-greensboro.all

Figure 11 This shows a spam page in which key words are stuffed without putting any meaning in its context to get indexed into the Google indexer. Thus it is a spam page having score 1.

4.2 Based on Link Based Features

We view our set of Web pages as a Web graph, that is, a graph G = (V,E) in which the set V corresponds to Web pages in a subset of the Web, and every link (x, y) ∈ E corresponds to a hyperlink from page x to page y in the collection. For concreteness, the total number of nodes N = |V | in the full Web index able by search engines is in the order of 10^{10}, and the typical number of links per Web page is between 20 and 30.

For the link based web spam detection the following steps are followed [1]:-

- Data set is obtained by using web crawler.
- For each page, links and its contents are obtained.
- From data set, a full graph is built.
- For each host and page, certain features are computed.
- Link-based features are extracted from host graph.

Link Based classifier operates on the three features of the link farm which are as follows[1]:-

4.2.1 Based on the Estimation of Supporters

Link analysis algorithms assume that every link represents an endorsement, in the sense that if there is a link from page x to page y, then the author of page x is recommending page y. Following, we call x a supporter of page y at distance d, if the shortest path from x to y formed by links in E has length d. The set of supporters of a page are all the other pages that contribute towards its link-based ranking.

A particular characteristic of a link farm is that the spam pages might have a large number of distinct supporters at short distances, but this number should be lower than expected at higher distances .We can see that the number of new distinct supporters increases up to a certain distance, and then decreases, as the graph is finite in size and we approach its effective diameter.

Highly-ranked pages have a large number of supporters after a few levels, while lowly-ranked pages do not. Note that if two pages belong to the same strongly-connected component of the Web, then eventually their total number of supporters will converge after a certain distance. In that case the areas below the curves will be equal.

Thus it has been observed that a normal webpage have their graph of the supporter increasing exponentially and the number of supporters increases with the distance. But in the case of the web spam their graph has a sudden increase in the supporters over a small distance of time and decreasing to zero after some distance. The distribution of the supporters over the distance has been shown in the figure below.

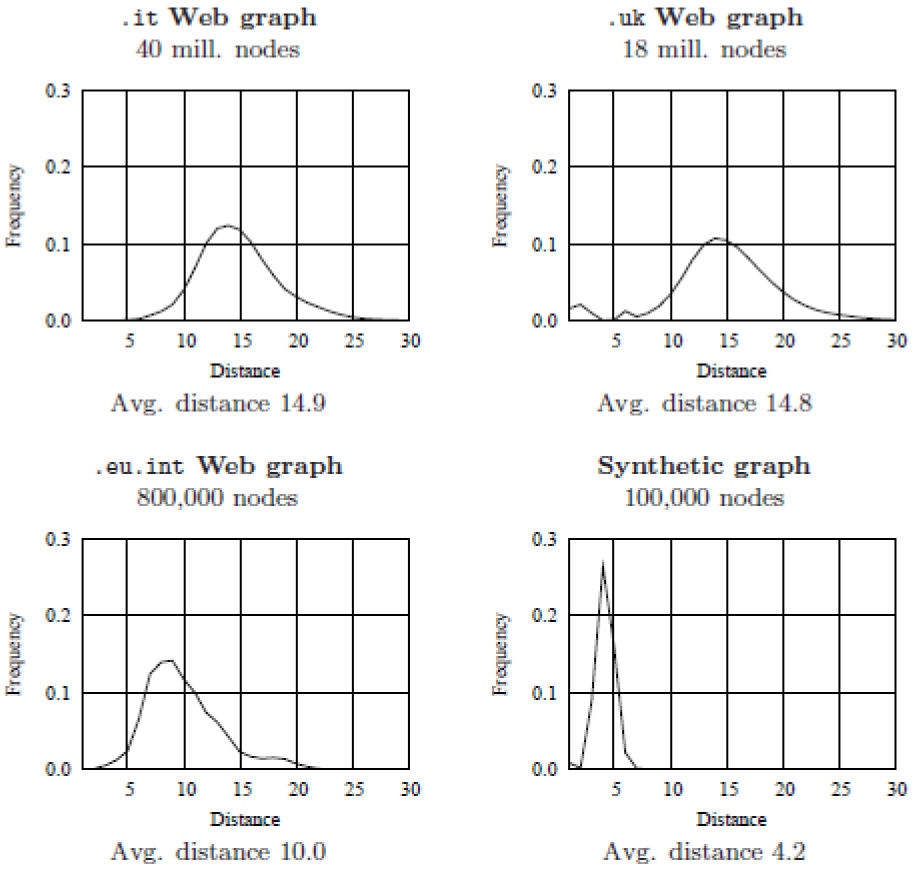


Figure 12:Distribution of supporters over a distance of the spam and non-spam page[1]

4.2.1.1 Algorithm Proposed

We start by assigning a random vector of bits to each page. We then perform an iterative computation: on each iteration of the algorithm, if page y has a link to page x,then the bit vector of page x is updated as x ← x OR y. In Figure 12, two iterations are shown. On each iteration, a bit is set to 1 in any page can only move by one link in distance. After d iterations, the bit vector associated to any page x provides information about the number of supporters of x at distance ≤d. Intuitively, if a page has a larger number of supporters than another, more 1s will appear in the final configuration of its bit vector.

```
Require: N: number of nodes, d: distance, k: bits
for node : 1 : : : N do [Every node]
    for bit : 1 : : : k do [Every bit]
        INIT(node,bit)
    end for
end for
for distance : 1 : : : d do [Iteration step]
    Aux   0k
    for src : 1 : : : N do [Follow links in the graph]
        for all links from src to dest do
            Aux[dest]   Aux[dest] OR V[src,_]
        end for
    end for
    for node : 1 : : : N do
        V[node,_]   Aux[node]
    end for
end for
for node: 1 : : :N do [Estimate supporters]
    Supporters[node]   ESTIMATE( V[node,_] )
end for
return Supporters
```

The estimation of the number of supporters for all vertices in the graph can be computed concurrently with the execution of Truncated PageRank and PageRank. Thus if number of supporters increases upon a short distance and reduces to zero after some distance then it is classifies as a **SPAM PAGE.**

4.2.2 Based on Trust Rank and Page Rank

TrustRank is a well-known algorithm for separating high-quality reputable pages/hosts from the rest, that is, spam and low-quality nodes. It provides a metric that can be used for ranking, and when combined with PageRank, can also be used for Web spam demotion.

The TrustRank calculation starts from a set of pages manually labeled as trusted, and does a random walk for a few steps (typically around 20), returning to the original set with a certain probability at each step (usually 0.15). The obtained probability of reaching a page is called its estimated non-spam mass. Pages with very low estimated **nonspam mass** compared to their **PageRank** should be considered suspicious. The intuition is that **a page with a high PageRank, but with no connection with the most relevant pages on the Web, can be considered suspicious.**

4.2.2.1 PageRank

PageRank is a link analysis algorithm used by the Google Internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. Google interprets a link from page A to page B as a vote, by page A, for page B.

A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself. If there are no links to a web page there is no support for that page.Google assigns a numeric weighting from 0-10 for each webpage on the Internet; this PageRank denotes a site’s importance in the eyes of Google.

PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size

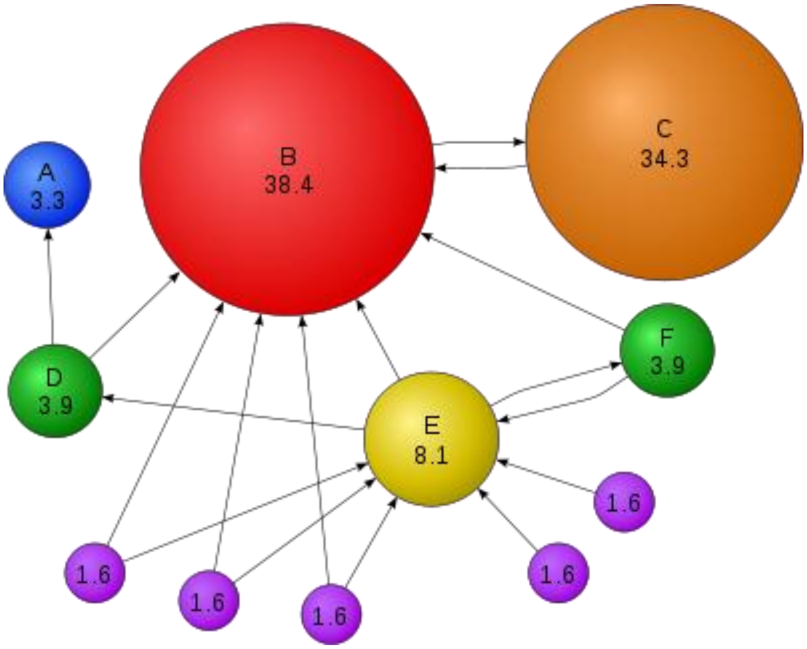


Figure 13 Link Structure of the Web page

(Source :www.wikipedia.org)

Assume a small universe of four web pages: **A**, **B**, **C** and **D**. The initial approximation of PageRank would be evenly divided between these four documents. Hence, each document would begin with an estimated PageRank of 0.25.

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$$

Where PR(A) is PageRank Of web page A and L(B) is the normalized number of outbound links. In the general case, the PageRank value for any page **u** can be expressed as:

$$\boldsymbol{PR(u)} = \sum_{v \varepsilon \mathbf{B}u} \frac{PR(v)}{L(v)}$$

The PageRank value for a page **u** is dependent on the PageRank values for each page **v** out of the set $\mathbf{B_u}$ (this set contains all pages linking to page **u**), divided by the number $L(v)$ of links from page **v**. The PageRank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor d .Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.Thus PageRank is determined by

.

$$\mathbf{PR(A)}= \frac{1-d}{N} + d \left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(B)} + \ldots \right)$$

OR

$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

Where the adjacency function $\ell(p_i,p_j)$ is 0 if page p_j does not link to p_i, and normalized such that, for each j and

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix}$$

A page having high PageRank but having no connection with the rest of the graph is a spam page [1].

4.3 Future approach on web spam detection

Till now Google is using these methods to detect web spam but now they are thinking to develop some new methods to detect the web spam. Till now they had used **Eigen Vector Centrality Method but** they are planning to use all the other centrality measures. The Centrality measures of a node are:

- **Degree centrality**- Degree centrality is defined as the number of links incident upon a node (i.e., the number of ties that a node has).
- **Betweenness centrality**- Betweenness is a centrality measure of a vertex within a graph .Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not.
- **Closeness centrality**- closeness is a centrality measure of a vertex within a graph. Vertices that are 'shallow' to other vertices (that is, those that tend to have short geodesic distances to other vertices with in the graph) have higher closeness.
- **Eigenvector centrality**- It is a measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes**.**

These centrality measures can be found by using igraph library in linux operating System

```
#include"igraph.h"
#include"math.h"
int main()
{
igraph_t g;igraph_vector_t v,result;long int i;igraph_integer_t
value,retcode;igraph_arpack_options_t options;
igraph_erdos_renyi_game(&g,IGRAPH_ERDOS_RENYI_GNP,1000,0.5,1,0);
igraph_arpack_options_init(&options);
 igraph_vector_init(&result, 0);
igraph_vector_init(&v,0);
igraph_eigenvector_centrality(&g,&v,&value,0,0,&options);
igraph_degree(&g, &result, igraph_vss_all(), IGRAPH_ALL, IGRAPH_NO_LOOPS);
if(options.info!=0)
return 1;
printf("EIGEN VECTOR\n");
for(i=0;i<igraph_vector_size(&v);i++)
printf("%.3f",fabs(VECTOR(v)[i]));
printf("DEGREE\n");
for(i=0;i<igraph_vector_size(&result);i++)
printf("%.3f",fabs(VECTOR(result)[i]));
igraph_closeness(&g, &result, igraph_vss_all(), IGRAPH_ALL);
printf("CLOSENESS\n");
for(i=0;i<igraph_vector_size(&result);i++)
printf("%.3f",fabs(VECTOR(result)[i]));
igraph_betweenness(&g, &result, igraph_vss_all(),IGRAPH_DIRECTED);
printf("BETWEENESS\n");
for(i=0;i<igraph_vector_size(&result);i++)
printf("%.3f",fabs(VECTOR(result)[i]));

printf("\n");
igraph_vector_destroy(&v);
igraph_destroy(&g);
return 0;
}
```

CHAPTER 5

CONCLUSION

In this thesis we had shown how two different features of the web farm: Context based and link based can be viewed as instances of a relevancy computation in search engines. We had applied algorithms for Web spam detection based on these features of the web farm i.e Context based(Naïve Bayesian Classifier) and link based(PageRank Algorithm).

Most modern search engines use a combination of factors for ranking search results. These factors include information aggregated from different sources, such as user clicks, text similarity and link analysis. The “spamicity" of a page is another factor and we must find how to combine it with the other sources of information. If we can estimate the probability of a page being spam, then a natural combination could be to rank the pages according to the product of their score and the probability of page being not spam {if we are 100% sure that a page is spam, then its final score will be zero.

The performance of web spam detection algorithms in general is still modest when compared with the error rate of modern e-mail spam detection systems. Fortunately, web spam detection can be more strict than e-mail spam detection. While losing a relevant e-mail message is very bad, demoting a relevant Web page in the search results is not so bad, because if the page is relevant, it can be found later by following links, or it can be moved to the second or third page of results.

However, the current precision and recall of Web spam detection algorithms can be improved. An ambitious goal we propose is to try to achieve the same precision and recall of e-mail spam detection algorithms. In our opinion, this is unfeasible with link-only or content-only classifiers, and requires the combination of both methods. An algorithm better than Naïve Bayesian Classifier can be proposed. We should use the other centrality measures to find better methods to detect link based web spam pages.

References

1. **Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Using rank propagation and probabilistic counting for link-based spam detection. In Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD), Pennsylvania, USA, August 2006. ACM Press.**

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

3. **Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock. Know your Neighb**
ors: Web Spam Detection Using the Web Topology In SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (2007), pp. 423-430

4. **Zoltán Gyöngyi, Hector G. Molina, and Jan Pedersen. Combating web spam with trustrank. In Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB), pages 576–587, Toronto, Canada, August 2004. Morgan Kaufmann**.

5. "Using Spam Farm to Boost PageRank" by Ye Du, Yaoyun Shi, Xin Zhao

6. Sergey Brin and Lawrence Page The Anatomy of a Large-Scale hypertextualWeb Search Engine, ,Stanford University, Computer Networks and ISDN Systems,1998.

7. Baeza-Yates, R., Boldi, P., and Castillo,Generalizing pagerank: Damping functions for link-based ranking algorithms. C. 2006. In Proceedings of ACM SIGIR. ACM Press, Seattle,Washington

8. A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In Proceedings of the World Wide Web conference, pages 83–92, Edinburgh, Scotland, May 2006.

9. http://www.googleguide.com/google_works.html