Sādhanā Vol. 41, No. 2, February 2016, pp. 193–202

Detection of spam web page using content and link-based techniques: A combined approach

RAJENDRA KUMAR ROUL*, SHUBHAM ROHAN ASTHANA, MIT SHAH and DHRUVESH PARIKH

BITS, Pilani-K.K.Birla Goa Campus, Goa, India
e-mail: rkroul@goa.bits-pilani.ac.in; asthana.st.francis@gmail.com; mitk1shah@gmail.com; parikhdhruvesh1@gmail.com

MS received 27 March 2014; revised 11 September 2015; accepted 6 November 2015

Abstract. Web spam is a technique through which the irrelevant pages get higher rank than relevant pages in the search engine's results. Spam pages are generally insufficient and inappropriate results for user. Many researchers are working in this area to detect the spam pages. However, there is no universal efficient technique developed so far which can detect all spam pages. This paper is an effort in that direction, where we propose a combined approach of content and link-based techniques to identify the spam pages. The content-based approach uses term density and Part of Speech (POS) ratio test and in the link-based approach, we explore the collaborative detection using personalized page ranking to classify the Web page as spam or non-spam. For experimental purpose, WEBSPAM-UK2006 dataset has been used. The results have been compared with some of the existing approaches. A good and promising F-measure of 75.2% demonstrates the applicability and efficiency of our approach.

Keywords. Collaborative filtering; content spam; link spam; personalized page rank; pos ratio; term density.

1. Introduction

With the unprecedented rise of information on the World Wide Web, the amount of textual data available to any end user has become very huge. According to the latest survey, the size of the Web is at least 4.8 billion pages[1]. Thousands of Web pages are being added to the Web corpus every day in which many of them are either duplicate or spam (Ntoulas *et al* [1]). Spammers take advantages of the internet users by attracting them to their websites using various intelligent spamming techniques. Their ultimate aim is to improve the ranking of their Web pages in the search results.

The intension of creating a spam page is to mislead the search engine so that it returns those results which are not useful for the user. A robust and efficient Information Retrieval system can be built if one can identify and eliminate all the spam pages. This is the reason why efficient search engines are required which can provide high quality and promising results as per the user query. The next job is to rank the retrieved Web pages either by using content or semantic similarity between the query entered by the user and retrieved Web pages. At the end, the ranked Web pages are returned to theuser. Web spam has many negative effects on both end user and search engine. This is because spam pages not only waste space but also waste time. As search engine needs to index and store a large number of Web pages, hence more space is required. Similarly, when search engine needs to search Web pages based on a user query, the searching will take place in a large corpus and hence more time is required. This in turn reduces the effectiveness of the search engine and weakens the trust of the end user on search engine.

According to Ghiam & Pour [2], there are generally two kinds of Web spamming techniques: Content spam and Link spam.

1. *Content spam*: The content spam is most popular type of Web spam. This is popular because most of the search engines take the help of information retrieval models such as vector space model, probabilistic model and statistical language model which are applied to the content of a Web page to rank it. Spammers try to understand the weakness of these models and try to manipulate the content of the target page. For example, increasing the term frequencies for terms appear in a page, repeating important terms many times on a target page, putting all dictionary terms on a target page are some of the techniques by which the spammers try to increase the score of a page in the search results. Generally, there are five subtypes of content spamming based on the structure of

*For correspondence

[1]http://www.worldwidewebsize.com

a Web page. They are Title spam, Body spam, Meta-tag spam, Anchor text spam and URL spam.

2. *Link spam*: Outgoing and Incoming are two common categories of link spam. In outgoing link spam, the spammer has direct access to the target page and hence can add many links to the target page to change the search results. In incoming link spam, spammers try to increases the number of sites that link to a spam page which is also known as link farms where a network of Web pages are densely connected to each other. This is because most of the search engines rely on both quality as well as quantity of Web pages that are link to a target page. By doing this the importance of the target page will be increased.

In our work, we proposed a method to detect spam Web pages by using either content or link-based techniques or combination of both. In the content-based approach, we have used term density and POS ratio test to detect the spam pages. The term density test classifies a Web page into three different categories based on the percentage of the distinct terms present on that Web page. Then based on the category, a sequence of tests will be conducted on that Web page to detect it as spam or non-spam. POS ratio test is based on linguistic features by which we detect a page as spam or non-spam. In the link-based approach, we have used collaborative detection using personalized page ranking to detect the spam pages. First personalized page rank has been calculated for all the Web pages and then using an optimization function which is same as used in collaborative filtering [3], we detect a page as spam or non-spam. WEBSPAM-UK2006 dataset has been used for experimental purpose. The empirical results with a promising F-measure witness the effectiveness of our approach compared to other existing approaches.

The remaining sections are organized as follows. Section 2 summarizes the literature survey on different Web spam techniques and their detection. Section 3 reports the background details of the content and link-based approach. In Section 4, the proposed approach is described to detect the spam Web pages. Section 5 covers the experimental results. Section 6 highlights the conclusion and future enhancement of the work.

2. Literature survey

Web spamming is a technique which helps to increase the rank of spam Web pages returned by search engine. Recently the research in this area has become more popular and has grown dramatically. Content analysis and link-based detection are two main strategies on which the anti-spam works rely on. Well known anti-spam techniques using link-based analysis are discussed by Gyöngyi *et al* [4, 5], Benczúr *et al* [6], Becchetti *et al* [7]. Working in content-based analysis, a decision tree classifier has been used by Ntoulas *et al* [1] to detect the spam pages. For content-based techniques, they have proposed many features like the amount of anchor text, the number of words, the average length of the words, the fraction of the visible content and independent of n-grams of likelihood within the page. A semi-supervised and combinatorial feature fusion method for detecting spam pages has been proposed by Tian *et al* [8]. Their generated TF-IDF feature vectors over 100 Web pages for a host are all sparse vectors. Semi-supervised learning has been used in order to exploit the unlabeled examples. To reduce TF-IDF content-based features and to construct new features, combinatorial feature-fusion method has been used. Experimental results witness the efficiency of their approach.

A combined approach of content and link-based features to detect the spam pages is proposed by Abernethy *et al* [9]. In their work, they proposed an algorithm called WITCH where Support Vector Machine (SVM) with graph regularization classifier have been used to detect the spam pages. Egele *et al* [10] proposed a new approach using J48 decision tree classifier which differentiates the spam sites from other authorized sites. Their experimental results lower the false positive to zero and able to detect one out of five spam pages. Ahmed & Abulaish [11] consider 14 set of generic statistical features which are identified from Twitter and Facebook to identify spam profile on online social networks. Empirical results on two different datasets demonstrate the effectiveness of their approach. A new spam detection technique called SAAD (Spam Analyzer and Detector) based on set of heuristics is proposed by Prieto *et al* [12]. Two public datasets (Email Spam (Web Spam Corpus), Yahoo!) have been used to test and compare SAAD with other previous studies. They claimed that their proposed technique can generate a safer client environment, protecting users from attacks. Luckner *et al* [13] proposed a new lexical-based features for a stable Web spam detection. They have compared SVM classifiers which are trained and tested on both WEBSPAM-UK 2006 and 2007 datasets. Experimental results show that both the accuracy and the specificity of their approach are statistically stable. Goh *et al* [14] proposed a link-based approach using weight properties to detect the spam pages. They have defined the weight properties as the influence of one Web node towards another Web node. Their test results on WEBSPAM-UK2007 outperformed the benchmark algorithms up to 30.5% improvement at the host level and 6.11% improvement at the page level.

Combining the content-based with the link-based technique can enhance the performance of search engine for detecting the spam pages. But very few research work has been done on combined approach. Our approach combined both content and link-based features to detect the spam pages. Term density and POS ratio test have been used for content-based technique while collaborative detection using personalized page ranking has been used for link-based technique to detect the spam pages. The experimental results on WEBSPAM-UK2006 dataset show the importance of the proposed combined approach over other standard existing approaches by obtaining a good F-measure.

3. Background

3.1 *Content-based technique*

3.1a *Term density test*: This test is used to find the density (in %) of each distinct term on a Web page i.e. the % of occurrences of each distinct term compared to all other terms on a Web page. The following steps discuss how the term density test will be done on a Web page k.

1. Find out the term density of all the distinct terms of a page k. The term density of a distinct term i on page k is defined as follows:

$$Term\ density(i) = \frac{number\ of\ times\ i\ appears\ on\ page\ k}{total\ number\ of\ terms\ on\ page\ k}. \quad (1)$$

2. Based on the term density of each distinct term $i \in k$, the Web page k will classify into one of the three different categories mentioned below.

   (i) Category 1 (Low Term Density)
   If the term densities of all the distinct terms of a page k are less than $threshold\_category1$ (determined by experiment)[2] then page $k \in$ Category 1. For this category, first the POS ratio test (discussed in section 3.1b) followed by collaborative detection using personalized page rank which is a link-based test (discussed in section 3.2) are performed on page k to detect it as spam or non-spam. The reason to do POS ratio test first is that from experiment it has been found that more number of pages falls into 'Category 1' because of low term density, and link-based test consumes more time compared to POS ratio test. Hence in order to save time, we ran first the POS ratio test on this category of pages to filter out the spam pages (if a page fails in any of the two tests then it is declared to be spam and further testing is not required) so that the number of pages left for link-based test is less.

   (ii) Category 2 (Medium Term Density)
   If the term density of any of the distinct term of page k laying between $threshold\_category1$ and $threshold\_category2$ (determined by experiment)[2] then page $k \in$ Category 2. For this category, first the collaborative detection using personalized page rank followed by POS ratio test is performed on page k, because it can happen that the term having frequency greater than the $threshold\_category1$ might present in text of links to other Web pages which belong to some class of Web pages on a particular topic, and that topic word is more occurring word on that Web page.

   (iii) Category 3 (High Term Density)
   If the term density of any distinct term of page k is greater than $threshold\_category2$ then page $k \in$ Category 3. For this category, Web page k should be directly eliminated because this clearly depicts the case of *term or keyword stuffing* (i.e. overload keywords onto the Web page k). *Keyword stuffing* means, the number of distinct terms in k is very less and the spammer has repeated some of the particular distinct terms too many times in k. This is done by repeating some particular distinct terms (or a particular distinct term) again and again in many places of k such as in content, Meta tags, Alt attributes and comment tags. The main aim of the spammer is to make the search engine believe that the page is relevant to the distinct terms present in the user query and thus increase its ranking. Hence, k will be categorized as spam.

3.1b *POS (Part of Speech) ratio test*: POS Ratio test is a linguistic technique to detect content re-purposing. Some of the spam pages are machine generated and hence technique of content re-purposing can be employed by spammers. This is done by incorporating large parts of a single Web page or incorporating multiple small parts of a Web page into a single Web page. Linguistic features are generally applied on the Web page to detect content re-purposing. This method relies on the assumption that spammers cannot replicate all aspects of the natural language while doing content re-purposing. Piskorski *et al* [15] in their work have shown the usefulness of using these wide ranges of linguistic features. The main aim of using these linguistic features is to identify the authorship and originality of the content of a Web page. For doing this test, we measure multiple aspects of text quality and later combine them into content spam classifier by using supervised learning. Ratio of different grammatical forms like noun-singular, verb, adverb, adjective, conjunction, pronoun, preposition and determiner is calculated to extract maximum information from different part of speech.

The POS ratio test conducted on a Web page k is as follows:

1. Finding different grammatical forms[3] in k:
2. Calculation of ratio of each grammatical form:
   Ratio of each grammatical form, $g \in k$ is calculated as follows:

$$Ratio(g) = \frac{x}{y} \quad (2)$$

[2] Threshold values of different types can be obtained by iteratively running the script on the training dataset of WEBSPAM-UK2006, over a range of values and finally selecting those values as threshold at which the evaluation parameters have better results. The results have been evaluated by plotting histograms shown in section 5.1 on various parameters.

[3] http://infomotions.com/blog/2011/02/forays-into-parts-of-speech/

where, x = number of occurrences of g in k and y = total number of terms present in k

3. Divergence calculation:

   Next, $divergence$ of the ratio of g from the standard ratio of occurrence of g available in normal English text has been calculated. The website[3] has been referred for standard ratio figures.

4. Average divergence calculation:

   After calculating the $divergence$ of each $g \in k$, the average $divergence$ of the Web page k is calculated as follows:

$$average\ divergence = \frac{m}{n} \tag{3}$$

   where, m = sum of $divergence$ of each $g \in k$ and n = total number of grammatical forms considered on the Web page k.

5. Spam or non-spam detection:

   If *average divergence* > *pos_divergence_threshold* (determined by experiment)[2] then

   i. the Web page k fails to qualify the POS ratio test
   ii. declare k as spam

To tag every word in the dataset, Penn Treebank POS tagger[4] has been used. Penn Treebank annotates naturally occurring text for linguistic structure. Algorithm 1 discussed the details of the POS ratio test.

Algorithm 1: POS ratio test

```
Data: A set of .html Web pages < P1, P2, P3....Pn >, where n is the total number of
      Web pages in the corpus
Result: A set of Web pages < P1, P2, P3....Pn >, where each Pi is either categorized
        as spam or non-spam and i = 1 to n

page ← Pi //stores all the pages
Form ← {'noun', 'pronoun', 'verb', 'adverb', 'adjective', 'conjunction', ... }
//considering r different grammatical forms
for i in 1 to n do
    // 'n' number of Web pages
    for j in 1 to m do
        // 'm' number of terms in each page
        for k in 1 to r do
            // 'r' number of grammatical forms
            if Page[i][j] ∈ Form[k] then
                // i^th page's j^th term match with k^th form
                count[i][k] ++ // increment the corresponding grammatical form
                present in i^th page
                break
            end
        end
    end
    for k in 1 to r do
        Ratio[i][k] = count[i][k] / m
        Divergence[i][k]=| Ratio[i][k] − standard[k] |
        Average_Divergence = Σ_{k=1}^{r} Divergence[i][k] / r
        if Average_Divergence[i] ≥ threshold_divergence then
            Page[i] ← spam // classify page i as spam
        end
    end
end
```

3.2 *Link-based technique*

3.2a *Collaborative detection of spam web page using personalized pagerank*: In Link-based techniques, we have worked with a basic assumption that spam pages contribute more to the page rank of spam pages and non-spam pages contribute more to the page rank of non-spam pages. Unlike others, we are not manually classifying some seed pages to propagate label of 'spam' or 'non-spam' further. We calculate personalized page rank for all Web pages. From that, we partition the Web pages in two parts, spam and non-spam, by using an optimization function, similar to one used in collaborative filtering. We see the page rank contribution of Web page i to Web page j, as a rating, and try to learn, in what kind of Web pages j, Web page i is interested, (i.e. spam or non-spam). Also, from page rank contributions of incoming links from Web pages j to Web page i, we try to learn which type of Web pages j are interested in Web page i. Based on that learned parameter, we classify the Web page i as spam or non-spam.

The following steps are used to develop the link-based approach:

(1) Construction of adjacency matrix for the Web graph:

The Web can be modeled by a directed graph $G = \{V, E\}$ where V are Web pages and $(U \rightarrow V) \in E$ represents the hyperlink that U references to V. Initially we stored the outlinks information from one Web page to others as shown in table 1. Here, we considered the format of the outlinks information is as per the link information available in WEBSPAM UK-2006 dataset.

Each row of the table contains the outlink information. For example, in the first row, 0^{th} Web page has one outlink to Web page 2, two outlinks to Web page 3. Now the above outlinks information will be converted to an adjacency matrix of dimension $n \times n$, where n is the total number of Web pages. In matrix A, β_{ij} represents the number of outlinks from Web page i to Web page j and its corresponding matrix $A'[i][j]$ shows a typical scenario of the number of outlinks from Web page i to Web page j by using the information available in table 1, where $n = 6$.

$$\boldsymbol{A} = \begin{bmatrix} \beta_{11} & \dots & \beta_{1n} \\ \beta_{21} & \dots & \beta_{2n} \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ \beta_{n1} & \dots & \beta_{nn} \end{bmatrix}_{n \times n} \qquad \boldsymbol{A'} = \begin{bmatrix} 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 \\ 2 & 0 & 1 & 0 & 0 & 1 \\ 0 & 3 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{6 \times 6}$$

Next, we row normalized this adjacency matrix A as it is required as the input for personalized pagerank. H

Table 1. WEBSPAM UK-2006 data format.

Web Page	Outlinks to
0	2:1, 3:2
1	3:2, 4:1
2	3:2, 4:2, 5:2
3	0:2, 2:1, 5:1
4	1:3, 2:3
5	4:1

[4] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

represents the row normalized matrix of A, and H' is the corresponding row normalized matrix of A'.

$$\boldsymbol{H} = \begin{bmatrix} \frac{\beta_{11}}{\beta_{11}+...+\beta_{1n}} & \cdots & \frac{\beta_{1n}}{\beta_{11}+...+\beta_{1n}} \\ \frac{\beta_{21}}{\beta_{21}+...+\beta_{2n}} & \cdots & \frac{\beta_{2n}}{\beta_{21}+...+\beta_{2n}} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \frac{\beta_{n1}}{\beta_{n1}+...+\beta_{nn}} & \cdots & \frac{\beta_{n1}}{\beta_{n1}+...+\beta_{nn}} \end{bmatrix}_{n\times n}$$

$$H' = \begin{bmatrix} 0 & 0 & \frac{1}{3} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{2}{6} & \frac{2}{6} & \frac{2}{6} \\ \frac{2}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ 0 & \frac{3}{6} & \frac{3}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{6\times 6}$$

(2) Calculating personalized page rank:

Personalized page rank gives the contribution of each page in the page rank of each other page. In other way, personalized page rank of a page i is calculated by taking into consideration the contribution of all other pages to the page rank of i. The contribution that a vertex U makes to the page rank of vertex V of a graph G (discussed in Step 1) is defined in terms of personalized page rank [16]. For a row normalized adjacency matrix H and teleportation vector ϑ, we calculate the pagerank for page i (i.e. P_i)as follows:

$$P_i \leftarrow \alpha * P_i * H + (1-\alpha) * \vartheta, \tag{4}$$

where $\alpha \in [0, 1]$ is scaling parameter and is usually set to a standard value of 0.85 [17]. If we need to calculate personalized page rank contribution vector for i^{th} Web page, then i^{th} bit of teleportation vector ϑ should be made 1 and all other bits are set to 0. Initially P_i is assigned [1/n 1/n ... 1/n] and we iteratively calculate it by Eq. (4). The page rank of page i is then stored in the i^{th} row of personalized page rank matrix as $ppr[i,:] \leftarrow P_i$. The details of the personalized page rank calculation are discussed in Algorithm 2.

(3) Collaborative detection of spam page:

Now ppr[i][j] of Algorithm 2 denotes the contribution of Web page i in the page rank of Web page j. From that we try to learn the features (spam and non-spam) for each Web page say k by the features of Web pages contributing to k and the Web pages to which k is contributing. Hence, we select the two features (spam and non-spam) values for each of the n Web pages and create the matrix x of dimension $n \times 2$.

Let $x(j,:)$ represent the j^{th} row of matrix x. So $x(j,:)$ will denote these two features for j^{th} Web page. Hence, multiplying $x(j,:)^T \times x(l,:)$ will give the similarity of features between j^{th} and l^{th} Web page. More this similarity will be, more their personalized page rank contribution to each other. So, in order to obtain the matrix x, we need the following function to be minimized:

$$\min_{x^{(1)}...x^{(n)}} \left[1/2 \sum_{j=1}^{n} \sum_{l=1}^{n} (x(j,:)^T \times x(l,:) - ppr(j,l))^2 + \lambda/2 \sum_{j=1}^{n} \sum_{k=1}^{m} (x(j,k))^2 \right] \tag{5}$$

where λ is the regularization value used in linear regression. To minimize this function, we perform a usual update of features of j^{th} Web page in the direction of Gradient descent.[5]

$$x(j,k) = x(j,k) - \left(\sum_{l=1}^{n} [x(j,:)^T \times x(l,:) - ppr(j,l)] \times x(l,k) \right) + \lambda x(j,k) \tag{6}$$

Algorithm 3 discussed the details.

(4) Identifying the spam and non-spam category:

At the end of Step 3, all Web pages have been classified into two categories (i.e. category 1 and category 2). One category contains spam pages and the other one contains non-spam pages. But, we do not know which category contains which type of pages; as Algorithm 3 only separates all Web pages into two categories, based on the link structure of the Web graph. So, we need to figure out which category is spam and which is non-spam. To do this, we followed the steps mentioned below:

1. Let 'n' be the total number of Web pages, which needs to be classified as spam and non-spam. We add extra 10% (0.1*n) spam Web pages to the input

Algorithm 2: Personalized page rank

Data: $n \leftarrow$ size of the adjacency matrix
$P_{i0} \leftarrow$ initial $1 \times n$ vector at iteration 0 // set to [1/n 1/n...1/n]
$H \leftarrow$ Row-normalized adjacency matrix
$\vartheta \leftarrow$ teleportation vector
$\alpha \leftarrow$ scaling parameter
$\beta \leftarrow$ convergence tolerance // set to 0.001(user dependent)
Result: $ppr \leftarrow$ personalized pagerank matrix of dimension $n \times n$

$\vartheta \leftarrow$ [0,0....,0]
for *i in 1 to n* **do**
  ϑ[i]$\leftarrow$ 1 // set the i^{th} bit of ϑ[i] to 1 to calculate the page rank contribution vector of i^{th} Web page
  $k \leftarrow 0$ //number of iterations
  residual $\leftarrow$ 1
  $P_i \leftarrow P_{i0}$
  while *residual* $\geq \beta$ **do**
    $P_{i_{prev}} \leftarrow P_i$
    $k \leftarrow k+1$
    $P_i \leftarrow \alpha * P_i * H + (1-\alpha) * \vartheta$
    residual$\leftarrow$ norm-1 distance between P_i and $P_{i_{prev}}$
  end
  $ppr[i,:] \leftarrow P_i$ // values of P_i stored in i^{th} row of ppr
  ϑ[i] $\leftarrow$ 0
end

[5] http://en.wikipedia.org/wiki/Gradient

Algorithm 3: Link-based spam detection

```
Data: personalized page rank matrix (ppr) of n × n
Result: n× 1 vector (denoted as 'a' in the algorithm) // '0' denotes the Web page
        belongs to category '1' and '1' denotes the Web page belongs to category '2'
num_pages ← total number of pages considered(n)
num_features ← number of features // spam and non-spam
λ ← 10 // determined by experiment
p ← number of iterations
x ← random(num_pages, num_features) // create a random matrix of num_pages ×
2 for features
x_grad ← zeros(num_pages, num_features )// create a matrix of initially all entries
are zero
for i=1 to p do
    // to update the features of j^th Web page
    for j = 1 to num_pages do
        // to update the k^th feature of j^th Web page
        for k=1 to num_features do
            // to update the k^th feature of j^th Web page, consider k^th feature of
            l^th Web page
            for l=1 to num_pages do
                x_grad(j,k) ← x_grad(j,k) + [(x(j,:)^T ×x(l,:)) − ppr(j,l)] × x(l,k)
            end
            x_grad(j,k) ← x_grad(j,k) + λ × x(j,k)
        end
    end
    x ← x_grad
end
a← zeros(num_pages, 1) // vector of size num_pages × 1, initially filled with 0s
for i=1 to num_pages do
    if x(i,1)≥ x(i,2) then
        a[i]=1
    end
end
return a
```

initially for which we know their classification labels.

2. Now, for each spam Web page added extra, we check whether it belongs to category 1 or category 2. Depending on, in which category majority of spam Web pages falls, we mark that category as the "spam category" and the other one as "non-spam category". Reason for adding extra spam Web pages instead of non-spam Web pages to label the two categories outputted by our approach is that our algorithm is more accurate at classifying all spam Web pages in a category than classifying all non-spam Web pages in that category, as the main aim of complete approach is to detect the spam Web pages.

4. Proposed approach

In this section, we discussed how to detect a spam page by combining both content and link-based approach. A set of .html Web pages will be given as the input to the approach and the output will be detection of each page as spam or non-spam. For every page, we need to find out when a page P_i is qualify for term density test, POS ratio test and link-based test as discussed below:

Step 1: Find the ratio (R) of total number of distinct terms to total number of terms (excluding stop-words) on P_i. Then the following conditions need to be check on P_i:

i. If $R > upper\_threshold$ (determined by experiment)[2] then it indicates that most of the distinct terms are present in P_i. This is a typical case when spammers filled up their Web pages with large number of distinct terms in order to attract queries related to different topics in a variety of unrelated fields. Thus, P_i received a large number of distinct terms which improves its ranking in the search results. This categorized P_i as spam.

ii. If $R < lower\_threshold$ (determined by experiment)[2] then it is a perfect case of *term or keyword stuffing* as the number of distinct terms in P_i is very less. This case arises when the spammer has repeated some of the particular distinct terms too many times in P_i by repeating them over and over again in content, Meta tags, Alt attributes and comment tags of P_i. The ultimate intention of the spammer is to attract the search engine and makes it to believe that the page P_i is relevant to the distinct terms and thus inflate P_i's ranking higher in the search results. Hence, P_i will be categorized as spam.

Step 2: Calculate the total term count (TC) of P_i (including stop-words)

i. If $TC < count\_threshold$ (determined by experiment)[2] then P_i does not qualify for Term density test, hence first apply POS ratio test followed by Collaborative detection test using personalized page rank on P_i to check it as spam or non-spam.

ii. If $TC \geq count\_threshold$ then P_i is categorized based on the term density test and then according to the category, the sequence of tests will be conducted on P_i to check it as spam or non-spam.

Algorithm 4 discussed the details.

5. Experimental results

For experimental purpose, verification set which consists of randomly selected Web pages, labeled as spam and non-spam from the WEBSPAM-UK2006 dataset[6] has been taken into consideration. This dataset is well suit for spam page detection and has the following properties:

(i) the dataset is used as a benchmark measure in detecting spam Web pages and is available freely
(ii) the dataset consists of diverse quality of spam and non-spam Web pages
(iii) it contains spam Web pages produced by using a variety of spamming techniques

[6] http://chato.cl/webspam/datasets/uk2006/

Algorithm 4: Combined content and link-based approach to detect the spam pages

```
Data: A set of .html Web pages < P_1, P_2, P_3....P_n >, where n is the total number of
      Web pages in the corpus
Result: A set of Web pages < P_1, P_2, P_3....P_n >, where each P_i is either categorized
        as spam or non-spam and i = 1 to n

T_i ← total number of terms in P_i (excluding stop words)
TC_i ← total number of term counts in P_i (including stop words)
for i in 1 to n do
    pre-process P_i by removing HTML tags, digits, and special characters
end
for i in 1 to n do
    R ← 0
    R = (total number of distinct terms in P_i) / T_i
    if (R < lower_threshold) or (R > upper_threshold) then
        // Categorize P_i as Spam
    end
    else
        if TC_i < Count_threshold then
            // P_i does not qualify for Term density test
            POS ratio test on P_i followed by Collaborative detection using
            personalized page rank on P_i
            //If in any of the above two tests P_i fails then categorize P_i as spam
        end
        else
            //Term density test on P_i for finding the category to which it belongs.
            if P_i ∈ Category 1 then
                POS ratio test on P_i followed by Collaborative detection test using
                personalized page rank on P_i.
                //If in any of the above two tests P_i fails then categorize P_i as spam
            end
            else
                if P_i ∈ Category 2 then
                    Collaborative detection test using personalized page rank on P_i
                    followed by POS ratio test on P_i.
                    //If in any of the above two tests P_i fails then categorize P_i as
                    spam
                end
            end
            else
                P_i ∈ Category 3
                // Categorize P_i as spam
            end
        end
    end
end
```

(iv) the sample Web pages in the dataset are random and uniform

(v) the dataset divides the Web pages into training and testing set with both spam and non-spam labels so that it can efficiently used for any content and link-based approach

(vi) the training set has been utilized to obtained the optimized threshold values used for our proposed content-based approach. Hence, there is no use of any holdout samples as the dataset itself provides the training and testing set

We took a collection of 11,402 different hosts as available from WEBSPAM-UK2006 dataset (Host graph format). We converted it into an adjacency matrix and resulting again to a normalized adjacency matrix of dimension 11,402 × 11,402. Out of 11,402 Web pages, we filtered out the dataset to 1000 Web pages by using the following pre-processing techniques where each Web page has many outgoing links:

1. We only considered those Web pages whose human assigned labels are available
2. Among those Web pages, we have selected all working/existing links.
3. Next, we extracted content from those links and stored in text file format
4. We selected those Web pages whose content are at least 1KB, which is required for our content-based technique

Table 2. Performance evaluation of the proposed approach.

Different techniques	Precision (%)	Recall (%)	F-Measure (%)
Content-based	71.3	69.3	70.2
Link-based	67.3	73.2	70.1
Combined	72.9	77.6	75.2

The proposed approach is implemented in python language and a machine with Intel Core 2 Duo Processor, 2.1 GHz, with 64 GB RAM and running Ubuntu 14.04 has been used to execute the algorithms.

F-measure[7] has been used for evaluation of the proposed algorithm and comparison of our work with other related works, as it is a standard technique which combine both precision and recall. Here, the true positive rate 'TPR' (also called recall) represents the amount of spam pages that is detected by the algorithm. The proposed spam page detection approach obtained the following results shown in table 2.

5.1 *Obtaining appropriate threshold values*

WEBSPAM-UK2006 dataset has been used to obtain the appropriate threshold values. The dataset divides the Web pages into training and testing set with both spam and non-spam labels. In order to obtain the optimized threshold values of different types which have used for content-based technique (discussed in section 3.1) including the upper_threshold, lower_threshold and count_threshold (discussed in section 4), the proposed algorithm has run iteratively over the training dataset of WEBSPAM-UK2006 over a range of values. Finally, we selected those values as appropriate threshold at which the evaluation parameters have better results. In other words, these obtained values are then plotted on different histograms to decide the most appropriate threshold that provides least false positive ratios and high F-measure. Figures 1–2 show the threshold used for content-based technique to decide the non-spam and spam category for a Web page. Similarly figures 3–4 show the threshold frequency for non-spam and spam used for POS ratio test.

5.2 *Comparisons with existing approaches*

Our experimental results have been compared with the following existing approaches:

1. Our Approach vs. Egele *et al* [10]

   We compare our obtained results with the work of Egele *et al*. For detecting spam pages, they have used link-based features as keywords in URL and in domain name, number of inbound links to name a few features. J48 decision tree as classifier has been used for their experimental work. As per the confusion matrix listed in table 3 of Egele *et al* [10], they achieved an F-measure

[7] http://www.gabormelli.com/RKB/F-Measure

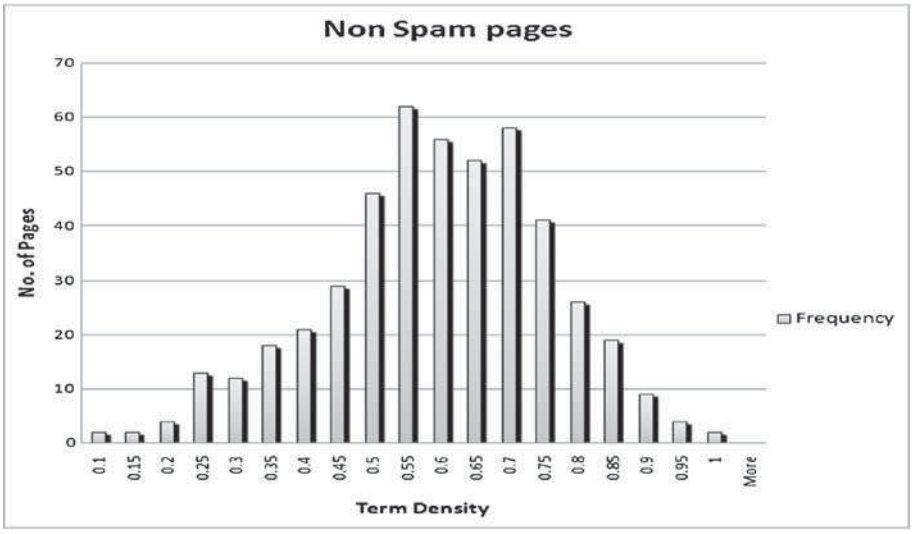

Figure 1. Number of Web pages vs. Fraction of distinct terms on the Web page.

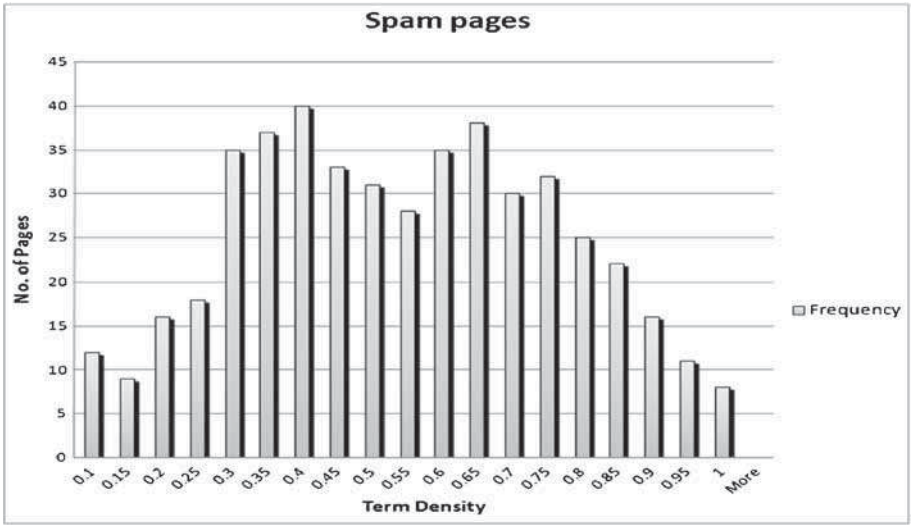

Figure 2. Number of Web pages vs. Fraction of distinct terms on the Web page.

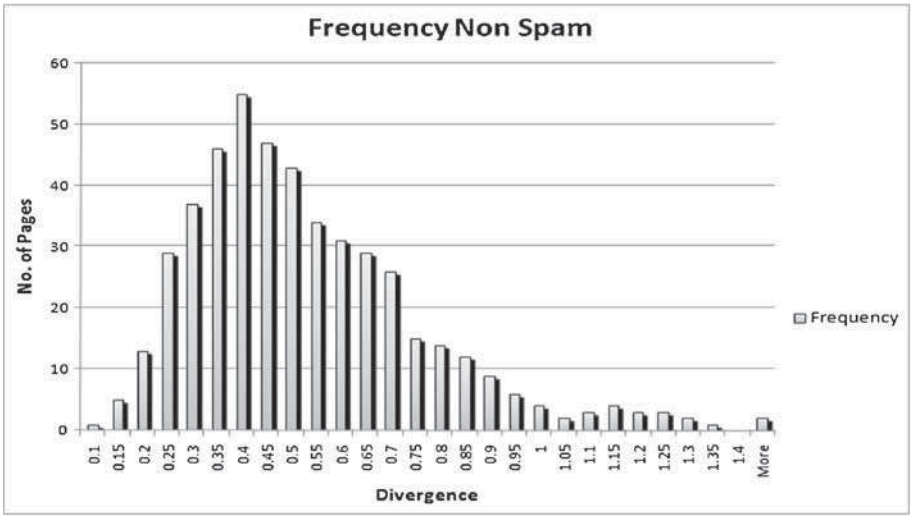

Figure 3. Number of Web pages vs. Average POS ratio divergence.

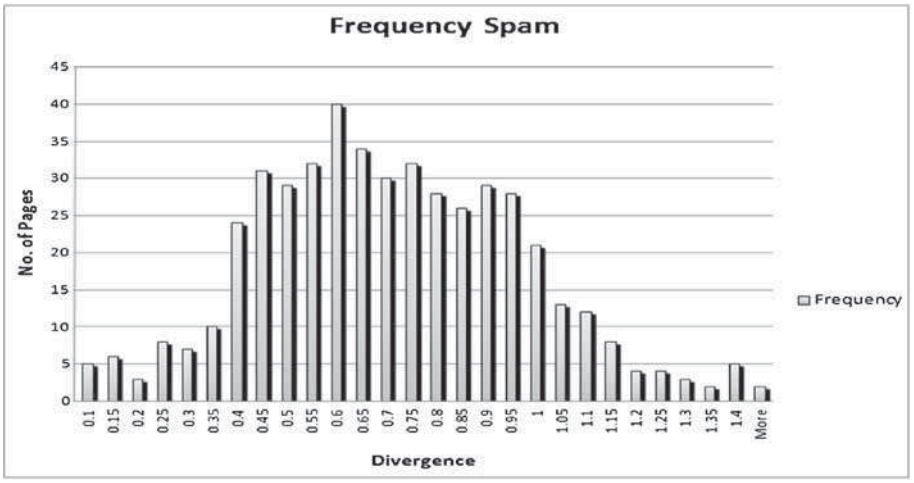

Figure 4. Number of Web pages vs. Average POS ratio divergence.

of around 0.419 and precision of around 0.512, which is significantly lesser than our results.

2. Our Approach vs. Dai *et al* [18]

   Next, our experimental results are compared with Dai *et al*. In their work, historical Web page information for spam detection has been considered. To improve the spam classification, content features from historical version of Web pages are used. Using supervised learning mechanism they combine the classifiers based on the current page content and on temporal features. Their approach extracts a variety of temporal features from archival copies of the Web provided by internet Archive's Way back Machine. WEBSPAM-UK2007 dataset has been used for experimental purpose. From table 5 listed in Dai *et al* [18], it has been found that by using their approach, they achieved an F-measure of 0.527 and its corresponding precision of 0.650 which is lesser than our results

3. Our Approach vs. Becchetti *et al* [19]

   Further, our work has been compared with Becchetti *et al*. In their work, ratio of hidden text, presence of keywords in URL, etc. has been used for content-based features. Similarly for link-based features, they plotted the Web graph and found various page ranks like 'degree-based measures', 'truncated page rank' and 'trust rank', of Web pages to detect the spam. They have considered C4.5 decision tree as the base classifier. WEBSPAM-UK2002 and WEBSPAM-UK2006 datasets have been used for experimental purpose. It is found from table 6 of Becchetti *et al* [19] that by using both these feature techniques, they achieved an F-measure of around 0.723 with a precision rate of 0.668 which is lower compared to our results.

4. Our Approach vs. Benczúr *et al* [20]

   Finally, we compared our results with Benczur *et al*. They proposed a number of features for Web spam filtering based on the occurrences of keywords. Those keywords are either high advertisement or highly spammed.

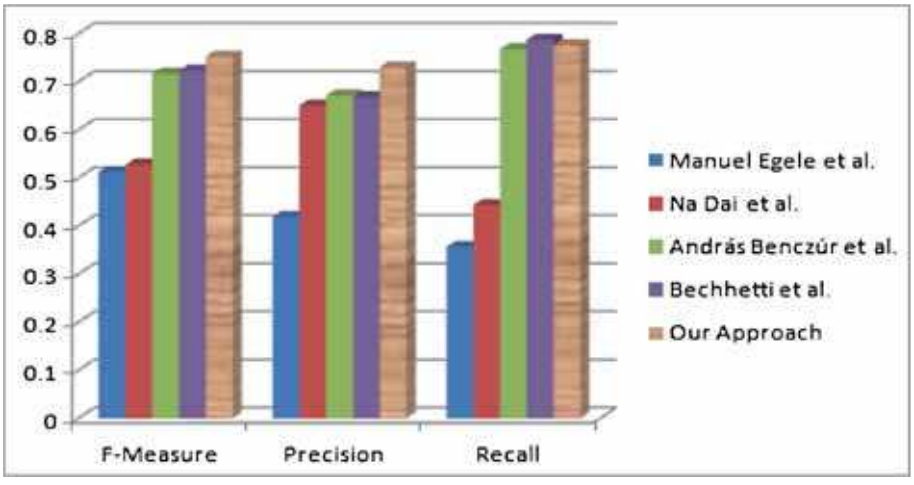

Figure 5. Comparison of proposed approach with other standard approaches.

Their features for Web spam detection include "online commercial intention", "the Yahoo Mindset classification of Web pages", "Google AdWords advertisement keywords suggestions", "the distribution of Google AdSense ads over pages of a site," etc. They have used WEBSPAM-UK2006 dataset for spam detection and achieved an F-measure of around 0.716 with precision around 0.671 which is still lesser than our results.

From the above comparisons, it is understood that the proposed approach clearly outperforms the above four spam detection techniques and is shown in figure 5.

6. Conclusion and future work

In this paper, a combined approach of content and link-based techniques is proposed to detect the spam Web pages. In content-based technique, we explored two methods namely term density and POS ratio test to identify a Web page as spam or non-spam. Similarly, our link-based technique used personalized page rank along with collaborative detection to identify a Web page as spam or non-spam. The experimental work has been carried out on WEBSPAM-UK2006 dataset of 11,402 Web pages. A very good and promising F-measure of 75.2% compared to other existing techniques demonstrates the strength of our approach. This work can be further extended by finding topical spam patterns to understand different tricks played by spammer in different Web pages. We believe that the bidirectional links between spam and unknown pages may help us to understand the spamming in a better manner. Also working in a distributed environment using map-reduce such as Hadoop for identifying the spam pages further reduce the running time of the algorithm.

References

[1] Ntoulas A, Najork M, Manasse M and Fetterly D 2006 Detecting spam web pages through content analysis. In: Proceedings of the 15th international conference on World Wide Web, ACM, pp 83–92

[2] Ghiam S and Pour A N 2012 A survey on web spam detection methods: Taxonomy. *Int. J. Netw. Security Appl.* 4(5): 119–134

[3] Ekstrand M D, Riedl J T and Konstan J A 2011 Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction* 4(2): 81–173

[4] Gyöngyi Z, Garcia-Molina H and Pedersen J 2004 Combating web spam with trustrank. In: Proceedings of the Thirtieth international conference on very large data bases-Volume 30, VLDB Endowment, pp 576–587

[5] Gyongyi Z, Berkhin P, Garcia-Molina H and Pedersen J 2006 Link spam detection based on mass estimation. In: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, pp 439–450

[6] Benczúr A A, Csalogány K and Sarlós T 2006 Link-based similarity search to fight web spam. In: AIRWEB, Citeseer

[7] Becchetti L, Castillo C, Donato D, Baeza-Yates R and Leonardi S 2008 Link analysis for web spam detection. *ACM Transactions on the Web (TWEB)* 2(1): 2

[8] Tian Y, Weiss G M and Ma Q 2007 A semi-supervised approach for web spam detection using combinatorial feature-fusion. In: Proceedings of the Graph Labelling Workshop and Web Spam Challenge at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, pp 16–23

[9] Abernethy J, Chapelle O and Castillo C 2010 Graph regularization methods for web spam detection. *Mach. Learn.* 81(2): 207–225

[10] Egele M, Kolbitsch C and Platzer C 2011 Removing web spam links from search engine results. *Journal in Computer Virology* 7(1): 51–62

[11] Ahmed F and Abulaish M 2013 A generic statistical approach for spam detection in online social networks. *Comput. Commun.* 36(10): 1120–1129

[12] Prieto V M, Álvarez M and Cacheda F 2013 Saad, a content based web spam analyzer and detector. *J. Syst. Softw.* 86(11): 2906–2918

[13] Luckner M, Gad M and Sobkowiak P 2014 Stable web spam detection using features based on lexical items. *Comput. Security* 46: 79–93

[14] Goh K L, Patchmuthu R K and Singh A K 2014 Link-based web spam detection using weight properties. *J. Intell. Inf. Syst.* 43(1): 129–145

[15] Piskorski J, Sydow M and Weiss D 2008 Exploring linguistic features for web spam detection: A preliminary study. In: Proceedings of the 4th international workshop on Adversarial information retrieval on the web, ACM, pp 25–28

[16] Andersen R, Borgs C, Chayes J, Hopcraft J, Mirrokni V S and Teng S H 2007 Local computation of pagerank contributions. In: Algorithms and Models for the Web-Graph, Springer, pp 150–165

[17] Langville A N and Meyer C D 2004 Deeper inside pagerank. *Internet Math.* 1(3): 335–380

[18] Dai N, Davison B D and Qi X 2009 Looking into the past to better classify web spam. In: Proceedings of the 5th international workshop on adversarial information retrieval on the web, ACM, pp 1–8

[19] Becchetti L, Castillo C, Donato D, Leonardi S and Baeza-Yates R 2008 Web spam detection: Link-based and content-based techniques. In: The European Integrated Project Dynamically Evolving, Large Scale Information Systems (DELIS): proceedings of the final workshop, vol 222,pp 99–113

[20] Benczúr A, Bíró I, Csalogány K and Sarlós T 2007 Web spam detection via commercial intent analysis. In: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web, ACM, pp 89–92