International Journal of Current Engineering and Technology
ISSN 2277 - 4106

Available at http://inpressco.com/category/ijcet

Research Article

Classifying Web Spam Using Block-based TrustRank

M. Sree vani[a*], R.Bhramaramba[b], D.Vasumati[c] ,O.Yaswanth Babu[d]

[a]Dept of CSE, MGIT, Gandipet, Hyderabad -500075
[b]Dept of IT, GITAM University, Vizag-530045
[c] Dept of CSE, JNTUH, Hyderabad-500032
[d]CMC Limited(A TATA Enterprise), Hyderabad-

Accepted 8 Nov.2012, Available online 1Dec. 2012, **Vol.2, No.4(Dec. 2012)**

Abstract

Web spamming refers to actions intended to mislead search engines into ranking some pages higher than they deserve. TrustRank is a recent algorithm that can combat web spam. However, the seed set used by TrustRank may not be sufficiently representative to cover well the different topics on the Web. In this paper, We propose the use of Combined page segmentation for selecting seed set in TrustRank algorithm and uses Block-level retrieval to rank the seed pages so that we can use highly multiple–topic ranked pages as seed set. Experimental results show that our approach deals effectively with the problem of multiple drifting topics and identify highly desirable pages for seed set and thus improve the performance of TrustRank.

Keywords: *spam, page segmentation, TrustRank.*

1. Introduction

Web spam is behavior that attempts to deceive search engine ranking algorithms. Many kinds of spam have been discovered (A. Perkins et al, 2001), but there is no universal method that can detect all kinds of spam at the same time. Gyongyi et al. present the TrustRank algorithm to combat web spam. The basic idea of this algorithm is that a link between two pages on the Web signifies trust between them; i.e., a link from page A to page B is a conveyance of trust from page A to page B. Initially, human experts select a list of seed sites that are well-known and trustworthy on the Web. Each of these seed sites is assigned an initial trust score. A biased Page Rank algorithm is then used to propagate these trust scores to their descendants. After convergence, good sites will have relatively high trust scores, while spam sites will have poor trust scores.

TrustRank may suffer from the fact that the coverage of the seed set used may not be broad enough. Many different topics exist on the Web and there are good pages within each topic. The seed selection process used in the TrustRank algorithm cannot guarantee that most of these topics are covered. So, it is possible that in using TrustRank to detect spam, we may get good precision but suffer from unsatisfactory recall. We propose that the trustworthiness of a page should be differentiated by different topics. i.e., the page should be more trusted in the topics that it is relevant to. This relies on the fact that a link between two pages is usually created in a topic-specific context (B. Wu et al, 2006).

In order to address the above issues, we propose the use of combined page segmentation which tries to take advantage of both visual information and fixed length, is very effective in dealing with the multiple-topic and varying length problems of web pages, and therefore can significantly improve the overall retrieval performance. Our approach called Block-based TrustRank retrieves highly multiple–topic ranked pages, which further uses as seed set. The final ranking is based on these seed set pages and Trust scores.

The rest of this paper is organized as follows: The related work introduced in Section 2.A brief introduction about combined page segmentation is presented in Section 3.The details of our approach given in Section 4.The experiments and results are shown in Section 5. Conclusion is presented in Section 6.

2. Related Work

Gyongyi et al. introduced TrustRank. It is based on the idea that good sites seldom point to spam sites and people trust these good sites. This trust can be propagated through the link structure on the Web. So, a list of highly trustworthy sites are selected to form the seed set and each of these sites is assigned a non-zero initial trust score, while all the other sites on the Web have initial values of 0. Then a biased Page Rank algorithm is used to propagate these initial trust scores to their outgoing sites.

M. Sree vani, R.Bhramaramba, D.Vasumati are working as Associate Professor and O.Yaswanth Babu is working as Senior IT Engineer

Jeh and Widom dwelled on the global notion of importance that Page Rank provides to create personalized views of importance by introducing the idea of preference sets. The rankings of results can then be biased according to this personalized notion. For this, they used the biased Page- Rank formula.

Chakrabarti et al. characterized linking behaviors on the Web using topical classification. Using a classifier trained on ODP topics, they generated a topic-topic citation matrix of links between pages that showed a clear dominant diagonal, which meant that pages were more likely to point to pages sharing their topic.

Recently, Chirita et al. described the method of combining ODP data with search engine results to generate a personalized search result. Based on a predefined user profile le, the distance of this profile to each URL received from a search engine's response list is calculated and these URLs are resorted to generate a new output for the user. Our approach makes similar use of human-edited directories, but our goal is to demote spam.

Guha et al. study how to propagate trust scores among a connected network of people. Different propagation schemes for both trust score and distrust score are studied based on a network from a real social community website.

Baoning Wu et al described Topical TrustRank, partition the set of trusted seed pages into topically coherent groups and then calculates TrustRank for each topic. The final ranking is based on a balanced combination of these individual topic specific trust scores. Deng Cai et al described Block based Web search, shows that semantic partitioning of web pages effectively deals with the problem of multiple drifting topics and mixed lengths, and thus has great potential to boost up the performance of current web search engines.

3. Combined Page Segmentation

CombPS tries to take advantage of both visual information and fixed length. The CombPS method is processed as the following two steps [4]:

Step 1. Vision-based Page Segmentation

People view a web page through a web browser and get a 2-D presentation which provides many visual cues to help distinguish different parts of the page, such as lines, blanks, images, colors, etc. For the sake of easy browsing and understanding, a closely packed block within the web page is much likely about a single semantic. We have previously proposed a vision-based page segmentation method called VIPS in (B. Wu et al, 2006). Similar to semantic passages, the blocks obtained by VIPS are based on the semantic structure of web pages. Traditional semantic passages are obtained based on content analysis which is very slow, difficult and inaccurate. VIPS discards content analysis and produce blocks based on the visual cues of web pages. This method simulates how a user understands web layout structure based on his or her visual perception. The DOM structure and visual information are used iteratively for visual block extraction, visual separator detection and content structure construction. Finally a vision-based content structure can be extracted. Since the method is totally top-down and the permitted degree of coherence can be pre-defined, the whole page segmentation procedure is efficient, flexible and more accurate from semantic perspective.

In Figure 1, the vision-based content structure of a sample page is illustrated. Visual blocks are detected as shown in Figure 1(b) and the content structure is shown in Figure 1(c). It is an approximate reflection of the semantic structure of the page. In VIPS method, a visual block is actually an aggregation of some DOM nodes. Unlike DOM-based page segmentation, a visual block can contain DOM nodes from different branches in the DOM structure with different granularities. Structural tags such as <TABLE> and <P> can be divided appropriately with the help of visual information, and wrong presentation of DOM structure can be reorganized to a proper form. Therefore, VIPS can achieve a better content structure for the original web page. After the vision-based content structure is obtained, all the leaf visual blocks are taken as the input to the next step for block extraction.

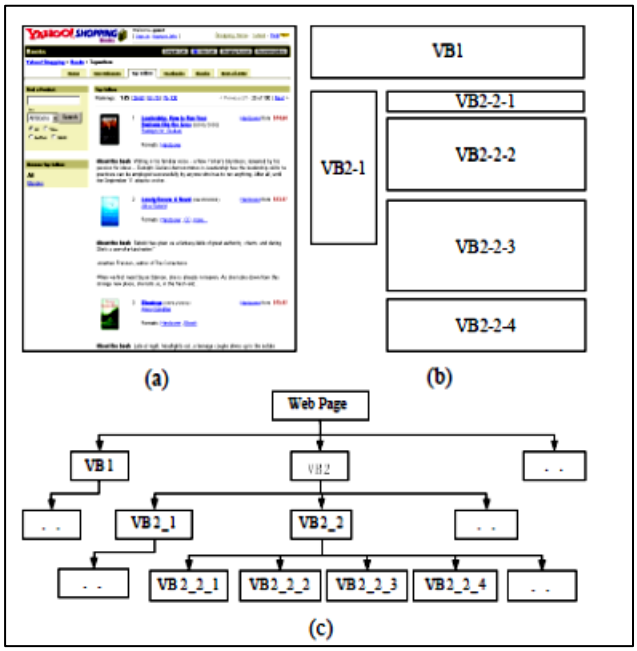

Figure 1. Vision-based content structure for the sample page

For each visual block obtained in the previous step, overlapped windows are used to divide the block into smaller units. The first window begins from the first word of the visual block, and subsequent windows half-overlap preceding ones till the end of the block. For visual blocks that are smaller than the pre-defined length of the window, they are directly outputted as final blocks without further partition. Upon this strategy, large visual blocks are departed into smaller ones and thus greatly reduce the impact of varying length. Compared with fixed-length approach FixedPS, CombPS utilizes semantic information in partitioning and makes page segmentation insensitive to

queries. By allowing small semantic blocks to directly be parts of segmentation results, CombPS intuitively obtains a more diverse and "correct" segmentation result set.

4. Block-based TrustRank

Block Retrieval – Similar to passage retrieval, block retrieval performs the retrieval task at the block level and aims to adjust the rank of documents with the blocks they contain.

4.1. The TrustRank Algorithm

Function TrustRank, shown in Figure 2, computes trust scores for a web graph. The input to the algorithm is the graph (the transition matrix T and the number N of web pages) and parameters that control execution (L, M_B, α_B). As a first step, the algorithm calls function SelectSeed, which returns a vector s. The entry s(p) in this vector gives the "desirability" of page p as a seed page. In step (2) function Rank(x, s) generates a permutation x^1 of the vector x, with elements $x^1(i)$ in decreasing order of $s(x^1(i))$. In other words, Rank reorders the elements of x in decreasing order of their s-scores. Step (3) invokes the oracle function on the L most desirable seed pages. The entries of the static score distribution vector d that correspond to good seed pages are set to 1. Step (4) normalizes vector d so that its entries sum up to 1. Finally, step (5) evaluates TrustRank scores using a biased Page Rank computation with d replacing the uniform distribution.

```
Function TrustRank
                    Input
                        T     transition matrix
                        N     number of pages
                        L     limit of oracle invocations
                        α_B   decay factor for biased Page Rank
                        M_B   number of biased Page Rank iterations
                    Output
                        t*    TrustRank scores
                    begin
                    //evaluate seed-desirability of pages
                    ( 1 )   s=SelectSeed(…)
                     //generate corresponding ordering
                     ( 2)   σ =Rank({1,….,N},s)
                      //select good seeds
                     ( 3 )  d=0_N
                        For i= 1 to N do
                    if O(σ (i))= = 1 then
                            D(σ (i)) = 1
                     //normalize static score distribution vector
                     ( 4 ) d = d/|d|
                     //compute TrustRank scores
                  ( 5 ) t* = d
                        For i=1 to MB do
                        t* = α_B * T * t* + (1- α_B) * d
                        return t*
                        end
```

Figure 2:The TrustRank algorithm

4.2.Selecting Seeds

The goal of function SelectSeed is to identify desirable pages for the seed set. That is, we would like to find pages that will be the most useful in identifying additional good pages. Our approach for selecting a seed set is to give preference to pages with high Block-based Rank using Combined page segmentation. Our approach contains the following steps:

Step 1. Initial Retrieval

An initial list of ranked web pages is obtained by using the Okapi system. The document rank obtained in this step is called DR.

Step 2. Page Segmentation

A Combined page segmentation method is applied to partition the retrieved pages into blocks. All of the extracted blocks form a block set.

Step 3. Block Retrieval

This step is similar to Step 1, except that documents are replaced by blocks. The same queries are used to get a block rank BR. After obtaining the block rank, pages can be re-ranked based on the single best-ranked block within each page, though we can also consider several top blocks of each page to re-rank the page. Besides this simple approach, a combined rank is also presented in our experiments like in [5], in which the rank of each web page d is determined by

$$\alpha \cdot rank_{DR}(d) + (1-\alpha) \cdot rank_{BR}(d)$$

Since high- Block based Rank pages are likely to point to other high-block based Rank pages, then good trust scores will also be propagated to pages that are likely to be at the top of result sets. Thus, with Block-based Rank selection of seeds, we may identify the goodness of fewer pages (as compared to inverse Page- Rank), but they may be more important pages to know about.

5. Experiments and Results

5.1 Data Set

We perform a number of experiments on a real web graph which is a partial set of pages crawled by Tianwang search engine (developed by network lab, Peking University) in

Nov. 2005. The data set is consists of 13.3M pages and 232M links among these pages. We divide these pages into 358,245 hosts according to their URLs, most of which belong to .cn domain.

5.2 Seed Set

We chose the top 40 pages based on (BR+DR) rank as seed set for TrustRank algorithm. For comparing with Trust-Rank we picked the top 40 pages based on inverse page rank. We found that we could benefit substantially from a larger seed set. Also we used the common α value of 0.4 .We strongly feels that our approach of seed selection gives better results than inverse Page Rank because its execution time is polynomial in the number of pages, while determining the maximum coverage is an NP-complete problem. As a first step, we conducted experiments to compare the Block-based retrieval and inverse Page Rank seed selection.

5.3 Results

We use the number of spam sites within the top 10 buckets as our first metric to measure the performance of an algorithm. The distribution of spam sites within the top ten buckets for these two algorithms is shown in Figure3.

We used the spam bucket distribution to evaluate the performances of the algorithms. Given an algorithm, we sorted the 5.6-million websites in descending order of the scores that the algorithm produces. Then we put these sorted websites into 15 buckets. The numbers of the labeled spam websites over buckets for Page Rank, TrustRank, and Block-based TrustRank are listed in Table1

Table1: No. of Spam websites over buckets

Bucket No	No. of websites	TrustRank	Block-based TrustRank
1	15	0	0
2	148	1	1
3	720	11	4
4	2231	20	18
5	5610	34	39
6	12600	56	88
7	25620	112	87
8	48136	128	121
9	87086	177	156
10	154773	294	183
11	271340	320	198
12	471046	366	277
13	819449	443	323
14	1414172	424	463
15	2361420	328	756

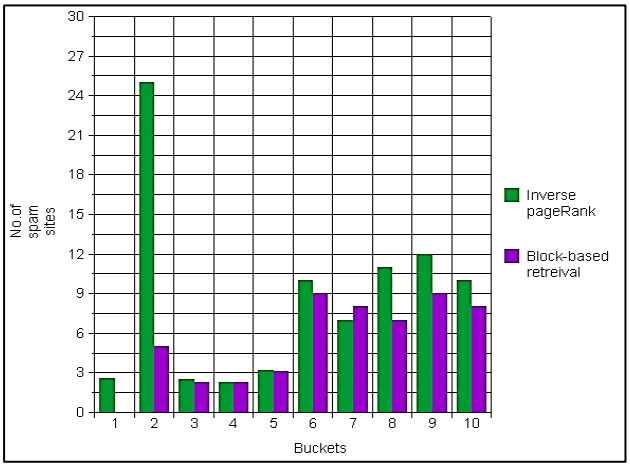

Figure 3. Spam Distribution

We see that Block-based TrustRank can successfully push many spam websites to the tail buckets, and the number of spam websites in the top buckets in Block-based TrustRank is smaller than TrustRank. That is to say, Block-based TrustRank is more effective in spam fighting than Page Rank and TrustRank.

6.Conclusion

In this paper , we used Combined page segmentation rather than inverse Page Rank algorithm for selection of seed set in TrustRank algorithm so that we can use Vision-based and fixed length properties to rank the web page. Our experimental results show that our approach deals effectively with the problem of multiple drifting topics and identify high desirable pages for seed set and thus improve the performance of TrustRank.

References

G. Jeh and J. Widom (May 2003),Scaling personalized web search, Proceedings of the Twelfth International World Wide Web Conference, pp. 271- 279, Budapest, Hungary.

Z. Gyöngyi, H. Garcia-Molina, J. Pedersen(2004), Combating Web Spam with TrustRank,
VLDB, pp. 576-587.

A. Ntoulas, M. Najork, M. Manasse and D. Fetterly (2006), Detecting spam web pages through content analysis, Proceedings of the 15th International Conference on World Wide Web.

B. Wu, V. Goel and B. D. Davison (2006),Topical TrustRank: using topicality to combat web spam, Proceedings of the 15th international conference on World Wide Web.

P. Chirita, W. Nejdl, R. Paiu, and C. Kohlschutter. (Aug. 2005),Using ODP metadata to personalize search. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 178-185, Salvador, Brazil.

J. Hirai, S. Raghavan, H. Garcia-Molina, and A. Paepcke.(2000), WebBase: a repository of Web pages, *Computer Networks*, 33(1-6):277-293.

Räber Information Management GmbH. The Swiss search engine, 2005.
http://www.search.ch

S. Chakrabarti, M. Joshi, K. Punera, and D. Pennock. (2002), The structure of broad topics on the web, Proceedings of 11th International World Wide Web Conference, pp. 251-262,Honolulu, *ACM Press Hawaii*, US..

Deng Cai , S.YU and Wei-Ying Ma (2004),Block-based Web Search, SIGIR'04,july 25-29,Sheffield,South Yorkshire,UK.

R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. (May 2004),Propagation of trust and distrust. In Proceedings of the 13th International World Wide Web Conference, pp. 403-412, New York City.

Z. Gyongyi, H. Garcia-Molina, and J. Pedersen (Sept.2004), Combating web spam with TrustRank. In Proceedings of the 30th International Conference on Very Large Data Bases (VLDB), pp. 271-279, Toronto, Canada.

A. Perkins (Sept.2001), White paper: The classification of search engine spam,Online at http://www.silverdisc.co.uk/articles/spamclassification/.