Spam Detection with a Content-based Random-walk Algorithm

F. Javier Ortega
Departamento de Lenguajes y
Sistemas Informáticos
Universidad de Sevilla
Av. Reina Mercedes s/n
41012, Sevilla (Spain)
javierortega@us.es

Craig Macdonald
Department of Computer
Science
University of Glasgow
Glasgow, G12 8QQ, UK
craigm@dcs.gla.ac.uk

José A. Troyano
Departamento de Lenguajes y
Sistemas Informáticos
Universidad de Sevilla
Av. Reina Mercedes s/n
41012, Sevilla (Spain)
troyano@us.es

Fermín Cruz
Departamento de Lenguajes y
Sistemas Informáticos
Universidad de Sevilla
Av. Reina Mercedes s/n
41012, Sevilla (Spain)
fcruz@us.es

ABSTRACT

In this work we tackle the problem of the spam detection on the Web. Spam web pages have become a problem for Web search engines, due to the negative effects that this phe-nomenon can cause in their retrieval results. Our approach is based on a random-walk algorithm that obtains a ranking of pages according to their relevance and their spam likelihood. We introduce the novelty of taking into account the content of the web pages to characterize the web graph and to ob-tain an a-priori estimation of the spam likekihood of the web pages. Our graph-based algorithm computes two scores for each node in the graph. Intuitively, these values represent how bad or good (spam-like or not) is a web page, according to its textual content and the relations in the graph. Our experiments show that our proposed technique outperforms other *link-based* techniques for spam detection.

Keywords: Information retrieval, Web spam detection, Graph algorithms, PageRank, web search

1. INTRODUCTION

Web spam is a phenomenon where web pages are created for the purpose of making a search engine deliver undesirable results for a given query, ranking these web pages higher than they would otherwise [16]. Spam web pages have become a problem for Web search engines, causing negative effects in their retrieval results [10]. Basically, there are two forms of spam intended to cause undesirable effects: Self promotion and Mutual promotion [5]. Self promotion tries to create a web page that gains high relevance for a search engine, mainly based on its content. This can be achieved through many techniques, such as word stuffing, in which visible or invisible keywords are inserted in the page, in order to improve the retrieved rank of the page for the most common queries. Mutual promotion is based on the cooperation of various sites, or the creation of a wide number of pages that form a *link-farm*, that is a large number of pages pointing one to another, in order to improve their scores by increasing the number of in-links to them. This method is effective against search engines that employ the co-citations between pages as features (i.e. PageRank [18]).

There are different approaches to deal with the problem of web spam, using different information sources to decide whether a web page is spam or not. These approaches can be classified into two groups, depending on the spam mechanism that they attempt to identify. *Content-based* techniques use the textual content of the web pages to classify them. These methods usually examine the distribution of statistics about the contents in spam and not-spam web pages, such as the number of words in a page, the HTML invisible content, the most common words in a page in relation with the ones in the entire corpus, etc.[6, 7]. In general, content-based tackle self promotion.

On the other hand, *link-based* techniques focus on the structure of the graph made up of the web pages and the hyperlinks among them. These methods study the relations of the pages in the web graph, aiming to detect the link-farms of spam web pages. The basic assumption to deal with link-farms is that similar objects are related to similar objects in the web graph [11]. In the context of web spam, it means that non-spam web pages are frequently related to other non-spam web pages, and vice versa. Link-based methods are intended to deal with the mutual promotion mechanisms.

In this work, we introduce a method that integrates concepts from both techniques for spam detection. Intuitively, our approach uses some content-based heuristics to characterize a link-based algorithm, in such way that the information provided by the heuristics improves the ranking of pages obtained by the graph-based algorithm. The promotion of the good pages and the demotion of the bad ones in the final ranking are produced according to their contents and their relations in the Web graph. The aim of our approach is to build a ranking of web pages according to their relevance, using content-based metrics to demote the spam pages, in order to avoid their presence in the first positions of the ranking.

The organization of the rest of the paper is as follows. In the next section, we discuss other works that tackle the problem of web spam detection from different points of view. In Section 3, we introduce the intuition behind our approach, and explain the components of our method: the set of content-based metrics, and the way in which these heuristics are used in the creation of the graph model. The experimental design and results are shown in Section 4. Finally, we remark on our conclusions concerning the present work, and talk about some ideas for future works.

2. RELATED WORK

We firstly review one of the most popular methods for link analysis. PageRank [18] is a well-known method intended to obtain a ranking of nodes according to their centrality in a network. Let $G = (V, E)$ be a directed graph with the set of vertices V and set of directed edges E. For a given vertex $v_i \in V$, let $In(v_i)$ be the set of vertices that point to it (predecessors), and $Out(v_i)$ the set of vertices that v_i points to (successors). The PageRank score for each node can be defined as follows:

$$PR(v_i) = (1-d)e_i + d \sum_{j \in In(v_i)} \frac{1}{|Out(v_j)|} PR(v_j)$$

where $d \in [0, 1]$ is the damping factor that represents the probability of a random surfer to jump randomly to a page not pointed by the current one. Intuitively the e vector corresponds to the distribution of web pages that a random surfer periodically jumps to. It can be used to give views of the Web which are focussed or biased on some aspects [9].

There are many methods intended to tackle the problem of the spam detection, based on link analysis techniques like PageRank. Next, we review some of them.

The key assumption in [2] is that supporters of a non-spam page should not be overly dependent on one another. In other words, if the supporters of a web page have a large numbers of links between them, they likely form a *link-farm*, and could be spam web pages. An example of suspicion is the previously mentioned case of a page that receives its PageRank from a large number of very low ranked pages. The proposed algorithm obtains the supporters of each page, and then studies the distribution of their PageRank scores, in order to compute a PageRank biased according to a vector of penalizations.

Truncated PageRank [1] tackles the problem of the link-farms. It penalizes pages that obtain a large share of their PageRank from the nearest neighbors, avoiding the effect of the supporters that are topologically very close to a given node.

TrustRank [8] is based on the idea that a high PageRank score held on a huge amount of links from pages with low PageRank, is suspicious of being Spam. It means that a node with high PageRank and no relations with others pages with high PageRank is likely to be a spam web page. They obtain an estimator for this metric by calculating the *estimated non-spam mass*, that is the amount of PageRank received from a set of (hand-picked) trusted pages. In contrast, [14] proposes an algorithm with the same idea, but taking as input a set of spam web pages. This technique, called Anti-TrustRank, computes the *estimated spam mass* for each node.

In [19], Wu et al. propose an approach based on trust and distrust propagation. This work consists in an algorithm that computes two scores for each node in the graph, indicating the levels of trust and distrust of a page. The process starts from two seed sets, trustworthy and spam pages, respectively.

In contrast to link-based techniques, content-based techniques are focused on determining whether a page is spam or not according to its textual content. Mishne et al. [15] propose the comparison of language models to classify texts as spam or non-spam. In [13], Kolari et al. present a machine learning technique based on SVM, taking as features different heuristics, such as the anchor text of the links in a web page, the tokenized URL of the page, or the meta-tag text. In [17], Ntoulas et al. proposed several spam detection metrics. They compare the values of this content-based metrics for spam and not-spam web pages, and discuss the discrimination ability of each metric to detect spam. Some of the proposed heuristics are the number of words in the title of a web page, the average length of words, the amount of invisible content, the compression rate of the web pages, the fraction of anchor text with respect to the total amount of text in a page, etc.

A system that combines content-based and link-based features is proposed in [4]. They discuss three methods to include features related to the web graph topology, into a classifier. Some well-known algorithms are used in this work, such us TrustRank and PageRank. A variant of this algorithm, called Truncated PageRank, is also proposed. It does not include the direct neighbors of a node in calculating its score, in order to avoid the effects of the link-farms in the ranking.

The impact of spam in information retrieval systems, and the effects of some anti-spam filters are studied in [5]. They use three filters in this work, and a naive Bayes classifier to combine all of them. The first filter is a classifier built from a labeled corpus with spam and non-spam pages. The second one consists of a set of documents retrieved by some of the most popular queries to a web search engine. And finally a set of documents extracted from the Open Directory Project[1]. They show the improvements achieved in some of

[1] http://www.dmoz.org

the systems participants in TREC 2009[2] applying a spam detection technique.

3. OUR APPROACH

The random-walk algorithms have been shown to be reliable methods to obtain a ranking of nodes according to its relevance in a graph [12, 18]. However, these methods can fail dealing with the problem of the mutual promotion of spam pages or link-farms. Some of the works mentioned above present variations of random-walk algorithms, in order to succeed against these kind of attacks. These works also show that, due to the link-farms phenomenon, most of the pages pointing to a spam page, are spam pages, and vice versa.

Link-based methods present interesting strategies to detect link-farms, and to demote the pages within them. Truncated PageRank [1] does not take into account the nearest neighbors of a web page in the computation of its score, assuming that in the case of a bad page, those neighbors are also bad pages. In other words, they form a link-farm. TrustRank [8] assumes that a page that gains high score from many low scored pages, is likely to be a spam. This method needs a set of trusted web pages to obtain the amount of PageRank score that a web gains from them. The Anti-TrustRank [14] algorithm computes the opposite score, using as seeds a set of hand-picked spam web pages.

Our approach consists of a PageRank-based algorithm that computes two scores for each node: a positive score representing the authority of a web page, and a negative score which represents the spam likelihood of a page. The difference between both scores is taken into account in order to build a ranking of web pages. Intuitively, this value represents the overall authority of a web page. In this way, web pages with high negative scores are demoted in the final ranking, because they are likely to be spam. In our algorithm, the positive score of a web page must be affected only by good pages, and the negative score must only change according to the relation of this page with spam-like web pages.

With this aim, we compute a *spam-biased* random-walk algorithm that gives more relevance to a specific set of seeds, in a similar way as TrustRank. Since we compute two scores for each page, we need two sets of seeds, each of them intended to reinforce the positive or negative relevance of each type of web pages in the graph. Thus, the first set of seeds must contain a group of non-spam pages, and the second one consists in a group of spam pages. At this point, we propose an automatic process to obtain the seed sets for our algorithm, instead of relying in human-picked ones. Our method is based on some simple content-based heuristics for spam detection. These metrics give an intuition about the spam likelihood of a page, according to its textual content.

Isolated pages, i.e. pages without any in-links or out-links are also a problem for random-walk algorithm. Indeed, despite not being able to use any link-based heuristics for these pages, the textual content provides useful information to obtain a spam likelihood score for them.

In the remainder of this section, we discuss the algorithm proposed in this work (Section 3.1), the content-based metrics used to characterize our algorithm(Section 3.2), and the automatic methods to obtain the sets of seeds(Section 3.3).

3.1 Algorithm for Spam Detection

As mentioned above, we propose a random-walk algorithm to obtain a ranking of the web pages, according to their relevance. This algorithm is intended to demote the spam web pages in the overall ranking by computing two scores for each page, PR^+ and PR^-. Given a page A, it is desirable that its positive score, PR^+, depends on the good pages pointing to A, and analogous for the negative score, PR^-. In other words, we want the spam web pages to propagate their negative scores to their neighbors, and the positive pages do the same with their positive scores. With this aim, two vectors, e^+ and e^-, are built based on a set of content-based metrics from each page. These spam-biased vectors are used in the computation of our random-walk algorithm, representing the non-spam and the spam likelihoods of a page, respectively. They can be thought of a reinforcement for the positive and negative score of each page. Having said that, the proposed scores are obtained as shown in Equations (1 and 2) below:

$$PR^+(v_i) = (1-d)e_i^+ + d \sum_{j \in In(v_i)} \frac{PR^+(v_j)}{|Out(v_j)|} \quad (1)$$

$$PR^-(v_i) = (1-d)e_i^- + d \sum_{j \in In(v_i)} \frac{PR^-(v_j)}{|Out(v_j)|} \quad (2)$$

where v_i is a node of the graph (a web page), $In(v_j)$ is the set of nodes pointing to v_j, and $Out(v_j)$ is the set of nodes which v_j points to. Both scores, PR^+ and PR^-, are obtained with a PageRank-like algorithm. The algorithm iterates over the nodes in the graph, applying Equations (1) and (2). This process is performed until the maximum difference between the scores in one iteration and the previous one, is lower than a given threshold, T. This algorithm has the same time complexity as the original one. In the next section, we discuss the content-based metrics computed to obtain the spam-biased vectors.

3.2 Content-based metrics

The intuition behind the use of content-based metrics in conjunction with a random-walk algorithm lies on the characterization of the web graph with the information provided by the content of the pages. It is possible to determine some features of the graph by using some simple metrics. The aim behind this idea is to increase the ranking of the good web pages and penalize the bad ones.

The content-based metrics that we use in the experiments of this work have been chosen according to their discrimination capacity, distinguishing between spam and not-spam web pages, as identified by [17]. Another important factor to select these metrics is their computational complexity. Following these criteria, we have implemented three heuristics:

- **Compressibility**: is defined as the fraction of the sizes of a web page, x, before and after being compressed.

  $$Compressibility(v_j) = \frac{GZIPSize(v_j)}{TotalSize(v_j)}$$

  A web page with a very high compressibility value, is likely to be a spam. This heuristic is intended to detect repeated content or words in a web, because more redundant content leads to a higher compression ratio.

[2] `http://trec.nist.gov`

- **Fraction of globally popular words**: a web page with a high fraction of words within the most popular words in the entire corpus, is likely to be a spam. This metric scores spam self promotion techniques such as the word stuffing mechanisms.
- **Average length of words**: non-spam web pages have a bell-shaped distribution of average word lengths, while malicious pages have much higher values. Hence, this heuristic penalizes the use of word stuffing mechanisms.

In the next section, we explain how we use these heuristics to obtain the spam-biased vectors for our algorithm.

3.3 Obtaining the seed sets

As we previously explained, our approach uses the content-based heuristics to automatically select the sets of seeds that will be used in our algorithm to obtain a final ranking of web pages by spam likelihood. A given page will be demoted or promoted in the ranking according to its relations with this bad and good web pages. We use seed sets to ensure that negative scores of negative pages will be propagated over the graph, and analogous for the positive seeds. The seed sets are represented in our approach by two spam-biased vectors, e^+ and e^-. The vectors contain the spam and non-spam likelihoods of the web pages taken as seeds in our algorithm, giving higher positive or negative scores to those nodes that have higher e^+ or e^- (see Equations (1) and (2)).

In this section, we introduce three different ways to build the spam-biased vectors, given the heuristics of each web page.

3.3.1 N most positive/negative pages

This first method chooses the N most positive and negative pages in the graph as seeds, according to their metrics. Formally, given a page v_j, let $M(v_j)$ be a vector with the content-based metrics for v_j. The spam likelihood of v_j will be determined by the norm of $M(v_j)$, as shown in Equation (3):

$$Spaminess(A) = \sqrt{\sum_{h \in M(v_j)} h^2} \qquad (3)$$

where A is a web page, and h represents the heuristics which $M(v_j)$ contains.

In this way, we take the N nodes with highest *Spaminess* as negative seeds, and the N nodes with lowest *Spaminess* as positive seeds. The spam-biased vectors can be defined as follows:

$$e_i^+ = \begin{cases} 1/N & \text{if } i \in S^+ \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

where N is a parameter that specifies the number of seeds that will be taken. S^+ is the set of the N nodes with lowest Spaminess in the graph. The formula is analogous for vector e^-.

3.3.2 N most positive/negative pages with content-based weights

Following the previous schema, we can take advantage of the content-based metrics by including the actual scores directly in the computation of the weights of the seeds, as shown in Equation (5):

$$e_i^+ = \begin{cases} \frac{Spaminess(i)}{\sum_{j \in S^+} Spaminess(j)} & \text{if } i \in S^+ \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where $Spaminess(i)$ is the norm of the vector built from the metrics of the page i (see Equation 3).

3.3.3 Content-based nodes characterization

The last method consists in applying the previous formula to every nodes in the graph. We can rely on the thresholds proposed in the study in [17], and use them to determine whether a page must be a negative or a positive seed. The thresholds for the metrics considered in the present work are shown in Table 1. Given a page, if one of its metrics is above the corresponding threshold, we include the page in the set of negative seeds, and in other case it will be taken as a positive seed. Once the sets of nodes have been defined, we apply the same formulas shown in Equation (5).

Heuristics	Threshold
Compressibility	6.0
Fraction of globally popular words	0.75
Average length of words	9.0

Table 1: Thresholds for the content-based metrics.

4. EXPERIMENTS

In this section, we show the experimental design defined to show the performance of our technique, the dataset used and the results obtained. We also detail the values of the parameters for each set of experiments, and the different variants proposed in this work.

The aim of the experiments is to show the performance of our approaches in terms of demotion of spam web pages in a ranking, and to compare them to a state-of-art spam detection technique. The experiments are intended to assess the usefulness of including content-based heuristics into a random-walk algorithm, in order to build a ranking of web pages according to their relevance, penalizing the spam web pages.

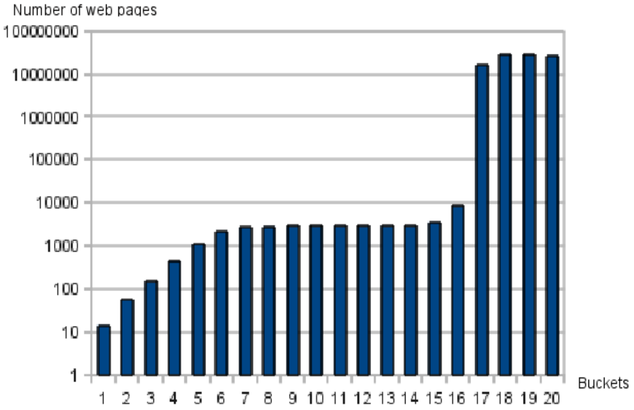

Figure 1: Web pages per bucket according to PageRank, in a log scale.

4.1 Dataset

The corpus used in the experiments for the paper is the WEBSPAM-UK2006 Dataset [3] for spam detection. It contains more than 98 million pages. The collection is based on a crawl of the .uk domain performed in May 2006. It was collected by the Laboratory of Web Algorithmics, Università degli Studi di Milano, with the support of the DELIS EU-FET research project. The collection was labeled by a group of volunteers and/or by domain-specific patterns such as .gov.uk or .ac.uk. Of the 11,402 hosts in UK2006 dataset, 7,423 are labeled as spam or non-spam. For the evaluation purposes, we have considered as spam any web page that belongs to a host labeled as spam. There are about 10 million spam web pages in the collection.

4.2 Evaluation

The purpose of our algorithm is to build a ranking of web pages, demoting the spam web pages in the ranking and trying to put them as far as possible from the first positions of the ranking. Since our approach does not classify the web pages between spam or non-spam, it do not make sense to perform an evaluation in terms of classification accuracy. We use in our experiments the same evaluation method followed in other works on the application of graph-based algorithms to the spam detection task [2, 8]. Our intuition is that it is more important to correctly detect the spam in high PageRank valued sites, because they will often appear in the first positions in the search results for many queries. The aim of this evaluation method is to easily determine the number of spam web pages detected mainly in the highest positions of the ranking of web pages.

The evaluation method is implemented as follows. First, a list of pages is generated in decreasing order of their PageRank score. This list is segmented into 20 buckets, in such way that each of the buckets contains a different number of sites, with scores summing up to 5% of the total PageRank score. The nodes per bucket are plotted in Figure 1. Once we obtain the size of the buckets according to the PageRank scores of the web pages, we use them to build a set of buckets with the rankings computed in each experiment. For evaluation purposes, buckets of the same sizes as the ones in Table 1 are built with the results of each method.

The number of spam web pages per bucket is our evaluation metric. It is obtained by counting the number of pages in each bucket that are labeled as "spam" in the dataset. The aim of a spam detection technique is to avoid spam pages into the first buckets, demoting these pages in order to put them into the lastest buckets. In the next sections, we show the performance of the proposed approaches. The parameters of the PageRank, the damping factor and the threshold, have been set to 0.85 and 0.01, respectively, for all the experiments shown in this work.

4.3 TrustRank

TrustRank algorithm [8] is a link-based algorithm that takes as input the web graph and a set of non-spam web pages, chosen in a semi-supervised way, that are the seeds for the algorithm. The output is a ranking of web pages according to their relevance, in which the spam web pages are demoted. TrustRank computes a score for each web page, as PageRank, using the seed set to include a bias in the random-walk algorithm. In order to build the seed set, they propose an *inverse PageRank*, taking into account the out-links of the web pages, instead of their in-links. Then they choose by hand a number N of non-spam web pages from that ranking. In this way, they try to take as seeds the N good pages that reach as many nodes as possible.

The results shown in Figure 2 have been obtained by performing 20 iterations of the algorithm with a damping factor of 0.85, as suggested in [8]. Since they use a set of 178 seeds with a dataset of 13 million pages, we have taken a seed set with $178 * 3 = 534$ pages from our dataset of 99 million pages.

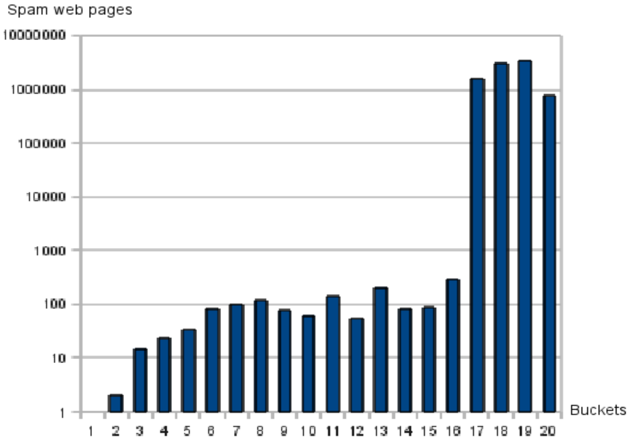

Figure 2: Spam web pages per bucket, using TrustRank algorithm.

We test this technique with the UK2006 dataset, and it achieves good results. There are less than a 3% of spam web pages in the two first buckets. However, more than the 10% of pages in the third bucket are spam.

4.4 N most positive/negative pages

The first set of experiments corresponds to the method introduced in Section 3.3.1. We have selected the 5% of the most positive and negative nodes as the positive and negative seed sets, respectively. The results are shown in Figure 3.

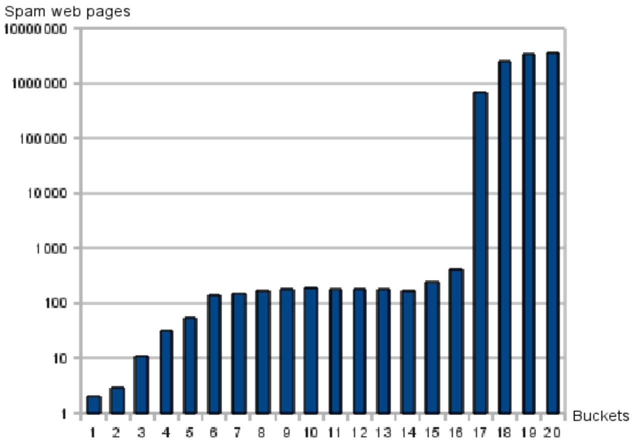

Figure 3: Results using 5% most positive and negative nodes as seeds.

In the figure, we can see that the 14% of web pages in the first bucket are spam, and the amount of spam pages per bucket is around the 6% of the total number of pages in the rest of the dataset, except for the last buckets. These results are worse than the ones with TrustRank, mainly in the first buckets which are the most important.

4.5 N most positive/negative pages with content-based weights

The results applying the method in Section 3.3.2, that includes the content-based heuristics directly in the algorithm are shown in Figure 4, presenting the number of spam web pages in each bucket, taking the 5% of the most positive and negative nodes as seeds.

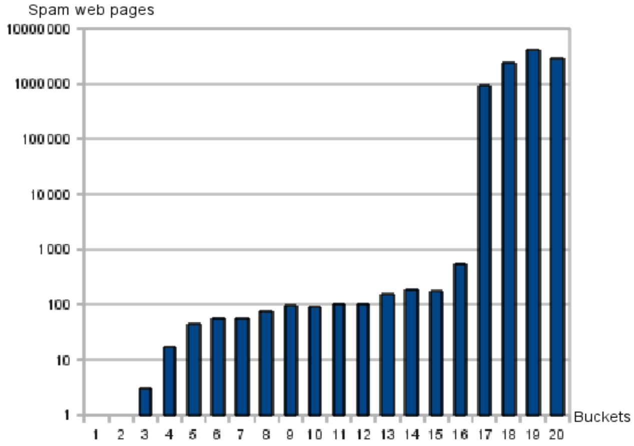

Figure 4: Results using the 5% most positive and negative nodes as seeds. The seed weight is set according to the content-based metrics.

Comparing the charts in Figures 3 and 4 we can see the improvement achieved by the inclusion of the metrics in the weights of the seeds. Our second approach is more effective at demoting the spam web pages in the high PR buckets. Furthermore, it outperformes the results of TrustRank algorithm, allowing a maximum of 4% of spam web pages into the 12 first buckets.

4.6 Content-based seed characterization

In this section we present the experimental results achieved with the method explained in Section 3.3.3. This approach applies the same method as the previous section, but taking all the nodes in the web graph as seeds for the algorithm. The results can be seen in Figure 5.

The results show a noticeable improvement with respect to TrustRank, although the approach in Section 3.3.2 achieves a better performance in terms of demotion of spam web pages in the final ranking.

4.7 Comparative study

A recap of the results presented in this work is shown in Table 2. The first column represents the buckets of pages and the second one contains the number of web pages from the first bucket to the current one, inclusive. The rest of the columns shows the accumulated number of spam web pages for each technique, that is the total number of spam web pages from the first bucket to the current one, inclusive. In the table, TR corresponds with TrustRank algorithm; PNS (Positive/Negative Seeds) is our first approach, that takes the N most positive and negative pages as seeds and assigns to each of them a weight of $1/N$; PNS+M (Positive/Negative Seeds + Metrics) is our second approach, based on the previous one, but setting the weights of the seeds according to the content-based metrics; and finally, CbC (Content-Based Characterization) corresponds with our third approach that takes every node as a seed for the random-walk algorithm.

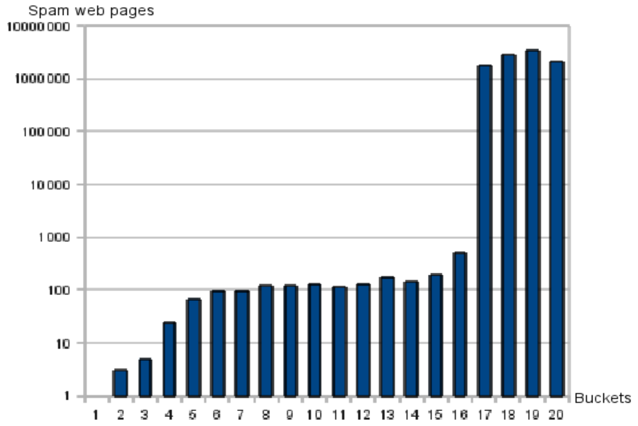

Figure 5: Results using all the nodes in the web graph as seeds. The seed weight is set according to the content-based metrics.

#	Pages	TR	PNS	PNS+M	CbC
1	14	**0**	2	**0**	**0**
2	68	*2*	5	**1**	3
3	212	17	16	**4**	*8*
4	649	40	48	**21**	*32*
5	1719	*73*	104	**66**	101
6	3849	*155*	244	**124**	199
7	6513	*254*	392	**180**	297
8	9291	*371*	557	**255**	416
9	12102	*448*	742	**350**	537
10	14914	*511*	937	**440**	650

Table 2: Accumulated number of spam web pages for each method: TrustRank (TR), Positive/Negative seeds (PNS), Positive/Negative Seeds with metric-based weights (PNS+M) and Content-based characterization (CbC)

Since the first positions of the ranking are the most important for us, only the first 10 buckets are presented in the table. We can see that PNS+M achieves the best results within each bucket, outperforming the TrustRank. In contrast, the first approach does not improve the TrustRank algorithm. The relevance of including the content-based metrics in the random-walk algorithm is clear regarding the difference between these two experiments. The content-based characterization method also presents good results, with only 32 spam web pages in the first 649 pages, outperforming TrustRank in those first buckets as well.

5. CONCLUSIONS

In this work, we have introduced a novel method to deal with the web spam pages. Our approach combines concepts from known link-based and content-based techniques to avoid the negative effects of spam web pages in a web search engine. We use a graph-based algorithm to obtain a ranking of pages according to their relevance. In addition to this algorithm, we propose the inclusion of some metrics into the graph in order to promote good pages and demote the bad ones, regarding the textual content of the pages. This is done by the implementation of some simple heuristics that provides the algorithm with information about the spam likelihood of the pages, based on their content. Our method achieves good experimental results applied to a big dataset of more than 98 million pages.

We plan to further our research by studying the effects of including other content-based metrics in the method. It is interesting to find out the relation between the improvement achieved by the inclusion of new heuristics and the time complexity of our algorithm. On the other hand, the spam-biased selection of the seeds could be improved in many other ways, for example taking into account the amount of inlinks of the nodes, as proposed in [8, 14]. Another idea is to use the content-based metrics to characterize not only the seeds of the algorithm, but also the edges of the web graph. In this way, we could automatically set the weights of the edges according to the heuristics, giving more relevance to the relations between some kinds of pages, or even decreasing the negative effects that some sort of links could cause in the algorithm, such us links between spam-like web pages (link farms). We intend to integrate it in a spam classifier, using many features in order to perform a binary classification of the web pages. We also plan to apply these ideas to the task of finding the trustworthiness users in a social network. This problem can be modelled as a spam-like task, in which spam web pages are now malicious or untrustworthiness users.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Education and Science (HUM2007-66607-C04-04). Special thanks to the Terrier Team[3] of the Department of Computer Science of the University of Glasgow.

6. REFERENCES

[1] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of web spam. In *AIRWeb'06: Adversarial Information Retrieval on the Web*, 2006.

[2] A. A. Benczur, K. Csalogany, T. Sarlos, M. Uher, and M. Uher. Spamrank - fully automatic link spam detection. In *In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb*, 2005.

[3] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[4] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: 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*, pages 423–430, New York, NY, USA, 2007. ACM.

[5] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Computing Research Repository*, 2010.

[6] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, January 2005.

[7] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases*, pages 1–6, New York, NY, USA, 2004. ACM.

[8] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. Technical Report 2004-17, Stanford InfoLab, March 2004.

[9] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. Technical Report 2003-29, 2003.

[10] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[11] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, New York, NY, USA, 2002. ACM.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[13] P. Kolari, T. Finin, and A. Joshi. Svms for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, March 2006.

[14] V. Krishnan. Web spam detection with anti-trustrank. In *ACM SIGIR workshop on Adversarial Information Retrieval on the Web*, Seattle, Washington, USA, 2006.

[15] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[16] M. Najork. Web spam detection. In *Encyclopedia of Database Systems*, pages 3520–3523. Springer US, 2009.

[17] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 83–92, New York, NY, USA, 2006. ACM.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[19] B. Wu, V. Goel, and B. D. Davison. Propagating trust and distrust to demote web spam. In *Proceedings of Models of Trust for the Web (MTW), a workshop at the 15th International World Wide Web Conference*, Edinburgh, Scotland, 2006.

[3] `http://terrierteam.blogspot.com`