Identification of Web Spam through Clustering of Website Structures

Filippo Geraci
Institute for Informatics and Telematics, CNR
Via G. Moruzzi, 1
Pisa, Italy
filippo.geraci@iit.cnr.it

ABSTRACT

Spam websites are domains whose owners are not interested in using them as gates for their activities but they are parked to be sold in the secondary market of web domains. To transform the costs of the annual registration fees in an opportunity of revenues, spam websites most often host a large amount of ads in the hope that someone who lands on the site by chance clicks on some ads. Since parking has become a widespread activity, a large number of specialized companies have come out and made parking a straightforward task that simply requires to set the domain's name servers appropriately.

Although parking is a legal activity, spam websites have a deep negative impact on the information quality of the web and can significantly deteriorate the performances of most web mining tools. For example these websites can influence search engines results or introduce an extra burden for crawling systems. In addition, spam websites represent a cost for ad bidders that are obliged to pay for impressions or clicks that have a negligible probability to produce revenues.

In this paper, we experimentally show that spam websites hosted by the same service provider tend to have similar look-and-feel. Exploiting this structural similarity we face the problem of the automatic identification of spam websites. In addition, we use the outcome of the classification for compiling the list of the name servers used by spam websites so that they can be discarded before the first connection just after the first DNS query. A dump of our dataset (including web pages and meta information) and the corresponding manual classification is freely available upon request.

Categories and Subject Descriptors

[**Information systems - World Wide Web**]: Spam detection

General Terms

Clustering, Metric space, classification

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742127.

Keywords

Web spam, adversarial Information Retrieval

1. INTRODUCTION

Spam websites are not necessarily long lists of commercial links pointing to some sponsor web site, but they can be fake web portals where ads are diluted with false free services such as: weather forecasts, news, personalized search. Most often, spam websites look like web directories where one side of the site contains links with anchors in such a way related to the domain topic and the other side contains ads organized as search results. The purpose of this layout is to capture clicks that distracted users make by chance. Among the main goals of spam websites there is to park domains keeping them reserved for selling purposes. As a result, parked domain owners typically have to manage a large portfolio of names. In order to allow a simple management of large groups of domains, specialized companies have come out in recent years. The parking procedure is straightforward; it is suffice to change the name server entries of the domain redirecting them to those of the domains parking provider (DPP) or to insert a redirection from the home page to an appropriate page of the DPP.

This simple procedure, the low price of domains, and the potentially high revenues [**?**] are among the causes for the increase of the number of networks of spam websites. These websites have shown to be able to compromise the quality of results of search engines [**?**]. For example, they are responsible of a considerable waste of resources spent in crawling, indexing and ranking [**?**].

Spam websites hosted by a certain DPP tend to be similar to each other, but they are not identical. In fact, DPPs offer a range of possible customizations of the web template and color scheme. In certain cases, the web template can be inspired to commonly used CMS templates with the purpose of appearing more familiar to the users and, thus, increasing the probability of receiving some clicks on their ads. In some cases spam websites do not look like web directories, but they look like fake web portals or free mail services. The visual similarity between spam domains and web directories or mail portals makes the task of identifying them by inspecting their content challenging.

According to the above procedure, it is sufficient a single DNS query or the download of the sole home page to discriminate whether a website is spam or not. In fact, if the name servers belong to a DPP or the home page contains a redirection to a page attributable to a DPP, the website can be safely classified as spam. To the best of our knowledge,

lists of DPP name servers are not publicly available. It is possible to find on the web only partial and sloppy lists frequently not updated. As a result, an automatic toot able to compile and update a reliable list of DPP name servers has become a need.

In this paper, we propose a novel approach, compared to content based ones, to the identification of spam websites. We do not try to classify a website according to the presence/absence of common structural characteristics of spam websites, but we exploit the fact that websites hosted by a certain DPP tend to be similar to each other and, thus, they can be partitioned obtaining clusters with a homogeneity factor much higher than normal websites hosted by general internet service providers (ISPs).

To partition the websites in clusters, we used an improved version of the FPF [**?**] algorithm for the k-center problem. For each cluster we used the average distance from the center as an approximate measure of the homogeneity of the cluster. We empirically observed that service providers completely devoted to spam have a very low average distance (as low as less than 0.2) for all clusters, while name servers used for both web hosting and spamming, still maintain the low average distance for the clusters of spam websites and have a sensible higher average distance for all the other clusters. As a result, we can use this measure as a criterion to discriminate and classify each cluster.

Another important contribution of this paper is the definition of a distance among pairs of web pages able to capture their structural differences regardless the text contained in the pages. Moreover, since our distance definition requires a quadratic computation in the number of tags of the web pages, we defined an upper bound that can be computed in linear time.

With the purpose of facilitating reproducibility of our experiments and allowing future development of novel approaches to web spam detection, we made all the collected web pages and the corresponding manual classification available upon request.

The remainder of the paper is as follows. Section **??** gives a brief survey of related work, while section **??** describes in detail our approach, the distance function and the clustering algorithm. Section **??** shows our experimental results. In section **??** we draw some conclusions.

2. RELATED WORK

Due to its economic and user-related implications, web spam detection has attracted the attention of scientists in last ten years. Different types of spam require different approaches [**?**]. The most common approach is based on the text content analysis exploiting the fact that spam pages tend to share some common characteristics. In [**?**] the author presents a text content based approach to classify adsportals. The author defined a set of markers (anchor text ratio, anchor text ratio, etc.) that are more likely to be revealing of spam websites. In [**?**] the authors analyze the ".biz" TLD developing a classifier based on the use of regular expressions to identify specific patterns that are more likely to be present in spam websites. In [**?**] the authors train a pool of classifiers to detect a variety of web spam types all belonging to the category of parking.

In [**?**] the authors combine link-based and content-based features. They observed that linked hosts tend to belong to the same class: either both are spam or both are non-spam.

The only work comparable with our approach is [**?**] where the authors test several similarity measures based on the identification of web templates. They also propose a fingerprinting technique and a clustering algorithm for large collection of documents based on these measures. The similarity of the two approaches consists in the assumption that spam pages tend to cluster together.

3. OUR APPROACH

3.1 Metric space

A visual inspection of websites hosted by the same DPP and using the same template, shows that they have an impressively similar look-and-feel even if they have no text in common. For example they can have a similar menu bar in the same position with the same number of bold entries, but, potentially, with completely different labels depending on the commodity sector of the website. It is sometimes possible that two websites have some slight local differences. For example one displays a forecast weather box and the other has a fake mail login form. A distance function, therefore, has to capture all these aspects.

We derived our distance function from a modified version of the global sequence alignment score as defined in [**?**]. In short, the global alignment of two sequences consists of filling them with gaps in the appropriate positions in order to minimize their hamming distance. Instead of using a character-wise comparison that can be subjected to noise due to the insertion of spurious characters on the web pages, we strip out the text and split the web pages into tokens using HTML tags as base elements for the comparison.

Let $p_1 = \{p_{1,1}, \ldots, p_{1,n}\}$ and $p_2 = \{p_{2,1}, \ldots, p_{2,m}\}$ be two web pages such that $p_{i,j}$ is the j-esim tag of page p_i; the global alignment algorithm builds a $n+1 \times m+1$ matrix S (called *similarity matrix*) containing in position (i, j) the alignment score of the i-long prefix of page p_1 and j-long prefix of p_2. Each local judgement consists of maximizing the alignment score by deciding if matching/mismatching $p_{1,i}$ and $p_{2,j}$ or inserting a gap in one of the two pages. A score value, depending on the application goal, is assigned to each decision. In our case, since we want to define a similarity proportional to the number of matched elements between the pages, we assign a score of $s_{match} = 1$ to the match and a score of 0 to the other operations, thus $s_{mismatch} = s_{gap} = 0$.

The matrix is filled from left to right and from top to bottom. Consider the case where we are aligning $p_{1,i}$ and $p_{2,j}$. If we fill p_1 with a gap, we have $S[i,j] = S[i,j-1] + s_{gap}$. Symmetrically, by inserting a gap in p_2 we obtain $S[i,j] = S[i-1,j] + s_{gap}$. The score in case of alignment depends on the content of the aligned tokens. In fact, if $p_{1,i} = p_{2,j}$, then the score $S[i,j] = S[i-1, j-1] + s_{match}$, otherwise $S[i,j] = S[i-1, j-1] + s_{mismatch}$. In order to maximize the overall alignment score the algorithm decides for the maximum of the above values. By using our scoring scheme we have:

$$S[i,j] = \begin{cases} \max(S[i-1,j], S[i,j-1], S[i-1,j-1]) & \text{if } p_{1,i} \neq p_{2,j} \\ S[i-1,j-1]+1 & \text{if } p_{1,i} = p_{2,j} \end{cases}$$

The element $\hat{S} = S[i+1][j+1]$ is the global alignment score for pages p_1 and p_2. As defined here, $\hat{S}$ is a measure of similarity dependent on the size of the smallest web page and it is bounded in the range $[0, min(|p_1|, |p_2|)]$. To normalize $\hat{S}$

to be size-independent and turn it into a distance we define the following:

$$D(p_i, p_j) = 1 - \frac{\hat{S}}{\max(|p_1|, |p_2|)}$$

3.1.1 Efficient Distance Approximations

The distance D has two practical disadvantages: 1) it takes $O(n^2)$ time and space even for comparing pages that are evidently different, 2) it requires that the entire HTML pages are loaded in the main memory. We describe here an approximated distance function that requires a linear time pre-processing to produce compact fingerprints of the pages that can be maintained in RAM and a constant time computation of the distance. For a page p_i we build its fingerprint f_i as a vector of 11 elements so that $f_i[k]$ counts the number of tags of length k in the page p_i and $f_i[11]$ counts the number of tags longer than 10 characters. In this case we consider only the tag keywords ignoring possible parameters.

We define the distance between fingerprints f_i and f_j as:

$$F(p_i, p_j) = 1 - \left(\frac{(\sum_{k=1}^{11} \min(f_i[k], f_j[k])}{\max(|p_1|, |p_2|)} \right)$$

We observe that the factor $\sum_{k=1}^{11} \min(f_i[k], f_j[k])$ is upper bounded by the number of tags of the smallest page and it is an over-estimation of the number of matched tags in the quadratic alignment. The divisive factor of the above formula, instead, is used to normalize the measure with respect to the number of tags of the bigger page (as in the quadratic distance).

A more rough approximation of the above distance functions can be obtained by the following definition:

$$R(p_i, p_j) = 1 - \frac{\min(|p_i|, |p_j|)}{\max(|p_i|, |p_j|)} \quad (1)$$

In this case the numerator of the fraction is the size of the shortest page that is the upper bound of the number of possible matches. Although the formula in (**??**) could be considered a too rough approximation of the structural distance between two pages, it can be used as a filter for our purposes where the target is not the clustering in itself. In fact, since it holds that $R \leq F \leq D$, if $R(p_i, c_j)$ is high enough to decide that page p_i is too far from the center μ_j, then the computation of $D()$ becomes unnecessary. According to the above consideration, we performed our clustering by using the three distance definitions $R()$, $F()$ and $D()$ in cascade stopping when one of these distances returns a value higher than a certain threshold δ. Although different values of the threshold can influence the selection of the centers, our experiments have shown that δ has no influence on the final prediction.

3.2 Clustering algorithm

Clustering is the activity of dividing a set of objects into homogeneous groups according to their distance. Results strongly depend on: the definition of a distance function able to capture the differences among objects as well as the objective function that the clustering algorithm attempts to minimize/maximize.

In our application the goal of clustering is to highlight highly homogeneous clusters. As observed in [**?**] an hint of the homogeneity of a cluster can be obatained from its cluster radius. As a result, we used an algorithm optimizing the k-center problem (i.e. the problem of selecting from a set S a subset μ of k elements $\{\mu_1, \ldots, \mu_k\} \subset S$ that induces a non-overlapping partitioning such that the radius of the widest cluster is minimized) for clustering.

In [**?**] the author shows that the k-center problem is NP-hard and gives a 2-competitive algorithm (*Furthest point first (FPF))* that requires $O(nk)$ distance computations. This algorithm is proven to be optimal unless $P = NP$. As in [**?**] we used a heuristic version of the furthest point first algorithm that, exploiting the triangular inequality in metric spaces, is able to speed up the computation.

3.2.1 The M-FPF algorithm

The FPF algorithm builds its clustering incrementally: at each step it selects as a new center the element maximizing the distance to its assigned center. Once a new center is chosen, the clustering is updated. This latter step dominates the computational cost requiring $O(n)$ distance invocations.

The heuristic version of the FPF algorithm (later referred as M-FPF) attempts to reduce the number of distance computations for both the selection of a new center and the clustering update.

Let $C_i = C(\mu_i)$ be the set of elements of S closest to a center μ_i. By keeping C_i as a list, sorted according to the distance to the center μ_i, we can easily compute each cluster radius in time $O(1)$ and select the new center by simply ranking the clusters' radiuses.

Once a new center μ_y is selected, the clustering can be updated by scanning each list C_i in decreasing order of distance from μ_i. By the triangular inequality we can stop scanning the list C_i when we reach an element x satisfying the condition:

$$D(x, \mu_i) \leq \frac{1}{2} D(\mu_i, \mu_y) \quad (2)$$

because the remaining elements of C_i cannot be closer to μ_y than μ_i. Notice that the distances among all the pairs of centers are required, thus introducing an extra $O(k^2)$ computational cost. Figure **??** shows an example where there are two centers (in red) and an element (in blue) satisfying the condition in (**??**).

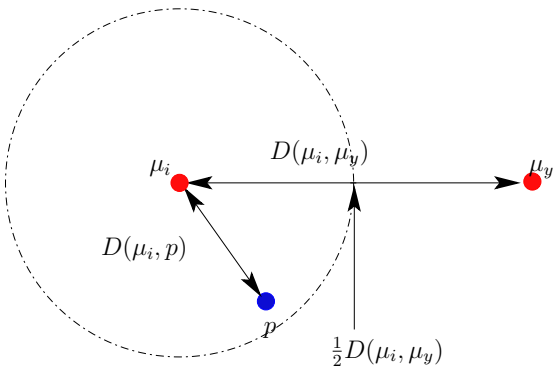

Figure 1: Example of element satisfying the condition in (**??**)

We experimentally observed (see sec. **??**) that the condition in (**??**) is more likely to be quickly satisfied refining a cluster of spam websites where most of the elements are very similar (when not identical) to the cluster center. As

a result, our heuristic tends to be more effective when the quadratic distance function is mostly used.

4. EXPERIMENTAL EVALUATION

In this section we show the results of an empirical evaluation of our websites classification system. We performed a crawling with our proprietary crawler running on a cluster of 6 *Mac mini* endowed with 16Gb of RAM and a 2.4 Ghz *Intel*$^{\text{TM}}$processor. Clusterings were made on a *Mac pro* endowed with 64Gb of RAM [1] and *Intel*$^{\text{TM}}$XEON 12-cores.

In order to assess our algorithm performances against the state of the art, we compared our method with the *LSH-fingerprint* clustering algorithm described in [**?**] which, at the best of our knowledge, is the most similar approach to our. In our experiments we do not make any claim on the running time of the competitor algorithm since in the original work there are no references to the execution time. Since the algorithm in [**?**] is influenced from the choice of some parameters, we decided to use the same values suggested by the authors in their work.

4.1 Dataset description

Evaluating performances of spam classification algorithms poses many thorny problems. The first important issue is the absence of a recent representative labelled snapshot of home pages including both DNS and HTML information. Recrawling existing datasets (i.e. [**?**]) can result in an inaccurate classification due to possible changes of the content or owner of some web pages after the manual classification. In our case, this caused the need of an expensive manual evaluation of the dataset.

Another critical aspect is the exact definition of spam website. In fact, according to the most restrictive definition, these websites contain ads. However, in some cases we found that certain websites are clearly lacking of useful content except for the contact information of website's owner even if these websites do not contain any ads. In some other cases a web site consists only of a courtesy page of the service provider (most often containing ads). In these cases, discriminating whether a domain is spam or it has been registered for the impending publication of a new website is impractical. In our manual labeling we used a conservative approach. When it was possible to establish with certainly that a domain was reserved for potential future use (even without ads) we classified it as spam, otherwise we classified it as not spam even if it was lacking of content.

To build our collection we performed an extensive web crawling obtaining a collection of $1,076,079$ valid home pages belonging to 93 distinct internet service providers. These home pages and service providers have been evaluated using a human-supervised semi-automatic procedure.

We iteratively requested a pool of volontiers to evaluate a certain number of random unlabelled pages in the dataset. In order to avoid influencing the human evaluator, she had no information about the service provider. Similarly to [**?**], these labelled pages have been used as seeds to induce a clustering where two pages belong to the same cluster if they have approximately the same tag structure. Notice that pages not similar enough to some seed are not assigned to any cluster. Singletons (namely clusters with only the seed element) and clusters of duplicates have been labelled according to the human evaluation. For the remaining clusters (i.e. clusters with radious higher than 0) we asked the human expert to evaluate at least another random page. If all the evaluations of a cluster agree to each other the pages belonging to the cluster have been labelled. The procedure has been repeated until all the pages had been labelled. At the end of the evaluation procedure our dataset consisted in $917,581$ non-spam pages and $158,498$ spam pages. Service providers have been divided into categories (namely: SPAM, ISP) according to the label of the majority of hosted domains. As a result of the classification, our dataset consists of 77 providers labelled as ISP, and 16 classified as SPAM.

[1] a single clustering process could allocate only 32Gb of RAM due to manufacturer limitations.

4.2 Parameters tuning

Parameter and threshold setting can become a complicated task because they can profoundly affect the outcome of an algorithm. As for most clustering based tasks we have to choose the number k of clusters in which to partition the domains belonging to a service provider. Moreover we need to set a threshold δ to classify each cluster as containing spam or regular domains. The prediction of the number of clusters is a well studied problem in the literature [**?**] even if a generally accepted approach is still on the horizon.

In this section we experimentally show that the choice of the number of clusters is not critical for us because our algorithm proven to be robust to the change of this parameter. Furthermore, we show that the average radiuses of spam website clusters and regular website clusters belong to well-separated ranges. As a result, setting the value of δ arises as a natural choice.

We select 7 among the biggest service provider in our dataset. Three of them are explicitly used for spamming (two of them have the substring *parking* as part of their name and one is a well-known domain parking provider). The other four NSs belong to general ISPs. In order to better assess a good estimation of δ we included the NS of one of the most used content management system (CMS) provider (namely *wordpress*) in this latter set. This choice is motivated by the fact that we expect a higher degree of structural similarity among web pages created using the same CMS and consequently a lower average radius.

Let $R(C_i)$ be the radius of the cluster C_i and let $|C_i|$ be the number of elements of the cluster. Fixing the number of desired clusters to k, we compute the average radius as the weighted sum of the cluster radiuses:

$$\hat{R_k} = \sum_{i=1}^{k} |C_i| R(C_i)$$

Since our distance definition is normalized to belong to the range $[0, 1]$, $\hat{R_k}$ belongs to the same interval.

Figure **??** shows $\hat{R_k}$ for various assignment of k. We observe that spam service providers have an average radius constantly much lower than general ISPs for each possible assignment of k. The figure also shows a smooth trend of the average radius that, for $k > 16$ decreases quite slowly for both spam and non-spam websites. This confirms that the choice of k is not critical and suggests that $k = 16$ could be a good choice in general.

Figure **??** also gives an important hint about the choice of the value δ used as threshold to discriminate spam clusters versus general ISPs. All the spam service providers show

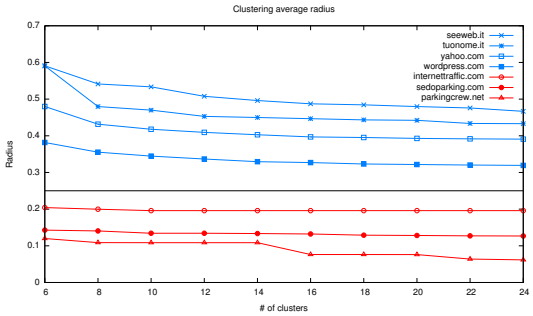

Figure 2: Average cluster radius. In red results for NS devoted to SPAM, in blue results for general purpose ISP.

an average radius lower than 0.2 while general ISPs have an average radius higher than 0.3. According to this behaviour we set $\delta = 2.5$.

Finally, figure **??** confirmed that our similarity function is also able to classify web sites generated by means of CMSs correctly. In fact, despite that the average distance for the NS *Wordpress* is lower than the other general ISPs, it is consistently higher than 0.3 for all the assignments of k.

4.3 Running time evaluation

Although clustering time is not a critical performance parameter, it still influences the overall classification time and thus, the possibility to keep the list of spam websites updated. In fact, the overall classification time is the sum of the clustering times of all the service providers.

A formal assessment of the ability of our sequence of upper bounds to reduce the number of quadratic distance computations, and thus the clustering time, is impractical because it completely depends on the input data. Thus, we carried out an experimental performance assessment. For each service provider we measured the distribution of the calls to the three distance functions $R()$, $F()$ and $D()$. Although calling $D()$ is subordinate to calling $F()$ which in turn is subordinate to invoking $R()$, for the sake of comparison we counted only the number of invocations as a last distance function of the sequence. In table **??** we report these distribution as percentages dividing them according to the name server classification. We also include the overall number of distance calls and the running time.

Class	Metric			# dist. calls.	Running time
	R	F	D		
ISP	71.41	14.84	13.75	33.93M	17:35:55.49
SPAM	15.49	5.26	79.25	8.69M	40:25.43
Total	60.02	12.88	27.10	42.62M	18:16:20.92

Table 1: Distribution of the number calls to the three distance measures. Last two column reports: the overall number of distance calls (in million) and the overall running time in hh:mm:ss.cent.

As table **??** shows, for the most time-demanding class (ISP) most of the times (more than 70%) the approximation returned by calling the function $R()$ is a fair accurate approximation of the distance between two web pages. Furthermore, although the overall contribution of the function $F()$ to the reduction of the number of calls to $D()$ is limited if compared to $R()$, it is still good enough for more than $1/2$ of the distances not fairly approximated by $R()$. As a result of the combination of the two approximated distances, the expensive quadratic distance $D()$ is called only the 13.75% of the times.

As expected, the class SPAM shows a different distribution. In fact, the number of calls to distance $D()$ dominates the overall number of distance invocation. Nevertheless, in this case the overall running time is not heavly affected from these calls because of the beneficial effect of the optimization of equation **??** in the M-FPF algorithm.

4.4 Classification Evaluation

In this section we report the classification performance in terms of f-measure and accuracy of our method and compare it with the *LSH-fingerprint* algorithm described in [**?**].

We made two levels of evaluation:

- **Service provider level**: we labelled each service provider according to the class predicted by the majority of the corresponding websites. This evaluation is aimed at measuring the ability of each algorithm to predict the list of service providers devoted to spam.
- **Website level**: we compared the prediction for each website with the corresponding human judgement. This analysis allows the measuring of the probability of a website to be misclassified.

In regards to the first evaluation, as shown in table **??**, our method has demonstrated to be able to approximate the human judgement with a very high confidence level obtaining, in terms of both f-measure and accuracy, an increase of about 5% over the state of the art. This result is particularly important for all those applications exploiting the web as a source of data to be used to build a knowledge base [**?**]. In fact, in this case what really matters is not the complete list of regular websites, but the avoidance of the introduction of low quality documents into the knowledge base.

	f-measure	Accuracy
Our	0.9178	0.9062
LSH	0.8670	0.8541

Table 2: F-measure and accuracy of our method and LSH clustering for the service provider level assessment.

In other cases, for example for spam filters, classifying all the domains belonging to a certain service provider with the same label could be too rough and could potentially introduce a non-negligible amount of false positives and false negatives. While for filtering applications the misclassification of a spam website is typically non-problematic because its effect is only the potential visualization of some unwelcome website, false positives may represent a problem because they can cause the inaccessibility of legal contents. In contrast, for parental controll appllications the presence of false positive can represent a problem.

According to table **??** the rate of false negative of both methods is in the order of 3%. An in-depth examination of these missclassified websites show that they are often clusters of nearly empty pages, HTTP errors and other non-informative pages that, however, can not be considered as spam. Instead, the two methods show a large difference in

		Our		LSH	
		ISP	Spam	ISP	Spam
Human eval.	ISP	82.21	3.06	61.56	3.35
	Spam	0.65	14.08	23.70	11.39

Table 3: Confusion matrix of the website level assessment of both methods expressed as percentages.

the rate of false positive predictions. In fact, our algorithm is able to bound the misclassified websites under the threshold of 1% while the LSH exceed the 20%. We believe that this low rate of false positive predictions makes our algorithm a relialable tool for critical applications such as parental control filters.

For the reader convenience we report in table **??** the results of the website level assessment also in terms of f-measure and accuracy. As seen before, the large difference between the two algorithms is due to the impact of false positive.

	f-measure	Accuracy
Our	0.9640	0.9628
LSH	0.6925	0.7294

Table 4: F-measure and accuracy of our method and LSH clustering for the website level assessment.

5. CONCLUSIONS

In this paper we presented a novel clustering-based approach to the identification of spam websites. The novelty of our approach stands in the fact that we observed that these websites are not similar to each other in general, but this similarity is highly evident among websites hosted by the same DPP. Moreover, we do not make a-priori assumptions about the content of spam websites, this makes our approach robust to changes due to technological evolution. We also defined a metric able to capture the structural similarity among pairs of web pages. We believe that our metric can find other applications in Web information retrieval. We show how the quadratic computation of our distance can be easily overestimated in linear time without affecting the overall clustering quality. As a result our metric can be employed in practical applications where hundreds of thousands of websites have to be clustered. Finally, we built a manually curated dataset of spam websites that can be used for further research.

6. REFERENCES

[1] M. Almishari and X. Yang. Ads-portal domains: Identification and measurements. *ACM Trans. Web*, 4(2):4:1–4:34, Apr. 2010.

[2] 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, Dec. 2006.

[3] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 423–430, 2007.

[4] M. Crane and A. Trotman. Effects of spam removal on search engine efficiency and effectiveness. In *Proceedings of the Seventeenth Australasian Document Computing Symposium*, ADCS '12, pages 1–8, New York, NY, USA, 2012. ACM.

[5] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial intelligence*, 118(1):69–113, 2000.

[6] F. Geraci, M. Pellegrini, P. Pisati, and F. Sebastiani. A scalable algorithm for high-quality clustering of web snippets. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, SAC '06, pages 1058–1062, New York, NY, USA, 2006. ACM.

[7] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[8] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, April 2005.

[9] T. Halvorson, J. Szurdi, G. Maier, M. Felegyhazi, C. Kreibich, N. Weaver, K. Levchenko, and V. Paxson. The biz top-level domain: Ten years later. In N. Taft and F. Ricciato, editors, *Passive and Active Measurement*, volume 7192 of *Lecture Notes in Computer Science*, pages 221–230. Springer Berlin Heidelberg, 2012.

[10] P. Hayati, N. Firoozeh, V. Potdar, and K. Chai. How much money do spammers make from your website? In *Proceedings of the CUBE International Information Technology Conference*, CUBE '12, pages 732–739, New York, NY, USA, 2012. ACM.

[11] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. Wang. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. *2012 IEEE Symposium on Security and Privacy*, 0:112–126, 2013.

[12] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[13] S. Pevtsov and S. Volkov. Russian web spam evolution: Yandex experience. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, pages 1137–1140, 2013.

[14] V. M. Prieto, M. Álvarez, R. López-García, and F. Cacheda. Architecture for a garbage-less and fresh content search engine. In *KDIR*, pages 378–381, 2012.

[15] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: (Statistical Methodology)*, 63(2):411–423, 2001.

[16] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Tracking web spam with html style similarities. *ACM Trans. Web*, 2(1):3:1–3:28, Mar. 2008.

[17] J. Wang and J. Chen. Clustering to maximize the ratio of split to diameter. In *29th International Conference on Machine Learning ICML*, 2012.

[18] S. Webb, J. Caverlee, and C. Pu. Characterizing web spam using content and http session analysis. In *CEAS*, 2007.