# Measuring and Visualizing the Scrappiness Level of a Website

Darius BUFNEA
*Department of Computer Science*
*Babeş-Bolyai University*
*Cluj-Napoca, Romania*
*Email: bufny@cs.ubbcluj.ro*

Diana ŞOTROPA
*Department of Computer Science*
*Babeş-Bolyai University*
*Cluj-Napoca, Romania*
*Email: diana.halita@ubbcluj.ro*

***Abstract*—Scraper sites are questionable quality sites that copy their content partially or entirely from other websites and sometimes gain more ranking and popularity to the detriment of the original websites. This usually happens from a search engine point of view. Misleading a user to a scraper site almost always implies an unhappy, time consuming user experience, the scraper site being an unnecessary link in the user's navigation path. In this paper we present a method through which one can numerically measure and quantify the scrappiness level of a website and also visually display this level. In the same time, this paper wants to advert to the web and research communities about this type of websites and to urge actions against them.**

***Keywords*-scraper site; scrappiness level; link spam; web spam detection; content spam; document similarity; search engine; web search.**

## I. Introduction

With the development of the Internet, the number of websites has considerably increased. This can easily explain the plenty of information that exists on the web and the fact that search engines are amid the most accessed type of websites if one wants to locate and access specific information. Unfortunately, lately, both the user and the search engine have to face an increased number of scrapers.

Scraper sites are questionable quality sites that copy their content partially or entirely from other websites and sometimes gain more ranking and popularity to the detriment of the original websites. Content copying from other sources is one of the cheapest method used by a scraper to get its content. In fact, the content is one of the most precious resources, creating original and valuable content implying the most effort by the publishers or content providers. Scrapers' appearance is closely tied to more than one factor:

- financial factor and pursuing easy and rapid earnings through displaying ads, i.e. web monetization [1];
- publishers / content providers migration towards web / online platforms to the detriment of classical platforms (newspapers, radio, television);
- a much higher availability of the web content, mostly plain text, and a higher ability to get, store, modify and display it by the means of relatively low investments;
- the appearance of online advertising platforms, i.e. intermediate agents such as Google which are the link between publishers / content providers and advertisers [2]. Such platforms commission their share providing in exchange through smart ads placement a much higher targeting and conversion.

Scrapers can be classified as follows [3], [4]:

- sites that copy the entire content from another site, without adding any original information that should highlight the new page, and which publish the information in the same form as it is harvested;
- sites that copy the content from other websites and post them in a new form, automatically altering some words of the content (for example by using synonyms);
- sites which reproduce information taken from RSS feeds (Rich Site Summary) without adding relevant information that are important to the site visitors. In this category we can also include sites that aggregate information from other sites (e.g. sites that aggregate torrents files or products).
- sites that take multimedia files (images, movies or any other type average) and provide them to the users without additional information.

Through scraper one should think of a website that presents an amalgam of content taken from other sources, most often without permission. Such websites are usually full of ads, and their goal is to be interposed between the user and the website which really provides the information that one may be looking for. Often, scraper sites outrank in SERP the original websites (sites where the content is taken from). Well positioned scrapers in SERP [5] decrease both the search engine's performance and the user's navigation experience satisfaction, as they operate as an additional step in the user's navigation path: user - search engine - scraper - destination site. Considering this situation, search engines generally disapprove this kind of sites, but often a search engine tolerates or validates a scraper as a genuine publisher.

In this paper we present a method through which one can numerically measure and quantify the scrappiness level of a website and also visually display this level. In the same time, our goal is to advert to the web and research communities about this type of websites and to urge actions against them.

## II. State of Art and previous work

In the effort of identifying a scraper's spam pages [3], [6], [7] and omit their appearances within the SERP, different methods were advanced by the research community and search industry [8]. While paper [9] presents a link based semi-supervised learning algorithms for web spam detection and paper [10] presents a label propagation algorithm on click-through bipartite graphs to detect web spam, paper [11] provides a comparison of some algorithms to detect spam link. However, the fact that such websites are still present in a search engine results page [4], lead us to believe that, so far, these research didn't achieve the desired results, leaving the door open for further advances in the field. From this point of view, more is to be done in order to offer better and more reliable search results to users.

One common feature to almost all scraper sites is the extreme search engine optimization performed within such a website. This is also the main reason why a scraper is top ranked within the SERP. However, one cannot rely only on this characteristic to classify a website as a scraper or not: such a feature is not exclusively common to scrapers, using search engine optimization techniques is also a common practice among webmasters of genuine websites for gaining more traffic. The fact is that there is a large gray area between ”ethical” search engine optimization, that is making sure that a page can be found by search engines, and ”unethical” spamdexing [12] that is used by scraper for deceiving search engines.

Some methods of identifying scraper sites were previously suggested, but because of millions of new web pages are being published every day, no method is proving good enough. Most scraper sites are self-evident [13]. There are some markers included in almost every scraper site: ads everywhere, stolen content, pop-ups or suspected malicious executable files offered for download. While these characteristics of a scraper can be easily spotted and recognized by a human user, there have to be found a general enough numerical methods through which a website can be classified as a scrapper by a computer program (for eg. by the search engine's crawler or by a browser plugin provided by an Internet security suite installed on the user's computer).

Some search engines announced in 2014 a policy to identify scraper sites based on users experience [14] and through their feedback. Moreover, Google wants to involve webmasters community in order to qualify which sites are scrapers and which are the original sources of certain information [15]. Reported sites, once classified, are not automatically downgraded in SERP, the results obtained from users classification are used in testing automatic algorithms which will be able to decide if a site is scraper or not. Such an algorithm implemented by the search engine will generally look for known patterns in a site's content and structure in order to automatically classify it [1], [3].

The disapproval of scrapers by search engines is not always firm. Some search engines also act as intermediate agents in the web advertising industry, their duplicity being the result of the profitability of this type of business. Some search engines still tolerate scrapers' existence, allowing ads inside a scraper's content (ads delivered through the search engine's API). It's a win-win situation for everybody: scrapers, search engines (by brokering ads delivery) and advertisers to the detriment of the user's free time spent on consuming thin or low quality content.

Authors of [16] present a method through which one may identify improper placed outgoing links such as ads or spam. This method is based on studying the content similarity between linked and original content. Scrapers identification process may use a similarity based approach too.

The techniques of fighting and identifying web scrapers can be implemented by different actors on the WWW scene. Hence, we distinguish three main types of techniques:

- fighting and identifying scrapers at search engine's level - this would be the more natural method, since scrapers in general rely on traffic coming from search engines. All the above mentioned actions fall in this category.
- identifying web scrapers at genuine site's level - these are the most inefficient methods, consisting in different techniques of limiting the scraper's crawling process using captcha or even actually forbidding its access. These methods can be easily avoided by User-agent spoofing and limiting the crawling rate or using a large pool of IP addresses in the crawling process.
- identifying web scrapers by a third party (for example by an Internet security suite provider or by an experimental research project) exposed web API called by a browser plugin that can highlight to the user, or even completely hide, low quality websites before the user's actual visit to the scraper. For example, such a plugin can provide an additional visual low mark for a scraper in SERP. This last technique is the most suitable to be implemented by the research community, including for testing the method proposed in this paper.

Regardless of the method, the process of identifying scraper sites seems to be a consistent job, the amount of time and energy implied being in general very high [3]. Looking from a genuine website's perspective, if one scraps its content, the website's owner has to deal in some ways with the content scraper. Some techniques are based on constructing some in-situ solutions, such as captcha, rate limiting or IP blacklists, but all this techniques are not strong enough and might be circumvented. Another solution may be the use of some add-on modules which try to combat such ”attacks”. In general, due to the regularity of the provided content, a scraper will crawl a website built over a content management system rather than other sites. Such websites dominate the web and have a common structure that can be easily parsed, i.e. once the scraper wrote a parser for a CMS based website, it can use that parser to crawl almost every other website that was build based on the same CMS. Depending on the CMS used for building a website, there were developed some additional, reusable, CMS integrated techniques in order to deal with content scrapers [17], [18]. Although, a website built over a content management system will be in a spotlight for content scrapers, it has the advantage that, server side, its owner can use additional modules developed in order to avoid content scrapers.

An effective and relatively easy to implement technique would also be building a browser plugin through which one can rank a site returned in SERP. The purpose of using such a plugin is to gather visitors reviews (explicit feedback provided by visitors) or implicit feedback (by analyzing users' behavior). These reviews are valuable assets leading to a fair scrappiness / quality level of a website returned in SERP. The main advantages of such a plugin is that its functionality does not depend of any of the big actors involved on the web scene (i.e. search engines or Internet security providers) and it can be easily implemented and tested by small entities such as research groups. Through its disadvantages are the dependability of test subjects (users which opt to install the plugin), the dependability of JavaScript (which can easily be turned off in the client's browser), or the fact that such a plugin might partially mimic a spyware like behavior rising user privacy concerns.

## III. Research Methodology, Experiments and Results

Every site usually consists in multiple pages, every page being integrated in the site's general look and feel. Through this, one can understand that every page use the same structure and design in order to properly present the information. Therefore, every page presents general information which can be found on every other page, such as menus, header, footer or sidebars, i.e. the so-called master page template and specific information related to the title of the page, i.e. the so-called absolute content. To accomplish the intended purpose we rely on the analysis of the absolute content extracted from every page of the site in order to properly classify it. The classification results obtained by absolute content analysis can be validated using two heuristics.

The first heuristic is based on website's traffic report captured in Squid logs. Due to the entire amount of data extracted from the captured logs we were able to identify different user navigation patterns. The most interesting navigation pattern shows that scrapers are only intermediate agents between users and common websites (i.e. genuine sites that are properly presenting information). One who might follow this pattern, usually make an initial request to a search engine using an initial query which is related to a specific information that the user wants to find. In the generated Search Engine Results Page (SERP) one may find a site which seems to present the desired information (the scraper site). After only a few seconds of visiting the scraper site, the user find within the scraper's thin content the link to the page where the content was taken from, and follows it in order to find the complete, original and accurate desired information. We will rely on this heuristic later in the paper in order to validate our results.

The second heuristic is based on the advertising strategy adopted by a scraper site. Relying on ads for website and business sustainability it's a common practice for genuine websites too, but a scraper's volume of placed ads usually is higher when compared with the quantity and quality of its content. Therefore, it is an interesting approach to determine for a scraper what kind of ads it delivers, their quantity and their spread within the thin content of the website.

Next, we will present a method of numerically measure and visualise the scrappiness level of a website. This method is based on the similarity between the absolute content of pages that are part of the suspected scraper site and their corresponding source pages - i.e. pages where the content was taken from. We are expected to find within a scraper site backlinks to the original content. Because of the copyright rules and laws [19], at least in UE, scraper sites usually link back on each page to the corresponding source page where the content was initially published. Thus, the process of numerically measure and visualise the scrappiness level of a website will consist in the following:

- crawling the tested website;
- comparing the content of every page with the content that can be reached by following the external links within that page (hyperlinks that point at any domain other than the domain the link exists on); in order to determine the absolute content of a web page we have used a Java library named boilerpipe, which is able to remove or extract full text from HTML pages;
- computing the similarity between those two contents; we have used Cosine Similarity [20] measure;
- observing that for a scraper site the obtained similarity is higher when the external links are placed in the absolute content of a webpage, rather than when they are placed in the site wide template.

To reach the goal of this article, we analyzed several test sites that are frequently updated. Using a web crawler we indexed all pages of these sites, and then, using the collected information we identified the absolute content of every page along with the source where its content was taken from. For each test site, after the crawling step, we observed that a large data set was created. In order to avoid a large data set and to speed up the entire process, we will test and present in a future paper how feasible is to use only a subset of pages, i.e. we will compare only a subset of absolute contents with their sources. By applying the steps of the algorithm

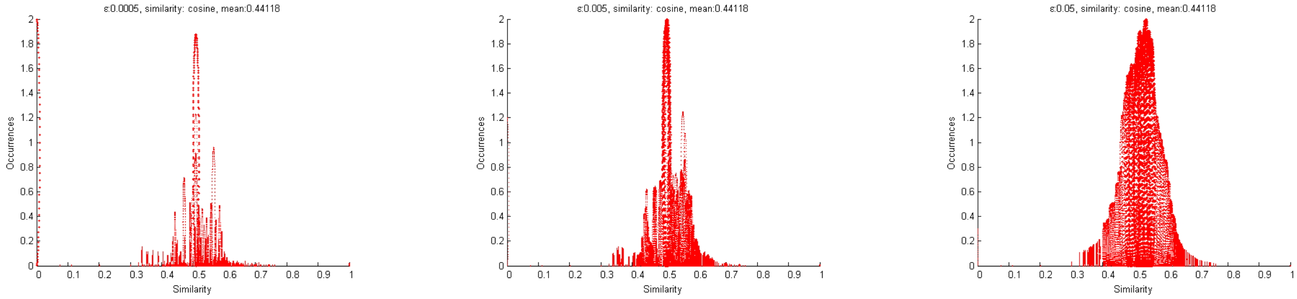

Figure 1: Common website C1

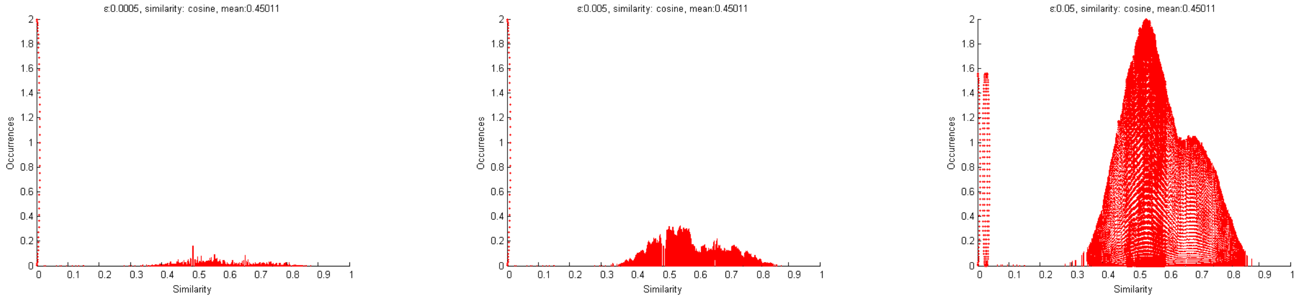

Figure 2: Common website C2

presented below, there were generated triplets (internal link, external source link, similarity). These triples are the primary data used in the scraper sites identification process.

In this paper we describe the scraper identification and visualisation processes using a generic similarity measure. Our tests were performed using a simple similarity measure, the Cosine similarity. Generally, for scraper site classification, different types of similarity functions might be used. Future evaluation of different text similarity functions such as character based similarity functions or term based similarity functions [21] should be performed in order to check how they perform and fit in this scenario. Also, one can choose a similarity measure from geometric similarity functions or from semantic similarity functions category. Regarding Cosine, even if it is a character based similarity, it presents a series of advantages. It is well appreciated as a string similarity measure especially because it gives a good complexity over a large data set and because if offers quick answers regarding matching the pairs.

The idea of discovering a scraper site using numerical methods can be split into two parts:

(i) Firstly, we are looking to the statistically reports obtained by comparing absolute contents from a website with their corresponding sources. The statistical tool used in order to generate these reports is the arithmetical mean, i.e. the sum of all similarity measurements divided by the number of pairs of links in the data set.

$$mean = \frac{\sum_{k=1}^{n} sim_k}{n}, \tag{1}$$

where n is the number of pair of links considered for the analysis. We will call the above mean the scrappiness score of a website and we will denote it by $SS_{site}$ through the rest of this paper.

Generally, while analyzing websites, we expect three different types of behaviour:

- data is spread out more to the right, which corresponds to a scraper site. This will be supported by a relatively high value of the scrappiness score;
- data tends to be around the mean value and it gets close to a normal distribution, which corresponds to common genuine websites;
- data is spread out more to the left, which corresponds to a website whose content is poorly similar with the external linked content. This behaviour will be supported by a relatively low value of the scrappiness score. A site full of ads unrelated to its content or a parked domain might fall in this category.

Any of the extreme cases - either a high value of the $SS_{site}$, either a low one - would indicate a problem with that site's content. A high value of the scrappiness score indicates a scraper site behaviour and the lack of original content, while a low value indicates improper placed outgoing links whose content it's not related with the linking content within the website (i.e. possible ads or other forms of links selling).

(ii) Secondly, using the obtained numerical results, one may want to visually recognize scraper or regular sites. To better observe a scraper site's behaviour we will graphically represent all the points obtained in a cartesian system,

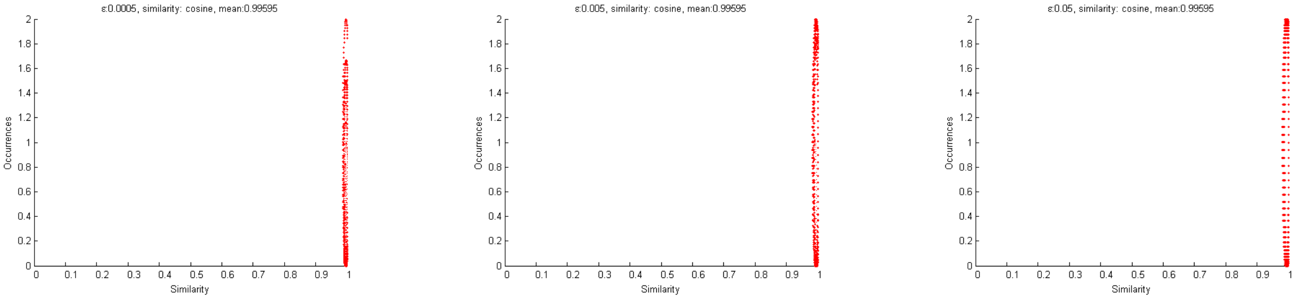

Figure 3: Scraper website S1

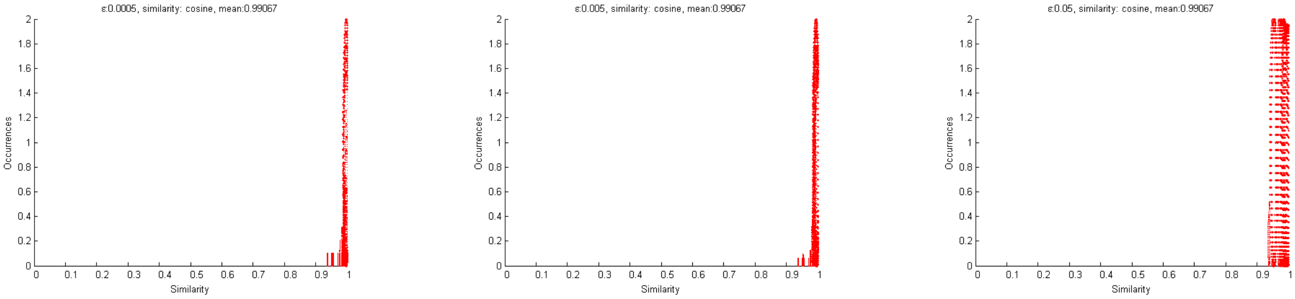

Figure 4: Scraper website S2

following the model:

- the x-axis represents the similarity between the two links; we use the similarity measure to compare the two pages' absolute contents;
- the y-axis will represent a number directly proportional with the number of occurrences, which means the number of pairs (internal link, external source link) which have a similarity that is in the confidence interval of the abscissa of the considered point. Through a confidence interval of a number s we understand an interval $[s - \epsilon,\ s + \epsilon]$, where $\epsilon$ is a threshold which we have experimentally chosen. The experiments we performed involved different values for $\epsilon \in \{0.05, 0.005, 0.0005\}$, different values might provide a better visualization of data. The number of occurrences grows inverse proportionally with $\epsilon$, but the mean value (the scrappiness score of a website) is not affected by the chosen $\epsilon$. That said, we want to observe from this analysis how many pages from the test site have almost the same similarity with their cited sources. This number which will be represented on the y-axis is the double of the normalized value with respect to the maximum number of occurrences for the analyzed site - we chose this approach only from data visualization considerations.

For every point on the graphical representation, we represent a circle having its center in the considered point and the radius equally with half of the point's ordinate. The entire purpose of this strategy is to highlight the number of pairs for which we obtained very high similarity between the content found at the internal link and the one found at the external link. For each pair $P(x_0, y_0)$ we represent in the $XOY$ system the circle of equation:

$$C : (x - x_0)^2 + (y - y_0)^2 = (\frac{y_0}{2})^2 \tag{2}$$

Listing Algorithm 1 contains the pseudocode for measuring and visualizing the scrappiness level of a website. In this algorithm we intentionally choose not to use thresholds to judge a site either a scraper or a common one because these thresholds would be similarity dependent. Additionally, we plan further research on a large websites data set in order to properly fine tune such thresholds.

Figure 1, 2, 3 and 4 depict our test results. We applied our method on two obvious scraper sites denoted by S1 and S2 (scrapers built based on the RSS feeds of some genuine online newspapers) and, also, on two common websites, denoted by C1 and C2, one of them being the website of our university.

From our test results, we concluded that for a scraper site data is spread out more to the right (the obtained graphical representation is right shifted). This is supported mathematically by the numerical interpretation of the scrappiness level of a website, i.e its scrappiness score, defined by using the arithmetic mean in Equation 1. In the case of a scraper site, the similarities of the pairs (internal link, external source link) are high, but without using a confidence interval to represent their occurrences, most of the ordinates will overlap. In the confidence interval of a similarity there will be located all the pairs having almost identical similarities

**Algorithm 1:** Algorithm for measuring and visualizing the scrappiness level of a website

```
for each analyzed site do
    Discover and read all web pages on the website;
    for each $internalLink$ do
        for each $externalLink$ do
            • Compute the $similarity$ between the content on the $internalLink$ and the content found on the $externalLink$;
            We denote it by: $sim[internalLink][externalLink]$;
            • Compute the number of occurrences in the confidence interval;
            We denote it by: $occurrences[internalLink][externalLink]$;
            • Set the maximum number of occurrences;
            We denote it by: $maxOccurences[internalLink]$;
    for each internalLink do
        for each externalLink do
            • The x coordinate equals the previous found similarity;
            Be that: $x_0 = sim[internalLink][externalLink]$;
            • The y coordinate equals the double of the previous found number of occurrences divided by the maximum y-value obtained;Be that:
            $y_0 = 2 * \frac{occurrences[internalLink][externalLink]}{maxOccurences[internalLink]}$;
            • The radius equals half of the y coordinate;
            Be that: $r_0 = \frac{y_0}{2}$;
            • Plot the circle: $C : (x - x_0)^2 + (y - y_0)^2 = r_0^2$;
    Compute the scrappiness score of the website: $SS_{site}$.
    Observe the pattern of the graphical representation using the following conditions:
    if $SS_{site}$ is close to 1 then
        the website is a scraper site;
    else if $SS_{site}$ is close to 0 then
        the website's content is poorly similar with the external linked content;
    else
        the website is a common website;
```

(based on the chosen $\epsilon$). The graphical representation of these pairs should reflect the amplitude of the phenomenon (i.e. their number). Mainly, this is why we chose to use circles with different radius in the graphical representation. If we had given up to this type of graphical representation, then many of the points would overlap, especially due to the very small differences between their corresponding similarity. In addition to that, the shifting effect to the right would have been considerably reduced.

Albeit not in the main scope of this paper, we tested a poor quality website (a parked domain) full of ads not related with its thin content, expecting a left shift behaviour. Unfortunately, the Cosine similarity did not perform very well in this scenario, other similarities offering better results. Figure 5 shows how Cosine and Bigrams similarities performed in this situation. Bigrams similarity led to a more left shifted result sustained by a lower mean. This similarity is more sensitive to the context but it has his own disadvantages: it fails when strings are very much alike, but the bigrams sets are disjoint or when strings are not alike at all and still their bigrams sets are the same [22].

## IV. Results Validation

After we analyzed the absolute content for the chosen test sites, we managed to confirm each site either as a common (genuine) or as a scraper one. In order to prove our method, we further studied proxy logs containing anonymous traffic data, looking for predictable web access patterns in users' behaviour. The proxy logs were obtained from the proxy server of our university which is running Squid, the most popular proxy server in the Internet [23]. Squid is able to generate logs containing details about every request made by a user who is connected to the Internet through it. Some interesting data from our perspective recorded in these logs are: timestamp, referrer, URL or client IP. These data can also be collected from other sources, such as a browser plugin provided by an Internet security suite installed on the user's computer or analytics data gathered by the big actors on the WWW scene (such as Google, Facebook, etc). Each of these data collectors could easily implement similar techniques, having the benefit of a bigger data set. From the analyzed logs we determined the set of IPs corresponding to clients that visited a scraper site as indicated by our previous method. The pattern observed in most situations revealed, as expected, that scrapers are only intermediate agents between users and common sites that properly present information, as depicted in Figure 27. Most visits, as extracted from Squid logs, were following the same pattern. Initially, one makes a request to a search engine using a search term which is related with a specific information that the user is interested in. After submitting the query, the visitor is offered the appropriate results page that contains a small set of results that match the query. Based on the fact that Squid logs contain timestamps of each request, we observed (Figure 7) that the user spends a short amount of time reading the SERP (or at least the first entries from SERP). We denoted by $\Delta t <<$ the short time period that the user spends on a page, usually a few seconds. The next entry in the client visit list, as revealed by the logs, is the request for a scraper site hosted web page. Within the scraper, the user finds the link to a page hosted on the original source site and follows it in order to access the complete and accurate information he or she is interested in. The transition to the new website is also a quick step - usually the amount of time spent to transit the scraper is much shorter (corresponding to the scraper's

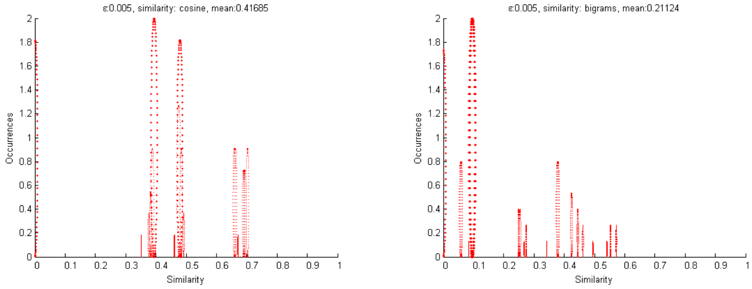

Figure 5: Parked domain P1

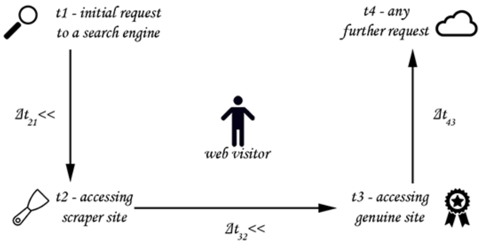

Figure 6: Path of a user transiting a scraper

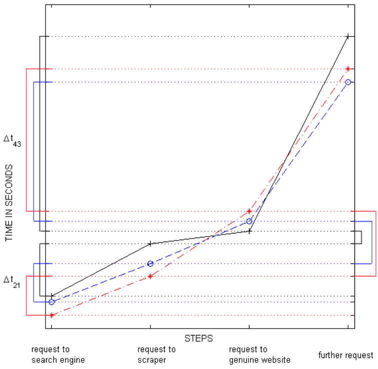

Figure 7: Patterns of scraper transiting sessions

thin content) than the time spent accessing the last requested resource in this chain (i.e. the genuine website).

Another approach to test our proposed mechanism is reversing the original process. Instead of analyzing known scrapers' right shift and look for expected traffic patterns in users' behaviour, we search for the aforementioned pattern in Squid logs for a suspected transit site accessed via SERP having $\Delta t_{32} <<$. Then, we test the suspected transit site's content similarity against external referred genuine content. Besides previously already known scraper sites, we were able to detect from a relatively small Squid data set a new scraper, its scrappiness level depicted in Figure 8 also being right shifted and supported by a mean value of 0.96337.

## V. Conclusions and Future Work

Scraping, as one of the most frequent form of web spamming, is a challenging to combat phenomenon, making further advances in the field absolute necessarily. Content stealing on the web is quickly proliferating and it is becoming more profitable while affecting content providers and publishers. The challenge comes from studying how to avoid content scrapers through enough general methods that can be widely deployed and easily adapted to new types of scrapers.

We have presented in this paper a method through which one can numerically measure and quantify the scrappiness level of a website and also visually display this level. The basic insight of this paper is that we have succeeded in validating our results by reversing the process used in analyzing known scrapers, also proving the effectiveness of this method. Our work attempts to formalize the scraping identification problem and to present a numerical method through which search engines or any other third party can be assisted in the detection of web spam.

As a future work we plan to create an experimental browser plugin through which a web client can be notified about the scrappiness level of a dubitable quality website. Such a plugin could easily and automatically route the user directly to the genuine website, avoiding precious time to be wasted, or can gather anonymous data based on user behaviour, that can be used to further fine tune the scraper identification mechanism. Also, considering other text based or semantic similarity functions and testing how they perform and fit in this scenario is a must in order to increase the quality of the process. Further research is to be done in order to speed up the scraper identification process, considering that it is a time consuming logic that requires high data availability. Any other proposed similarity function should be tested regarding its performance (processing time vs. data volume) and how it performs when it is used to test different type of websites (genuine websites, scrapers or left shifted sites).

Possible false positives and false negatives cases should also be investigated, together with possible methods that might be used by an adversary in order to evade the scrappiness score.

## References

[1] C. Castillo *et al.*, "A reference collection for web spam," in *ACM Sigir Forum*, vol. 40, no. 2. ACM, 2006, pp. 11–24.

[2] D. S. Evans, "The online advertising industry: Economics, evolution, and privacy," *The journal of economic perspectives*, vol. 23, no. 3, pp. 37–60, 2009.

[3] M. Najork, "Web spam detection," in *Encyclopedia of Database Systems*. Springer, 2009, pp. 3520–3523.

[4] N. Spirin and J. Han, "Survey on web spam detection: principles and algorithms," *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 2, pp. 50–64, 2012.

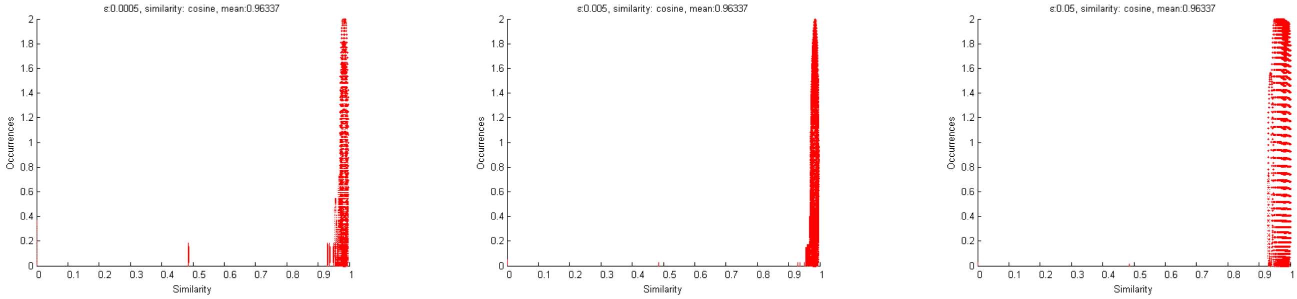

Figure 8: Scraper website S3

[5] R. Patel, Z. Qiu, and C. T. Kwok, "Classifying sites as low quality sites," Apr. 7 2015, uS Patent 9,002,832.

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

[7] M. Erdélyi, A. Garzó, and A. A. Benczúr, "Web spam classification: a few features worth more," in *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality*. ACM, 2011, pp. 27–34.

[8] M. Daiyan, S. K. Tiwari, and M. A. Alam, "Mining product reviews for spam detection using supervised technique," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 8, pp. 619–623, 2014.

[9] G.-G. Geng, Q. Li, and X. Zhang, "Link based small sample learning for web spam detection," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 1185–1186.

[10] C. Wei *et al.*, "Fighting against web spam: a novel propagation method based on click-through data," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 395–404.

[11] C. P. Bharatbhai and K. M. Patel, "Analysis of spam link detection algorithm based on hyperlinks," *IFRSA International Journal of Data Warehousing & Mining*, vol. 4, pp. 67–72, 2014.

[12] L. Araujo and J. Martinez-Romo, "Web spam detection: new classification features based on qualified link analysis and language models," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 581–590, 2010.

[13] D. Fetterly, M. Manasse, and M. Najork, "Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages," in *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004*. ACM, 2004, pp. 1–6.

[14] Y. Liu *et al.*, "Identifying web spam with the wisdom of the crowds," *ACM Transactions on the Web (TWEB)*, vol. 6, no. 1, pp. 2:1–2:30, 2012.

[15] J. Beel and B. Gipp, "On the robustness of google scholar against spam," in *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*. ACM, 2010, pp. 297–298.

[16] D. Haliţă and D. Bufnea, "A study regarding inter domain linked documents similarity and their consequent bounce rate," *Studia Universitatis Babeş-Bolyai, Informatica*, vol. 59, no. 1, 2014.

[17] N. Poggi, J. L. Berral, T. Moreno, R. Gavalda, and J. Torres, "Automatic detection and banning of content stealing bots for e-commerce," in *NIPS 2007 workshop on machine learning in adversarial environments for computer security*, 2007.

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

[19] D. S. Market, "The eu copyright legislation," https://ec.europa.eu/digital-single-market/en/eu-copyright-legislation, Last visited on 25.05.2017.

[20] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. Siam, 2007, vol. 20.

[21] W. H. Gomaa and A. A. Fahmy, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, no. 13, 2013.

[22] G. Kondrak, "N-gram similarity and distance," in *International Symposium on String Processing and Information Retrieval*. Springer, 2005, pp. 115–126.

[23] "Squid: Optimising web delivery," http://www.squid-cache.org/, Last visited on 25.05.2017.