The Evolution of a Crawling Strategy for an Academic Document Search Engine: Whitelists and Blacklists

Jian Wu[†], Pradeep Teregowda[‡], Juan Pablo Fernández Ramírez[†], Prasenjit Mitra[†], Shuyi Zheng[*] and C. Lee Giles[†]
[†]Information Sciences and Technology, Pennsylvania State University, PA, 16802
[‡]Computer Science and Engineering, Pennsylvania State University, PA, 16802
[*]Facebook Inc., Menlo Park, CA, 94025
jxw394@ist.psu.edu

ABSTRACT

We present a preliminary study of the evolution of a crawling strategy for an academic document search engine, in particular CiteSeerX. CiteSeerX actively crawls the web for academic and research documents primarily in computer and information sciences, and then performs unique information extraction and indexing extracting information such as OAI metadata, citations, tables and others. As such CiteSeerX could be considered a specialty or vertical search engine. To improve precision in resources expended, we replace a blacklist with a whitelist and compare the crawling efficiencies before and after this change. A blacklist means the crawl is forbidden from a certain list of URLs such as publisher domains but is otherwise unlimited. A whitelist means only certain domains are considered and others are not crawled. The whitelist is generated based on domain ranking scores of approximately five million parent URLs harvested by the CiteSeerX crawler in the past four years. We calculate the F_1 scores for each domain by applying equal weights to document numbers and citation rates. The whitelist is then generated by re-ordering parent URLs based on their domain ranking scores. We found that crawling the whitelist significantly increases the crawl precision by reducing a large amount of irrelevant requests and downloads.

Author Keywords

Information retrieval; web crawling; search engine

ACM Classification Keywords

H.3.3 Information Search and Retrieval: Retrieval models

General Terms

Design; Performance

INTRODUCTION

Web crawling is an essential part of a search engine. In addition to a high performance crawler, an appropriate crawling policy is critical to optimize the crawl efficiency. Because the number of URLs increases exponentially with the crawling depth, the seed selection is an important factor to achieve a high crawling efficiency [5].

A typical crawler includes a seed scheduler or submitter, a link extractor, a URL distributor, duplicate URL eliminator, a URL prioritizer, a URL frontier, a fetcher, and a series of URL filters [4]. For a typical vertical (focused) crawler, such as the CiteSeerX crawler, the filters are especially customized.

CiteSeerX is a search engine that provides free access to millions of academic and research papers and books. Its specially designed crawler, *citeseerxbot*, is designed to harvest publicly accessible academic documents in PDF and postscript. Because of this, the crawler resources are highly concentrated on research institutions, i.e. home pages with university domains as well as free online publishers. For a general breadth-first crawling, the crawler can visit any URLs linking to a web page. However, the links to PDF and postscript documents may cover various topics including but not limited to presentation slides, class notes, manuals, government reports, schedules and product advertisements. Most of these documents show similar properties to academic papers in terms of content types, document sizes, and URLs patterns, which makes it difficult for the crawler to identify and filter them out. Although we have tools to perform the content-based classification, the original documents need to be fetched and the classification time scale is usually longer than the typical URL enqueueing time scale. Furthermore, the crawler needs to download a large amount of unnecessary HTML pages when visiting irrelevant sites which link to a small number of PDF/postscript files. These HTML files can consume a large fraction of disk space and bandwidth.

An intuitive solution is to create a blacklist, which contains names of hosts to avoid. When the crawler receives the header information from each URL request, it extracts its host name and skips the current URL if it matches any of the host names in the blacklist. It is order linear time and trivial to perform this comparison (currently, the blacklist is less than 1,000 hosts). However, the blacklist solution has several disadvantages. First, it has to be created and edited

WebSci 2012, June 22–24, 2012, Evanston, Illinois, USA.

manually because it is difficult to predict and decide if a host should be blacklisted. Second, as the blacklist grows longer, so does the time the crawler spends on blacklist filtering, which may significantly slow down the crawl speed.

An alternative solution is to create a "whitelist" which contains URLs that provide the resources. In addition, these URLs are most likely to provide new useful documents. Here, we construct a whitelist which contains a list of high quality URLs and compare the crawling efficiencies alternating between using the blacklist and whitelist.

GENERATING THE WHITELIST

Whitelist Properties

The whitelist is generated based on a long crawling history. It must have the following properties: (1) URLs in the whitelist provide the majority of useful documents; (2) the URLs are sorted based on their rank indicators so that top ranked URLs have the highest priority to be crawled.

To satisfy (1), we need to select URLs based on the number of documents they link to. The document usefulness can be parameterized by their citation rates. However, the citation rate is not necessarily a sufficient indicator of URL quality, since most recently posted/published articles are not well cited or cited at all.

Because the number of documents that each specific URL links to are usually small (typically < 100), it is more representative to cluster all URLs with the same web domain (e.g., cmu.edu). In our work, we focus on selecting and ranking web domains and constructing the domain whitelist first. The URL whitelist is then generated by sorting URLs in the order of their domains.

Since CiteSeerX web crawling was launched in 1998 (beginning with the original CiteSeer), we have accumulated over 700,000 parent seed URLs, each of which provides at least one PDF/postscript document. These seed URLs are first grouped by their web domains. We then count the total number of PDF/postscript documents harvested (n_{d}) and the total number of citations received (n_{c}) for each domain. Based on these two numbers, we can generate two domain ranking lists, namely, List D, which is sorted by n_{d} and List C, which is sorted by n_{c}. In this study, we use the first 8,000 domains in either ranking list. This is because we put weights on the URLs which provide the majority of documents ($\sim 98\%$ from List D and $> 99\%$ from List C). The remaining part of List C has $n_{\mathrm{c}} \leq 2$ and the remaining part of List D $n_{\mathrm{d}} \leq 8$.

List D and C can contain different domains, but there are 4706 domains ($\sim 60\%$, hereafter List F) shared by both, which constitutes the *core* of the domain whitelist because these domains have both relatively high numbers of documents harvested and citation rates. The remaining 3294 domains in List D must have very low citation numbers ($n_{\mathrm{c}} \leq 1$), most likely because their documents are not academic papers. The remaining 3294 domains in List C have relatively low numbers of documents crawled ($n_{\mathrm{d}} \leq 8$), but because these domains still have at least one citation, we add them to the the core of the whitelist. As a result, the final domain whitelist contains 8,000 web domains adopted from List C but re-ordered based on both of number of documents and citations. These 8,000 domains contain over 99.99% of the total number of documents we have ever crawled.

The F_1 Rank Indicator

Because of the large scale of the web, it is essential to rank the seed list to prioritize the best URLs [2]. Existing page ordering techniques include prioritizing URLs by indegree [2], PageRank [3], and/or site size. Because the CiteSeerX crawler performs a vertical crawl focusing on a certain type of document, our domain ranking must balance the number of documents and citations, which represent both of the *quantity* and *quality* of the URL seeds. Here, we propose a new ranking indicator, which is defined based on the traditional F score. Traditionally, the F_1 score is defined as the harmonic mean of recall r and precision p. In order to use F_1 as a ranking indicator, we modify the traditional definition and calculate F_1 value by substituting r and p with p_{d} an p_{c}, which are defined respectively as the fraction of crawled documents n_{d} among the total number of documents crawled in List D N_{d} and the fraction of citations n_{c} among the total number of citations received in List C N_{c} for a given domain. The *modified* definition of F_1 is then

$$F_1 = \frac{2p_{\mathrm{d}}p_{\mathrm{c}}}{p_{\mathrm{d}} + p_{\mathrm{c}}} \tag{1}$$

in which $p_{\mathrm{d}} = n_{\mathrm{d}}/N_{\mathrm{d}}$ and $p_{\mathrm{c}} = n_{\mathrm{c}}/N_{\mathrm{c}}$. Because the expression of F_1 contains both of n_{d} and n_{c}, it can only be calculated for the core of the whitelist (List F).

In Figure 1, we plot the p_{c} against p_{d} for all the 4706 domains in List F and color-code data points with their F_1 values[1]. This plot shows that although p_c and p_d exhibits a "correlation", the dispersion becomes larger for low F_1 web domains and this dispersion increases as the F_1 becomes lower (note that the y-axis is logarithmic). This implies that p_c or p_d are equivalent to F_1 only for high ranking domains. For intermediate and low ranking domains, it is better to use F_1 which balances these two parameters. In Table 1, we listed some examples of web domains in the F_1 ranking list. We also tabulate the number of URLs belonging to certain domains. The rankings in the List D and List C can be very different from the new rankings. In addition, there are a significant number of domains which have high rankings in List D but low rankings in List C and vice versa. In Figure 2, we plot the value of F_1 as a function of ranking. The value of F_1 drops by approximately three orders of magnitude for the first 1,000 domains but only about one order of magnitude for the next 2,000 domains. The supplemental part of the whitelist should have even lower values of F_1 indicating that the "long tail" of the whitelist does not make a significant contribution to the harvesting of documents for the whole crawl collection.

The final domain whitelist contains 8,000 web domains. We then select seed URLs from our parent seed URL collection

[1]Because the F_1 values are typically very small, we use $\log 100F_1$.

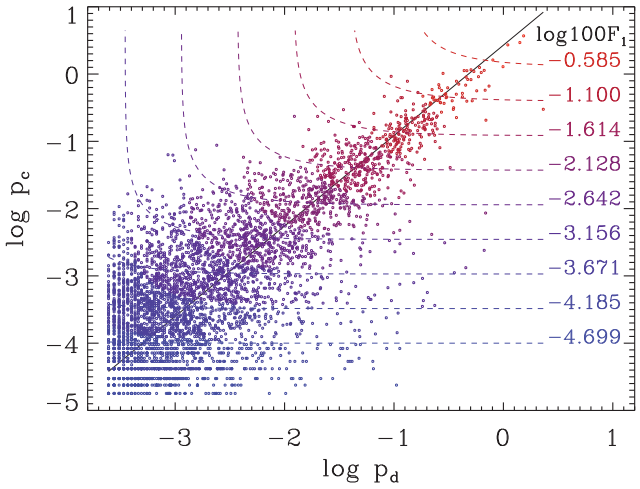

Figure 1. Plot of p_c vs. p_d for all 4706 domains in List F. The data point colors change gradually from red to blue as F_1 value decreases, which are represented by color contours. Numbers on the right side of each contour are corresponding F_1 logarithmic values. The solid black line is the linear regression using the ordinary least squares bisector method.

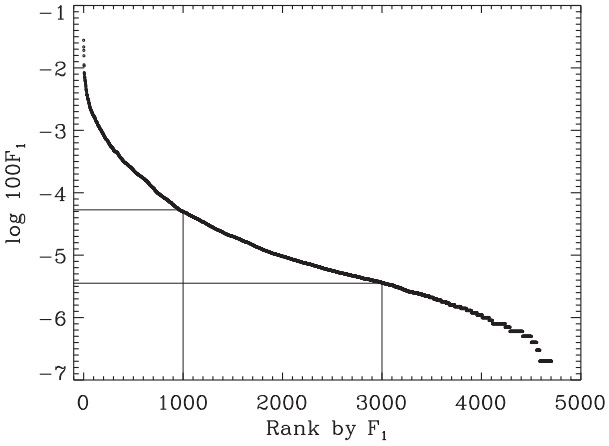

Figure 2. The F_1 values (in logarithmic scale) of all web domains in List F. The "step" patterns at > 4000 ranks are because of domains which have equal number of documents and citation rates (typically less than 10).

Table 1. Examples of domains ranked by the F_1 indicator. Columns are ranks based on F_1 values, number of URLs in a given domain, ranks by p_d, ranks by p_c and domain names.

Rank	log $100F_1$	N(URL)	Rank(D)	Rank(C)	Domains
1	0.443	12723	2	1	cmu.edu
2	0.335	6454	3	2	mit.edu
3	0.281	7209	4	3	berkeley.edu
4	0.194	5041	5	4	stanford.edu
5	0.049	4451	6	6	utexas.edu
203	−1.075	543	99	314	auckland.ac.nz
262	−1.226	351	320	313	unl.edu
347	−1.375	205	481	355	open.ac.uk
513	−1.657	249	678	551	jucs.org
630	−1.805	163	492	833	ntu.edu.sg
1089	−2.377	13	1732	1238	ufpb.br
1256	−2.513	1	2483	1123	ieee-icnp.org
1623	−2.784	7	3904	1035	unibas.it
2061	−3.049	3	2346	2766	jurix.nl
3313	−4.602	2	6836	4034	yeungnam.ac.kr

and sort them by their F_1 rank. The entire URL whitelist contains $> 500,000$ parent seed URLs. We also noticed that the F_1 rank indicator works best for domains with n_d and and n_c greater than 10. When both of them are small, different domains are equally unproductive as they may have the same F_1 values. Other factors need to be considered (e.g., update frequency) when ranking these domains. Though these domains do not provide the majority of the document collection, they are there for completeness and possibly future document harvesting.

CRAWLING THE WHITELIST

The seed URLs in the whitelist are crawled up to a depth of two. It was shown that a crawler needs to visit no more than a depth of 3 to 5 to reach 90% of the pages that users actually visit [1]. Our seed URLs were initially selected based on the number of documents they *directly* link to so a depth of two was found to be sufficient.

To restrict the crawl to be within the whitelist domain, we implement a *ParentDomain* filter which only accepts URLs that have the same domain name as their parents. For example, if the host domain of a seed URL is `cmu.edu`, only subsequent URLs inside the `cmu.edu` is crawled. For the purpose of CiteSeerX, only certain content types are downloaded. Besides HTML/text, the crawler accepts PDF, postscript and their zipped documents with less than 10-megabytes. If a URL contains links to more than one acceptable document, it is treated as a "good" parent URL and saved to the parent URL table for future crawls. The URL information of the fetched documents is saved to another table in the crawl database. The documents themselves and their associated metadata files in XML format are saved to the crawl repository. The crawler strictly obeys the `robots.txt` exclusion rule.

The seed URLs are submitted to the crawling frontier queue by a `cron` scheduler daily at midnight. Our multi-threaded crawler, *citeseerxbot*, can crawl about 10,000 to 20,000 seed urls a day depending on the URL ranking. Typically, high rank seed URLs take more time to crawl as they link to more resources. It takes about a month to finish crawling the entire whitelist.

BLACKLIST AND WHITELIST STRATEGIES

Because we constrain the crawling scope within the whitelist, the crawler does not explore new URLs beyond the whitelist domain. As a result, we may not obtain as many new documents as before. However, we expect that the *crawling efficiency*, which is parameterized by the *fraction* of fetched documents F among the total number of requests Q and the *fraction* of *useful* documents G among the fetched documents F, should increase. We also expect that the seed URLs the crawler discovers to have higher quality on aver-

age, which is parameterized by the number of useful documents per parent URL ($D_{P,G}/P$) and the percentage of useful parent URLs among all parent URLs (P_G/P). Here, we focus on comparing the *average* efficiencies of the blacklist and whitelist crawls.

We use crawling results from 2011. The blacklist and whitelist periods both last for five months but do not overlap. We do this so that the update rates of the online academic and research resources do not change significantly. The useful documents are defined as documents which passed our document content filter and were classified as academic papers/books. The values of F/Q and G/F are computed for the blacklist and whitelist periods, respectively, which are presented in Column 2 and 3 in Table 2.

Table 2. Comparison for the blacklist, whitelist and hybrid periods.

Statistics	Blacklist	Whitelist	Hybrid
Q per day	900002	839284	1402988
F per day	137531	319678	219496
N per day	2855	1399	1738
F/Q	15.28%	38.09%	15.64%
N/F	1.98%	1.24%	0.79%
G/F	0.29%	0.97%	0.29%
G/N	20.33%	49.61%	32.47%
G/Q	0.11%	0.52%	0.07%
$D_{P,G}/P$	0.750	1.082	1.021
P_G/P	0.902	0.934	0.740

The comparison results in Table 2 implies that in the whitelist period, while the total number of requests per day does not change significantly, the number of fetched documents (F/Q) nearly triples compared to the blacklist period. This is because crawling "good" seed URLs reduces the number of failed and filtered requests. The number of new documents per day decreases by 50% and the percentage of N/F decreases little, which is likely caused by the whitelist domain constraint policy. In addition, the fraction of good documents among those fetched G/F increases by a factor of three. This is because our seed selection is biased towards high quality URL resources. In the blacklist period, only $\sim 20\%$ of new documents are useful; in contrast, in the whitelist period, $\sim 50\%$ of crawled documents are useful. The whitelist policy is also better at searching for useful parent URLs (P_G): the fraction of P_G/P increases from 90.2% to 93.4% and the average number of useful documents per parent URL increases from 0.75 to 1.08.

CONCLUSION AND DISCUSSION

Here, we constructed a whitelist containiing over 500,000 URLs from 8,000 web domains based on $> 700,000$ parent URLs accumulated by the CiteSeerX crawler since 1998. The URLs in our whitelist provided us 99.99% of all documents we crawled. These URLs are sorted based on the F_1 ranking indicator which is calculated based on the document numbers and citation rates for each domain. A comparison of the crawling efficiencies between the blacklist and whitelist periods shows that implementing the whitelist significantly reduces the number of useless URL requests and unnecessary downloads and increases the fraction of useful documents.

The whitelist policy includes two essential factors: a ranked seed list, and a domain constrained crawling rule. While this policy reduces crawling irrelevant URLs, it could miss opportunities to discover new resources as well. It is possible to make a *hybrid* policy by combining the blacklist and whitelist policies. In this policy, we still crawl the ranked seed list but remove the crawling constraint rule. The last column of Table 2, presents the statistical results of a test crawl for 10 days using this new policy. The crawling results though noisy reflect the average trends.

The numbers of crawled documents increase by about 25% compared to the whitelist period. This is because the crawler attempted to request URLs in other domains. Although this number is still well below its counterpart in the blacklist period, the document quality is higher: The ratio of G/N in the hybrid period beats the blacklist period by more than ten percent, but is still about seventeen percent lower than the whitelist period. In addition, the values of F/Q and G/F in the hybrid period are also similar to the blacklist period. The number of useful documents per parent URL ($D_{P,G}/P$) and the average percentage of useful parent URLs (P_G/P) are comparable or even lower than the whitelist period. The test results demonstrate that although the hybrid policy allows the crawler to discover more resources beyond the seed domain scopes, the discovery rate is slow, i.e., most of useful documents are still located inside the seed domains.

Although our study is based on the CiteSeerX project, our conclusions can be generalized to generic topic focused crawlers. Seed URLs are essential to improve the crawling efficiency. Increasing the crawling depth is not the best way to harvest useful resources. Seed selection should balance both quantity and quality, which is what can result from using both blacklist and whitelist seeds.

ACKNOWLEDGEMENT

We gratefully acknowledge Jose San Pedro Wandelmer and members of the CiteSeerX group plus partial support from NSF and DTRA.

REFERENCES

1. Baeza-Yates, R. A., and Castillo, C. Crawling the infinite web. *J. Web Eng. 6*, 1 (2007), 49–72.
2. Cho, J., Garcia-Molina, H., and Page, L. Efficient crawling through url ordering. *Computer Networks 30*, 1-7 (1998), 161–172.
3. Cho, J., and Schonfeld, U. Rankmass crawler: A crawler with high pagerank coverage guarantee. In *VLDB* (2007), 375–386.
4. Olston, C., and Najork, M. Web crawling. *Foundations and Trends in Information Retrieval* (2010), 175–246.
5. Zheng, S., Dmitriev, P., and Giles, C. L. Graph-based seed selection for web-scale crawlers. In *CIKM* (2009), 1967–1970.