SiteRank-Based Crawling Ordering Strategy for Search Engines

Qiancheng Jiang, Yan Zhang †
State Key Laboratory of Machine Perception
Peking University
100871 Beijing, China
{jiangqc, zhy}@cis.pku.edu.cn

Abstract

Search engines are playing a more and more important role in discovering information nowadays. Due to limitations of time-consuming, network bandwidth and hardwares, we cannot obtain the whole information on the web and have to download important information first. In this paper we propose a novel crawling ordering strategy which is based on SiteRank. Experimental results running on over 15 million pages indicate that it can work efficiently in discovering important pages under the PageRank evaluation of page quality. Furthermore, it exhibits the ability of anti-spamming.

1 Introduction

Search engines are becoming a predominant tool for information discovery and retrieval on the infinite web. But as the huge volume of web pages and limitations of hardwares, a search engine can only fetch a fraction of the web. Therefore, which pages should be downloaded first is significantly crucial. Usually this problem may take many factors into consideration. No doubt that the quality of the downloaded page set is one of the most important factors. In common sense, people always prefer to download the most important pages first.

For example, when a web crawler prepares to download a page set, it starts with an initial set of seeds. Then all the URLs that are parsed from the seed pages are added into the URL waiting list for further visit. A main problem is how to choose the pages that should be visited next. This is so-called Crawling Ordering Strategy. There are several representative crawling ordering strategies such as Breadth-First[2, 3, 6], Backlink-Count[3], Partial-PageRank[2] and Larger-Sites-First[2].

Breadth-First selects page according to the order of the URL in the list. This method is very easy to implement and can work effectively in most situations. Backlink-Count always visits the page with the most backlink count first, that is to say, the number of backlink is the metric of page “quality”. Partial-PageRank uses the web graph of pages that have been downloaded so far to compute the PageRank[1, 8] values of pages in the list, and then chooses the page with the highest score for further visit. Obviously, this is a time-consuming process although it sounds good in discovering high quality pages. Larger-Sites-First is mainly based on the assumption that a large-scale site may have a high possibility of high quality. So this strategy prefers large sites to small sites. Experimental results indicate that this strategy outperforms the three methods above. However, it has a great bias against small sites and how to define “large” is still a subtle problem.

Although all of these strategies have been used by some search engines, they share the same problems. They do not take effective page quality metric and seldom make use of history information of previous crawling. So we propose a novel strategy based on SiteRank which prefers the pages from sites with high SiteRank scores. This method not only makes good use of the history information, but also gives an equivalent chance to every page. Meanwhile, it has the ability of anti-spamming.

The reminder of this paper is structured as follows. Section 2 gives an overview of the related work. Section 3 introduces our algorithm in detail. Section 4 shows our experimental results running on 15M pages. We draw our conclusions and give some future work in the last section.

2 Related work

Much attention has been drawn on the crawling ordering strategy so far. The explosion of the web makes the crawlers unable to crawl the whole pages on the web. They have to download the high quality web pages first. Then there comes the challenges: what does the “high quality” mean

---

*Support by NSFC under Grant No.60673129 and 60573166, Beijing Natural Science Foundation under Grant No.4073034
†Corresponding author
and how to select the page for next visit during crawling process.

Breadth-First[2, 3, 6] makes a simple breadth-first traversal of the web graph. This strategy can slightly promote the discovery of the high quality pages, and can avoid the overload of network on a specific site. Mark Najork et al. [6] performed a crawl of 328 million pages over the entire web, covering more than 7 million distinct hosts. When using PageRank[4, 8] as the page quality metric, they found that Breadth-First downloaded the hot pages first and the average quality of pages decreased over the duration of the crawling process.

Backlink-Count was proposed by Junghoo Cho et al.[3]. This strategy regards a page with the most backlink count as the candidate of the next visit. Junghoo Cho’s experiments on 179,000 pages of stanford.edu domain showed that Backlink-Count had poorer performance than Breadth-First on the page quality metric of PageRank.

Partial-Pagerank uses the partial pagerank score to select the “important” pages. Obviously this is a time-consuming process which makes this strategy quite inefficient. At the same time, the experimental results on the .cl and .gr domain given by Ricardo Baeza-Yates[2] suggested that the Partial-Pagerank was less effective than Breadth-First. So Ricardo Baeza-Yates et al. proposed a Larger-Sites-First strategy[2]. They prefer pages of large sites to those of small sites. It works more efficiently than all the other strategies on the page sets of 3.4 to 3.6 million pages of .gr domain and 2.4 to 2.8 million pages of .cl domain.

Since most crawling process is periodic, the crawler is always rich in history information. Thereby people can use these useful information to guide the crawling. In common sense, a strategy with history information is usually performing better than without it. This is verified in Baeza-Yates’s experiment.

3 SiteRank-Based Crawling Ordering Strategy

We introduce SiteRank first and then illustrate how to employ it to direct the crawling process. At last, we give a crawler simulator model.

3.1 SiteRank

There are two key problems we care about. The first is how to define the concept of site, the second is how to compute SiteRank score. People can define a site according to either the URL or the host address of each site. When we use site URL to define a site, we regard the URL before the first slash as an independence site in order to avoid sites with too many pages. For example, http://www.lib.pku.edu.cn and http://www.lib.pku.edu.cn are considered as two different sites. When using host address, we just consider different server IPs as different sites, although a site may deploy to several servers or a server may support many sites. In our experiment, we use site URL to divide sites. Because this way is not only reasonable but also easy to operate. Further, it is fit for the concept of sites in common sense.

SiteRank is an algorithm which is similar to PageRank. It exploits the relation of citations among sites. For a Site-Graph \( G = \langle V, E \rangle \) which consists of two parts: a set \( V \) of vertices where each vertex represents a web site and a set \( E \) of edges where each edge represents a direct link between two sites. Note that the links between two sites’ pages are accumulated, i.e., if there are three sites \( S_1, S_2, S_3 \), where 100 direct links between \( S_1 \) and \( S_2 \), 10 between \( S_1 \) and \( S_3 \). Obviously, the recommendations of \( S_2 \) and \( S_3 \) by \( S_1 \) are not equivalent. So we weight the link count when compute the SiteRank. When using random surf model, we can gain the transition matrix \( M \) as follows:

\[
M(i, j) = \begin{cases} 
C_{ij}/C_i & \text{if } S_i \text{ points to } S_j \text{ and } C_i \neq 0 \\
0 & \text{if } S_i \text{ does not points to } S_j \\
1/N & \text{if } C_i = 0 
\end{cases}
\]

In the above definition, \( C_{ij} \) represents the direct link count that site \( S_i \) points to site \( S_j \) and \( C_i \) represents all the outlink of site \( S_i \). \( N \) is the total number of web sites. Considering that some sites do not have outlinks, we use a decay factor to ensure the matrix \( M \) have a non-trivial Eigenvector. This is the same way used in PageRank. So we can get a formula for SiteRank computation just the same as PageRank:

\[
M = \alpha M + (1 - \alpha)/N 
\]

Jie et al. demonstrate the convergence and distribution of SiteRank in detail in the article [9]. He gives a concrete algorithm to compute it and makes an experiment on a dataset of EPFL domain of a campus-wide Web graph which contains 600 independent Web sites identified by their hostnames or IP address. The experimental results indicate that the distribution of SiteRank scores follows the power-law. At the same time, SiteRank can better identify spam pages than PageRank.

3.2 Advantages of SiteRank

The SiteRank-Based crawling ordering strategy has an assumption that the pages included in high SiteRank sites are always good in quality. This assumption is reasonable and this strategy has some advantages in discovering high quality pages.

Stableness: Compared with PageRank, stableness is the main advantage. As we know, SiteRank ignores the intra-site links and it only considers the inter-site links. The statistics by Jie Wu et al.[9] on the web links suggests that
76% of the web links comes from intra-site. And the web is varying all the time, the architecture of a site is changing as well. The lifespan of a page is getting shorter[7]. The links among pages are not stable enough. Therefore PageRank is lack of stableness. However, the links between sites are varying less frequently. So the SiteRank score can keep stable for a longer period and we can use the history information of SiteRank to guide the next crawling. The stableness can ensure the significance of history information when using SiteRank.

**Fairness:** Baeza Yates et al. propose the Larger-Sites-First strategy which chooses the sites having the most pending pages for the next visit [2]. This inevitably leads to the injustice to small sites which are good in quality. Also, this strategy has the limitation of introducing more sites to the user. The SiteRank prefers the quality to the size of a site to some extent. SiteRank can give all the sites equivalent opportunities.

**Anti-Spamming:** Link spamming is one of the common spamming techniques[5]. Spammers often build link farms on their own sites to boost their target pages. When we use links to analyze the quality of the pages or sites, inter-site links are more significant. So SiteRank can greatly weaken the action of link farms. Thereby, the SiteRank-Based strategy can improve the quality of the crawled pages.

### 3.3 The Crawler Simulator

In our experiment, we do not crawl Web pages in the real web. Instead, we use a simulator to imitate the crawling process on a static dataset. The reason is as below. First, the real web is changing all the time. To ensure the justness to all the crawling ordering strategies, we must make sure that all the algorithms are running on the same dataset. So a static dataset is necessary. Second, it is known to us that a real crawler need to take many factors into consideration, such as time, network bandwidth and resource distribution. While in our experiment, we only focus on the ability of discovering high quality pages. So a crawler simulator is more reasonable and operable.

Figure 1 gives a view of the work flow of a crawling simulator. In this figure, there are mainly two modules. We just omit some details in realization and generally give a rough view of the simulator. The crawling ordering module is the core of the architecture. We aim at this module to improve our crawling ordering algorithm. At the very beginning of the process, we select some seed sites. Then we have such an ordering strategy: there are two queues in the waiting list, one is a queue of pages with history information of SiteRank which is labeled as \( Q_{ws} \), the other is a queue without history information which is labeled as \( Q_{wo} \). We sort these two queues respectively. The \( Q_{ws} \) is sorted by SiteRank and the \( Q_{wo} \) is sorted by the number of pending pages at that moment. We only need to get the max of these two queues’ first sites according to the number of pending pages at that time. Thus, we crawl the pages selected by the crawling ordering module, add the newly discovered page URLs into the waiting list. We iterate this process until the waiting list is empty. Finally we will get a crawling sequence. We analyze the sequence and make some statistics on it to evaluate the performance.

### 4 Experimental Evaluation

We run our crawling simulator on a 15M-pages dataset and compare our method with the previous strategies. We present our experimental results in this section.

#### 4.1 Data Set

Our data set is a partial set of pages crawled by Tianwang search engine (developed by network lab, Peking University) in Nov. 2005. It contains 15.3 M pages with about 232 M links on 358,245 sites, most of which belong to .cn domain. Table 1 gives some statistical information about this data set.

<table>
<thead>
<tr>
<th>Sites</th>
<th>358,245</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages</td>
<td>15.3M</td>
</tr>
<tr>
<td>Links</td>
<td>232M</td>
</tr>
<tr>
<td>Avg. Pages per site</td>
<td>42.7</td>
</tr>
</tbody>
</table>

#### 4.2 Performance Evaluation

We give a performance evaluation model to evaluate the performance of all the algorithms. In this model we mainly have two rules to evaluate whether a strategy is effective or not.
1. The strategy should discover high quality pages as early as possible.

2. The pages of spam sites should be identified and thereby downloaded at the later stage of the crawling process.

Based on these two rules, we can get a generally qualitative analysis for each strategy. Obviously, how to evaluate the quality of a page is always a challenge. It is difficult for us to give an objective judgement on a page. PageRank\[8\] is an acceptable and reasonable metric on page quality. We know there are many pages using some illegal ways to gain an undeserved PageRank. We call these sites Spam and crawl these sites is a vain work. Thus, we have two metrics in our experiments. They are Cumulative Pagerank and Spam sites in buckets. The details are discussed as below:

**Cumulative Pagerank**: This metric gives us a prospect of the process of discovering high quality pages. We equally set ten detecting points in our experiment. At each point, we calculate the sum of PageRank of the crawled pages. Usually, at a random crawling order, the distribution of cumulative PageRank is a diagonal. On an ideal occasion, the distribution curve has a rapid increase at the beginning of the crawling process and varies to flat at the middle of the process.

**Spam sites in buckets**: Before we run the crawler simulator, we first label 100 spam sites manually. Then we also set ten detecting points as that for the cumulative PageRank. At each point, we count how many spam sites appear. Under this metric, it is better that the spam pages appear in the later phase of the crawling process.

### 4.3 Experiments

We implement Breadth-First, Backlink-Count, Larger-Sites-First as well as our SiteRank-Based strategy. The crawling is a serial process to avoid the extra factors which may affect the experimental results. At the same time, we do not make any optimization for all of the algorithms. We just use the naive algorithms because we do not care about running time.

#### 4.3.1 Seed Selection

Seed selection is an important problem in this experiment. The quality of the seed set will affect the experimental results. If the seed set is not sound, the crawled part may poorly represent the whole data set and the analysis on the results is useless and makes no sense. In order to guarantee the propagation and coverage of the seed set, we manually choose 30 high quality site as the initial seed set. We have two principles for the seed set. First, the site itself should have high PageRank to ensure the richness in links and the propagation. The second is the seeds should be domain dispersed which can avoid the bias of the crawling pages. Based on these two principles we can get a good coverage of the data set. In our experiment, the coverage can reach to 70% which is pretty good.

4.3.2 Experimental Results

Figure 2 shows the performance of each crawling ordering strategy on discovering high quality pages. We use PageRank as the metric of page quality. We set ten detecting points at each 10% increment of pages and compute the sum of PageRank of the crawled pages. From Figure 2, we can draw some conclusions:

1. All the four strategies make an optimization for the naive crawling more or less. At the beginning of the crawling process, all the strategies can crawl the pages with high PageRank. When downloading 30% of the pages, the sum of PageRank is over 50% even for the worst one. At the later phase of crawling, the sum of PageRank varies slowly and reach to 1 finally.

2. Larger-Sites-First performs slightly better than Breadth-First and Backlink-Count. The result is same as what Baeza-Yates et al. described in his paper[2]. At the first 20% fraction of pages, the sum of PageRank is 10% higher than Breadth-First. Some noises appear at the end of 40%, Backlink-Count surpasses the Larger-Sites-First. The reason is that the Larger-Sites-First strategy often crawls pages of small sites which are low in PageRank at the end of the crawling.

3. Our strategy makes a notable improvement and works effectively in discovering high quality pages. From Figure 2 we can find that at the first 20% fraction, the
sum of PageRank can reach to 80%. It is 1.5~2 times higher than the other three. At the half of the crawling process, the sum of PageRank reaches to 90%.

(4) The Backlink-Count strategy is more effective than Breadth-First in our dataset. It is opposite to the Baeza-Yates’s experiment in [2]. It is most likely that different the data set leads to this phenomenon.

Besides discovering high quality pages, the SiteRank-Based strategy has the ability of anti-spamming. Our strategy can effectively delay the crawling of spam pages. Figure 3 shows the result of the distribution of spam sites in each crawling phase. We labeled about 100 spam sites at first. At the end of the crawling, the ratio of crawled spam sites is from 49.5%~64%. The SiteRank-Based crawls the fewest spam sites and the Larger-Sites-First gets the most. We can get more details from the Figure 3.

![Figure 3. The Distribution of Spam sites in a Crawling Process](image)

(1) At the first 60% fraction of the pages, we can see that Backlink-Count introduces the most spam sites. The main reason for this is that link spam is the generally used spamming technique we mentioned above. Larger-Sites-First is also performing badly in delaying the crawling of spam pages. It not only introduces many spam sites at the beginning but also the most spam sites in total.

(2) The SiteRank-Based and Breadth-First have the similar performance for this metric. The curve of the SiteRank-Based is lower than the other strategies all the time at the same phase of the crawling. Note that the Breadth-First can also perform well in delaying crawling spam pages. So Breadth-First strategy is a good choice to crawl pages when there is no restrict quality requirements.

5 Conclusions and Future Work

The explosion of the web information and many restrictions of the crawler force the crawler have to download the high quality pages first. In our experiment on a 15M-pages data set, we implement three strategies which are representative and considered to be effective. At the same time we propose a novel ordering strategy which is based on SiteRank, since SiteRank has advantages of stableness and the ability of anti-spamming.

In order to evaluate the performance of all the strategies, we propose two evaluation metrics: cumulative PageRank and the ability of anti-spamming. The experimental results show that the SiteRank-Based strategy has the best performance in discovering high quality pages. Moreover, the SiteRank-Based strategy is verified that it has the best performance of anti-spamming. It can effectively delay the crawling of the spam pages.

There are also some limitations in our experiment. For a real web crawler, the initial seed set is a crucial factor to the final crawled page set. We put this discussion as our future work. At the same time, we will do more validation experiments on a larger data set. The most important open problem is how can we apply the ordering strategies into real use.

References


