PRIORITY BASED FOCUSED WEB CRAWLER

Jaytrilok Choudhary¹ and Devshri Roy²
¹(Department of CSE, MANIT, Bhopal, India)
²(Department of CSE, MANIT, Bhopal, India)

ABSTRACT

The World Wide Web (WWW) has billions of documents and these documents are attached to each other using hyperlinks. Web crawler is a heart of Search engine that gathers these documents from WWW. Maximum documents present on WWW are dynamic and changes periodically. Hence, Crawler needs to refresh these documents to update database of search engine. In this paper, we have proposed a priority based focused web crawling algorithm. The web pages corresponding to URL (Uniform Resource Locator) are downloaded from web and calculated the relativity score of downloaded page with focus word. We store URL and its relativity score with focus word in priority queue instead of normal queue. So, every time priority queue returns maximum Score URL to crawl next. The overall performance gain over simple crawler is 87% and over focused crawling is 24%.

Keywords: Priority, focus word, web pages, downloader, search engine.

1. INTRODUCTION

Web search engine is designed to find information which is related to search query specified by user from WWW. Search engine stores millions of web pages and their links. These web pages are needed to be refreshed that makes it more reliable. Search engine uses web crawler for this purpose. Web crawler is a continuous running program which downloads web pages periodically from WWW. The downloaded pages are indexed and stored in a database as shown in Fig. 1.[1]
There are two types of web crawling: breadth-first crawling and best-first crawling [2].

1.1 Breadth first crawling

Breadth-first crawling method is same as breadth-first search in a graph. Web crawler starts with initial seed URLs. It downloads web pages for given URLs. Then extract new URLs from the downloaded pages, add them into queue and pick up URL one by one and repeat same process for specific count or until queue is empty. The architecture of classical web crawler is shown in fig. 2 which has the following four components:

![Architecture of Web Crawler](image-url)

**Figure 2:** Architecture of Web Crawler
1.1 Queue
Queue is a data structure used by crawler to store URLs extracted from the downloaded pages for further used in crawling process.

1.1.2 Scheduler
Scheduler selects URL from the queue and sends it to downloader for downloading the web page at given URL.

1.1.3 Downloader
Downloader downloads web page at given URL.

1.1.4 Storage
After downloading page from the web, crawler stores page in stable storage.

1.2 Best first crawling
The best first crawling focuses on to download only relevant pages of a particular given topic. A crawler using a best first crawling strategy is known as a focused crawler [2]. In other words, focused crawling is a variation of breadth first crawling where web pages related to particular topic or set of topics are downloaded only.

In this paper, we are presenting a priority based focused web crawler that will download relevant pages related to a particular topic or focus string only. We have used priority queue instead of simple queue to store web pages and their similarity score with focus string. Every time when a delete operation performed on queue will return maximum score web page.

The reminder of this paper is organized as follows: section 2 contains related work in this area. Section 3 is the architecture of priority based focused web crawling. Section 4 contains the algorithm of priority based focused web crawling. Section 5 describes the experimental results and Section 6 is the conclusion and future work.

2. RELATED WORK

The various crawling algorithms has been proposed which are as follows:

N. Singhal et. al.[1] have designed an incremental web crawler. The incremental crawler visits the internet periodically to update its database. Based upon updation web documents, web documents are categorized and grouped as very frequently, frequently less frequently. The crawler visits a site frequently and the frequency of visits may be adjusted according to the category of the site. This architecture is more suitable for parallel web crawler.

S. Ganesh et. al.[3] have proposed an ontology based web crawler. In this approach, a new metric called association-metric has been proposed. The association-metric analyzes the semantic content of the URL based on the domain dependent ontology. After downloading the page, the association metric estimates the relevancy of the links in that page. Finally, reordering of URL is done based on relevancy of web page.

D. Mukhopadhyay et. al.[4] have proposed a domain specific web crawler which crawls domain specific Web pages from the World Wide Web(WWW). Crawler uses ontology of a domain for which web pages has to be crawl.

X. Chen et. al.[5] gave the methodology for focused crawling. They have focused on content of web page to improve page relevance and also used link structure to improve the coverage of a specific topic. They considered only two factor, content of web page and link structure, to get relevancy of web page.

D. Hati et. al.[6] have proposed an adaptive focused crawling based on link analysis. In this approach, they first calculate the score of unvisited URL based on its anchor text relevancy score, relevancy score of its parent, its description in Google search engine and calculate the similarity
score of description with topic key words. The major issue of this technique is URL queue optimization.

S. Thenmalar et al.[7] have proposed an algorithm for focused crawling based on ontology. They are preparing topic as an overall conceptual vector that is obtained by combining concept vectors of individual pages associated with seed URLs. Here the role of ontology is to obtain concepts associated with seed page. The next URL to be crawl is based on the conceptual rank of the web page at that level which is obtained by conceptual matching between conceptual vectors of all web pages at each level.

3. PRIORITY BASED FOCUSED WEB CRAWLING

The overall crawling process is shown by figure 3.

![Figure 3: Architecture of Priority based focused crawler](image)

The crawling process begins with initial seed URL and focus word.

Step 1: Crawler downloads web page corresponding to given URL.
Step 2: Now, it extracts all the URLs present in downloaded page.
Step 3: Again, crawler downloads all the pages corresponding to all new extracted URLs.
Step 4: Now, we calculate cosine similarity between focus word and all downloaded pages that works as relativity score.
Step 5: We add page and its relativity score into the priority queue and every time when we delete a page from priority queue, queue will return a maximum similarity page.
Step 6: Now, repeat step 2-5 for either specified number of pages or until queue is empty.

Let crawler starts crawling with initial seed URL, we assign score 1 to initial seed URL. Web page is downloaded from web for seed URL and new URLs are extracted from downloaded page i.e. URL 1, URL 2,..., URL N as shown in figure 4.

Now, crawler again downloads web pages for every new URL which are Page 1, Page 2,..., Page N. We calculate similarity score between web pages and focus word. Let Page 1, Page 2,..., Page N has score 0.7, 0.5,..., 0.8 respectively where 0.8 is maximum score. Downloaded pages and
their score are inserted into priority queue. Now a page is deleted from priority queue, page with highest score is selected. The maximum score URL is URL N, crawler will extract all the URL from Page N first.

Figure 4: Priority based focused crawling process

Again, it will repeat the same process i.e. extracting URLs (URL N1, URL N2, ..., URL NM) and downloading pages (Page N1, Page N2, ..., Page NM). Calculate the similarity score between newly downloaded pages and focus word. Let Page N1, ..., URL NM has score 0.25, ..., 0.6 respectively where 0.6 is maximum. Downloaded pages and their score is inserted into priority queue. Now a page is deleted from priority queue at this time Page 1 is selected because it has maximum score then remaining other pages present in queue. Every time maximum score page is selected for crawling. This will be the advantages of using priority queue over simple queue. This will defiantly improve performance of crawling process over normal crawling process.

4. CRAWLING ALGORITHM

The priority based focused crawling algorithm works as follows:

**Input:** Initial seed URL, Focus_String and PQqueue.

**Output:** Web_Pages related to Focus.String.

**Step 1:** Page := downloadPage(URL);

**Step 2:** addPQueue(Page, 1);

**Step 3:** While PQqueue is not empty do

**Step 4:** Page := dePqueue();

**Step 5:** newURLs := extractURL(Page);

**Step 6:** for each i^th^ URL in newURLs do

**Step 7:** Page[i] := downloadPage (newURLs[i]);

**Step 8:** RScore[i] := SimScore(Page[i], Focus_String);

**Step 9:** addPQueue(Page[i], RScore[i]);

**Step 10:** end for;

**Step 11:** end while;
Descriptions of various modules used in algorithm are as follows:

1.1 addPQueue(Page, Score)
addPQueue module add a new downloaded page and its similarity score with Focus_String into Priority Queue.

1.2 dePQueue()
dePQueue module returns a page which has maximum score from Priority Queue.

1.3 downloadPage(URL)
downloadPage module downloads web page from WWW corresponding to given URL.

1.4 extractURL(Page)
extractURL module extracts all URLs which are present in given Page.

1.5 simScore(Page, Focus_String)
simScore module calculates cosine similarity score between Page and Focus_String.

5. EXPERIMENTAL RESULTS

The performance of Focused crawler is measured by harvest rate. Harvest Rate measures the rate at which relevant pages are crawled and how effectively irrelevant pages are filtered off from the crawl [8].

\[
\text{Harvest ratio} = \frac{R}{N} \quad (1)
\]

Where
R: No. of relevant web pages crawled, and
N: Total No. of web pages crawled

This harvest ratio must be high. We have used standard data set present on WWW in the form of open directory named “http://www.dmoz.org”. We have evaluated and compared the harvest rate of our crawler with simple crawler and focused crawler. The figure 5 shows the harvest rate of breadth first crawler, focused crawler [2] and priority based focused crawler. The 1000 web pages crawled on focus word Computer, Science, Regional and Sports, and average of harvest rate are taken.

![Priority based focused crawling process](image)

**Figure 5:** Priority based focused crawling process
6. CONCLUSION

In this paper, we proposed a priority based focused crawler which keeps all URLs to be visit in priority queue along with their relativity score. When we delete URL from priority queue, it returns maximum score URL. Thus, every time a highest priority URL is return for crawling. Experimental results shows that our crawler gives 24% improved results over focused and 87% improved results over simple crawler. The main problem of this crawling strategy, it is more time consuming. In future, we will try to reduce the crawling time by implementing algorithm parallely.

REFERENCES


