An Effective Method for Ranking of Changed Web Pages in Incremental Crawler

Arvind Kumar
Assistant Professor
Vidya college of engineering, Meerut, India

Km. Pooja
VICST, Meerut, India
Vidya college of engineering, Meerut, India

ABSTRACT
The World Wide Web is a global, large repository of text documents, images, multimedia and much other information, referred to as information resources. A large amount of new information is posted on the Web every day. Web Crawler is a program, which fetches information from the World Wide Web in an automated manner. The crawler keeps visiting pages after the collection reaches its target size, to incrementally update/refresh the local collection. By this incremental update, the crawler refreshes existing pages and replaces less-important pages with new and more-important pages. Incremental web search requires a much smaller amount of data processing of the web. There is a problem in searching new information from the web in incremental web search to evaluate ranking of changed web pages. We developed an effective solution to this problem. In order to evaluate ranking of changed web pages. An Integrated ranking framework combining three metrics: Popularity Ranking, Content-based Ranking and Evolution Ranking which produce good Ranking for the changed web Pages.

Keywords— Popularity Ranking, Content-based Ranking, Evolution Ranking, Integrated Ranking.

1. INTRODUCTION
The World Wide Web is a global, large repository of text documents, images, multimedia and much other information. It is estimated that web contains more than 2000 billion visible pages. Due to the extremely large number of pages present on Web, the search engine depends upon crawlers for the collection of required pages. The general architecture of the web crawler is shown in Fig 1.

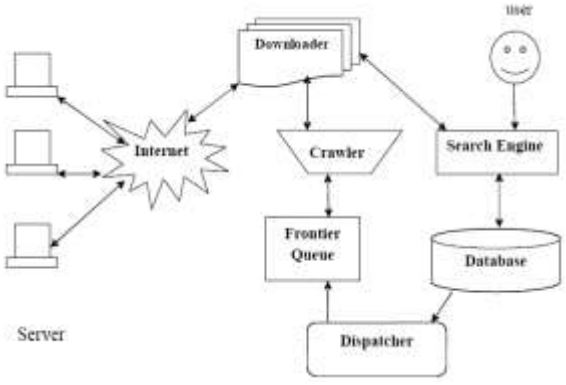

Fig.1 Architecture of the web crawler

In order to download a document, the crawler picks up its seed URL, and depending on the host protocol, downloads the document from the web server. For instance, when a user accesses an HTML page using its URL, The crawler simply sends HTTP requests for documents to other machines on the Internet, just as a Web browser does when the user clicks on links. A single URL Server serves lists of URLs to a number of crawlers. Web crawler starts by parsing a specified Web page, noting any hypertext links on that page that point to other Web pages. Then parse the pages for new links, and so on, recursively. When the crawler visits a Web page, it extracts links to other Web pages. So the crawler puts these URLs at the end of a queue, and continues crawling to a URL that it removes from the front of the queue.

The Algorithm of the Web crawler is given below:

1) Read a URL from the set of seed URLs.
2) Determine the IP address for the host name.
3) Download the Robot.txt file which carries downloading permissions and also specifies the files to be excluded by the crawler.
4) Determine the protocol of underlying host like http, ftp, gopher etc.
5) Based on the protocol of the host, download the document.
6) Identify the document format like doc, html, or PDF etc.
7) Check whether the document has already been downloaded or not.
8) If the document is fresh one
   Then Read it and extract the links or references to the other Cites from that document.
9) Else
   Continue.
10) Convert the URL links into their absolute IP equivalents.
11) Add the URLs to set of seed URLs.

2. INCREMENTAL CRAWLER REVIEW
The crawler visits the web until the collection has a desirable number of pages, and stops visiting pages. When it is necessary to refresh the collection, the crawler builds a brand new collection using the same process described above, and then replaces the old collection with this brand new one. This type of crawler is called a *Periodic crawler*. The crawler may keep visiting pages after the collection reaches its target size, to incrementally update/refresh the local collection. By this incremental update, the crawler

refreshes existing pages and replaces less-important pages with new and more-important pages. When the crawler operates in this mode, then this is called an *Incremental crawler*.

The architecture of the Incremental Crawler is shown in Fig 2.

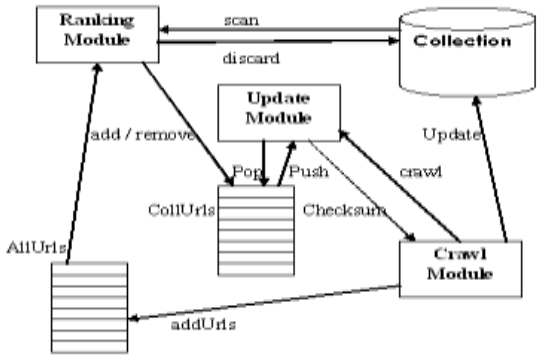

Fig. 2. Architecture of Incremental Crawler

The URLs in CollUrls are chosen by the Ranking Module. The Ranking Module Constantly scans through AllUrls and the Collection to make the refinement decision. When a page not in CollUrls turns out to be more important than a page within CollUrls, the Ranking Module schedules for replacement of the less-important page in CollUrls with that more-important page. The URL for this new page is placed on the top of CollUrls, so that the Update Module can crawl the page immediately. Also, the Ranking Module discards the less-important page from the Collection to make space for the new page while the Ranking Module refines the Collection; the Update Module maintains the Collection fresh (update decision). It constantly extracts the top entry from CollUrls, requests the Crawl Module to crawl the page and puts the crawled URL back into CollUrls. To estimate how often a particular page changes, the Update Module records the checksum of the page from the last crawl and compares that checksum with the one from the current crawl. From this comparison, the Update Module can tell whether the page has changed or not. The Crawl Module crawls a page and saves/updates the page in the Collection, based on the request from the Update Module.

3. EVALUATING PAGE RANK CHANGES BETWEEN DIFFERENT WEB PAGES

In searching new information over the web there are exiting problems are as follows:

- In Popularity Ranking, there exists a problem of How to rank changes between different web documents?
- In Content-based Ranking, there exists a problem of How to rank changes appearing at different locations on a single web page?
- In Evolution Ranking, there exists a problem of How to rank changes appearing at different time on a single web page?

In order to evaluate the New Information Fragments (NIF) in the local database, combine the following metrics to produce an integrated ranking: Popularity, Content-based and Evolution Ranking. As shown in Fig 3

Fig.3 The framework of ranking web changes

3.1 Popularity Ranking

The popularity ranking uses the link structure of URLs to infer which pages are important in the web graph. The popularity ranking algorithm uses the page ranking is as follows:

Qp(A) =(1-d)+d(Qp(T1)/C(T1)+……..+Qp(Tn)/C(Tn)) (3.1)

Where
- Qp(A) is the Page Rank of page A,
- Qp(Ti) is the Page Rank of pages Ti which link to page A,
- C(Ti) is the number of outbound links on page Ti
- d is a damping factor which can be set between 0 and 1.

So, the Page Rank does not rank web sites as a whole, but it determined for each page individually. Further, the Page Rank of page A is recursively defined by the Page Ranks of those pages which link to page A. The Page Rank of pages Ti which link to page A does not influence the Page Rank of page A uniformly. Within the Page Rank algorithm, the Page Rank of a page T is always weighted by the number of outbound links C (T) on page T. This means that the more outbound links a page T has, the less will page a benefit from a link to it on page T.

3.1.1 Example

The following example illustrates to calculate page rank of the following web graph as shown in fig 4.

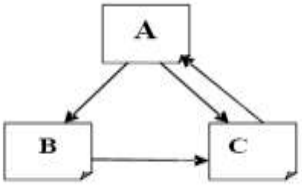

Fig 4 Web graph for pages

Web consisting of three pages A, B and C, whereby page A links to the pages B and C, page B links to page C and page C links to page A. let take the value of damping factor d is 0.5.

Qp(A) = 0.5 + 0.5 Qp(C)
Qp(B) = 0.5 + 0.5 (Qp(A) / 2)
Qp(C) = 0.5 + 0.5 (Qp(A) / 2 + Qp(B)

These equations can easily be solved. We get the following Page Rank values for the single pages:

Qp(A) = 14/13 = 1.07692308
Qp(B) = 10/13 = 0.76923077
Qp(C) = 15/13 = 1.15384615

3.2 Content-based ranking

In evaluating the content of new information carried by web changes, there are two considerations: How much information carried and how timely that the information is. The amount of information can be evaluated simply from the length of the NIFs. In order to evaluate how timely web changes are from the content of web changes, we need to know what kind of words can be good indicators of new information.
Then count the word frequency of the web changes found as well as the word frequency of all web pages. After removing the stop words, general verbs and too common words that are unrelated to new information. Then divide the words into groups: Time information words, Time related words, related words, popular topic words and the rest go to misc words. It shows that web changes are more likely to contain time related information. News or event information in web changes often have short lifetime. Such information also has more significance when we present web changes. Therefore, Count the appearances of time related words in NIFs as a metric of quality ranking.

Let L be the length of an NIF; and N be the number of time related words in the NIF.

The quality score is defined as:

$$\mathbf{Q_c = W_1 L + W_2 N} \quad (3.2)$$

Where W_1 and W_2 are weights determining the scale of quality ranking score.

3.2.1 Example

The following example illustrates the working of Q_c of page A.

For page A, Let take L = 3, N = 5 and w = 1/3
$Q_c A = 3/3 + 5/3$
$Q_c A = 2.64$
So, the ranking $Q_c A$ is 2.64.
Like this, QcB is 1.85 and QcC is 1.50. This shows that this ranking method increases the page rank of the pages of A, B, C.

3.3 Evolution ranking

Popularity ranking and content ranking can produce static ranking of information. However, the importance of each web change decreases as time goes on. The evolution ranking reflects this decrease over time. Let evaluate the web changes using an exponential function. Let Q_0 be the score of static ranking. The evaluation of new information at query-time goes as follows:

$$\mathbf{Qev = Q_0 \, . \, e^{-\alpha t}} \quad (3.3)$$

Where α is a parameter to determine how fast the score decreases.
t = 0 corresponds to the timestamp when the modification is made or when the change is detected.

- How should α be set?

When a NIF has been removed from the web, we consider the information it carries is obsolete and of little importance. Denote the average lifetime of web changes on a single page as Tc, the ranking score of a web change at its creation as Q0 and the score when it is removed as pQ0
The parameter α satisfies:

$$Q0 \cdot e^{-\alpha t} = pQ0$$

Therefore

$$\alpha = -\log(p)/Tc$$

Where Tc denote the average lifetime of versions for a single web pages.

3.3.1. Example

To find out the time the importance of web page changes decreases.
Let take α = .346 and time (t = 1,2 ,3 , 4….n)

For page A
$Qev = 1.07692308 * e^{-.346*1}$
$Qev = 0.76193$
$Qev = 1.07692308 * e^{-.346*2}$
$Qev = 0.539$
$Qev = 1.07692308 * e^{-.346*3}$
$Qev = 0.3814055$

This shows that the importance of page A is decreases over the time.

For page B,
$Qev = 0.76923077 * e^{-.346*1}$
$Qev = 0.544240$
For page C,
$Qev = 1.15384615 * e^{-.346*1}$
$Qev = 0.8163599$
This shows that the importance of page B and Page C.

3.4 Integrated ranking

The ranking scores of popularity ranking, quality ranking and evolution ranking, Denote Qp as the normalized score of popularity ranking, Qc as the normalized score of content-based ranking and the α of evolution ranking. Integrated Ranking score of New Information Fragments at query time is:

$$\mathbf{Q(T) = (Qp + Qc) \cdot e^{-\alpha(t-t0)}} \quad (3.4)$$

Where t0 is the creation time of the web change.
Let choose the sum of Qp and Qc rather than the product as static score. Qp measures the importance of the page that contains the change. Such combination can well distinguish the significance of multiple changes of the same page at the same time. For changes on different pages, which is evaluated more than one another is determined by the configuration of the numerical scale between Qp and Qc.

3.4.1 Example

The following example is to illustrate working of integrated ranking for page A.

From above the Fig 3.2, for page A calculated Qp is 1.07692308 from the equation 3.1, Qc is 2.64, $\alpha = 0.346$, t0 = 1 and t1=2.

These value put into equation in 3.4

Q = (1.07692308 + 2.66)* e-0.346(1)
Q= 3.74*.7075
Q=2.65
For t1 = 3,
Q = (1.07692308 + 2.66)* e-0.346(2)
Q= 3.74*0.50057
Q=1.8721464
For t1 = 4,
Q = (1.07692308 + 2.66)* e-0.346(3)
Q= 3.74*0.35416
Q=1.32456
For page B,
Let t0 = 1 and t1 = 2
Q = (0.76923077 + 1.85) * e-0.346(1)
Q = 1.853084
For page C,
Q = (1.15384615 + 1.50) * e-0.346(1)
Q = 1.8775961

Above result show the rank of page A has been improved by comparing all the ranking system. Also increases the freshness time of the page. It has been solved all above problems.

4. RESULT ANALYSIS

We shows that integrated ranking combining three metrics: Popularity Ranking, Quality Ranking and Evolution Ranking produce good Ranking for the web Pages and degree of freshness of web page.
Degree of freshness for page A

Time(t)	Evolution Ranking	Integrated Ranking
1	0.76193	2.65
2	0.539	1.8721464
3	0.3814055	1.32456

Table 4.1 Degree of freshness for page A over the time

Comparison of page rank of web pages with integrated ranking with other ranking system

Page	Popularity Ranking	Content-based Ranking	Evolution Ranking	Integrated ranking
A	1.07692	2.64	0.7619	2.65
B	0.76923	1.85	0.5442	1.85308
C	1.15384	1.50	0.8163	1.87759

Table 4.2 Comparison of integrated ranking with other ranking

5. CONCLUSION

In order to bring new information to users in a timely manner, Incremental web search requires a much smaller amount of data processing than full indexing of the web. Therefore, new information carried by changes can be updated in the web index more quickly. The Problems of searching for new information over the web by the Incremental Crawler are solved by an Integrated ranking framework is proposed combining three ranking metrics: Popularity Ranking, Content-based Ranking and Evolution Ranking. The Page Rank score is a good predictor of change frequencies of web pages. Such predictor can be combined with change history date of web pages to improve the effectiveness of the estimator of change frequencies.

6. REFERENCES

[1] Cho, J. Ntoulas, A. and Olston, C. In Proc. 13th International World Wide Web Conference, 2004. What's new on the web? : the evolution of the web from a search engine perspective.

[2] Cho, J. and Roy, S. In Proc.13th International World Wide Web Conference, 2004. Impact of search engines on page popularity.

[3] Wang, Z. In Proc.5th International Conference of Web Age Information Management, 2004. Improved link-based algorithms for ranking web pages.

[4] Fretterly, D., Manasse, M., Najork, M. and Wiener, J. In Proc. 12th International World Wide Web Conference, 2003. A large-scale study of the evolution of web pages.

[5] Edwards, J., McCurley, Kevin S. and John A. In Proceedings of the Tenth Conference on World Hong Kong, May 2001. An adaptive model for optimizing performance of an incremental web crawler.

[6] Brewington, Brian. , Bharat, Krishna. , Cybenko, George., Maghoul, Farzin. , and Stata, Raymie. In Proceedings of the Ninth Conference on World Wide Web Amsterdam, Netherlands, May 2000. How dynamic is the web?

[7] Cho, J. and Garcia-Molina, H. In Proc. 26th International Conference on Very Large Data Bases, 2000. The evolution of the web and implications for an incremental crawler.

[8] Brewington, B. and Cybenko, G. IEEE Computer, 33(5), 2000. Keeping up with the changine web.

[9] Dean, J. and Henzinger, M. In Proceedings of the 8th International World Wide Web Conference (WWW8), 1999, "Finding related pages in the world wide web.

[10] Fred Douglis, Anja Feldmann, and Balachander Krishnamurthy, 1999.Rate of change and other metrics: a live study of the world wide web.

[11] Cho, J., Garcia-Molina, H. and Page, L. In Proc. 7th International World Wide Web Conference, 1998. Efficient crawling through URL ordering.

[12] Page, L. (1998). The Page Rank Citation Ranking: Bringing Order to the Web.