Web Searching With Logarithmic and Probability Measure

S. Subatra Devi
Assistant Professor
PSVP Engineering College
Chennai, Tamil Nadu, India

P. Sheik Abdul Khader, PhD.
Professor & HOD
BSA Crescent Engineering College
Chennai, Tamil Nadu, India.

ABSTRACT
The web is a huge and highly dynamic environment which is growing exponentially in content and developing fast in structure. No search engine can cover the whole web, but it has to focus on the most valuable pages for crawling. Many methods have been developed based on link and text analysis for retrieving the pages. In this paper, an algorithm based on link, text, logarithmic distance and probabilistic measure is presented to find the relevancy of the web pages. Here, the most relevant pages are retrieved. It has been proved experimentally that this method provides more number of relevant pages.

Keywords
Search engine, Logarithmic distance, Probabilistic measure.

1. INTRODUCTION
The web contains more than 350 million pages and continues to grow rapidly by creating million pages per day. Such growth and fluctuation generate essential limits of scale for today's generic search engines [7]. Since the number of web sites is growing rapidly, the number and size of stored documents are increasing faster and also the contents of websites are often getting updated increasingly [1]. A preliminary set of web pages (seed pages) are given as input to the web crawlers and it extracts the outgoing links emerging in the seed pages and decide what links to visit next, based on certain criteria. Crawlers continue visiting the Web pages until a desired number of pages have been downloaded or until local resources such as storage are exhausted [4].

In this paper, an efficient web crawling algorithm is presented by combining text content, link analysis, logarithmic distance and probabilistic measure. Initially, the seed URL is extracted from the web by inputting the relevant keyword. Then, the relevant pages are identified from it according to the relevancy score computed for all the outgoing web pages. Once the outgoing links are filtered, the crawling process is then continued within these links until the relevance of the corresponding page is not satisfied by the threshold value. The advantage of the proposed algorithm is that it makes use of the link, text content as well as the logarithmic distance and the probabilistic measure to find the similarity. Here, the similar and also the dissimilar keywords are found with the probabilistic measure for improving the efficiency.

The rest of this paper is organized as follows. Section 2 specifies the related work. Section 3 proposes the algorithm for web crawling process. Section 4 shows the experimental results and performance evaluation of the proposed algorithm. Finally, the conclusion of the result is given in section 5.

2. RELATED WORK
In the literature, numerous algorithms have been proposed for web crawling process. Fish Search is one of the first dynamic search heuristics, that capitalizes on the intuition in which relevant documents often have relevant neighbors [14]. It does not provide enough relevant search directions, under time limit constraints. A shark search algorithm have been developed which is an improved and more powerful version of that algorithm. The shark-search algorithm overcomes the limitations of the fish search algorithm, by better evaluating the significance of neighboring pages, even before they are accessed and analyzed [12]. They assigned starting URLs, which are relevant to an interested topic to the crawler. Initial step is to determine the starting URLs or the starting point of a crawling process. The crawler is unable to traverse the Internet without starting URLs. Similar to focused crawler [16], a user has to define some starting URLs to the crawler.

It is claimed that the crawler does not need any starting URLs [18], and the crawler is able to find a direction to target pages by starting at non-related web pages. But, it is believed that [10] the crawler with good starting URLs is capable of gathering more relevant web pages.

For the content-based method, the proposed shark-search [12], motivated by fish-search [14], which used topic similarity of a vector space model technique as a parameter in URL ordering process.

With the huge growth of World Wide Web, the existing general-purpose search engines have presented much more limitations [2]. The search results are ranked based on user preferences in content and link and integrated to rank the results [3]. A search engine with a specialized index has more structured content and provides higher accuracy than a generalized search engine, as it has already been intelligently extracted from the web [20]. Searching based on hyperlink and content relevance strategy is discussed in [15].

Web crawlers retrieve the web pages and include them or their representations into a local repository. The processing of crawler begins from a seed page and then it uses the external links within the seed page to deal with other pages [8]. The algorithm [10], covers the discussion of both the initial and successive crawling.

3. PROPOSED METHOD FOR WEB CRAWLING
The seed page is the most important page for extracting the relevant information. This seed URL is selected by giving the keywords as input to the most popular search engines such as Google, Yahoo. Then, the resulting URLs that are common in

these search engines are considered to be more relevant to the query and such URLs are taken as seed URLs.

3.1 Calculating the Relevancy Score

The seed URL that is calculated in the previous step and the keywords are given as input to the crawler. From the seed URL, the outgoing links are extracted. For each outgoing link, the relevancy score is calculated by using the methods specified.

3.1.1 Link Weight

The category of a web page is determined based on the keywords on the link, anchor text on the link and other attributes. Here, the anchor text of link is used to calculate the link weight based on division method. Link weight is used to find how many anchor text are presented in link content. If all the anchor text keywords are presented in the URL belongs, then its division score is 1. Otherwise, the link weight depends on the percentage value of anchor text appearing in the current link. The link weight is calculated using the following formula.

$$W_t = \frac{U_L}{L_T}$$

Where, U_L represents the total number of links that contain the anchor text and L_T represents the total number of links presented in the parent node.

3.1.2 Text content similarity

The text content similarity of the two pages, the seed URL page and the child page are computed using Levenshtein Distance. The keywords are extracted from the web page content using the text mining processes such as stop word removal and stemming method. The extracted tokens are given to the Levenshtein Distance, which calculates the text content similarity between the pages by using the word distance and the length of word forms. This is calculated based on the following formula.

$$D_{lev}(s1,s2) = \frac{\sum \text{EditDist}(s1,s2)}{\text{length}(s1)+\text{length}(s2)}$$

where EditDist is performed to calculate the number of insertion, deletion and substitution operation which are needed to transform one string s1 into the another string s2.

3.1.3 Logarithmic Distance

Here, the logarithmic distance which was proposed in FICA algorithm [20] is computed, between each links for finding the shortest path. The logarithmic distance is computed based on the total number of outgoing links in the current web pages along with the logarithmic distance of previous stage web pages. The seed URL obtains the logarithmic value of the total number of outgoing link. Then, the second level web pages obtain the distance by adding the logarithmic distance of previous level with the current level. A sample graph G of finding the logarithmic distance is show in Figure 1.

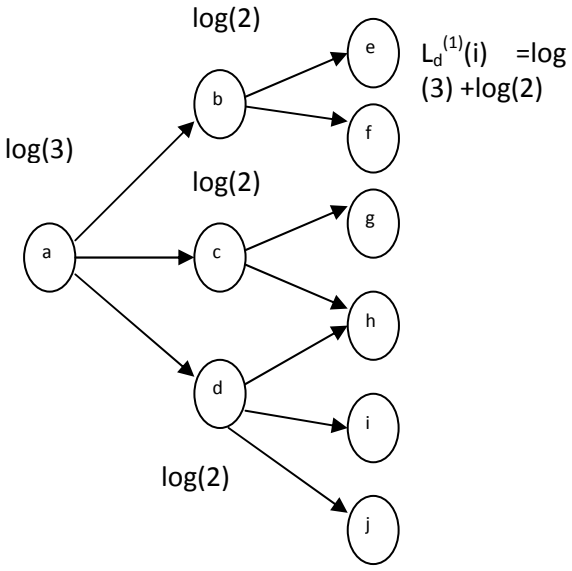

Figure 1: A sample graph of Logarithmic distance

The computation of the distance is done using the following formula.

$$L_d^{(2)}(i) = L_d^{(1)}(i) + \text{Log}(L_t(i))$$

where $L_d^{(1)}(i)$ is the logarithmic distance of the seed URL i , $L_t(i)$ is the total number of outgoing links for the web page i, $L_d^{(2)}(i)$ is the logarithmic distance of second level web page i.

3.1.4 Probabilistic Measure:

The proposed probability-based distance measure is used to find the similarity between the seed page with respect to the outgoing link. Initially, the preprocessing steps of text mining are applied and the tokens are extracted from the web pages and the tokens are sorted based on the frequency.

Subsequently, the top 'n' keywords are selected to find the probabilistic measure that mainly depends on the positive and negative occurrences of the keywords. For example, considering the seed URL 'u' that consists of a set of keywords W_i which is defined by the equation given below

$$W_i = \{ k_1, k_2, k_3, \ldots, k_n \}$$

where, k is the keyword of the URL 'u' and 'n' is the number of keywords. In the seed URL, there will be many other hyperlinks known as the sub links and of which, the best suitable link is selected. For this, the probabilistic measure is used by comparing the similar and dissimilar keywords in the seed URL and the sub link URL. In the set of output link, select one page denoted W_k, $(1 <= k <= n)$

$$W_k = \{ j_1, j_2, j_3, \ldots, j_n \}$$

Then, compute the probability of occurrence of similar keywords for both pages, W_i and W_k along with the probability of occurrence of dissimilar keywords in both the pages. The number of similar and dissimilar keywords are

sorted based on the frequency of its occurrence. Then, the probabilistic measure for the two pages W_i and W_k are given by

$$P_m = \mu [P (w_i \in w_k) + 1] - (1 - \mu) [1 - P (w_i \text{ not} \in w_j)]$$

where, $P(w_i \in w_k) = T / N$ and, $P (w_i \text{ not} \in w_j) = K / N$, T refers to the similar keywords of both the pages, N is the total number of keywords and K refers to the dissimilar keywords of both the pages.

3.2 Selecting the Outgoing Links using Relevancy Score

This step is used for finding the outgoing links that is then put into the URL queue to sequentially perform the above mentioned steps. Here, the outgoing link is selected from the seed URL page based on the relevancy score and the link was chosen from the outgoing link whether the relevancy score R_s satisfies the user specified value. The relevancy score for finding the relevant outgoing links is computed through the equation described as follows.

$$R_s = \alpha * W_t + \beta * L_d + \lambda * D_{lev} + \delta * P_m$$

After finding out the relevancy score, compare it with the specified threshold value, T_r. If the relevancy score value is greater than the threshold value, then it is given to the URL queue. The same process in the earlier steps is then sequentially performed for all the pages in the URL queue until the URL gets empty.

4 RESULTS AND DISCUSSION

In this section, experimental results of the proposed web crawling algorithm are presented. The proposed algorithm has been implemented in java (jdk 1.6) and the experimentation is performed on a 3.0 GHz Pentium machine with 2 GB main memory.

4.1 Experimental Results

The experimentation of the proposed algorithm is carried out by inputting the various keywords to the search engine Google, so that the seed URLs are extracted. With the help of the seed URL, the web pages are crawled from the web. The outgoing link from the seed URL are fetched and the value of the child URLs are determined by using the above specified method and the relevancy score is calculated for each link. This process is repeated until the threshold value of maximum depth is reached.

The value of the Relevancy score is calculated, by using the four different methods data for each outgoing link. The experimentation is performed for different threshold values and for different keywords. The relevancy score is compared with different threshold values. If the relevancy score is more, then the outgoing links are stored in folders and the relevant pages are determined from those links.

From the number of retrieved pages, the number of relevant pages are calculated and the graph is given in Figure 2 and Figure 3 for two different queries. The result is compared for different threshold values and the graph is generated for the proposed algorithm and the Integrated Page Rank algorithm [3]. The graph experimentally proves that the proposed algorithm fetches more number of relevant pages for the given query compared to the Integrated Page Rank algorithm.

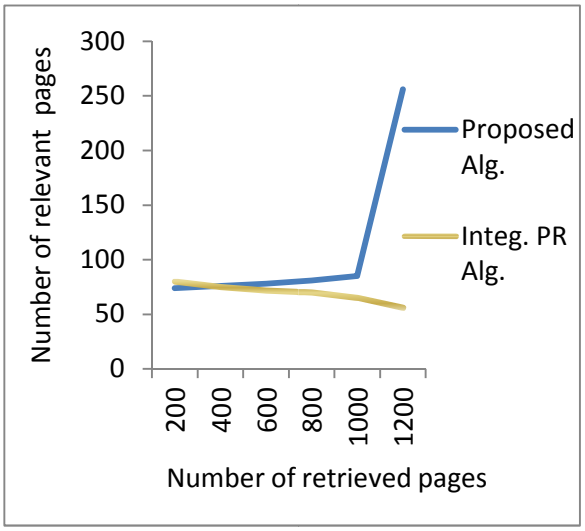

Figure 2: Result for the keyword "Computer books"

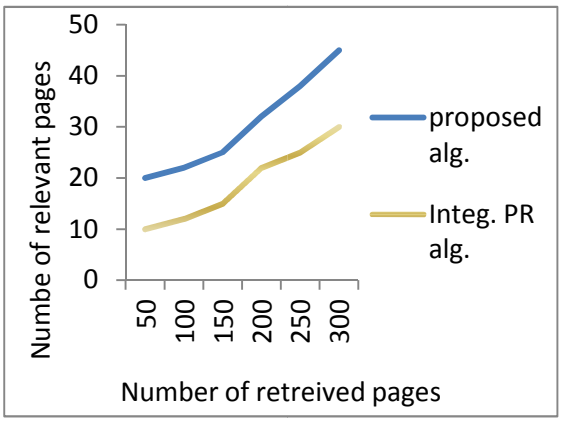

Figure 3: Result for the keyword "Java"

For different threshold value and for various keywords, the Relevancy score is computed by the crawler and the efficiency of the algorithm is determined.

5. CONCLUSION

In this paper, an efficient algorithm for web crawling method is proposed. The proposed algorithm efficiently utilizes the link, text content, logarithmic distance and probability measure to find the relevancy of the web pages. For experimentation, different keywords are given to crawl the web pages. In order to evaluate the effectiveness, the proposed algorithm is compared with the Integrated Page Ranking algorithm [3], which is proved in the experimental results.

6. REFERENCES

[1] G. Almpanidis, C. Kotropoulos, I. Pitas, September 2007. Combining text and link analysis for focused crawling—An application for vertical search engines. Information Systems, Vol 32, No: 6, pp: 886-908.

[2] Zhumin Chen; Jun Ma; Jingsheng Lei; Bo Yuan; Li Lian, Aug 24-27, 2007. An Improved Shark-Search Algorithm Based on Multi-information. Fourth

International Conference on Fuzzy Systems and Knowledge Discovery, pp: 659 – 658.

[3] J.Jayanthi, Dr.K.S. Jayakumar, January 2011. An Integrated Page Ranking Algorithm for Personalized Web Search, International Journal of Computer Applications, Vol 12 – No.11.

[4] Sotiris Batsakis, Euripides G.M. Petrakis, Evangelos Milios, Improving the Performance of Focused Web Crawlers, Data & Knowledge Engineering, Vol: 68, No: 10, pp: 1001-1013, October 2009.

[5] Blaž Novak, Survey of focused web crawling algorithms, in Proceedings of SIKDD, pp. 55-58, 2004.

[6] Shalin shah, Spe 2006. Implementing an Effective Web Crawler.

[7] Yang Yongsheng, Wang Hui , Implementation of Focused crawler, Journal of computers Vol. 6, No: 1, January 2011.

[8] Pant, G., Srinivasan, P., Menczer, F., Crawling the Web. Web Dynamics: Adapting to Change in Content, Size, Topology and Use, edited by M. Levene and A. Poulovassilis, Springer- verlog, pp: 153-178, November 2004.

[9] Debashis Hati and Amritesh Kumar, An Approach for Identifying URLs Based on Division Score and Link Score in Focused Crawler, International Journal of Computer Applications, Vol. 2, no. 3, May 2010.

[10] A. Rungsawang, N. Angkawattanawit, Learnable topic-specific web crawler. Journal of Network and Computer Applications, Issue no:28,page no:97-114,2005

[11] Sandeep Sharma, Mr. Ravinder Kumar, Web-Crawling Approaches in Search Engines, June 2008.

[12] Michael Hersovici, Michal Jacovi, Yoelle S. Maarek, Dan Pelleg, Menachem Shtalhaim and Sigalit Ur, The shark-search algorithm. An application: tailored Web site mapping, in Proceedings of the Seventh International World Wide Web Conference on Computer Networks and ISDN Systems, Vol. 30, no. 1-7, pp. 317-326, April 1998.

[13] Brin, S., & Page, L. The anatomy of a large-scale hyper textual web search engine. In Proceedings of the seventh international conference on World Wide Web (WWW), pp: 107–117, 1998.

[14] P. De Bra, G-J Houben, Y. Kornatzky, and R. Post, Information Retrieval in Distributed Hypertexts, in the Proceedings of RIAO'94, Intelligent Multimedia, Information Retrieval Systems and Management, New York, NY, 1994.

[15] Lili Yan, Wencai Du, Yingbin Wei, and Henian Chen, A Novel Heuristic Search Algorithm Based on Hyperlink and Relevance Strategy for Web Search, Advances in Intelligent and Soft Computing Volume 149, 2012, pp 97-102.

[16] S. Chakrabarti, M. van den Berg, and B. Dom, Focused Crawling: A New Approach for Topic-Specific Resource Discovery, In Proc. 8th WWW, 1999.

[17] M. Najork and J.L. Wiener. Breadth-first crawling yields high-quality pages, In Proceedings of the Tenth Conference on WorldWideWeb, Hong Kong, Elsevier Science, May 2001, pp. 114–118.

[18] Aggarwal C. Garawi F.Yu P. Intelligent crawling on the world wide web with arbitrary predicates, In: Proceedings of the 10th International World Wide Web Conference. Hongkong: 2001.p. 96-105.

[19] A. Rungsawang, N. Angkawattanawit, Learnable Crawling: An Efficient Approach to Topic-specific Web Resource Discovery, 2005.

[20] R. Steele, Techniques for specialized search engines, Directory of Open Access Journals, 2001.