IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 1 (Jul. - Aug. 2013), PP 102-106
www.iosrjournals.org

Topic-specific Web Crawler using Probability Method

S. Subatra Devi[1], Dr. P. Sheik Abdul Khader[2]
[1] *(Research scholar, PSVP Engineering College, Chennai, Tamil Nadu, India)*
[2] *(Professor & HOD, BSA Crescent Engineering College, Chennai, Tamil Nadu, India)*

Abstract : *Web has become an integral part of our lives and search engines play an important role in making users search the content online using specific topic. 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 content analysis for retrieving the pages. Topic-specific web crawler collects the relevant web pages of interested topics of the user from the web. In this paper, we present an algorithm that covers the link, text content using Levenshtein distance and probability method to fetch more number of relevant pages based on the topic specified by the user. Evaluation illustrates that the proposed web crawler collects the best web pages under user interests during the earlier period of crawling.*
Keywords - *Levenshtein Distance, Hyperlink, Probability Method, Search engine, Web Crawler.*

I. Introduction

The web is a very large environment, from which users provide the keyword, query or topics to fetch the required information. Such growth and fluctuation generate essential limits of scale for today's generic search engines [5]. The number of web sites keeps on increasing and hence the number of links and the documents keep on increasing day by day [1]. Crawlers are programs that penetrate through the hyperlinks to retrieve the required page. Based on the hyperlink, the crawler retrieves the document and the required numbers of web pages are retrieved. This process is repeated until the predefined numbers of web pages are downloaded or the storage of the local repository is exhausted [3].

In this paper, an efficient web crawling algorithm is presented by combining text content, link analysis and probabilistic method. Initially, the topic is given as input to various search engines. Here, Google, Yahoo and MSN are used. For the given topic the common URL's in all the three search engines or in any two, are retrieved and given as input to the crawler. This URL will be considered as the seed URL. The crawler, considering this as the seed URL will retrieve the web pages based on the hyperlink. The retrieving is done by considering the keyword on the link, the text content of the document and the probability method. The probability method is used to find the number of similar as well as dissimilar keywords occurring in a web page. Determining the probability of the dissimilarity in keywords allows filtering the irrelevant pages in the beginning of the crawling. This makes to consider the relevant pages more effectively and skews the search. The link content and the text content are the common methods used by most of the algorithms. Using this as the base, when it is implemented along with the probability method, this method provides more effective pages.

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.

II. Related Work

There are several algorithms based on content and link strategy. The algorithm based on hyperlink and content relevance and on HITS is presented as Heuristic search [14]. Another method specifies the ranking based on content and link [13]. In [9], the relevancy is based on 0 or 1, which consists of several drawbacks that are overcome in the Shark- search [8].

The division score and the link score is applied for each and every link in [6]. Some knowledge bases based on starting URL's, topic keywords and URL predictions are being discussed [7]. An importance of a page is computed based on the composition of PR score in [2]. Focused crawling analyzes its crawl boundary based on the links that are most likely to be relevant and avoids irrelevant regions [10]. An algorithm [22] on hyperlinks and content relevance strategy is based on topic-specific crawling.

The topic keyword is used as a base in several algorithms [2], [7], [10] and [12] by which the crawler crawls through the web to fetch the relevant pages. The crawler using breadth-first search [19] order discovers high quality pages during the early stages of the crawl. An algorithm in [20], combines text search with semantic search. To engineer a search engine Google with page rank method is applied in [17]. Different arbitrary predicates are applied [18] to perform an intelligent crawling. Accurately predict the relevance [15] of unvisited web pages by known URL's. In this method, this algorithm crawls the web by considering the

keyword in the hyperlink, the content of the document and the probability method. A latent semantic indexing classifier that combines link and text [21] is used to index domain specific web documents.

III. Proposed Method

The seed URL is the base path for retrieving the more relevant pages. This URL is fetched by giving the topic keyword [2], [7] to the three different search engines. The seed URL is given as root path to the crawler. Now, the outgoing links of this root path are fetched. For each of this link, three methods are applied as specified below.

1.1 Calculating the Relevancy Score

From the outgoing link as specified above, the relevancy score is calculated in an iterative method for each and every link. The calculation consists of the following methods.

1.1.1 Link Weight

The link weight is calculated by checking whether the topic keyword is presented in the outgoing links. If this anchor text is present the division method is used to determine the link weight. Here, if all the topic keywords are presented in the outgoing link then its link score is 1. Otherwise, the link score is based on the percentage of the topic keywords present in the link. Finally, the link weight is determined by the ratio of the total number of links containing the anchor text to the total number of links of the parent node. This Link Weight is given by

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

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.

For each and every parent URL, the link weight is determined and the crawling is performed based on the ranking of the parent URL's.

1.1.2 Text content similarity using Levenshtein Distance

Here, the Levenshtein Distance is used to compute the text content similarity of the two pages, i.e., the seed URL page and the child page. Before computing the similarity, the tokens are extracted from the pages. These tokens are nothing but the topic keywords of that page. For filtering these topic keywords, the two methods

- Stop word removal
- Stemming algorithm

Are used. The stop word removal method is, removing the base words from the document such as, a, an, the, etc. The stemming algorithm applied here is Porter Stemming algorithm, which skews the word with the stem of the word. For example, 'engineering', 'engineered', etc. are stemmed to the word 'engineer'.

The words extracted by these methods are known as tokens. These tokens are given as input to the Levenshtein Distance for 'EditDist'. Here, the distance is computed by determining the similarity between the two strings. For which, the following operations are performed to transfer one string s1 to another string s2.

- insertion
- deletion
- substitution

Finally, the text content similarity between the seed page and the child page is calculated based on the word distance and the length of the word.

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

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.

1.1.3 Probability Method

The probability based distance method is used to calculate the similarity between the two pages, the seed URL page and the child page. The tokens are extracted from these pages, said to be as keywords and the frequency of those keywords are found and represented in sorted order. The top n keywords are selected from the parent and the child pages. The similarity and the dissimilarity are calculated between these two pages.

$P_m = [P(w_i \in w_j) + 1] - (1 - \mu)[1 - P(w_i \notin w_j)]$

where, $P(w_i \in w_j) = T / N$, and $P(w_i \notin w_j) = K / N$, T refers to the number of similar keywords of both the pages, N is the total number of keywords and K refers to the number of dissimilar keywords of both the pages.

Based on the sorted order of the similar keywords, the documents are given the priorities in decreasing order. The probability of the similar keywords helps in retrieving the relevant pages effectively. Similarly, the probability of finding the dissimilar keywords supports in filtering the irrelevant pages effectively in the beginning stage of crawling. For each and every iteration, the irrelevant pages are filtered which makes the crawler to retrieve more number of relevant pages.

1.2 Determining the Relevancy Score

Here, the relevancy score is determined by using the above three methods. The computed value of the above three methods

$R_s = \alpha * W_t + \beta * D_{lev} + \gamma * P_m$

Where W_t represents the link weight, D_{lev} specifies the levenshtein distance and P_m specifies the probability method.

This relevancy score R_s is calculated for each and every link and it is compared with the threshold value. For different threshold value, the relevancy score is determined and the relevant pages are retrieved.

IV. Results And Discussion

In this section, it is proved with the experimental results that the proposed web crawling algorithm using link, text analysis and probability method is more effective. The proposed algorithm has been implemented in java (jdk 1.6) and the experimentation was performed for different categories.

The experimentation was performed with different topic keywords which are specified in Fig. 1 given below. The figure also shows that more number of relevant pages is retrieved by using the above mentioned method.

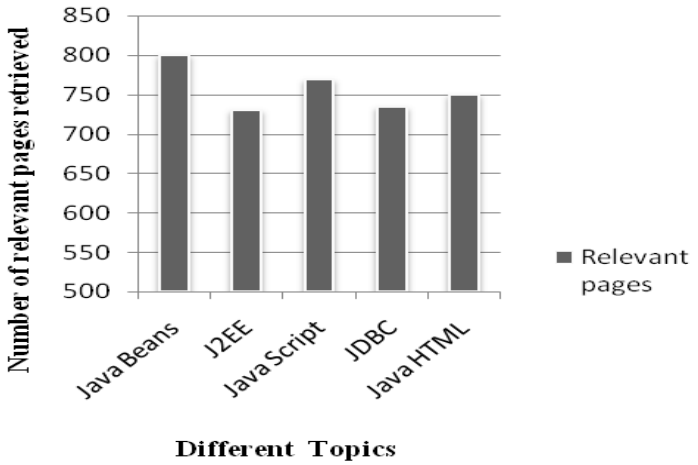

Fig. 1. Result for different topics

The analysis was made for different threshold values for different keywords. When the relevancy score value is less than the threshold value, then those pages are considered as the irrelevant pages and are removed from the URL queue. Otherwise, the pages are relevant pages and placed on the queue based on the priority. The more relevant pages that were retrieved for particular threshold values were taken for consideration. From these set, the average number of relevant pages are considered for each topic.

The Fig. 2 mentioned below specifies the relevant pages retrieved for the keyword 'Java Beans' on different threshold values. This process is applied for the different topics to determine the efficiency of the algorithm. The co-efficient factor α, β and γ was also tested with different values ranging between 0 and 1 for increasing the weight of the individuals.

Similarly, the experimentation was considered for different values of the co-efficient factor and the relevancy score was determined for each URL. It was proved that more relevant pages were retrieved by using this method for different topics.

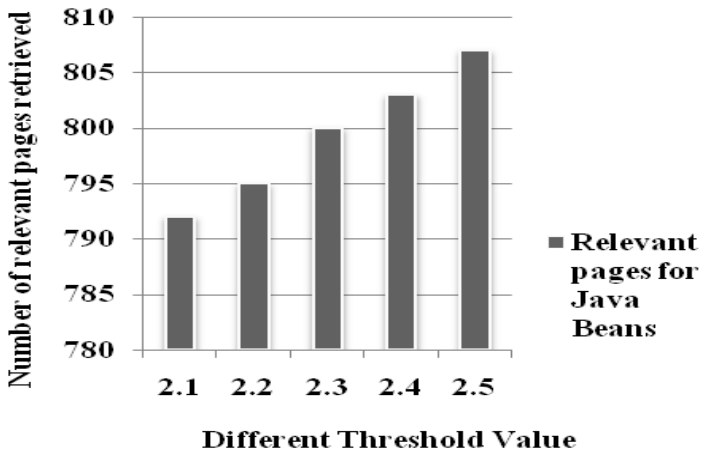

Fig. 2. Result for the topic 'Java Beans' on different threshold values

V. Conclusion

In this paper, the crawling method using the link, text content and probability method is more efficient and produces more relevant pages. The probability of finding the similar and dissimilar keywords makes the crawler to retrieve the pages effectively. For experimentation, several topic keywords were given and verified that the more relevant pages were retrieved. The experiment was compared with different threshold values to test the efficiency. The co-efficient factor of each method in the relevancy score computation was also tested with different values. Testing based on these different cases, it was proved that the proposed algorithm was more effective in retrieving the relevant pages efficiently.

References

[1] T.H. Have;owa;a Topic-Sensitive PageRamk, Proceedings of the 11th World Wide Web conference, pp.517-526.
[2] Blaž Novak, Survey of focused web crawling algorithms, in Proceedings of SIKDD, pp. 55-58, 2004.
[3] Shalin shah, Spe 2006. Implementing an Effective Web Crawler. 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.
[4] 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.
[5] A. Rungsawang, N. Angkawattanawit, Learnable topic-specific web crawler. Journal of Network and Computer Applications, Issue no:28,page no:97-114,2005
[6] 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.
[7] 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.
[8] S. Chakrabarti, M. van den Berg, and B. Dom, Focused Crawling: A New Approach for Topic-Specific Resource Discovery, In Proc. 8th WWW, 1999.
[9] A. Rungsawang, N. Angkawattanawit, Learnable Crawling: An Efficient Approach to Topic-specific Web Resource Discovery, 2005.
[10] K.Bharat and M.Henzinger, Improved Algorithms for Topic Distillation in a Hyperlinked Environment, In proc. Of the ACM SIGIR '98 conference on Research and Development in Information Retrieval.
[11] J.Jayanthi, Dr. K.S. Jayakumar, An Integrated Page Ranking Algorithm for Personalized Web Search, International Journal of Computer Applications, 2011.
[12] Lili Yan, Wencai Du, Yingbin wei and Henian chen, A novel heuristic search algorithm based on hyperlink and relevance strategy for Web Search, 2012, Advances in Intelligent and Soft Computing.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] M. Najork and J.L. Wiener. "Breadth-first search crawling yields high-quality pages", In Proceedings of the Tenth Conference on World Wide Web, Hong Kong, Elsevier Science, May 2001, pp. 114–118.
[18] Lixin Han and Guihai Chen, The HWS hybrid web search , Information and Software Technology, Elsevier Science, 2005.
[19] Rodriguez-Mier. P, Mucientes. M, Lama.M. Automatic Web service Composition with a Heuristic-based Search Algorithm, IEEE International Conference on Web Services, 2011.

[20] 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.
[21] Lili Yan, Wencai Du, Yingbin wei and Henian chen, "A novel heuristic search algorithm based on hyperlink and relevance strategy for Web Search", 2012, Advances in Intelligent and Soft Computing.