Research Article

iMedPub Journals
http://www.imedpub.com/

American Journal of Computer Science and Information Technology
ISSN 2349-3917

2017
Vol.5 No.2:07

DOI: 10.21767/2349-3917.100007

A Novel Approach on Focused Crawling with Anchor Text

S. Subatra Devi*

Assistant Professor, Hindustan College of Arts & Science, Chennai, Tamil Nadu, India

***Corresponding author**: S. Subatra Devi, Assistant Professor, Hindustan College of Arts & Science, Chennai, Tamil Nadu, India, E-mail: subesh_niru@yahoo.co.in

Received date: May 03, 2017; **Accepted date:** October 27, 2017; **Published date:** November 25, 2017

Citation: S. Subatra (2017) A Novel Approach on Focused Crawling with Anchor Text. Am J Compt Sci Inform Technol 5: 2. doi: 10.21767/2349-3917.100007

Abstract

Title: A novel approach with focused crawling for various anchor texts is discussed in this paper.

Background: Most of the search engines search the web with the anchor text to retrieve the relevant pages and answer the queries given by the users. The crawler usually searches the web pages and filters the unnecessary pages which can be done through focused crawling. A focused crawler generates its boundary to crawl the relevant pages based on the link and ignores the irrelevant pages on the web.

Methods and findings: In this paper, an effective focused crawling method is implemented to improve the quality of the search. Here, three learning phases are considered namely, content-based, link-based and sibling-based learning are undergone to improve the navigation of the search. In this approach, the crawler crawls through the relevant pages efficiently and more relevant pages are retrieved in an effective way.

Conclusion: It is proved experimentally that more number of relevant pages are retrieved for different anchor texts with three learning phases using focused crawling.

Keywords: Focused crawler; Hyperlink; Anchor text; Sibling; World wide web

Introduction

The World Wide Web grows exponentially with huge information and that leads to a great demand for developing an efficient and effective method in retrieving the information available on the web. Pant et al. [1-32] presents a general framework to evaluate topical crawlers. Topical crawlers, also known as topic driven or focused crawlers, are an important class of crawler programs that complement search engines. The focused crawling leads to significant savings in hardware and network resources, and helps in keeping the crawl more up-to-date [12]. The main job of the focused crawler is to retrieve the relevant pages that satisfy the anchor text. Focused crawler decides which URL to be selected based on the previously downloaded pages.

This paper illustrates the three learning phases applied with the focused crawling method. The three learning phases includes the content-based, link-based and the sibling-based learning. Applying these three phases, the crawler penetrates through the relevant pages. All the relevant pages and the relevant links are grouped as a set S1 and similarly all the irrelevant pages and irrelevant links are grouped as the set S2. If a parent URL consists of more number of outgoing links, then the sibling-based learning is used for determining the relevancy. This process is continued until the URL queue of downloaded pages gets empty or until the threshold value is reached.

The paper is organized as follows. In the next section, the works related to this paper are discussed. Section 3 discusses the novel method of web crawling. Section 4 describes the implementation of the approach. Section 5 specifies the various experimental results and the performance evaluation. Finally, the conclusion of the proposed method is given in Section 6.

Related Works

Focused Crawling was first developed by S. Chakrabarti in 1999 [12]. Focused crawlers [12,18] searches and retrieves only the subset of the web pertaining to a specific topic of relevancy. P. De. Bra [6] designed one of the first search algorithms called the Fish Search based on the content of individual nodes or documents and was improved in 2005 [23]. The Shark-search algorithm [16] was proposed in 1998 and was improved [10] in 2007. A focused crawler is implemented associating a score with each link [8,17] in the pages that it has downloaded. The algorithm that builds a model for the context within which topically relevant pages occur on the web [19]. A web search algorithm based on hyperlinks and content [9] relevance strategy performs well in topic-specific crawling. The focused crawler [15] of a special-purpose search engine aims to selectively seek out pages that are relevant to a pre-defined set of anchor texts, rather than to exploit all regions of the web. A method using a decision tree on anchor texts of hyperlinks [22] is used to utilize the anchor texts for determining the priorities.

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 [31]. A crawler must choose carefully at each step which pages to visit next. The novel concept of intelligent crawling [21] learns the characteristics of the linkage structure while crawling. To find the related pages in the web graph is to examine the siblings [29] of a starting node in the web graph. The likelihood of linked pages having similar textual content [30] finds to be high, the similarity of sibling pages increases when the links from the parent page are close together.

To efficiently compute the page rank for large numbers of pages, the method [14] considers the link structure of the web to produce a global importance ranking of every web page based on the graph of the web. A latent semantic indexing classifier [1] combines link analysis with text content in order to retrieve and index domain specific web documents.

Web crawlers are used to create a copy of all the visited pages [11] for later processing by a search engine, which will index the downloaded pages to provide fast searches. Link score is calculated based on average relevancy score of parent pages and division score [2]. An algorithm is used to find a minimal composition [5] that satisfies the user request. The web contains information on many related and unrelated topics. Such growth and fluctuation generates essential limits of scale for today's generic search engines [7]. Ranking based methods are applied in [3,4]. A review of the techniques of focused crawling is discussed by Blaz Novak [25].

A comprehensive analysis and critical comparison of various link-based similarity measures and algorithms are presented [27]. The method [26] discovers an efficient and better system for mining the web topology to identify authoritative web pages. The algorithm [28] efficiently utilizes the link and textual analysis, in which the four measures namely link-based measure, logarithmic distance measure, text content similarity and probabilistic measure are used to find the relevancy of the web pages.

The removing of stop words [33,34] is necessary because it reduces indexing file size and provides spin-off advantage for index compression. Bacchin et al. [35] propose and evaluate a statistical graph-based algorithm for stemming. Various approaches of focused web crawlers [36-40] are undergone. A design of distributed semantic web crawler is presented [41-48] to make crawling decisions.

Proposed Methodology

The seed page is the most important page for extracting the relevant information. This seed URL is extracted from the web by inputting the anchor text to the most popular search engines Google, Yahoo and MSN. Then, the resulting URLs that are common in any two or three of these search engines are considered to be relevant to the query, are taken as seed URL and placed in the URL queue. This URL is given as the input URL of the proposed algorithm. The most relevant URL is considered only with the text document discarding the video and image files. The most relevant URL is considered as seed URL since this leads to more number of most relevant pages. Three learning phases namely, content-based, link-based and sibling-based are undergone to predict the relevancy of the target pages. The seed URL is given as input to the crawler which crawls using focused crawling. Then, the outgoing links are obtained from the seed URL utilizing hypertext analysis. The working procedure of the proposed algorithm comprises the following steps: (1) Input the seed URL for the anchor text (2) Extract the outgoing links of the seed URL (3) Compute the relevancy score for each URL (4) Compare the relevancy score with the predefined threshold value (5) Select the outgoing link if the relevancy score is higher than the threshold value and (6) Place the link in the URL queue. These steps are reiterated until the URL queue gets empty.

The relevancy of the page is checked with the three learning phases and the focused crawler determines the relevant and irrelevant pages and groups the pages into two sets 'S1' and 'S2' respectively.

Procedural steps of the algorithm

The procedural step of the proposed algorithm is given in Table 1.

Table 1: Procedural Steps of the Algorithm.

Input: Input the seed URL u and the anchor text k. Initialize N, the co-efficients w1, w2 and w3 and the threshold value Tr.
Output: Set of outgoing links, O (URL).
1. Input the anchor text k and the number of pages to be crawled, N.
2. Input the seed URL u extracted for the anchor text.
3. Place the seed URL into the URL queue.
4. Perform steps 5 to 9 until URL queue is empty.
5. For each and every web page 'i' in the URL queue
6. Compute the learning phases:
6.1 Compute the content-based learning as Lc (C)={v/Ki Wi , v V }
6.2 Compute the link-based learning as Ll (C)={(a,b)/ti Ki , a S1 & b S1}
6.3 Compute the sibling-based learning as Ls (C)={v/v sibling of a crawled page}
7. Compute the Relevancy score Rs as Rs=w1 *Lc (C)+w2 *Ll (C)+w3 *Ls (C)
8. Select the outgoing link 'i', if the relevancy score Rs is greater than Tr.
9. Place the outgoing links in the URL queue.

The co-efficient w1, w2 and w3 are initialized with the constants for a balanced result and the threshold value Tr is assigned with a predefined value at which more numbers of relevant pages can be retrieved. From the seed URL, the outgoing links are extracted. For each outgoing link, the algorithm is applied and the relevancy score is computed.

Assumptions

A web graph is denoted as G=(V, E) for a particular domain, where V is the set of web pages and E is the set of hyperlinks between the web pages. Initially, crawl all the pages in G with the web pages V through the edges E and find the relevant pages and the irrelevant pages in the graph using the learning

phases. All the relevant pages are determined and the path is generated from the root node to the leaf node. The union of all these relevant pages and the relevant edges in the generated path is stored in 'S1' and the irrelevant pages, the irrelevant edges are stored in 'S2'.

A function F is represented as F(v)=true, if v is classified as a relevant page and F(v)=false, if v is classified as an irrelevant page. For each page 'p'in the graph, such that p Є {v/F(v)=true, v V}, which are the relevant pages, are fetched and all those pages are stored as a set 'P1'. Here, the irrelevant pages are also determined using p Є {v/F(v)=false, v V}. These irrelevant pages are stored in the set'P2'.

Similarly, the edges between the relevant pages are also considered. Let 'l' be the hyperlink between the pages denoted by l={(a, b) E} , where a and b denotes the source and target pages of the link 'l' respectively. The hyperlink 'l' consists of the relevant and the irrelevant links. For relevancy we denote the expression as, l Є {(a, b) E, a P1 & b P1}. These relevant hyperlinks are stored in 'L1'. A hyperlink is considered as an irrelevant link with the expression l Є {(a, b) E, a P1 & b P2}. The irrelevant hyperlinks are grouped in 'L2'.

In general, 'S1' is a set of 'P1' and 'L1' i.e., it contains all the relevant pages and the relevant links S1 = {P1, L1}

'S2' is a set of elements of 'P2' and 'L2' i.e., it contains all the irrelevant pages and the irrelevant links S2 = {P2, L2}

Decision graph

A focused crawler collects the web pages using the learning phases and prioritizes the pages and manages the hyperlink. It predicts the probability that an unvisited page is relevant or irrelevant before downloading the page. This is determined with the help of link-based learning and sibling-based learning.

The internet link structure is represented as a directed graph G=(V, E); the nodes V correspond to the pages and a directed edge (a, b) ϵ E indicates the presence of a link from web page 'a' to web page 'b' of hyperlinked pages. A sample web link graph is shown in Figure 2 to implement the web crawling algorithm. The nodes a, b, c, d, e, f, g, h, i and j represent the web pages and the links between the web pages represent the hyperlink in the web. Since each node contains more numbers of outgoing links, only few of the outgoing links are considered in the following Figure 1.

The link graph is used as a base model for determining the relevancy score. Using the web link graph, the learning phases are determined to compute the page importance. It consists of the set of relevant pages and the set of irrelevant pages along with their hyperlinks. That is, it contains the sets S1 and S2. S1 represents the set of crawled relevant pages. S2 represents the set of crawled irrelevant pages. In the above Figure, the node 'a' is the seed URL; 'b', 'c', 'e', 'f' and 'g' represents the set of crawled relevant pages; 'd', 'h' ,'i' and 'j' represents the set of crawled irrelevant pages.

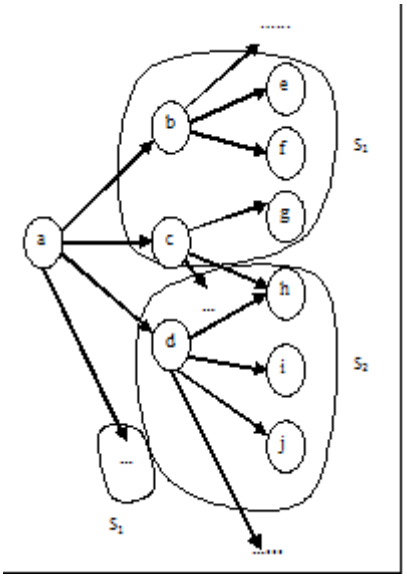

Figure 1: Decision Graph Specifying the Relevant and Irrelevant Pages.

Learning phases

The three learning phases are discussed below by which the relevancy of the page is determined.

Content-based learning: In content-based relevance, the page content of each web page V is checked for relevancy. From the page content, the keywords are extracted. Let W1, W2, W3, …., Wn be the set of words present in the page. If Ki represents the anchor text keywords, then Ki is compared with each Wi for existence. If so, then the crawler's content-based learning is determined as

Lc (C)={v / Ki Wi , v V}

Where v represents the set of vertices that satisfies the page content for the given keywords.

The set of web pages that satisfy the relevancy are stored in the set 'P1' and the irrelevant pages are stored in the set 'P2'.

Link-based learning: In Link-based learning, the URL is checked for relevancy. The URLs contain the tokens. The set of all tokens in the URL is specified as ti={ t1, t2, …., tn}. To fetch the tokens from the URLs, the URLs are first parsed with the separators "." and "/". These tokens are compared with the anchor text keywords Ki for existence. If one or more of the keyword Ki is present in the link, then the particular link is considered as the relevant link. If all the keywords are present, then it is determined as more relevant link, and it is given more priority.

If no anchor text is present in the URL link, then it is considered as an irrelevant link. All the relevant links are grouped to 'L1' and the irrelevant links are grouped to 'L2'. Then the crawler's link-based learning is determined as

Ll (C)={(a,b) / ti Ki ^ a S1 ^ b S1} where (a, b) is the set of hyperlinks containing the keyword Ki in the topic ti .

Sibling-based learning: The sibling of a page is determined whether the page is a relevant or an irrelevant page. If a parent URL has more number of child URLs, and if most of the child URLs satisfies the content–based learning, then the siblings is also considered as relevant pages before crawling.

Ls (C)={v / v sibling of a crawled page}

Here, if the crawler considers most of the crawled pages as relevant pages, then their siblings are considered as the relevant pages.

Relevancy Evaluation: The aggregate of these learning phases are formed by summing the weights of the individual relevance learning.

Relevancy-Score=w1 *Lc(C)+w2 *Ll(C)+w3 *Ls(C)

Here w1, w2 and w3 represents the weights which are used to normalize the different factor values. The values of these weights can be increased, to increase the importance of the individual factors. Here the implementation is made such that the weights are equally balanced and to get a balanced relevancy-score for each web page.

Implementation

In this section, the experimentation results of the proposed web crawling method are presented using the content, link and sibling-based learning phases. The proposed algorithm was implemented using java and the experimentation was performed with different anchor text. The efficiency of the proposed method was determined by comparing the relevant retrieved pages using focused crawling and without focused crawling. The tokens are extracted from the web pages using the text mining processes such as, stop word removal and stemming method.

The anchor text k is assigned as 'Computer science books' and the co-efficient are initialized as w1=0.1, w2=0.2 and w3=0.3. The seed URL in initialized as u="http://www.freetechbooks.com/" since the URL is common in three search engines Google, Yahoo and MSN. The value of N, the number of pages to be crawled is initialized with the different values varying from 2000 to 12000. Figure 2 shows the creation of folders and sub folders from the seed URL in the base path for each and every level of crawl. The output URLs are stored as folders in "E:\" drive starting with the base folder f0-0. So, the base path is initialized as "E:\f0-0" where the folders are created one by one during crawling based on the number of http links from the seed URL u.

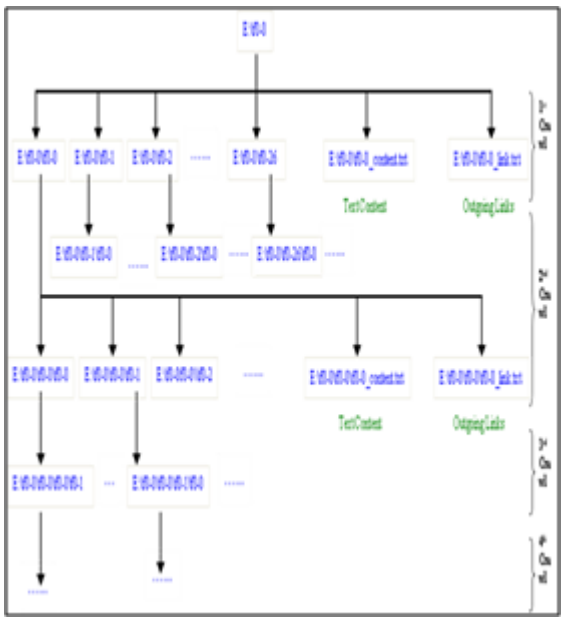

Figure 2: Creation of Folders from the Seed URL.

Two text files are created one containing the text content of the parent page and the other containing the outgoing links of the parent page. The crawler crawls into the web taking 'u' as the seed URL. The text content of the parent page is stored in the text file as "E:\f0-0\f0-0_content.txt". The outgoing links of the parent page are fetched and stored into the text file as "E:\f0-0\f0-0_link.txt". From the seed URL 'u' as initialized, there are 27 outgoing links during the first level of crawl and hence 27 folders are created in the base path from "E:\f0-0\f0-0" to "E:\f0-0\f0-26" and along with it the content and the link files are also created. Now considering these 27 outgoing links as the parent URLs, its child URLs are fetched and the sub folders are created on these path. The relevancy score is computed for each and every web page in the link file based on the relevant keywords present in the content file. The relevancy score is calculated for the 27 outgoing links. Then, the relevancy score for each of these links is compared with the predefined threshold value. The threshold value is the limit at which more numbers of relevant pages are retrieved. If the relevancy score is more than the threshold value, then the URL link is accepted for the second level of crawl. Otherwise, the link (folder) is removed (deleted) from further crawling. For example the path "E:\f0-0\f0-9\f0-3\f0-4\f0-20" shows the creation of sub folders for the seed URLs 10th outgoing link during the fourth level of crawl. This shows the fetching of the web page content and its URL link from the web. When this process is performed, the algorithm works on the web page content and the link file performing the procedural steps as described to fetch the relevant pages.

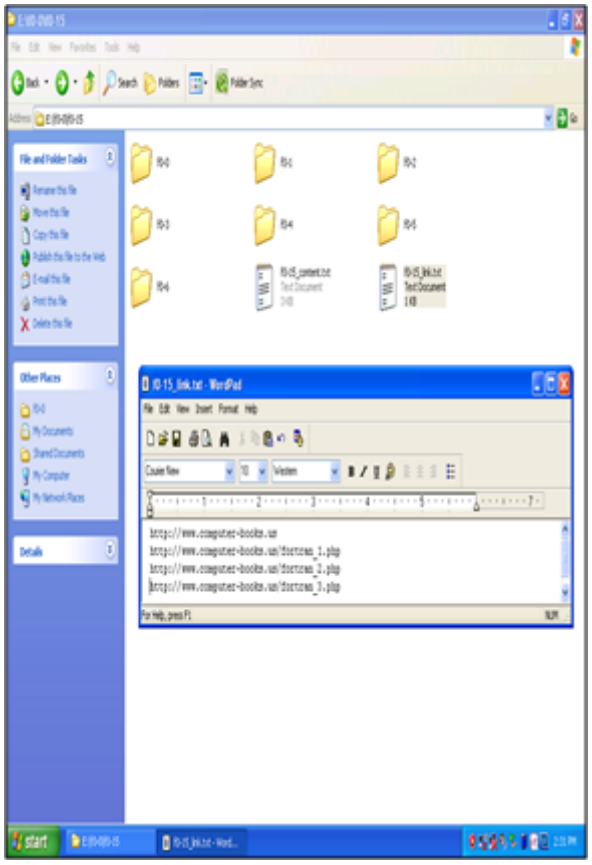

Figure 3: Stored Output URLs.

All the retrieved relevant pages and retrieved relevant links are stored in 'P1' and 'L1' and the retrieved irrelevant pages and retrieved irrelevant links are stored in 'P2' and 'L2', which are assigned to the sets 'S1' and 'S2'.

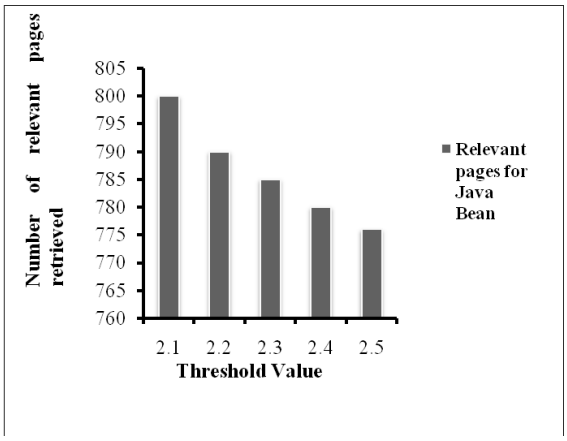

Figure 4: Number of Relevant Pages Retrieved for the Topic 'Java Bean' on Different Threshold Values.

The URL links in the text document f0-15 is shown in Figure 3. The existing folders f0-0, f0-1, f0-2, f0-3, f0-4, f0-5 and f0-6 indicate the 'retrieved pages' during the crawl. The omitted folders f0-7 to f0-15 indicates the 'not retrieved pages' during the crawl. The 'retrieved pages' comprise the 'relevant pages' and the 'irrelevant pages'. The URL links stored in the text document are checked if it contains the anchor text.

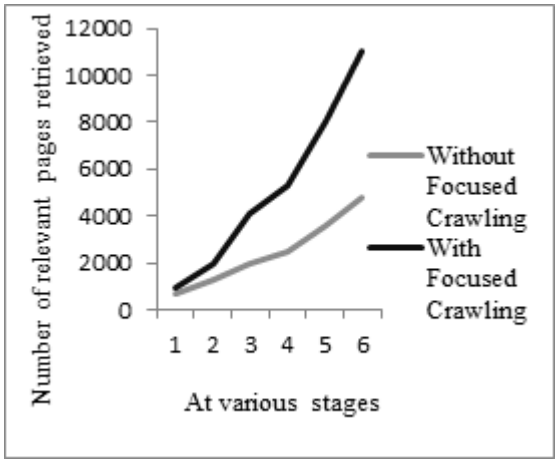

Figure 5: The performance of the Crawler for the Anchor Text 'J2EE'.

If so, the link is considered as a 'relevant page', else the link is considered as an 'irrelevant page'. Similarly, the 'not retrieved pages' contains the 'relevant pages' and the 'irrelevant pages'. The 'not retrieved pages' are stored as separate folders in different base path. The URL links stored in these folders are checked if it contains the anchor text. If so, the link is considered as a 'relevant page', else the link is considered as an 'irrelevant page'.

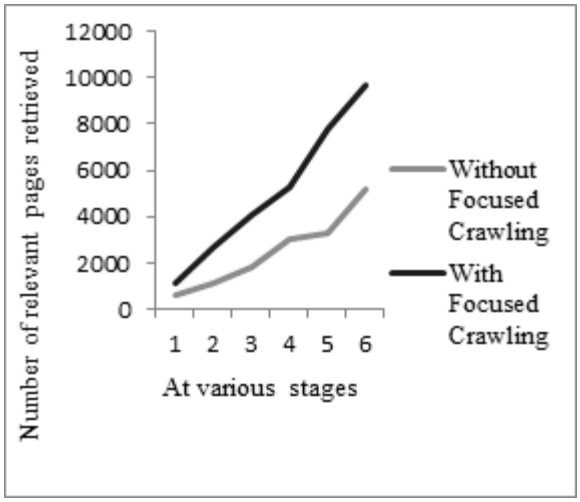

Figure 6: The performance of the Crawler for the Anchor Text 'Java Script'.

The set of relevant retrieved pages are stored in the set 'P1' and the irrelevant retrieved pages are stored in the set 'P2'.The output storage for the URL u="http://www.freetechbooks.com/" is shown in Figure 3.

Experimental Results

The keyword k is given with different anchor texts as 'Java Bean', 'J2EE', 'Java Script', 'Java AWT' and 'Java Swing'. For different topics, the numbers of pages retrieved by the proposed algorithm with focused crawling and without focused crawling are determined. For the topic 'Java Bean', the seed URL u is initialized as "http://docs.oracle.com/javase". The constants are initialized as w1 = 0.2, w2 = 0.2 and w3= 0.3. For different threshold values, the numbers of relevant pages retrieved are determined. These are determined by the count of the total URL links stored in the link files. If the particular URL contains the anchor text, then the URL is considered as relevant URL. Otherwise, the URL is considered as an irrelevant URL. Some of the URLs that are fetched and stored in the different path folders are shown in Table 2 when the threshold value is 2.1.

Table 2: Output URLs Stored in Different Folders on the Topics for the Threshold Value 2.1

http://docs.oracle.com/javase/tutorial/javabeans/quick/index.html
http://docs.oracle.com/javase/tutorial/javabeans/quick/project.html
http://docs.oracle.com/javase/tutorial/javabeans/quick/button.html
http://docs.oracle.com/javase/tutorial/javabeans/quick/wiring.html
http://docs.oracle.com/javase/tutorial/javabeans/quick/addbean.html
http://docs.oracle.com/javaee/7/tutorial/doc/partsupporttechs.htm#GIJUE
http://docs.oracle.com/javaee/7/tutorial/doc/cdi-basic.htm#GIWHB
http://docs.oracle.com/javaee/7/tutorial/doc/transactions.htm#BNCIH
http://docs.oracle.com/javaee/7/tutorial/doc/partcdi.htm#GJBNR
http://docs.oracle.com/javaee/7/tutorial/doc/security-advanced.htm#GJJWX
http://www.w3schools.com/js/tryit.asp?filename=tryjs_create_object
http://www.w3schools.com/js/tryit.asp?filename=tryjs_formattext
http://www.w3schools.com/js/tryit.asp?filename=tryjsref_doc_anchors
http://www.w3schools.com/js/tryit.asp?filename=tryjsref_doc_anchors2
http://www.w3schools.com/js/tryit.asp?filename=tryjsref_doc_getelementsbytagname
http://www.w3schools.com/js/tryit.asp?filename=tryjsref_doc_open2
http://docs.oracle.com/javase/7/docs/api/java/awt/image/ColorModel.html
http://docs.oracle.com/javase/7/docs/api/java/awt/color/package-summary.html
http://docs.oracle.com/javase/7/docs/api/java/awt/dnd/package-summary.html
http://docs.oracle.com/javase/7/docs/api/java/awt/datatransfer/package-summary.html
http://docs.oracle.com/javase/tutorial/uiswing/examples/components/HtmlDemoProject/src/components/HtmlDemo.java
http://docs.oracle.com/javase/tutorial/uiswing/examples/components/ButtonHtmlDemoProject/src/components/ButtonHtmlDemo.java
http://docs.oracle.com/javase/tutorial/uiswing/examples/components/TextDemoProject/src/components/TextDemo.java

The number of relevant pages retrieved for the different threshold values for the topic 'Java Bean' is specified in Figure 4. The number of relevant pages retrieved increases gradually as the threshold value decreases.

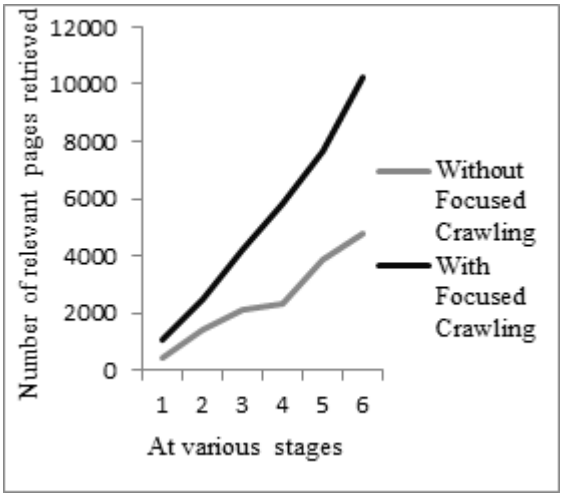

Figure 7: The performance of the Crawler for the Anchor Text 'Java AWT'.

The experimentation output at various stages is shown in the following figures (Figures 5-8). The figure shows the output for the anchor texts 'J2EE' 'Java Script', 'Java AWT' and 'Java Swing'. The algorithm is tested using various anchor texts and it shows that the proposed method yields more number of relevant pages with focused crawling.

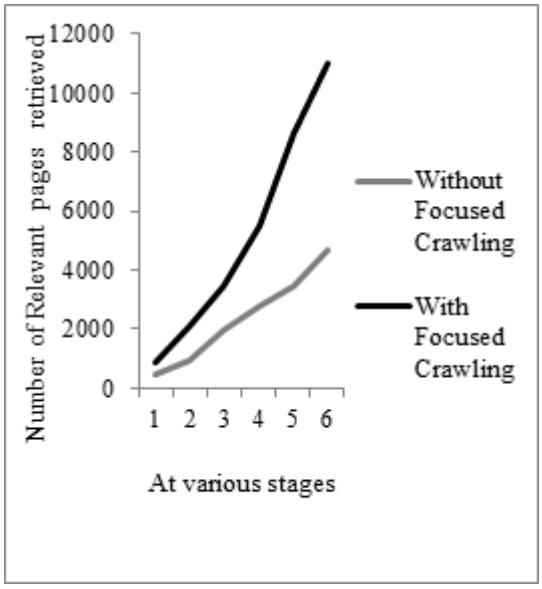

Figure 8: The performance of the Crawler for the Anchor Text 'Java Swing'.

The outcomes such as numbers of relevant pages retrieved, relevant pages not retrieved, irrelevant pages retrieved and irrelevant pages not retrieved are determined for the different anchor texts and the metrics are computed as shown in Table 3 for the algorithm with focused crawling and without focused crawling. It shows that for each and every topic, the precision, recall and accuracy of the proposed algorithm is higher with focused crawling.

Table 3: Metrics of the Algorithm for Different Anchor texts.

Metrics	Anchor texts	Precision (percent)	Recall (percent)	Accuracy (percent)
Algorithms				
With focused crawling	Java Bean	80	84.2	77
	J2EE	78.3	82.2	74.1
	Java Script	76.9	87.1	75
	Java AWT	70.6	75.3	72.8
	Java Swing	75.1	80	76.6
Without focused crawling	Java Bean	71.7	78.4	69.3
	J2EE	69.3	79.7	65.4
	Java Script	64.8	81.2	64
	Java AWT	65.4	70.9	60.1
	Java Swing	63.8	77.6	65.4

Conclusions and Future Work

In this paper, a technique with focused crawling using three learning phases namely, content-based, link-based and sibling-based learning are discussed to generate the more number of relevant pages. It is proved experimentally that more number of relevant pages is retrieved for different anchor texts using focused crawling. The comparisons of the results are done with focused crawling and without focused crawling. Here only the text document is considered for experimentation. The future

work is to retrieve more number of relevant pages for the documents other than texts namely, images, videos etc. The performance of the algorithm is to be tested with a large volume of web pages. Further extensions can be done on this work by analyzing and proposing semantic queries.

References

1. Almpanidis G, Kotropoulos C (2007) Combining Text and Link Analysis for Focused Crawling, An Application for Vertical Search Engines, Information Systems 326: 886-908.
2. Hati D, KumarA (2010) An Approach for Identifying URLs Based on Division Score and Link Score in Focused Crawler, Int J Computer App 2: 48-53.
3. Sudhaka P, Poonkuzhali G, Kishore KR (2012) Content Based Ranking for Search Engines, Proceedings of the International Multi Conference of Engineers and Computer Scientists.
4. Deore AD, Paikrao RL (2012) Ranking Based Web Search Algorithms, Int J Sci Res, 2
5. Rodriguez-Mier P, Mucientes M, Lama M (2011) Automatic Web service Composition with a Heuristic-based Search Algorithm, IEEE International Conference on Web Services.
6. Bra DP, Houben G-J, Kornatzky Y, Post R (1994) Information Retrieval in Distributed Hypertexts, Proceedings of RIAO '94, Intelligent Multimedia, Information Retrieval Systems and Management, New York. 481-491
7. Yongsheng Y, Hui W(2011) Implementation of Focused crawler, J Computer 6: 1.
8. Bharat K, Henzinger M (1998) Improved Algorithms for Topic Distillation in a Hyperlinked Environment, Proceedings of ACM SIGIR '98 conference on Research and Development in Information Retrieval 104-111
9. Yan L, Wencai D, Yingbin W, Henian C (2012) A Novel Heuristic Search Algorithm based on Hyperlink and Relevance Strategy for Web Search, Advances in Intelligent and Soft Computing, Springer, 149: 97-102
10. Chen Z, Ma J, Jingsheng L, Yuan B, Lian L (2007) An Improved Shark-Search Algorithm Based on Multi-information, Fourth International Conference on Fuzzy Systems and Knowledge Discovery.
11. Shah S (2006) Implementing an Effective Web Crawler.
12. Chakrabarti S, Van BM, Dom B (1999) Focused Crawling: A New Approach for Topic-Specific Web Resource Discovery, Elsevier Science.
13. Patel A, Schmidt N (2011) Application of structured document parsing to focused web crawling, Computer Standards & Interfaces, Elsevier. 33: 325-331
14. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine, in Computer Networks and ISDN Systems, 30(1-7), Elsevier. 107-117
15. Anshika P, Deepak ST, Shrivastava SC (2009) Effective Focused Crawling Based on Content and Link Structure Analysis, International Journal of Computer Science and Information Security. 2
16. Hersovici M, Jacovi M, Maarek Y, Pelleg D, Shtalheim M, et al. (1998) The Shark-Search Algorithm-an Application: Tailored Web Site Mapping, Computer Networks and ISDN Systems. Special Issue on the Seventh International World-Wide Web Conference, Brisbane, Australia, 30: 317-326
17. Kleinberg J (1998) Authoritative sources in a hyperlinked environment, proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms. 668-677
18. Cho J, Garcia-Molina H, Page L (1998) Efficient Crawling through URL Ordering, in Proceedings of the Seventh World-Wide Web Conference, Elsevier Science 161-172
19. Diligenti M, Coetzee F, Lawrence S, Giles C, Gori M (2000) Focused crawling using context graphs, Proceedings of 26th International Conference on Very Large Data Bases.
20. Najork M, Wiener J (2001) Breadth-first search crawling yields high-quality pages. In 10th Int. World Wide Web Conference, Hong Kong, ACM.114-118
21. Aggarwal C, Al-Garawi F, Yu P (2001) Intelligent Crawling on the World Wide Web with Arbitrary Predicates, Proceedings of the 10th International World Wide Web Conference, Hong Kong 96-105
22. Jun Li, Kazutaka F, Yamaguchi K (2005) Focused Crawling by Exploiting Anchor Text Using Decision Tree, 14th International Conference on WWW '05, ACM.
23. Fang-F Guo-long C, Wen-Zhong G (2005) An improved Fish- Search Algorithm for Information Retrieval, IEEE International Conference on Natural Language Processing and Knowledge Engineering, 523-528.
24. Tao P, Fengling H, Wanli Z (2006) A new Framework for Focused Web Crawling, Wuhan University Journal of Natural Science (WUJNS).
25. Blaz Novak (2004) A Survey of focused Web Crawling Algorithms, Proceedings of SIKDD, 55-58.
26. Jain A, Sharma R, Dixit G, Tomar V (2013) Page Ranking Algorithms in Web Mining, Limitations of Existing methods and a New Method for Indexing Web Pages, International Conference on Communication Systems and Network Technologies, 640-645.
27. Liu H, Jun He, Dan Zhu, Charles XL, Xiaoyong Du (2013) Measuring Similarity Based on Link Information: A Comparative Study, IEEE Transactions on Knowledge and Data Engineering, 25: 2823-2840.
28. Subatra DS, Sheik AKP (2013) A Comparative Study of Four Measures on Web Information Retrieval, Recent Science publications, International Journal of Internet and Web Technology, 38: 1107-1112.
29. Dean J, Monika RH (1999) Finding Related Pages in the World Wide Web, 8th World Wide Web Conference, Elsevier Science, 1467-1479.
30. Davidson BD (2000) Topical Locality in the Web, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 272-279.
31. Pant G, Srinivasan P, Menczer F (2004) Crawling the Web, Springer. 153-178.
32. Pant G, Srinivasan P, Menczer FA (2005) General Evaluation Framework for Topical Crawlers, Information Retrieval, Springer, 8: 417-447.
33. Wang Z (2004) Improved Link-Based Algorithms for Ranking Web Pages, Springer. 291-302.
34. Narayana VA, Premchand P, Govardhan A (2009) Effective Detection of Near Duplicate Web Documents in Web Crawling,

International Journal of Computational Intelligence Research 5: 83–96.

35. Bacchin M, Ferro N, Melucci M (2002) The Effectiveness of a Graph-Based Algorithm for Stemming, Proceedings of the 5th International.

36. Shexuebing, Fulei (2013) Keyword Extraction Algorithm Based on, International Conference on Computational and Information Sciences.

37. Gunjan HA, Nikita VM (2015) Keyword focused web crawler, International Conference on Electronics and Communication Systems.

38. Sharma S, Gupta P (2015) The anatomy of web crawlers, International Conference on Computing, Communication and Automation, IEEE.

39. Bai S, Hussain S, (2015) A framework for focused linked data crawler using context graphs, International Conference on Information and Communication Technologies. 1-6.

40. Gupta A, Anand P (2015) Focused web crawlers and its approaches, International Conference on Futuristic Trends on Computational Analysis and Knowledge Management.

41. Kumar N, Manjeet S (2015) Framework for Distributed Semantic Web Crawler, International Conference on Computational Intelligence and Communication Networks.

42. Gaur R, Sharma DK (2014) Review of ontology based focused crawling approaches, International Conference on Soft Computing Techniques for Engineering and Technology, IEEE.

43. Bai S, Hussain S, Khoja S (2015) A framework for focused linked data crawler using context graphs, International Conference on Information and Communication Technologies, IEEE.

44. Sharma DK, Khan MA (2015) SAFSB: A self-adaptive focused crawler, 1st International Conference on Next Generation Computing Technologies.

45. Deri L, Martinelli M, Sartiano D, Sideri L (2015) Large scale web-content classification, 7th International Joint Conference on Knowledge Engineering and Knowledge Management.

46. Gaur R, Sharma DK (2014) Review of ontology based focused crawling approaches, International Conference on Soft Computing Techniques for Engineering and Technology.

47. Goyal D, Kalra M (2014) A novel prediction method of relevancy for focused crawling in topic specific search, International Conference on Signal Propagation and Computer Technology, IEEE.

48. Bhardwaj A, Mangat V (2014) A novel approach for content extraction from web pages, Recent Advances in Engineering and Computational Sciences, IEEE.