A REVIEW ON WEB PAGE CHANGE DETECTION TECHNIQUES

Surbhi Chhabra
Research Scholar, Department of Computer Science &Applications, Kurukshetra University,
Kurukshetra-136119, Haryana, India
Email: f20.chhabra@gmail.com
Rajender Nath
Professor, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra-136119, Haryana, India
Email: rnath2k3@gmail.com

---Abstract---

The World Wide Web is growing exponentially adding billions of web pages daily. Most of the information on the Web is dynamic in nature. Web crawler crawls the new as well as the web pages which have changed since the last crawl and index them in the search engine database. In order to increase the efficiency, the web crawlers need to detect the changes in the web pages so that only those pages which have changed need to be index in the search engine. So, to do that many page changes detection techniques have been proposed in the literature which is systematically analyzed in this paper. Many weaknesses in the techniques are found and future research directions are drawn.

Keywords: Change Detection, Search Engine, Web Crawler, Web Page, WWW

1. Introduction

WWW (World Wide Web) contains billions of pages and billions of Internet users worldwide. Search engines are used to search information in an extensive WWW. Web crawlers are the information retrieval tool of search engine that crawl the data in the rapidly growing and changing web. It is a program that automatically traverses the web by following links of web page and downloads the pages for indexing. It starts with the seed URL, downloads the page corresponding to URL and extracts all the URLs from the downloaded pages and stores them in the queue. Then it picks the next URL from the queue and repeat the same process until the queue is empty. The information on the WWW is dynamic in nature. There is almost 60% of the web is dynamic. So the search engine should maintain their index up-to-date to provide fresh information to users. It is not possible for a web crawler to visit each and every page repeatedly to check whether a web page is changed or not. A lot of bandwidth is consumed if web crawler visits each and every web page. So, there should be a mechanism to detect change in web page. The crawler visit only those web pages that changes frequently. In literature, various techniques developed to detect change in web page that are categorised into three categories i.e. structural change, content change and hybrid.

The rest of the paper is organized as follows: Section 2 describes the three categorization of web page change detection. Section 3 describes various research works that has been done in web page change detection and weaknesses of various papers. Section 4 describes conclusion of the paper and future research directions in web page change detection

2. Research Methodology

Firstly, research papers relating to review the existing techniques of web page change detection will be collected from various sources such as IEEE, ACM, Elsevier, Science Direct, proceedings of International and National Conferences etc. Then these research papers will be classified into different broad categories. After that representative research papers under each category will be critically analyzed and research directions will be drawn.

3. Web page change detection Techniques

Total 18 research papers were collected and were classified into three broad categories (a) Structural change (b) Content change (c) Hybrid techniques. One research papers were found under the category of structural change, four research papers under content change and thirteen under hybrid techniques. The research papers under each category are discussed and analyzed below.

3.1 Page change detection techniques based on Web page Structure

Divakar Yadav et al. 2012 [1], proposed two techniques for structural change detection. The first technique was document tree based in which all the tags were extracted and arranged in the order of hierarchical relationship and document tree was constructed. Then level by level old page tree and modified page tree was compared to detect changes. Each node in the document tree contained various fields i.e. tag, child, level_no and no_of_siblings. The second technique was document fingerprint that generated two fingerprints. The first fingerprint contained first character of all the tags and second fingerprint contained last character of all the tags in the order they appeared in the web page.

Discussion- Weaknesses of these techniques are space and time complexities of the algorithms and difficulties in generating document tree due to misalignment and optional tags available in web page.

3.2 Page change detection techniques based on Web page Content

Sharma Chakravarthy et al. 2006 [2], designed Web Vigil, a system that automatically detect the changes at HTML/XML Pages and provided timely notification based on user specified change of interest. CH-DIFF and CX-DIFF change detection algorithms were developed to detect change in HTML and XML documents respectively. In CH-DIFF, change in any word can be detected using LCS (Longest Common Subsequence) algorithm. To detect change in any particular object, a formula was given S(A) - S(B) where S(A) be the set of objects extracted from old page and S(B) be the set of objects extracted from new modified page. In CX-DIFF, XML tree leaf nodes represented the content. The objects such as phrases, keywords were extracted from the leaf node and then signature was computed for each extracted leaf node of unique inserts/deletes was filtered and finds the common order subsequence between the leaf nodes of the given trees.

H. P. Khandagale et al. 2010 [3] proposed an algorithm for web page change detection. Signatures of root nodes of old page in repository and of modified page were compared to detect the change. Firstly, the HTML document was filtered into XML document and then XML document was transformed into trees using DOM. The algorithm assigned Hash value to all child nodes that were text nodes by using getHash( ) function. The signature was the function of hash value from the contents of the node. The summation of all the children node's signature was equal to the signature of interior node. The system was very useful for saving the browsing time and got result faster as it did not search the subtree if that subtree did not have any changes.

Vidushi Singhal et al. 2012 [4] proposed a formula of web page refresh policy for text based content. They assigned a code to the text content appeared in a web page word by word. The formula for text coding is (Summation of character frequency * UNICODE Symbol) / Distinct Symbol Count. UNICODE had developed a system in which each and every symbol can be coded. They compared the old page code with the new page code word by word till the first unmatched value rather than comparing whole page. When the single value was found to be different, it means the page needs to be updated. It saved the storage space and time. It worked for all the languages. It gave unique code for each and every content present on web page.

Hardik Trivedi et al. 2014 [5] designed a personalized focused crawler. They proposed a method for change detection of web pages that calculate the checksum of old page and new page. The formula for calculating Checksum is, Checksum (P) = ASCII Sum of Page Content /

Distinct Character Count. If checksum were same then there was no change in old page stored in database and new page. If checksum were different then old page was replaced with new modified page. It saved the storage space and time. It gave unique code for each and every content present on web page.

Discussion- Many limitations of page change detection techniques based on web page content are found. LCS algorithm that detects changes in words is very expensive in terms of computation. The Web Vigil system provides notification based on user's interest, so sentinels or user's request can be overloaded on the single server. The accuracy and efficiency of the node signature comparison algorithm is still unknown and needs to be calculated.

3.3 Hybrid Techniques

G. Cobena et al. 2002 [6] proposed a DIFF algorithm for detecting changes in XML Documents. It detected structural and content change. The algorithm found the most identical subtrees of old page and new page by using the ID attributes defined in the DTD. The two subtrees were matched using signatures. The signature was the function of a hash value that was computed using the nodes content and its children signature. The algorithm runs in linear time. It was efficient in terms of speed and memory space.

S. Flesca et al. 2002 [7] proposed a CMW system that provide a change monitoring service on the web. It detects changes in selected portions on the web documents by measuring the similarity between two documents. To find the similarity, three functions were used intersect() that returns the percentage of words that appear in both documents, attdist() measure the relative weight of attributes, typedist() was the difference between the complete types of element.

Latifur Khan et al. 2002 [8] proposed an effective algorithm that detect changes between old and modified version of XML document based on Signatures. Firstly, XML document represented as a tree, then by using top-down approach the signature value of root nodes of two versions were compared then signatures of children nodes were compared. Signature was the function of hash value. Signature of interior node was equal to XOR of its children nodes signatures.

Yuan Wang et al. 2003 [9] proposed X-Diff algorithm that detect change in XML document. The algorithm consists of three steps – Parsing and Hashing, Matching, Generating minimum-cost edit scripts. Edit scripts consists of a sequence of edit operations Insert, Delete, Update that convert one tree into another. It generated more accurate results.

Divakar Yadav et al. 2007 [10] proposed an algorithm for web page change detection that consists of three steps - document tree construction, document tree encoding and tree matching (based upon the R.M.S value of the content). It detects structural changes and content changes. It has linear time complexity. It reduced network traffic. It was simple and less cost.

Divakar Yadav et al. 2008 [11] proposed a parallel crawler architecture and web page change detection techniques. For page change, they proposed a three-step algorithm that detected page structure, text content and image changed or not. For change in page structure, an algorithm was given that include 2 phases – First the document tree was constructed, Second level-by-level tree parsed of old page and new modified page. For change in text content, a formula that calculated the RMS for page content i.e. RMS= [((a1)^2 + (a2)^2 + ------ (an)^2) / n] ^ ½ where ai was ASCII code of ith character. It had linear time complexity because it traversed only the changed portion of the tree rather than the whole tree and hence saved the time. It was simple, less cost and understandable.

Swati Mali et al. 2011 [12] designed a focused web crawler with web page change detection policy. They determined change in web page by detecting change in page structure at first level, change in text contents at second level, change in image at third level. Change in page structure was detected by creating two strings using the tags of that page. The first string stored the characters appearing at first position in the tag for all the tags in the order they appeared. The second string stored the characters that were at the last position in a tag. When the new page arrived, only these two strings were checked to determine a change in page structure. If the above method did not found any change. The change in text contents at second level was carried out. A code was assigned to all text contents appearing in a page. The formula for text coding was given by them. When a page was to be updated, text code of new page and old page was compared. The above two methods did not determined change in image. A code was given for images to determine whether they had undergone a change or not. Using the method, an Ival was calculated and stored. When the new page arrived, without actual download of image, the Ival was checked.

Srishti Goel et al. 2012 [13] proposed a new algorithm for web page change detection. It detects structural and content change. Tree was constructed by extracting tags, and then hash value was assigned to the leaf node. Hash function was equal to the node name and level no. For the non-leaf node tag value was assigned that was equal to the sum of hash value of its child. It was useful for saving the browsing time. It gave the user time to time update regarding the changes which will occur in a web page.

Naveen Kumar Varshney et al. 2013 [14] proposed an algorithm for web page change detection for structural change and content change. For structural change, the signature of nodes of old page and modified page are compared in top-down manner. The signature of node = Id value of that node + Branch count of all its descendant child’s. For content change, the formula used is Summation of (position value of ith character * ASCII value of ith character) / Distinct Character count. It was a simple method for detecting changes.

Neha Batra et al. 2013 [15] designed a technique to detect multiple change in a web page. This technique determined content change. Firstly, a tree was constructed from HTML page. The old page tree and modified page tree was compared using node signature algorithm. The signature was the function of hashing. Hash value of each node depends on level of tree and node name. The algorithm for change detection depends on certain parameters i.e. node name, node no., tag, childs, content where node no. was assigned according to BFS, tag contain hash value, child contain node no. of children and content contains content of leaf node. It saved the user browsing time.

K. S. Kuppusamy et al. 2014 [16] proposed an effective model CaSePer i.e. change detection based on segmentation with personalization. This model detects structural and content change. It focused on specific segment on which change occurred that narrow down the search space, so it reduced the complexity. It used MD5 hashing technique for computing the signature of the segments. It reduced the complexity of change detection by narrowing down the search space by focusing on specific segment on which changes occurred.

Sandesh D. Jain et al. 2014 [17] proposed a web page change detection system for selected zone based on tree comparison technique. The tree was generated for old page stored in repository and new page for selected zone. They developed generalized tree comparison algorithm. The two trees build by tree builder module was compared by comparator module that finds the most similar sub trees. It detected structural and content level changes at the minute level. It reduced browsing time.

Md. Abu Kausar et al. 2015 [18] proposed a novel web page change detection method. They compared hash value of new web page with the hash value of old web page previously stored in database. If the hash values were same it means web page was not updated and if hash values were different it means page was updated. The modified new page was stored in database with the updated hash value. The hash value was calculated by using sql server inbuilt function checksum(). The checksum() function can detect structural change, content change, cosmetic change or behavioural change. The above method worked in two steps. First, they constructed a HTML document tree using Document Object Model (DOM). Second, the hash value was calculated of new page arrived.

Discussion- Most of the algorithms are based on hybrid technique of page change detection as they combine the good features of the above two categories. The document tree method has high computational cost as it uses tree edit distances for extracting changes. The other limitation of these methods which are found is extracting changes retrieve the summary but not the complete content of the newly created page that creates insufficient information for the user. In document tree based approach, if the numbers of nodes in a tree are increased, comparison becomes longer, so it is difficult to compare signatures for each and every child node. The drawback of tree traversing algorithm is performance is not defined when depth or levels of the tree are increased.

4. Conclusion and Future Research Directions

Web page change detection techniques reported in the literature have been systematically analyzed. Many limitations have been found in the existing algorithms of page change detection and based on these limitations three research directions have been drawn. Hybrid techniques have been found more efficient than the other two techniques because they detect both changes in structure as well as changes in content of web pages. Least work has been done on structural change techniques because they are inefficient as they detect only structural changes and fail to detect the changes in the content.

The time and space complexity of the algorithms are high which needs to be improved. Still sufficient work has not been done to detect the changes in the images that can be a very good research direction for future work. More research need to be done to make the summary based page change detection technique more effective and efficient.

References

1. Yadav, S. C. M., “An Approach To Design Incremental Parallel Web Crawler”, Journal of Theoretical and Applied Information Technology, vol. 43, no.1, September 2012.

2. Chakravarthy, S., “Automating change detection and notification of web pages”, IEEE, 2006.

3. Khandagale, H., “A Novel Approach for Web Page Change Detection System”, International Journal of Computer Theory and Engineering, vol. 2, no. 3, pp. 364-367, June, 2010.

4. Singhal, S., “Text Content Based Web Page Refresh Policy”, JGRCS, vol. 3, no. 11, November 2012.

5. Trivedi, D. O. G. M., “An Approach to Design Personalized Focused Crawler”, JCSE, vol. 2, no. 3, 2014.

6. Cobena, A. M., “Detecting Changes in XML Documents”, Proceeding of 18th Int’l Conf. Data Eng., pp. 41-52, 2002.

7. Flesca, M., “Efficient and effective web change detection”, Data and Knowledge Engineering, pp. 203–224, 2003.

8. Khan, W. R., “Change Detection of XML Documents Using Signatures”, University of Texas at Dallas, Richardson, TX 75083-0688.

9. Wang, D. J., “X-Diff: An Effective Change Detection Algorithm for XML Documents”, Proceeding of 19th International Conference on Data Eng., pp. 519-30, 2003.

10. Yadav, S. G., “Change Detection In Web page” Proceeding of 10th international conference on information technology, pp. 265-270,2007

11. Yadav, S. Y., “Parallel crawler architecture and web page change detection” WSEAS, vol. 7, no. 7, July 2008.

12. Mali, M., “Focused Web Crawler with Page Change Detection Policy”, Proceedings published by International Journal of Computer Applications, IJCA, 2011

13. Goel, A., “An efficient for web page change detection”, IJCA, vol. 48, no.10, June 2012.

14. Varshney, S., “A Novel Architecture and Algorithm for Web Page Change Detection”, IEEE IACC, 782-787, 2013.

15. Batra, J., “A Novel Approach on Web Page Modification Detection System at multiple nodes”, IJARCCE, vol. 2, no. 9, September 2013.

16. Kuppusamy, A., “CaSePer: An efficient model for personalized web page change detection based on Segmentation”, JKSU, 2013.

17. Jain, K., “A Web Page Change Detection System For Selected Zone Using Tree Comparison Technique”, International Journal of Computer Applications Technology and Research, vol. 3, no. 4, 2014.

18. Kausar, D. S., “A Novel Web Page Change Detection Approach using Sql Server”, MECS, 2015.