IJCSN International Journal of Computer Science and Network, Volume 5, Issue 6, December 2016
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 1.02

An Efficient Indexing Approach on Hidden Web for AJAX Applications

[1] Beena Mahar, [2] Dr. C K Jha

[1] Research Scholar, AIM & ACT Department
Banasthali Vidyapeeth, Rajisthan, India

[2] HOD, AIM & ACT Department
Banasthali Vidyapeeth, Rajisthan, India

Abstract - In AJAX search engine, indexing techniques for the hidden web document is the major issue to optimize speed and performance to find out the related hidden data for a search query. This paper is based on indexing the hidden web documents for AJAX web search engine to emphasis on how to utilize the resources to fulfill the extent to save time and cost as compare to the existing frameworks for the organization details to understanding of the steps to be taken both in term of business process and technical implementation, evaluation and analysis. The searching process is basically directly based on the indexing technique. In web there are many indexing technique which index the web content retrieved by a general purpose of web crawler. But it's not necessary that all the indexing techniques are good for the hidden web. So that there are needs for the hidden web content must be indexed efficiently. In this research paper proposed a designed an efficient indexing technique. The main aim of this indexing technique is to reduce the query processing time based on domain and give more specific result html pages to the use's query.

Keywords - *Hidden Web, AJAX, Deep Web, Indexing Technique, Web Index, Indexer, Crawler.*

1. Introduction

Today indexing in the web search engine is a very active area of current research for the researchers. The aim of the web search engine is to provide the most relevant web documents to the users in least possible time. In simple indexing is the process of developing a document representation by assigning content descriptors or terms to the documents of collections. An indexer is the module in a web search engine system which is responsible to carry out the indexing process.

Hidden web which called deep web refers to the WWW content that is not part of the surface web and which is indexed by the standard search engines. Hidden web also called invisible web or the Undernet. It's describe the portion of the WWW that is not visible to the publically or has not been indexed by the search engine. Surfacing the hidden web poses several challenges [3, 26].

The main aim is to index the content behind many millions of HTML forms that span many languages and hundreds of domains. This approach is completely automatic, efficient and very scalable. The other aim is to large number of web forms have text input and require a valid input values to be submitted. So that the data retrieved through the hidden web is structured and in bulk. That's why we need of an efficient indexing technique and the data extraction architecture that extracts the data corresponding to different users based on their respective requirements in a specific to a particular domain [1, 2].

Generally in search engine the content on the hidden web is not accessible. The data is retrieved through hidden web is structured and the indexing techniques which is used to index the unstructured data are of no use in the case of the structured data.

In this paper we try to design and implement an efficient indexing technique for improves the access of hidden web documents for the AJAX search engines. The main aim of this indexing technique is to reduce the query processing time based on domain and give more specific result html pages to the use's query.

2. Requirement of Web Crawler

Index is the one type of data structure which is permitting a rapid identification of crawled pages which contain a particular words or phrases [4]. It's a set of web document with the words that they are contained. All

these words they contain where we need to processing all documents which is available in a local repository. Creating the index by accessing the web documents directly on the web is very impractical for a number of reasons. Collecting all web documents can be done by web browsing the web systematically. This is done by a web crawler and is used by a search engine [5]. Crawler, indexer and query processor are three main component of web search engine for searching a data. This is explaining below with the diagram.

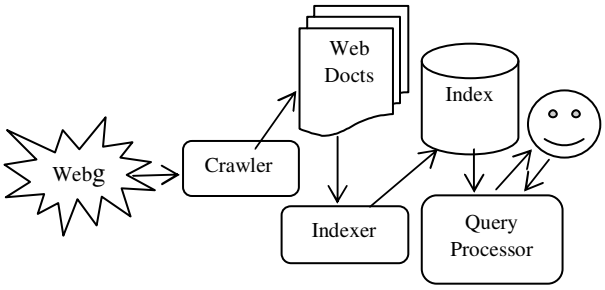

Fig. 1 Search Engine Process

When we crawler web pages, the copy which is made is returned to the search engine and to be stored in a data center which is simply we called local repository. Data centers are very huge and its built collection of servers which is act as a local repository of the all the copies of webpages being made by the web crawler. The repository of web pages is referred to the index. In index the data is store which is organized and used to provide the search result which you see on the web search engine. Indexing is the process of organizing the masses of data and the web pages so they can be searched very quickly for their relevant result to you search query

We have a very large collection of copies of web pages which are begin constantly updated and organize so that we can quickly find out what we are looking for. But we need a means by which they can be ranked in order of relevance to our search term. For this search term algorithm play a very important role. Because an algorithm is a very complex and lengthy equation which calculates a value for any given site in a relation to a search term [29].

The main purpose of storing an index is to optimize performance and speed in to finding relevant web documents for a web search query. Without an index the web search engine would scan each web document in the corpus, which would require considerable time and power. Web search engine architectures vary in the way indexing is performed and in method of index storage to meet the various design factors [6].

3. Fundamentals of Indexing Techniques

An index is a collection of web documents which is evaluated through user queries. Indexing is the process converting crawled web contents into a compressed searchable form thus its extract useful data from the source. The process of indexing starts by extracting each individual word from the text of a page [7, 8]. The simple indexing process explain with the below figure2.

The data structure of indexes includes two type of structure:

- Inverted index
- Forward index

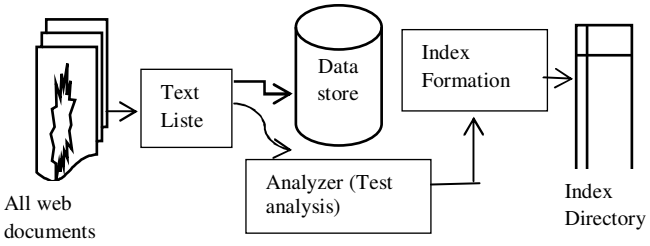

Fig 2. Indexing Process

Inverted Index: An inverted index stores a complete list of web documents which is made up of the search query where all the words in a web page and a pointer to the web page that are contains that search query. Because the inverted index carry a list of the web documents which contain each word the search engine can use direct access to find the web documents which is associated with each word in the query in order to make the matching web documents quickly [9]. The application sorts the data on the term that's why the search engine locates the matching terms extremely quickly. This index is used to find the record rather than the other way around.

Forward Index: The forward index stores a list of words for each web documents [30, 31].

4. Related Work

Since the web is growing fast and uncontrollably as the size of the search engines indices indicates several pre-indexing analysis techniques are performed. Such pre-indexing techniques are intended to reduce and also manage the size of the web data collected by web crawlers. The following are the pre-indexing analysis techniques:

Size Reduction Techniques: This includes representation of the high performance text indexing with

the index construction techniques and the algorithms for the evaluation of text queries. This improves the search engine's capability and efficiency to accommodate the size of all fetched web documents. Using this technique the space and time needed for text indexes and disk traffic required during the query evaluation are reduced. Basically this technique is used for construction of a document level indexing and for the ranked query evaluation. [10]

Lexical Analysis: This is a process of identifying the index terms from the input stream that is contents of documents where the user can find the desired information by submitting the queries to the search interfaces. It is to help to reduce the complexity and save time. [11]

Stemming Technique: This technique play a very important role to minimize the size of the index. For example, the terms "Connection", "Connected" and "Connect" are indexed by selecting one term "Connect" and eliminating the terms with the suffixes "ion" and "ed". In addition also removed the related suffixes, prefixes, adjectives etc. this helps to reduce the size of the index and make better searching performance. [12]

Compression and Query Processing Techniques: There are many indexing compression techniques which are used for the resize and minimize the size of the index. [13]

Many researchers have also emphasizing on fully on index the high quality data performance for various indexing techniques. There are various indexing techniques for the hidden web to be used by the researchers. These techniques for the hidden web documents are to optimizing speed and performance to find the related hidden data for a search query. Ali Mesbash [16] proposed an automated crawler called Crawljax, for AJAX based web applications and Cristian Duda [17] represent how various DOM states can be indexed. Ali Mesbash also conducted a quantitative study to measure the hidden web content induced by client side scripting which is implemented through a tool called Javis [24, 25].

One of the indexing techniques is Linguistic Technique [14]. This technique is for the capturing the semantic web of the hidden document which is used for the indexing and retrieved the related web documents from the internet. The advantage of this technique is to perform the keyword based indexing search and introduces a relationship between part of text using NP (Natural Language) and RST (Rhetorical Structure Theory).

Domain-specific key phrase extraction technique [22, 23] is another technique for creating the index. This technique is useful for clustering, topic search and document summarization. By this technique performance and quality can be boosted by automatically tailoring the extraction process to the particular document collection at hand.

Another is the domain-specific indexing technique for creating the index for the hidden web. But this technique is integrating them into the Oracle 8i server [15]. This framework especially based for the oracle 8i which is integrated by domain-specific indexing schemes. The main advantage of this technique to provide a set of ODCI index routines for index scan operations, index maintenance and index definition. The demerit of this technique is it is only for the oracle server.

One more domain-specific indexing technique for the creating the index the hidden web crawled documents, which is a domain specific indexing technique based on attributes of a query interface and their value sets [14]. The previous techniques are based on keywords in which first the crawler downloads the documents and then extracts the keywords from these downloaded documents to index them. The main advantage of this indexing technique this is not only optimizes speed for finding hidden web documents but also gives more specified results for a search query.

Hidden web data sources contain a high quality data which is hidden behind a query interface. Many efforts have been focused on querying and integrating the web data sources. The main focus of these efforts is to build a Global Query Interface which is refer [15] to the web data sources in the domain to make easy access to individual sources transparent to the users. This Global query interface is designed by the Interface Matching techniques. A separate Global Query Interface is designed for each domain.

All these literature indicates that many indexing techniques have been introduced to index the hidden web documents with their advantage and some drawbacks. Others more literatures are available which identify many frameworks have been introduced to crawl the hidden web documents. [18, 19, 20, 3, 22]

In reference [18] represents a system for domain-specific crawling for the hidden web. But the main disadvantage of this system is that it's only for the full text search forms. These forms search any web documents only through single text field which indicate a full text. When to crawl all the content behind a web form then automatically generate the problem of keyword query for

the hidden web. This is the problem which is addresses in reference [19].

Reference [20] indicates the problem of extracting the full contents behind a web forms. Due to this problem this system does not accept forms with the required "textbox" fields to be filled in.

In reference [3] which are totally based on a set of domain definitions where each one describing a data collecting task.

The one more system in reference [21] indicates not for the single attribute query but which is focuses on the multi attribute query forms used to query for structured data. The main advantage of this system is to automatically generate new queries from the result of previous one query. But the main drawback for this system is the crawling new result not in the indexing form and this is not applicable for the AJAX search application.

As we see all these related work focuses on measuring various indexing and crawling techniques on different ways. In this paper we focus on Task-specific and Human-assisted approach for indexing technique for the hidden web documents only for AJAX Search engine [27, 28]..

5. Proposed Work

An indexing technique for hidden web documents is proposed. It is very efficient indexing technique for crawler hidden web for AJAX applications. This technique uses URL and their states for indexing the hidden web pages. It helps us to improve the access to hidden web documents for AJAX based search engines. Improves the access to hidden web is very important factor for AJAX search engines from the user's view. Here indexing is performs on the basis of stats and their corresponding URL. It takes a less time to find the hidden document indexed by index files as combination of various states with their related URL.

In this technique a separate file is created for each states for their related URL.so that its helps us to improve the access to hidden web documents for AJAX based search engines. The framework of proposed indexing technique is as shown in below figure3.

AJAX data: we starting from the AJAX data which is crawls URL from the www. AJAX crawled data crawl the hidden web pages from the www which identify different states with same URL.

Hidden web crawler: AJAX crawled forward the AJAX data to hidden crawler than the hidden web crawler extracts the URL and then analyze with the different states.

Web indexer: Web indexing indexes to the URL or individual hidden web documents which are create of metadata to provide useful information for AJAX search engine. Indexer is the core part of the system which are used for searching AJAX data.

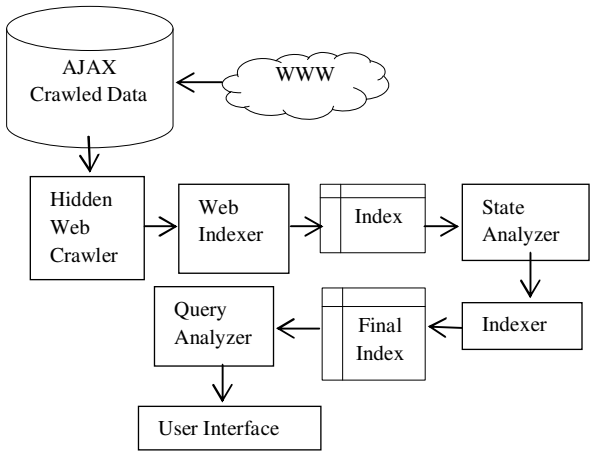

Fig. 3 Framework of Proposed Indexing Technique

Index file: An index file is a physical inverted file containing information about the URL and the state in which the keywords occur.

State analyzer: analyze the states and find out the relationship between the states and the user view of the querying.
Query analyzer: is the process of fetching a user query in to an efficient and correct way.

User interface: user has to various queries in the process of fetching of correct query through user interface.

6. Conclusion

The Hidden Web is important because it retrieves high-quality information. Therefore there is a need to implement an indexing technique to be more efficient to index the high quality data. Many research focus on the crawling and indexing algorithms for client side as well as server side DOM state changes, some are the hidden web behind forms, text and search query with their own advantages and disadvantage. In this paper proposed a framework for indexing technique which is performed by the various modules. This is a simple and efficient indexing technique for improving the access of hidden

web documents for the AJAX search engines. This research can also be expanded through adding multiple domain and their corresponding sub domains. Along with modification in implementation, new algorithms for better understanding and quick results of multiphase dynamic queries can be introduced.

References

[1] J. P. Lage, A.S.da Silva, P.B. Golgher and A.H.F. Laender, "Automatic generation of agents for collecting hidden web pages for data extraction" In Data & Knowledge Engineering, volume 49,issue 2, pages 177-196,2004.

[2] L. Barbosa and J. Freire,"An adaptive crawler for location hidden-web entry points" In Proc. of the 16th Int. Conf. on World Wide Web, pages 441-450, ACM-2007.

[3] S. Raghavan and H. Garcia-Moline, "Crawling the hidden web" In Proc. of the Conf. on Very Large Data Bases, pages 129-138, 2001.

[4] Hasan Mahmud, Moumie Soulemane and Mohammad Rafiuzzaman, " A framework for dynamic indexing from hidden web" In the Int .Journal of Computer Science, Volume 8, Issue 2, 2011

[5] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin and Craig G. Nevill-Manning, "Domain Specific Key phrase Extraction" " In the Proc. of the 16th Int. Conf. on Artificial Intelligence IJCAI, volume 2, pages 668-673, ACM-1999

[6] Moumie Soulemane, Mohammad Rafiuzzaman, Hasan Mahmud,"Crawling the Hidden web:An approach to dynamic web indexing", In the Int. journal of Computer Application, volume 55-no.1, Snn no: 0975-8887, 2012

[7] Steven s Skiena,"The Algorithm design", Manual 2nd edition , Springer, Verlag London , 2008

[8] Rahul Kumar, Anurag Jain and Chetan Agarwl,"Survey of web crawling algorithms", In the Int. Journal of Advances in Visions Computing(AVC), volume 1, Issue 2/3, 2014

[9] Aviral Nigam, NIT-Calicut, "Web Crawling Algorithms", In the International Journal of Computer Science and Artificial Intelligence, volume 4, Issue 3,Pages 63-67, 2014

[10] Justin Zobel and Alistair Moffat,"Inverted Files for Text Search Engines", In the Journal of ACM Computing Surveys(CSUR), Volume 38, Issue 2, July 2006

[11] Priyanka Gupta, Komal Bhatia and Kalpna Gupta,"Optimized method for indexing the Hidden web data", In the Int.Journal of Inforatmation Technology and knowledge Management, Volume 4, Issue 2, pages 673-678, July 2011

[12] Anjali Ganesh Jivani, "A Comparative Study of Stemming Algorithims" In the Int. Journal of Computer Technology and Applciation (IJCTA), Volume 2, Issue 6, Pages 1930-1938, 2011

[13] Hao Yah, Shuai Ding and Torsten Suel "Inverted Index Compression and Query Processing with Optimized Document Ordering", In the Int. WWW Conf. Committee –IW3C2 Madrid, Spain, Pages 20-24, ACM 2009

[14] Farhi Marir and lamel Houam,"RST Index: indexing and retrieving web document using computational and linguistic techniques" " In the Proc. of the 3rd Int. Conf. on Intelligent Data Engineering and Automated Learning, UK, pages 135-140, 2002 Available at http://portal.acm.org/citation.cfm?id=646288.686474.

[15] Jagannathan Srinivasan, Ravi Murthy, Seema Sundara, Nipun Aggarwal and Samuel DeFazio,"Extensible Indexing : A framework for integrating domain specific indexing schemes into Oracle 8i", In the Proc. of the 16th Int. Conf. on Data Engineering IEEE-2000

[16] A.Mesbah, A..an Deursen and S. Lenselink, "Crawling AJAX Based web applications through dynamic analysis of uder interface state changes", In ACM Transaction on the web- TWEB, volume 6, Issue 1, 2012

[17] C Duda, G. Frey, D. Kossmann, R. Matter and C Zhou,"AJAX Crawl: making Ajax applications searchable" In the Proc. of the Int. Conf. on Data Engineering, pages 78-78, 2009

[18] A. Bergholz, B. Chidlovskii, "Crawling for Domain-Specific Hidden Web Resources" In the Proc. of the 4th Int. Conf. on Web Information System Engineering,2003

[19] A. Ntoulas, P. Zerfos and J. Cho, "Downloading Textual Hidden Web Content through Keyword Queries" In the Proc. of the 5th ACM/IEEE Joint Conf. on Digital Libraris,2005

[20] S. Liddle, D. Embley, Del Scott and S. Ho Yau, " Extracting Data Behind Web Forms" In the Proc. of the 28th Int. Conf. on Very Large Data Bases, China, 2005

[21] Manuel Alvarez, Juan Raposo, Alberto Pan, Fidel Cacheda, Femando Bellas and Victor Cameiro, "Crawling the Content Hidden Behind Web forms " Department of Information and Communications Technologies, University of A Coruna, 15071, Spain,

[22] Ritu Shandilya, Sugam Sharma and Shamimul Qamar,"A Domain Specific Indexing Technique for Hidden Web Documents", In CISME, volume 2, issue 2, pages 37-41, 2012. Available at: www.jcisme.org

[23] A. K. Sharma and Komal Kumar Bhatia,"Merging "query interfaces in domain specific hidden web databases" In Int. Journal of Computer Science,2008

[24] A. Mesbah, E. Bozdag and A.V. Deursen, "Crawling AJAX by inferring user interface state changes", In the Proc. Of 8th Int. Conf. on Web Engineering (ICWE), Washington DC,USA, IEEsE-CSI , pages 122-134, 2008

[25] Zahra Behfarshad and Ali Mesbah**. "**Hidden-Web Induced by Client-Side Scripting: An Empirical Study", Springer Berlin Heidelberg, In Proceedings International Conference (ICWE 2013) Aalborg, Denmark, pages 52-67, 2013.

[26] Li Jie Cui, Hui He and Hong Wei Xuan "Analysis and Implementation of an Ajax-enabled Web Crawler" ,In the Int .Journal of Future Generation Communication and Networking , Volume 6, Issue 2, April 2013

[27] Paul Suganthan G. C.,"AJAX Crawler", In the Int. Conf. on Data Science and Engineering,(ICDSE), IEEE- 2012

[28] A. Mesbah, A. Van Deursen and S. Lenselink, "Crawling Ajax based web applications through dynamic analysis of user interface state changes", In ACM Transaction on the web- TWEB, volume 6, Issue 1,page 3, 2012

[29] A Rosaline Mary, B Visvanath, "Evaluation of web search engine-a comparative study", In Research Gate 2015

[30] Bhupendra Singh, Shashank Sahu,"Model for performance testing of AJAX based web applications", In Int. Journal of Research in Engineering and Technology, volume 3, Issue 4, eISSN 2319-1163,pISSN 2321-7308, 2014

[31] Bhupendra Singh, Shashank Sahu,"A Noval approach for evaluation of applying AJAX in the web site", In Int. Journal of Research in Engineering and Technology, volume 3, Issue 8, eISSN 2319-1163,pISSN 2321-7308, 2014