ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

Pre-Query and DUST based crawler for Hidden Web sites

Tejaswini Patil[1], Santosh Chobe[2]

P.G. Student, Department of Computer Engineering, DIT, Pimpri, Pune, India[1]

Associate Professor, Department of Computer Engineering, DIT, Pimpri, Pune, India[2]

ABSTRACT: Crawlers are used by many search engines to search data as per user requirements. It provides up to date data. Sometimes it gives irrelevant links which are not required. To achieve accurate or relevant result we have proposed web crawler. Our crawler searches deep web pages. Site locating and in-site exploring steps are used to achieve deep web pages and it also searches relevant links. To remove duplicate links Duster technique is used. Efficiency is achieved using pre-query approach.

KEYWORDS: Deep web, DUSTER, pre-query, ranking, reverse search.

I. INTRODUCTION

Search engines help users to find web pages on a given subject. The search engine maintains databases of web sites and use programs to collect information, which is then indexed by the search engine. Similar services are provided by directories which maintain ordered list of websites e.g. Yahoo, Google etc. A search engine is a searchable online database of internet resources. Web crawler is a program that searches data on web with help of this crawling technique. Crawler retrieves documents and that information is added to a combined index, the document is not stored, some search engines do a cache a copy of document to give clients faster access to the documents or links. These crawlers retrieve data much quicker and in greater depth than human searches.

The search engine does not cover all the data of the web. None of these engines search a data all overall the web. These data are called hidden data or deep data [4]. Domain specific search engines have however become equally popular as they offer a higher precision for someone looking for information specific to a domain of human activity. Deep Web refers to the data which are not indexed by any standard search engines such as Google, Yahoo. The Deep Web refers to all web pages that search engines cannot find such as user databases, registration required web forums, webmail pages and pages behind pay walls [16].These pages are only accessible by submitting queries to database and regular crawler are unable to find these pages. Locating the deep web database is a challenging because they are not registered with any search engines and they are constantly changing. The deep web is qualitatively different from the surface web. Most of the data is uploaded on web for creating online database. These databases are used to search information through internet. Deep web makes use of these databases and store their content in searchable database that produce results dynamically in response to a direct request.

II. RELATED WORK

Feng Zhao et.al in [1] shows that deep web grows very fast. These deep webs are vibrant in nature so large coverage and efficiency is not achieved. To defeat these drawbacks this paper proposes SmartCrawler. With the help of this crawler broad coverage and efficiency is easily achieved. From the study it shows that deep web site typically contain a small amount of searchable forms. Here, the crawler is divided into two steps: site locating is the first step and in-site exploring as a second step. In site locating it will accomplish broad coverage of sites and the in-site exploring step will achieve searching of forms inside the sites for efficient results.

Denis Shestakov et.al in [2] discussed that deep web classification is done in two ways i.e. overlap analysis and rsIP (random sampling of IP addresses). In overlap analysis it involves pair wise comparisons of deep web sites

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

and rsIP is easily generated and reproduced. This rsIP does not require any listings. Ignoring virtual hosting is the main drawback of rsIP. For accurate estimation of deep web parameters it samples the national web domain. These will implement the host IP cluster method that addresses the issues of old approach, and then it differentiates the deep web and resulted information is used for the survey of the deep web. The results of proposed sampling methods are estimated that might be helpful for further study to hold the deep web data.

Raju Balakrishnan et.al in [3] presented deep web database selects the relevant web databases to give the answers about particular query. The existing database selects the methods that will evaluate the source quality based query. When these methods applied to the deep web it has two difficulties. These approaches are doubter for the accuracy of the source and the query based results are not considered as important. For open collections like the deep web these considerations are essential. Many sources provide answer to the query, and then these answers are helpful for accessing the sources. It computes the contract of the source and the answer is return based on the agreements. While computing the agreement it allows measuring and compensating possible involvement between the sources. Here, they calculated SourceRank for the possibility of a random data.

Yeye He et.al [4] has shown deep web crawler refers the difficulty of developing hidden information following the web search interface of dissimilar sites transversely the web. Deep web sites preserve article based information which focuses on deep web journalism, this examine that a major section of deep web sites which includes shopping sites, company sites, business sites etc. Searching such site oriented content is valuable for many reasons. Existing crawling technique is not suitable for entity oriented sites then it is optimized for document oriented content. In this work, it describes a sample system that is crawling entity leaning deep web sites. It also proposes a technique which will deal with the significant sub-problems. Here, it leverages characteristics of these entity sites and proposed technique improves the efficiency and effectiveness of the crawler system.

Mustafa Emre Dincturk et.al in [5] presented new technique of crawling approach. This approach is used as a beginning to design a resourceful crawling strategy for RIAs (Rich Internet Application). A simple model crawling is called hyper-cube strategy. This crawling is compared with existing strategy for performance improvement. The performance can be checked using different methods such as breadth first search, depth first search, and a greedy method. Model based crawling approach is well organized and professional than any other standard strategy. Although the hyper-cube method is an excellent example which shows that this newly designed crawling approach works well and it has a fine performance compared to the existing strategy.

Soumen Chakrabarti et.al in [6] presented that the information which is related to a specific topic are gathered using focused crawlers. Focused crawlers allow searching relevant information in more depth and keeping the crawler fresh, because there is less to wrap for each crawler. The development over the focused crawler is that it assigns priorities to the unvisited URLs in the crawler database. This will lead a higher rate of fetching pages which are relevant to the specific topic. These features will be more numerous, sparser and might be harder to learn.

Cheng Sheng et.al in [7] describes a hidden database which will refer the dataset. An organization access these datasets on network by allowing developers to issue the queries during a search interface. Search engines ineffectively indexes hidden data and these are not able to express query to the relevant data in individual's repositories. Whenever the algorithm accesses data from hidden database in a row wise format then the problem is occurred. These algorithms are capable and they accomplish the assignment by performing little amount of queries in the worst case. And for larger queries it is impractical to recover the effectiveness of algorithm.

Nilesh Dalvi et.al [8] gives an indefinite set of items are enclosed in known and unknown group of elements. Estimation of sets can be done based on the set range and properties of the data within the defined set. This has many problems and to address such problems a capable method is defined that uses the set information to find solution of the defined problem. This approach might be used in the hidden web database environment to calculate the quantity of sets. Here, they believe in each set defined by this approach and allows the applications such as google-maps and APIs. The performance can be improved with the help of this newly defined approach.

Kaio Rodrigues et.al [9] describes that web crawler collects URLs these URLs may contain duplicate or near-duplicate contents. Accessing such duplicate data is wastage of time and resources because of this it gives poor results. To deal with this problem, author proposed a technique to detect and remove duplicate documents without fetching their contents. They used normalization rules to renovate all duplicate URLs into the similar canonical form. DUSTER approach is used with multi-sequence alignment strategy. This method leads to very effective results.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

Agarwal et.al in [10] discussed that existence of duplicate documents affects the performance of crawling and indexing. This paper presented a technique to extract rules from URLs and utilize these rules for de-duplication using URL strings without fetching the content explicitly. It contains crawl logs and similar pages are clustered to extract specific rules from URLs belonging to each cluster. Maintaining each rule is not efficient. To overcome this author used a technique to simplify the set of rules, which will reduce the resource foot- print to be usable at web-scale.

Kevin Chen et.al in [11] proposes objectives of the integration system which gives the dynamic results. This integration system constructs the methods such as query translation and discovery of dynamic sources. It presents the structural design and fundamental approaches while implementing the sub-systems. Integration of system efforts and performing function of sub-systems is the main task. Here, they have observed sub-systems which present the challenges and opportunities ahead of subsystem segregation.

Mangesh Manke et.al in [12] has shown that achieving wider treatment and high effectiveness is major issue. To overcome these problems this paper demonstrates a skeleton, specifically for efficient gathering deep interfaces. It defines two stages, in first step it will do site based arrangement of centre pages with respect to search information, avoid visiting an extra large page. To understand the results of crawling process, the crawler ranks these websites to instruct enormously relevant ones for a specific topic. In next step, most relevant linking is achieved by crawler in this stage.

Jayant Madhavan et.al in [13] discussed that surfacing deep web has several challenges. The HTML form will cover more languages and thousands of domains its main goal is to index the contents. This approach is fully automatic, extremely scalable and well-organized. A huge amount of searchable forms has phrasing input and it requires the suitable input ideas. These values should be sending properly for accurate results. Here, it will first select the input value for text inputs which will recognize key terms and an algorithm is used for identifying input values which will accept only standards of a specific data. This HTML form has more than one input and hence immature strategy is used.

Denis Shestakov et.al in [14] discussed web pages are dynamically generated. Web dynamism is the important container here web pages are generated as per the given query and they are submitted to the databases which is presented onsite. These pages represents piece of the web known as profound web. The study of web sites which are already estimated is based on deep webs. The main parameters of deep webs are not enquired so far. Thus, famous characters of the deep web possibly are unfair, which particularly outstand to enhance web content. Using sampling technique the deep web is estimated accurately.

Mohammadreza Khelghati et.al in [15] represented general approach which finds and extracts the information from search engines such as google, yahoo, bing via searching specific queries. Hidden web means all required data is not covered by search engines and this information is unavailable during the crawling process. From the estimation it is noticed that hidden web holds the data which is much more than the data available in search engines this process is called as surface web. The data available on deep web can be achieved by their own interfaces, resulting and querying all the sources of information that may be constructive that could be tricky to understand and are long task. When the large quantity of data is linked to useful data then it may not possible for a user to cover all data on web search engines.

Andre Bergholz et.al in [16] defines a technique to recognize domain relevant hidden web property. The hidden web crawler discovers exciting web pages, analyse it and search relevant information to take decision about hidden web. This paper describes a crawler which will starts at particular point in the hidden web. Pre-classified documents and relevant keywords are used to initialize domain specific crawler. These evaluate sequence of results using the top level category in the directory and report the study of hidden web assets. It shows the amount of hidden web assets is greatly domain dependent, the resources are established with slight crawling effort and these methods perform well in both the domain specific and accidental form of crawling.

III. SYSTEM ARCHITECTURE

Proposed crawler searches relevant data on internet as faster as possible and in efficient manner. If user wants faster and accurate data then there are different types of algorithm. This crawler works in following steps: Site Locating, In-site Exploring, Duster, and Pre-query.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

A. User Module:
Based on the given query crawler will search relevant results as per the user requirement.

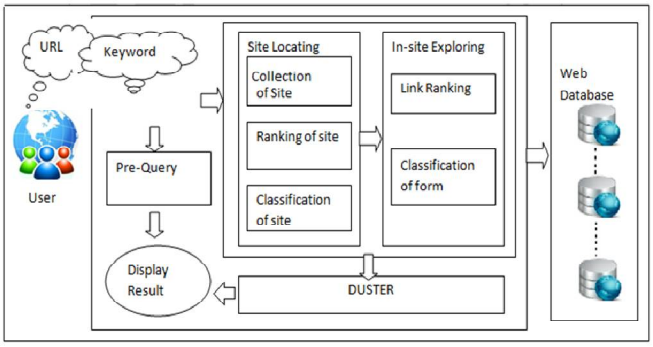

Fig.1 System Architecture

B. Site Locating:
In Site locating stage, user will find out the most relevant result for a given topic with the help of crawler. Site locating stage performs three steps:
1. Collection of site 2.Ranking of site 3.Classification of site. In this, it collects all website and find out center pages of unvisited sites using reverse searching technique. After collection, the site ranker assigns a rank to each unvisited site that corresponds to its relevance in ranking format and then classification of site is done. In this classification, site is categorized as topic relevant or irrelevant. If the site is relevant then it starts the crawling process. Otherwise, the site is ignored and new site is collected.

C. In-site exploring:
In in-site exploring stage it finds searchable forms from the site. It consists of two steps: 1. Link Ranking 2.Classification of form. Link Ranker assigns priority to links then crawler finds searchable forms. Classification of different subtypes collects input and finds linking of them and searches data within the site for accurate result. Display result in different form wise in ranking format.

D. Duster:
Duster is a technique which detects and removes DUST from search engines. DUST contains duplicate URL's with similar text these duplicate data implies waste of resources. To remove such duplicate data normalization rules are used in this Duster technique.

E. Pre-Query:
Pre-query approach is used to improve the efficiency. In this search is based on word by word bases. It maintains history and every time search query is checked in database if it resent in database it will fetch the result and give it to user. Otherwise, it searches data as per user query and then maintains data in database.

IV. ALGORITHMS

Following algorithms are used to crawl URLs in our system. Duster is used to remove duplicate URLs. Ranking algorithm assigns rank to the links found in first stage and with the help of ranking accurate results are achieved. Pre-Query algorithm is used to improve efficiency. It helps to find quick results by maintain database.

Algorithm 1. Duster Technique

Input: Get multiple URLs U= {u1....un}
Output: sorted/removed duplicate URLs

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

1. Domain=0; Sequence=0
2. for i =0;i< U.Size; i++
3. Domain[]=U.getdomain(i);
4. Sequence[]=U.getSequence(i);
5. End for
6. D_{dupli}=CalculateDuplicateDomain(Domain);
7. S_{dupli}=CalculateDuplicatesequence(Sequence);
8. $D_{original}$=Domain - $D_{dupli;}$
9. $S_{original}$=Sequence - $S_{dupli;}$
10. MergeUniqueUrl($D_{original,}$ $S_{original}$);
11. end

Algorithm 2. Ranking Technique

Input: Set of URL, search_url
Output: get ranking set.

1. Get input URL set.
2. Word=GetCenterWord(removeAfterDot(removeBeforeSlash(search_url)))
3. for(String s to Set of URL)
4. If(s. contain(Word))
5. Rank[]=s;
6. Else
7. Irrelevant[]=s;
8. End if
9. End for
10. Add all irrelevant[] array to Rank[] Array
11. Rank[].add(irrelevant[])
12. Return Rank[];
13. End

Algorithm 3. Pre-Query Technique

Input: All relevant data to query
Output: Fetched result from database

1. If(query==input)
2. DisplayCurrentResult();
3. Else
4. SearchWordbyWord();
5. Extract form in query for searching.
6. displayResult();
7. stop;

V. MATHEMATICAL MODEL

Start of System can be represented as set S1:
S1= {I, O, P, S, F}
Where,
I-Input, O-Output, P-Process/Function, S-Success, F-Failure
I= {Q}
P= {D, Rs, Rl,Cs ,Cf, Dust, Pre }

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

$$Rs = \frac{D}{Q} \quad (1)$$

$$Rl = \int_{i=0}^{N} RS \quad (2)$$

$$Cs = dx\, Rs \quad (3)$$

$$Cf = dx\, Rl \quad (4)$$

$$O = \int_{i=0}^{N} Cs + Cf \quad (5)$$

Dust = O – dupurl (6)

Pre= Q (PRE (D)) (7)

Where,

Q= user query, D= Dataset, Rl=Link ranker, Rs=Ranking site, Cs=Site classifiation, Cf= Form Classification, dupurl= duplicate Urls.

O= {Out} Out- resulted links/Urls.
S= {d} d – accurate and efficient results display to the user.
F= {f} f – internet connection lost.

VI. EXPERIMENTAL RESULTS

TABLE I
COMPARISON OF CRAWLERS

	Smart Crawler	**Web Crawler**
Total links	100	90
Relevant links	70	78

If we search for url-www.unipune.ac.in we collect above links. Here, our proposed web crawler searches more relevant links than smart crawler also it removes out duplicate links in final result.

Performance measure is calculated as:
Precision= (relevant links found) / (total number of links found)
Precision= 78/90*100 =86%.

VII. CONCLUSION

Proposed crawler searches for deep web pages and it maintains efficiency and accuracy. The crawler first performs site locating stage it collects sites and focusing on specific topic it will give relevant links. In in-site exploring stage, adaptive link ranking is used to search within the sites. This stage finds searchable and relevant forms. Here we collect deep web sites. Duster is a technique which will remove duplicate URLs. So, we achieve more accurate results. The efficiency of crawling is improved using pre-query approach.

REFERENCES

[1] Feng Zhao, Chang Nie, Jingyu Zhou and H.Jin, “Smart Crawler a two stage Crawler for efficiently harvesting Deep Web interfaces” ,Vol. 9, No. 4, July/August 2016.
[2] Denis Shestakov, Tapio Salakoski. “Host-ip clustering technique for deep web characterization” In Proceedings of the 12th International Asia-Pacific Web Conference (APWEB), IEEE, 2010.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

[3] Balakrishna Raju, Kambhampate Subbarao. "Sourcerank: Relevance and assessment for deep web based on inter-source agreement" In Proceedings of the 20th international conference on World Wide Web, pages 227–236, 2011.
[4] Yeye He, Sriram Rajaraman, Nirav Shah, Venkatesh Ganti, "Crawling Deep Web Entity Pages" Proceedings of the 6th international conference on web search and data mining. ACM, page 355–364, 2013.
[5] Mustafa Emmre Dincturk, Gregor Bochmann and Iosif Viorel Onut, "Model based approach for crawling rich internet applications." ACM Transactions on the Web, pages 1– 39, 2014.
[6] Soumen Chakrabarti, Kunal P., Mallela Subramanyam. "Accelerated Focused Crawling through Online Relevance Feedback". In proceedings of the 11th international conference. ACM, pages 148-159, 2002.
[7] Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin. "Optimal algorithms for crawling a hidden database in the web." Proceedings of the VLDB endowment, pages 1112–1123, 2012.
[8] Nilesh Dalvi, Ashwin Machanavajjhala, Ravi Kumar and Vibhor Rastogi. "Sampling hidden objects using nearest-neighbor oracles." In Proceedings of the 17th ACM international conference on Knowledge discovery of data mining, ACM, pages 1325– 1333, 2011.
[9] Rodrigues, Kaio, et al. "Removing DUST Using Multiple Alignments of Sequences." IEEE Transactions on Knowledge and Data Engineering 27.8 (2015): 2261-2274.
[10] Agarwal, Amit, et al. "URL normalization for de-duplication of web pages." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.
[11] Kevin Chen Chang, Bin He, Zhen Zhang. "Toward large scale integration: Building a metaquerier over databases on the web." pages 44–55, 2005.
[12] Mangesh Manke, Amit Kharade, Kamlesh Kumar Singh, Vinay Tak, "Crawdy: Integrated crawling system for deep web crawling" International journal of research in computer and communication engg. vol. 4, Issue-9, September 2015.
[13] Jayant Madhavan, Vignesh Ganapathy David Ko, Alex Rasmussen, Lucja Kot, "Google's Deep Web Crawl". In proceedings of the VLDB endowment ACM, pages1241-1252, 2008.
[14] Shestakov, Denis, and Tapio Salakoski. "On estimating the scale of national deep web." International Conference on database an Expert systems application. Springer Berlin Heidelberg, pages 780-789, 2007.
[15] Mohamamdreza Khelghati, Maurice V. Keulen, Djoerd Hiemstra. "Deep web entity monitoring." In proceedings of the 22nd international conference on World Wide Web, pages 377–382, 2013.
[16] Andre Bergholz and Boris Childlovskii. "Crawling for domain specific hidden web resources." In proceeding of 4th International conference on information systems, IEEE, pages 125–133, 2003.