International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

SmartCrawler: For Site Locating and Balancing

Sphurti S. Bhosale[1], Dr. S. B. Sonkamble[2]

[1]Computer Engineering Department, Pune University, JSPM NTC, India

[2]Professor, Computer Engineering Department, Pune University, JSPM NTC, India

Abstract: ***The number of web pages in Internet are changed day by day. Therefore relevant information searching is difficult task. Most of this data is hidden behind query forms which is interface to unexplored databases containing high quality structured data. Existing search engine has drawback that they cannot access and index this hidden data of the Web, in fact it is very hard to derive the unseen info. For this, we propose a framework, namely SmartCrawler, just to gather all the web interfaces in depth. Initially that is site locating, in this step, center pages are searched using search engines which is used to avoid visiting a large number of pages. For achieving precise results for a focused crawl, SmartCrawler gives ranks to websites to prioritize highly relevant ones for a given topic. The second step, adaptive link-ranking gives fast in-site searching by searching most relevant links. In third step, i.e. Web Navigation, Space complexity problem of other crawler are managed. The required relevant links are stored in database. Parsing of web pages is done in web navigation step. To reduce time in visiting some highly related links in unseen web directories, we have designed a link tree data***

Keywords: Ranker, Form Classifier, Deep web, Crawler, Adaptive learning

1. Introduction

A web crawler is known as robot or a spider. It is a system for downloading of large web pages. Web crawlers are used for a various purposes. Mostly, they are one of the main components of web search engines, systems that assemble a quantity of web pages, index them, and allow users to assign queries against the index and find the web pages that match the queries.

In web crawling, the crawler search around the web-pages, to gather and categorizes information on the World Wide Web. The crawler has three parts: The first part is the spider, also known as crawler. The task of spider is to visits the pages, the information is fetch and then follows the links in other pages within a site. The spider gives too crawled site in regular interval of time. First stage information is given to second stage, the index. It is called as catalogue. The index is act as a database, in which every copy of web-page that crawler finds are saved. When a web-page changes then the data is updated with new information in the database. Software is the third part which is nothing but a program that uses millions of web pages recorded in the index to find matches to search and level them according what it believes as most relevant.

Hidden data or invisible data are known as Deep web [2]. The contents of deep web [5] are not indexed in a search engine on the web. In Deep web the number of websites are collected that are publicly available but hide the IP addresses of a server that run on them. Deep web is visited by user but it is difficult to find information behind them. Deep web is cannot locate with a single search.

As they are not recorded by any search engines, it is difficult task to locate deep web interfaces. Deep web are usually rarely distributed and keep changing. To handle above problem, existing work has given two types of crawlers first is generic crawlers and second is focused crawlers [1][2].Generic crawler retrieves all the searchable forms and do not focus on a specific topic in Focused crawlers which focuses on a specific topic. Form-focused crawler (FFC) [5] and Adaptive crawler for hidden web entries (ACHE)[6] gives for efficiently and automatically detect other same domain forms. The main component of FFC includes link, page, form classifiers and frontier manager for focused crawling of web-forms. ACHE enhanced the focused strategy of FFC with additional parts as form filtering and adaptive link learner. For achieving higher crawling efficiency than the best-first crawler the link classifier is used [7]. The focused crawlers has low accuracy in terms of retrieving relevant forms. It is necessary to develop smart crawler which is able to quickly search similar contents deeply from web as much as possible.

SmartCrawler is a framework for efficiently harvesting deep web is designed in this paper. The main task of SmartCrawler is advanced level of data analysis and data extraction from the web.

Initially in the first stage, SmartCrawler performs site-based searching for centre pages with the help of search engines, and avoids visiting a large number of pages. To find more detailed results for a focused crawl, SmartCrawler used ranking method in which websites are rank to prioritized highly relevant once for a given data or topic. In-site exploring stage, SmartCrawler gives fast in-site searching to get most relevant links with an adaptive link-ranking.

In Site locating technique it uses reverse searching method and incremental two-level site ranking method for detecting relevant sites and to find more data sources. In second stage, a link tree is designed for a well-balanced link for prioritizing and eliminating bias towards web-pages in popular directories. In Web navigation step different hidden files are searched across the given website. The site is categorised as relevant, non-relevant, bad link, good link. The pages which has .aspx extension that file will be store in database.

2. Related Work

For area of Web crawlers many literature are available. In this section we survey the some papers which are given below: [2] The authors in the paper are Dong Xin, Yeye He, Venkatesh Ganti, Sriram Rajaraman, Nirav Shah gives method to crawl deep web information. Deep- web crawl is the problem of surfacing rich information behind the web search interface of different sites across the Web. Deep web crawling is used for various reasons such as data integration and web indexing. It is hard to crawl deep web data. Authors focuses on entity-oriented deep-web sites. The examples for these deep websites are online shopping sites (eg. Ebay, amazon etc) where each entity is a product which is associated with rich struc- tured information like brand name, item name, price, quantity and so forth.[7] There are differ- ent types of websites like movie site, job search- ing sites are included in entity oriented deep- web sites. [4]The authors in the paper are Bin He, Kevin Chen-Chuan Chang and Zhen Zhang propose MetaQuerier method. The Web has been quickly deepen by myriad searchable on- line databases, where data are hidden behind query interfaces. MetaQuerier system is used for both finding and integrating databases on the Web. The goal of the MetaQuerier system to access deep web systematically and to use deep web uniformly. [5] This will help users query online database. There are some challenges in copying of large scale data. The deep web is large collection of durable. Authors Juliana Freire, Luciano Barbosa propose method for searching hidden web databases. As per recent studies hidden web contains 7500 and 91850 terabytes of information. The volume of hidden web information is grows. Authors discuss the problem how to locate the searchable forms well. The searchable forms are entry points for hidden web data. [6] Authors Martin Van den Berg, Soumen Soumen Chakrabarti, and Byron Dom proposed focused crawling strategy.

As the web data grows day by day authors gives method hypertext resource detection system called focused crawler. The objective of this method to find out relevant information ac- cording to given topic. The topics are specified using exemplary documents not by keywords. To answer all possible ad-hoc queries, instead of collecting, indexing all available Web sites. Focused crawler used crawl boundary to catch the links that are similar to be most proper for the crawl, and escapes inappropriate regions of the Web. [4] For goal directed crawling authors proposed two hypertext mining programs that guide crawler: The first is a classifier which calculates the significance of a hypertext document according to the focus topics, and second is a distiller that identifies hypertext nodes which are a great access points to many appropriate pages within a few links.

3. Existing Work

Fig 1 shows working of SmartCrawler in two step. There are two stages of existing system first is Site locating and another is In-site exploring. In site locating stage discovers the most appropriate site for a given topic, and then the second in-site exploring stage discovers searchable forms from the site. The site locating stage choose a seed set of sites in a site database. Seeds sites are candidate sites given for SmartCrawler to start crawling, which be- gins by following URL from chosen seed sites to discover other pages and other domains

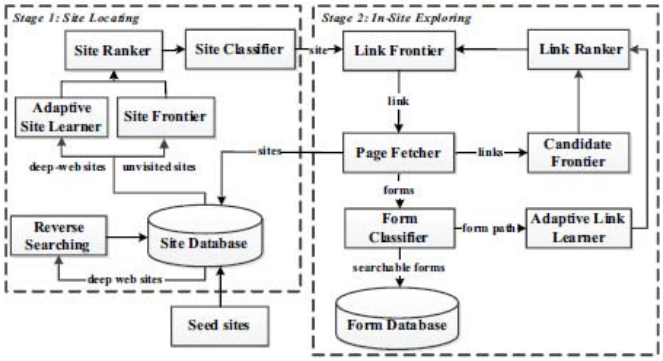


Figure 1: Existing System Architecture

4. Proposed Work

To answer the problem of searching for hidden- web information, SmartCrawler is given three- stage architecture. In proposed system there are following steps are included by which system performance is upgrade. In site locating step, it finds more relevant links for a given topic. In second step that is In-site searching or exploring discovers searchable data from website. In third step that is web navigation it search for more accurate result or websites which are actually worked. In proposed system Site frontier fetch all URL from site database. Site database contains all the URL of site. By default in any other crawler site database has large records because there database contains all other unwanted files like .css, .html, .php etc. But in Web navigation step only .aspx files are stored in site database so the site database size is small as compared to other crawler. Filter link module is used to filter the links in which .aspx files are stored in database. Other files are not stored on site database. There are various types are links are stored in database. Relevant links are those which has proper path of candidate links. It means that there is no missing links between candidate links. Bad links means which has not proper path of candidate link. It means when we click on bad link it will rise error message like could not find or forbidden error. Candidate links means it search from main link. For example in college site there are number of departments are included and each department has its own link so this links are candidates of main college link. Dataset generator is module of Web navigation method in which data are stored according its type. By designing this proposed system site database size will small in amount and time required to search new links will fast. The fig shows the architecture of proposed system of SmartCrawler.

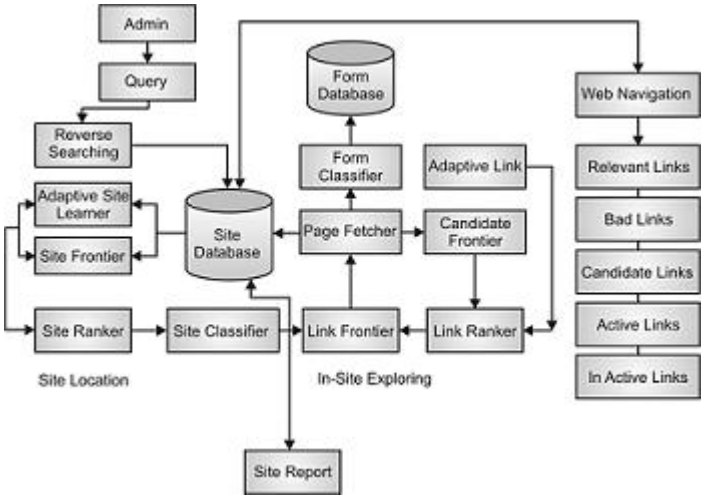


Figure 2: Architecture of Proposed System

5. Algorithm

Automatic Feature Selection

1: Input: set of links at distance d from a relevant form
2: Output: features selected in the three featureSpaces, anchor, URL and around
3: for each featureSpace do
4: termSet = getTermSet(featureSpace, paths)
{From the paths, obtain terms in specified feature space.}
5: termSet = removeStopWords (termSet)
6: stemmedSet = stem(termSet)
7: if featureSpace == URL then
8: topKTerms= getMostFrequentTerms(stemmedSet, k)
{Obtain the set of k most frequent terms.}
9: for each term t 2 topKTerms do
10: for each term t0 2 stemmedSet that contains the substring do
11: addFrequency(stemmedSet,t,t0)
{Add frequency of t0 to t in stemmedSet.}
12: end for
13: end for
14: end if
15: selectedFeatures = getNMostFrequentTerms(termSet)
{Obtain a set of the top n terms.}
16: end for

5.1 Mathematical Programming Model

1) MP model is applied on training data to obtain the set of new links and links to be improved.
2) From the testing data, mini session should be enhanced and that having two or more paths.
3) Each mini session any candidate link matches to one of the link if fine then user can pass through that link and improved the structure.

5.2 Mathematical Model

Let U be a universal set such that,
U = l, p, s, Tp, Id, Od, Cl, Wp, Ep, El Where,
l = [l0, l1…ln], is a set of links of a website.
p = [p0, p1 ...pn], is a set of paths traversed.
s = [s0, s1 ...sn], is a set of Sessions.
Tp = [Tp0, Tp1 ...Tpn], is a set of Target pages.
Id = [Id0, Id1 .Idn], is a set of In-degrees.
Od = [Od0, Od1, …Odn], is a set of Out degrees.
Cl = [Cl0, Cl1 ...Cln], is a set of candidate links.
Wp = [Wp0, Wp1,....Wpn], is a set of web pages.
El = [El0, El1, …Eln], is a set of Enhanced links.

6. Result

SmartCrawler works in different modules in first module that is User Query Request, user send any query with encrypted key to admin. Then admin check the query send by user and give appropriate link id to user. To answer the user query admin perform web navigation step, In this second module that is Web navigation, the url of any website is search, the number of sites hidden links associated with url are search. After that parsing is done in which links are classified on their specification? For e.g.: .aspx, .xml, .css. After parsing unique id is assigned to each link.

For eg. : the input url is http://www.metcolleges.ac.in
Parsing method output for this url is :

http://www.metcolleges.ac.in/achieve.php http://www.metcolleges.ac/contactus.php HTTP://WWW.METCOLLEGES.AC.IN/EVENT.HTML http://www.metcolleges.ac.in/ibm.php.

The next step is dataset generation. In this step ID of each link is assigned by the SmartCrawler. For eg. : for above given link following dataset is generated.

```
0 ( 0):
1 ( 9 ): 15 20 21  22 9 10 23  11  5
2 ( 9 ): 15 20 21  22 9 10 23  11  5
3 ( 9 ): 15 20 21  22 9 10 23  11  5
4 ( 9 ): 15 23 20  21 22  9 10 11  5
5( 9  ): 15 22 20  21 9 10 23  11  5
6 ( 9 ): 15 20 21  22 9 10 23  11  5
7 ( 9 ): 15 20 21  22 9 10 23  11  5
8 ( 8 ): 15 22 23  21 20  9 10 11
9 ( 9 ): 15 20 21  22 9 10 23  11  5
10 ( 9 ): 15 20 21  22 9 10 23  11  5
11 ( 9 ): 15 20 21  22 9 10 23  11  5
12 ( 8 ): 15 20 21  22 9 10 23  11
13 ( 8 ): 15 20 21  22 9 10 23  11
14(  8 ): 15 20 21  22 9 10 23  11
15(  9 ): 15 20 21  22 9 10 23  11  5
16 ( 1 ): 15
17 ( 1): 15
18 ( 1): 15
19 ( 1): 15
20 ( 9 ): 15 20 21  22 9 10 23  11  5
21(  9 ): 15 20 21  22 9 10 23  11  5
```

Analysis of each node is done by SmartCrawler. Following is the output of analysis step:

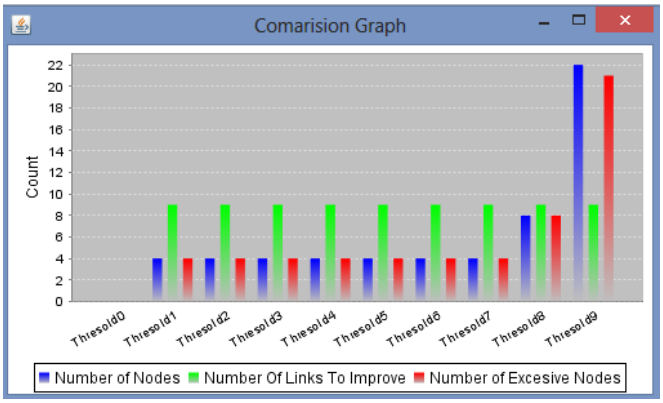
```
node =0 outDegree =0 Relevant link available =0
node =1 outDegree = 9
Relevant link available = {1, 15}
{1, 20} {1, 21} {1, 22} {1, 9}
{1, 10} {1, 23} {1, 11} {1, 5}
```

ID of eack node can be search by threshold value. For above given relevant link

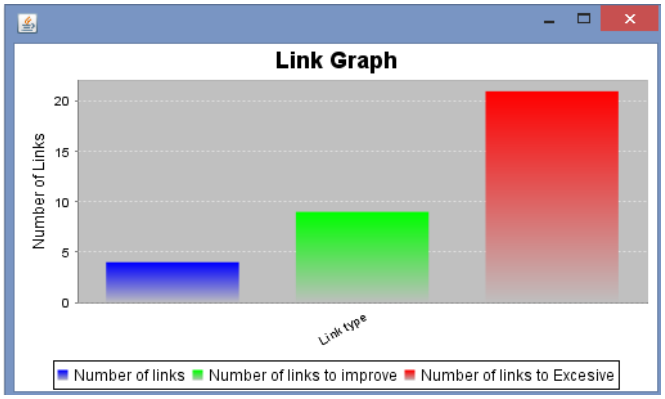
If Threshold value =3 & destination node = 9then output is :

```
Node: 1 link:   {1, 20, 9}
Node: 1 link:   {1, 21, 9}
Node: 1 link:   {1, 22, 9}
Node: 1 link:   {1, 9, 9}
Node: 1 link:   {1, 10, 9}
Node: 1 link:   {1, 11, 9}
Node: 1 link:   {1, 5, 9}
Node: 2 link:   {2, 20, 9}
Node: 2 link:   {2, 21, 9}
Node: 2 link:   {2, 22, 9}
Node: 2 link:   {2, 9, 9}
```

The graph is generated according threshold value of destination node. For above given url graph will be as follows:



One more graph of links are generated according to links to be improved and number of links and number of links to excessive.



7. Conclusion

Smart-Crawler is proposed for efficiently gathering deep web interfaces. SmartCrawler is a focused crawler which has two different stages: site locating and balanced in-site exploring. Deep websites are reverse searched by SmartCrawler for centre pages, when the threshold value is more than the number of not visited URLs in the database during the crawling process. The task of site frontier is to fetch web-page URLs from the site database. If there are un-visited sites, then they are handed to site frontier and are arranged by site ranker. Adaptive Site Learner improved the Site Ranker during crawling. The proposed system is used to advance the website structure for effectual user navigation and routing, while optimizing the cost of user's disorientation due to substantial changes in the current structure, a challenge in literature. The results shows that the modules of system are used to analyses the current website structure by displaying the available links as well as to determine the direct links to target pages, which will be further used to recommend the links to be added/enhanced as per the path threshold of mini sessions and optimized Out-degree threshold of web pages respectively. Existing System only shows the relevant links but parsing of links cannot be done. The size of existing system database is very large. The proposed system database can save only actual links so space complexity is reduced.

In future work, new method can be used to minimize the search time of reverse searching algorithm.

8. Acknowledgement

It gives me proud privilege to complete this survey paper under the valuable guidance of Prof. Dr. S. B. Sonkamble. I am also extremely grateful to Prof. R. H. Kulkarni (H.O.D of Computer Engineering) and Dr. D. M. Yadav. JSPM NTC Director who provided with all the facilities and helped for smooth progress of survey paper. For this I would also like to thank all the Staff Members and Management of Computer Engineering Department, friends and my family members, who have directly or indirectly guided and helped me for the preparation of this paper and gave me an endless support right from the stage the idea was conceived.

References

[1] Peter Lyman and Hal R. Varian. ", How much information? 2003." in ITechnical report, UC Berkeley, 2003.
[2] Roger E. Bohn and James E. Short."How much information?2009 report " in american consumers. Technical report, University of California, San Diego, 2009.
[3] Yeye He, Dong Xin, Venkatesh Ganti, Sriram Rajaraman, and Nirav Shah. "Crawling deep web entity pages " in In Proceedingsof the sixth ACM international conference on Web search and data mining, pages 355364. ACM, 2013.
[4] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang,"Toward large scale integration: Building a metaquerier over databases on the web " in In CIDR, pages 4455, 2005.
[5] Luciano Barbosa and Juliana Freire. " Searching for hidden-web databases," in In WebDB, pages 16, 2005.
[6] Soumen Chakrabarti, Martin Van den Berg, and By- ron Dom., "Focused crawling: a new approach to topic-specific web resource discovery," in Computer Networks, 31(11):16231640, 1999.
[7] Jayant Madhavan, David Ko, ucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy,""Googles deep web crawl,"" in Proceedings of the VLDB Endowment, 1(2):12411252, 2008.
[8] Olston and M. Najork, "Web Crawling,"in Foundations and Trends in Information Retrieval, vol. 4, No. 3, pp. 175246, 2010.
[9] M. Burner, ,"Crawling towards Eternity: Building an Archive of the World Wide Web,,"Web Techniques Mag- azine, vol. 2, pp. 37-40, 1997.
[10] Soumen Chakrabarti, Martin Van den Berg, and Byron Dom.,"Focused crawling: a new approach to topic-specific web resource discovery,,"Computer Networks, 31(11):16231640, 1999.
[11] Allan Heydon and Marc Najork,"Mercator: A scal- able, extensible web crawler.,"World Wide Web Conference, 2(4):219229, April 1999.
[12] Michael K. Bergman,"White paper: The deep web: Surfacing hidden value,,"Journal of electronic publishing, 7(1), 2001.
[13] D. Dhyani, W.K. Ng, S.S. Bhowmick, ,"A Survey of Web Metrics,,"ACM Computing Surveys, Vol.34, No. 4, pp. 469-503, 2002.

Author Profile

Sphurti Bhosale received B.E degree in Information Technology Engineering from Bharati Vidyapeeth's Women College of Engineering from Savitribai Phule Pune University, India and pursuing ME degree in Computer Science and Engineering from JSPMs Rajarshi Shahu College of Engi- neering NTC from Savitribai Phule Pune University, India.

Dr. Sulochana Sonkamble received B.E. degree in Computer Science and Engineering.in 1996, M.E. in 2002 and Ph. D. in 2013 from Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, Mahrashtra state, India. She is distinguished Professor in Computer Engineering Department at JSPM Narhe Technical Campus, under the Savitribai Phule Pune University Pune, Maharashtra state, India. This author became a Member of IEEE in 2006 is member of Computer Society of India and life member of Indian Society for Technical Education. The author have published and/or presented papers in International journal and National and International Conferences. Author has completed the Research Projects funded by SP Pune University, Pune. Also the author has organized National level, State level workshops / Seminars funded by SP University of Pune Maharashtra, India. Her research interest includes cloud computing, machine learning, computer vision and pattern recognition.