DwCB - Architecture Specification of Deep Web Crawler Bot with Rules Based on FORM Values for Domain Specific Web Site

S.G. Shaila, A. Vadivel*, R. Devi Mahalakshmi, and J. Karthika

Multimedia Information Retrieval Group,
Department of Computer Applications,
National Institute of Technology, TamilNadu, India
{shaila,vadi,devimaha,karthika}@nitt.edu

Abstract. It is well-known that obtaining deep web information is challenging task and it is required to choose suitable query values for crawling large data source. In this paper, we have proposed architecture specification of a deep web crawler with effective FORM filling strategy using rules. The rules are constructed by analyzing the FORM and combination of parameters. These FORM parameters are classified as most preferable, least preferable and mutually exclusive. For each successful FORM submission, the deep web data is extracted and indexed suitably for information retrieval applications. The performance of the crawler is encouraging when compared to a conventional surface crawler.

Keywords: Hidden web crawlers, Domain specific, Rule Set, Surface web, FORM values.

1 Introduction

Deep web sources can be dynamically retrieved based on the user's query using a well manageable crawler. The Deep web Crawler must interact with the FORM and have prior knowledge about the domain which improves in FORM filling capabilities and crawl more information. Since, the immense source of information buried into deep web, the first deep web crawlers is proposed with the complexity of interacting with web search interfaces [6]. Thus, crawlers are developed for extracting data from deep web databases [2] and to match deep web query interfaces [8]. A Source-Biased Approach [1] has used source-biased probing techniques, to allow interactions to decide whether a target database is relevant to the source database by probing the target with very precise probes. In [9], pages from Internet are generated dynamically and are structured to provide structured query interfaces and results. This approach works on single domain deep web resources. However, this process consumes longer time and sometimes valuable data may get ignored to find the probability of schema.

* Corresponding author.

V.V. Das and P.M. El-Kafrawy (Eds.): SPIT 2012, LNICST 117, pp. 191–196, 2014.

In [5], crawler autonomously discovers and downloads pages from the hidden web. The set covering problem is implemented for filling the FORM. However, working with such algorithm iteratively is a time consuming and downloading every page needs enormous amount of repository space. Deep web adaptive crawling based on Minimum Executable Pattern (MEP) [3] is proposed where minimal combination of elements in a query FORM is used for making successful query. Though, MEP is an efficient approach, it has a limitation on the size of result set. Hidden Web Exposer (HiWE) [6] has been developed, which interacts with FORM and a customized database to perform FORM filling. It is based on layout based information extraction technique. However, HiWE fails to recognize and respond to simple dependencies between FORM elements and lack of partial filling out the FORM. Vision based data-extraction (ViDE) [7] has focused on visual information of web pages and their implementation of web data extraction procedure. This approach involves both DOM tree analysis and visual analysis. When a website consists of web pages with visual dissimilarity, VIDE may not be automated. Fiva-Tech [4] is a page level web data extraction technique and applies tree matching, tree alignment and mining techniques. In a webpage, fixed pattern is identified and matched using a tree mining approach for extracting information. It has been observed that larger time is required to carry out all these tasks. Based on the above discussion, it is noticed that most of the work extracts information from deep web at the cost of huge processing time. Also, none of the approach has an efficient indexer, which can be effectively combined to use the crawler ideal time for indexing the already extracted information. Thus, it is imperative that a deep web crawler is required with architecture specification to crawl the surface web, deep web and with indexing capability. In addition, most of the approaches consume large time for processing FORM and it would be appropriate to analyze the FORM to acquire prior knowledge to reduce the FORM processing time. In this paper, these issues are handled and a suitable crawler architecture specification is proposed. This crawler has shown the ability in indexing, analyzing and mining web content from the hidden database.

The rest of the paper is organized as follows. The proposed work is presented in the next Section. In section 3, we present the experimental results and conclude the paper in the last section.

2 Proposed Work

In this paper, we propose an architecture specification of **D**eep **w**eb **C**rawler **B**ot (DwCB) and is shown in Fig. 1. From WWW, the URL pages are fetched and parsed with the help of URL fetcher and URL parser. The parsed web pages are verified for the FORM existence. The pages, which are not having FORM are crawled by Surface Web Crawler. The pages with the FORM are crawled by Deep web Crawler. Input elements of the FORM are analyzed and the Label-Value-Set (LVS) relationship of FORM is found out, using LVS manager.

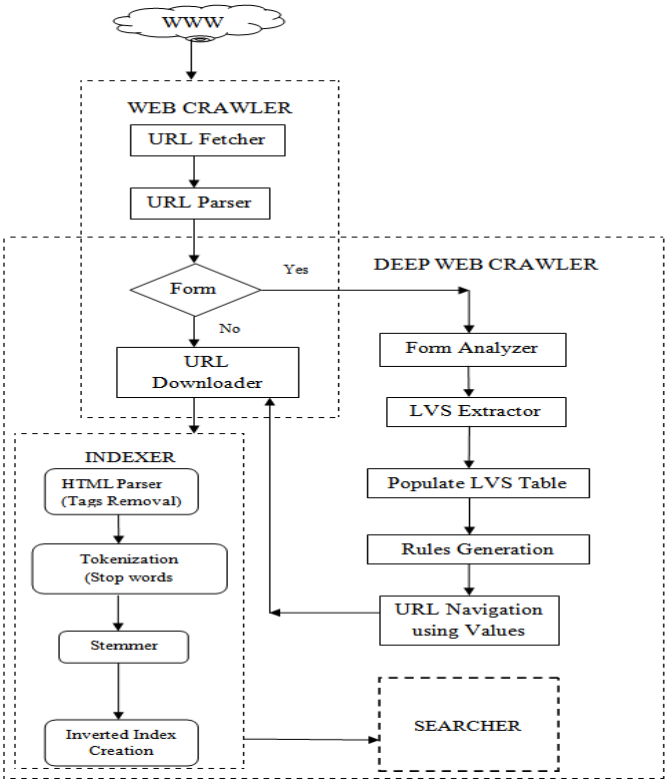

Fig. 1. Architecture Specification of DwCB

Using the LVS content of FORM, rules are generated and every FORM elements have their corresponding values. Based on the FORM analysis, prior knowledge is acquired and the values are populated to the FORM elements. Mathematically, we can represent L as label and *V={v1,v2,v3….vi}* is a fuzzy graded set of values. The values will be assigned to corresponding labels L using V. Intuitively, each *vi* represents a value that could potentially be assigned to an element ***e,*** if label: *e* "matches" L. Labels can be aliased i.e. two or more labels can share the same value set. Rules are generated depending on the LVS. The combinations of the values (Rules) that are populated should be navigated for crawling. The respective URLs are downloaded and the result is given to Indexer for indexing. We have considered real estate domain (http://www.99acres.com) and crawled the data from it. The rules for residential apartment and villa are shown in Table.1. Here, the rules depending on number of bedrooms in the apartment and villa are generated. The residential apartment (P_1) and villa (P_2) are considered in the property category, in bed room category eight levels [B_1-B_8] are considered and in the budget category seven levels of cost prices [R_1-R_7] are considered for generating the rules.

Table 1. Rules for real estate domain

`	**Rules**
1	If (P_1&&R_1)then B_1-MP; $B_{2,3}$ -LP; others-ME
2	If (P_1 && R_2) then $B_{1,2,3}$ - MP; $B_{4,5}$ - LP; others - ME
3	If (P_1 && (R_3 || R_4 || R_5 || R_6 || R_7)) then $B_{1,2,3,4}$ -MP; B_5- LP; others - ME
4	If (P_2 && R_1) then $B_{1,2,3}$- MP; B_4 – LP; others - ME
5	If (P_2 &&(R_2 || R_3|| R_4) then $B_{1,2}$- MP; B_3- LP; others - ME
6	If (P_2 && (R_5 || R_6) then $B_{1,2,3,4}$ - MP; $B_{5,6}$ - LP; others - ME
7	If (P_2 && R_7) then $B_{3,4,5,6}$ - MP; $B_{1,2,7}$ -LP; others – ME
Where, MP - Most Preferable, LP - Least Preferable, ME - Mutually Exclusive.	

For instance, as per rule 1, the FORM is filled with values (P_1) as "residential apartment" with price (R_1) as "twenty lacks". Given these values, the *most preferable* value for this field is 1-bedroom (B_1) and *least preferable* for 2-bedroom (B_2) and 3-bedroom (B_3). In contrast, B_4, B_5, B_6, B_7 and B_8 are *Mutually Exclusive* values, i.e. this combination returns no result. Here, we have constructed this rule by giving different property locations.

3 Experimental Results

We carried out a number of experiments to study and measure the performance of DwCB. We present experimental result to compare performance of surface web crawler and DwCB. The number of web pages crawled by surface web crawler and deep web crawler for a deep website with respect to crawling time is considered. It is observed from the below graph that for any Deep Web resource, DwCB provides better result compared to a surface web crawler. Fig. 2, represents the number of documents crawled with respect to time. We have experimented for website http://www.99acres.com. As indicated in Fig. 2, DwCB crawled 2,100 web pages and surface web crawled around 250 web pages in 2 hours.

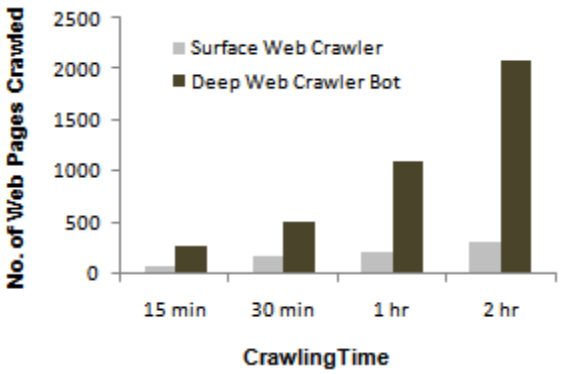

Fig. 2. Performance of DwCB compared to SWC

Another experiment is also carried out to find the number of relevant document retrieved in any particular context. A set of keywords is used as query and searched into both documents crawled by surface web crawler and DwCB. It is observed that the number of documents retrieved by DwCB is high compared to surface web crawler and is depicted in Fig. 3. However, as more FORMs are processed, the crawler encounters a number of different finite domain elements and is able to contribute new entries to the LVS table. In addition, the LVS manager uses these new entries to retrieve additional values from the data sources. As a result, at the end, the crawler successfully contributes crawling directly, almost a 2000 additional successful web pages.

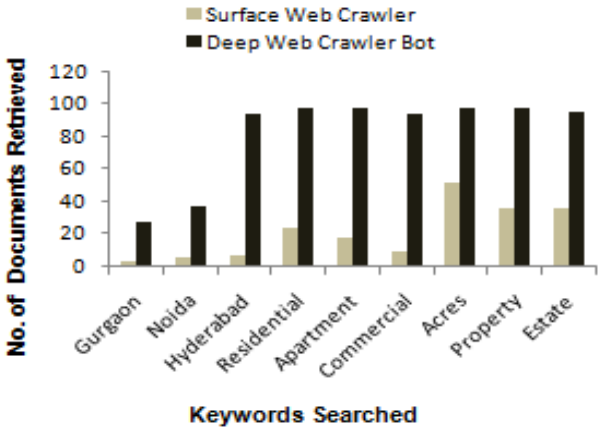

Fig. 3. Comparison ratio of response of DwCB Vs. Surface Web Crawler

4 Conclusion and Future Works

DwCB is provided with an enriched automated method to fill FORMs based rules for domain specific Web site. We have developed the system and extracted information from http://www.99acres.com. For this domain, the FORM is fetched, analyzed and a prior knowledge is acquired. During FORM filling procedure, rules are generated and the FORMs are filled with suitable value. The proposed architecture specification crawls more relevant documents in lesser time. The performance is superior compared to the conventional crawler. In future, we are trying to populate the LVS table dynamically depending on the domains chosen and thereby automating the rule generation.

Acknowledgement. The work done is supported by research grant from the Department of Science and Technology, India, under Grant DST/TSG/ICT/2009/27 dated 3rd September 2010.

References

1. James, C., Ling, L., Daniel, R.: Discovering Interesting Relationships among Deep web Databases: A Source-Biased Approach. World Wide Web 9(4), 585–622 (2006)
2. Craswell, N., Bailey, P., Hawking, D.: Server selection on the World Wide Web. In: Proc. of the Fifth ACM conference on Digital Libraries (ACM DL F00), San Antonio (2000)
3. Liu, J., Jiang, L., Wu, Z., Zheng, Q.: Deep Web adaptive crawling based on minimum executable pattern. Journal of Intelligent Information Systems 36, 197–215 (2011)
4. Mohammed, K., Chia-Hui, C.: FiVaTech: Page-Level Web Data Extraction from Template Pages. IEEE Trans. Knowl. Data Eng. 22(2), 249–263 (2010)
5. Alexandros, N., Petros, Z., Junghoo, C.: Downloading Hidden Web Content, Technical Report, UCLA (2004)
6. Raghavan, S., Garcia-Molina, H.: Crawling the hidden web. In: Proc. of the 27th International Conference on Very Large Databases (VLDB F01), Rome (2001)
7. Liu, W., Meng, X., Meng, W.: ViDE: A Vision-Based Approach for Deep web Data Extraction. IEEE Transactions on Knowledge and Data Engineering 22(3), 447–460 (2010)
8. Wu, W., Yu, C.T., Doan, A., Meng, W.: An interactive clustering-based approach to integrating source query interfaces on the deep web. In: Proc. of the 2004 ACM Conference on Management of Data (SIGMOD F04), Paris (2004)
9. Zhao, P., Li, H., Wei, F., Zhiming, C.: Organizing Structured Deep web by Clustering Query Interfaces Link Graph. ADMA, 683–690 (2008)