Deep Webpage Classification and Extraction (DWCE)

Meenakshi Sharma
Department Of Computer Science,
Sscet Badhani, Pathankot, India

Supriya
Department Of Computer Science,
Sscet Badhani, Pathankot, India
supriyaangra.87@gmail.com

Abstract

As the Deep web (or Hidden web) information is hidden behind the search query forms, this information can only be accessed by interacting with these forms. Therefore, development of automated system that interacts with the search forms and extracts the hidden web pages would be of great value to human users. To accomplish this task stated above, this paper proposes a novel method "*Deep Webpage Classification and Extraction*" which classifies the websites into appropriate domain, extracts their query interfaces and retrieves all result pages of deep websites using query building system.

Keywords: Deep Web, Search query interface, Query processors, Crawler.

1. INTRODUCTION:

Mining information from the Web and identifying relevant resources which match a query is studied in the field of Information Retrieval (IR)[1]. Today, it is impossible to imagine the Web without search engines. However, traditional search engines too have some limitations like they only returns the result pages that are already gathered and pre-processed by crawlers. This technique is efficient for the static web pages, which remain same for longer periods of time.

Let's suppose a user wants to buy a property. He/She fills the search box of search engine with "buy property". Search engine will return the result page containing result indexes as shown in figure 1.1

User has to open the link one by one and then he has to fill the search form of each and can get the result. But after getting these result pages, he has to analyze the page for his requirement. This shows that data actually resides inside the databases but it is not surfaced by the search engine. This means the result pages are not yet crawled by the web crawler. The paper has been organized as follows: The second section of this paper has proposed work in which the proposed architecture of DWCE and its working is described where a website classifier classifies the web pages into appropriate domain category and discards the surface web pages.

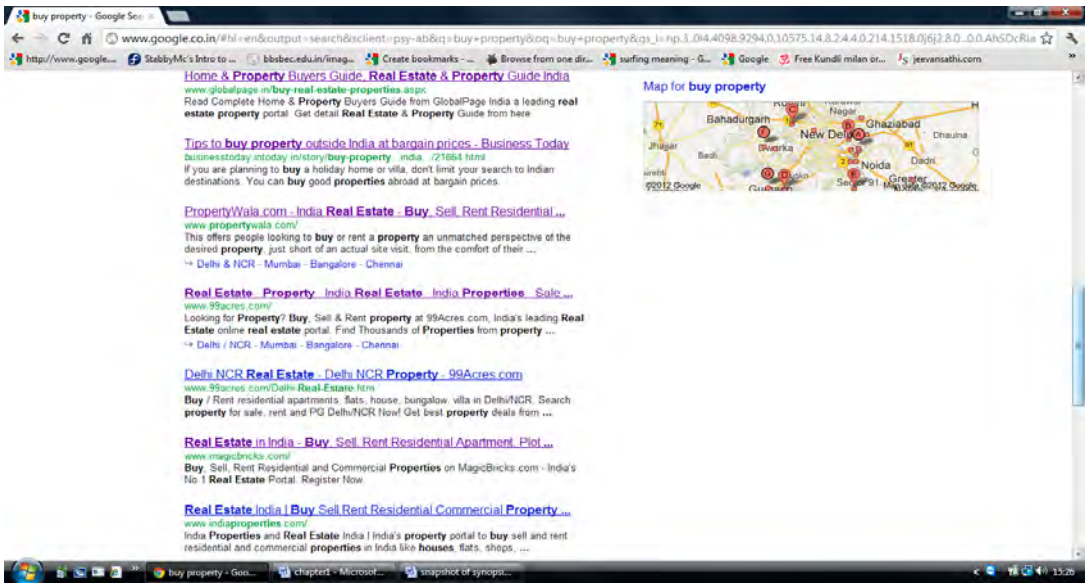

Fig 1.1: Google result page for query "buy property"

After collecting the domain specific pages, system will extract all the search query interfaces and then choose the interface with maximum number of fields. This interface becomes the base interface and user has to fill only this interface. This design is capable of extracting the result pages efficiently and accurately. Experiment and results are included in third section. Section 4 draws the conclusion and describes the future research.

2. PROPOSED WORK:

Search query interfaces and dynamically generated pages are helpful to users because users can get the desired information they want. However, it is really tiresome task for users to visit all the hidden web sites for the same domain and fill out different forms provided by each site. There is a need to develop a technique that can automatically fills the search interfaces of all the Deep websites and displays the result pages[16].The complete architecture of a Deep Webpage Classification and Extraction is shown in figure 2.1.

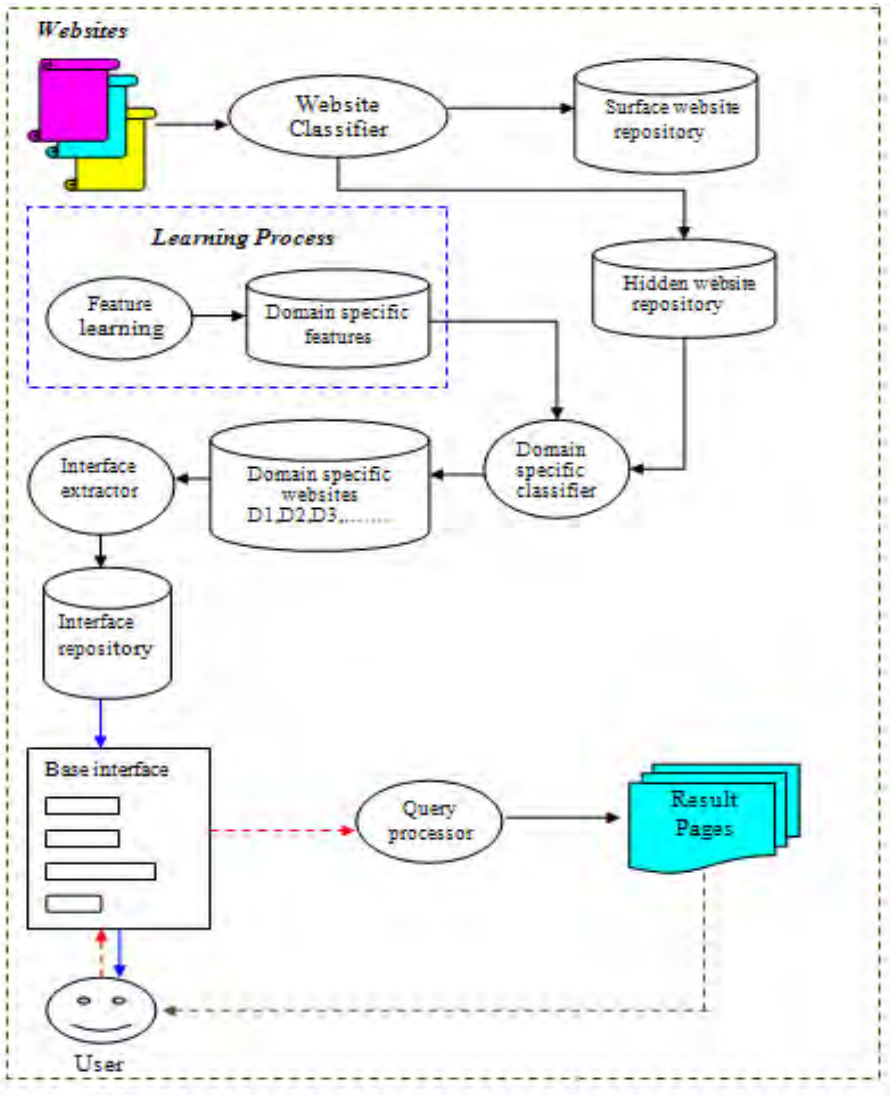

Fig 2.1: Architecture of DWCE

2.1 Website Classifier:

Algorithm web_classifier(URL list)

Step1. Pick one by one URL from URL list
2. Check for string for the "http:" in URL.
   If (present) then
   {"URL is valid "};
   Else
   {"URL is invalid "};
3. Download the source code and save.
4. check the source code for tag <form> and </form> // Form analyses
   If (present) then
   {"Site is Deep Website"};
   Else
   { " Site is Surface Website"}.
5. Collect list of URLs i.e list of Deep web sites.

Fig 2.2: Algorithm for website classifier

Predetermined URL list is given as input to *Website classifier* module. This module checks for http string in URL. If it is present then it is valid URL otherwise it cannot be opened. For valid URLs, module will download the source code. Since query interfaces will have <form> and</form> tags in the source code, this module will check for these tags in the code and classifies the sites according to that.

2.2 Learning Process:

Learning process starts by training next classifier i.e Domain Specific website classifier for domain specific properties. Now, Domain specific website classifier classifies the websites by looking at these features which will be discussed in next section.

2.3 Domain Specific Website Classifier:

This classifier classifies all Deep web sites into appropriate domains(Car, Real Estate, Shopping).Algorithm for this classifier is shown in figure 2.3.

Algorithm Domian_spec_classifier(URL list)

Step1. Pick one by one URL from URL list
2. String meta= Download the source code and save.
3. check the source code for attributes for all domains // Feature analyses

```
if (meta.Contains("new cars") || meta.Contains("used cars") || meta.Contains("car prices") ||
meta.Contains("compare cars")) then
        ("Add URL to Domain :Car");
else if (meta.Contains("property") || meta.Contains("properties") || meta.Contains("real estate") ||
meta.Contains("india property") || meta.Contains("rent property")) then
        ("Add URL to Domain :Real Estate");
else if (meta.Contains("online Shopping") || meta.Contains("india") || meta.Contains("electronics") ||
meta.Contains("clothing") || meta.Contains("mobile phones") || meta.Contains("digital cameras")) then
        ("Add URL to Domain :Shopping");
```

Fig 2.3: Algorithm for Domain Specific Website Classifier

2.4 Interface Extractor:

This component collects all interfaces of particular domain in repository. After selecting the particular domain, this extractor chooses the interface that contains maximum number of fields. This interface becomes the base interface and user has to fill only this interface.

2.5 User Interaction:

From all the interfaces for each domain, Base interface has been chosen which is the interface that contains maximum number of attributes. Users can enter their queries in this search form and the values for each field are provided to Query processor module The query is processed by *Query processor* and relevant results returned thereof are provided to the user.

2.6 Query Processor

When user fills the Base interface form with his/her query, *Query Processor* processes this query, forms the query string URL and then submits the form to find the result page. Algorithm for query processing is shown in figure 2.4.

Algorithm query_process (URL list)

Step1. Pick one by one URL from URL list

2. Extract the values of all text boxes T1,T2, T3,…..
3. Append URLs with the values of these text boxes

// URL/T1/T2/T3/……

4. Submit URL.
5. Get the result pages.

Fig 2.4: Query Processing algorithm

3. EXPERIMENT AND RESULTS :

This prototype is implemented on deep websites of real estate domain and results are shown below.

3.1 ***Website Classifier*** – It automatically classifies the websites into Deep web sites and Surface web sites.

Fig 3.1: Website Classifier

3.2 Interface Extractor and Base interface formation –

Fig 3.2: Base Interface

3.3 *Query Processor*

When user fills the Base interface form with his/her query, *Query Processor* processes this query, forms the query string URL and then submits the form to find the result page.

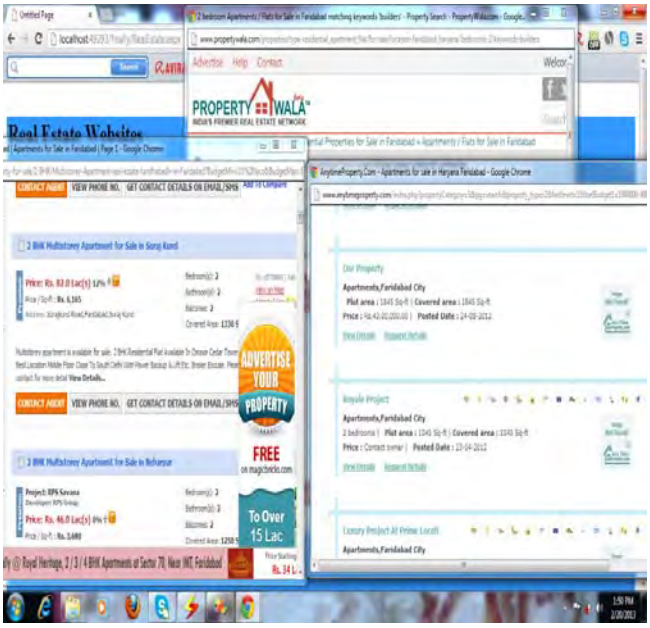

Fig 3.3: result pages from Live Deep websites after filling base interface form

4. CONCLUSION AND FUTURE SCOPE:

Deep Web data extraction is a major challenge nowadays. Because of inability of current crawler to interact with deep websites, traditional search engine has now become an ineffective way to search this kind of data. In this paper, an automatic and domain dependent prototype system is proposed. It automatically classifies the websites into Deep web sites and Surface web sites. The Graphical user interface is designed for user interaction where the user can fill the query in the form and find the desired result pages. Although this system extracts the web pages from various deep websites successfully, this work could be extended in near future for more domains and deep web directory or deep web search engine can be made.

REFERENCES:

[1]. Ricardo Baeza-Yates and Berthier Ribeirmobileo-Neto,"Modern Information Retrieval", ACM Press/ Addison-Wesley, 1999.
[2]. Madhavan, J., Cohen, S., Dong, X. L., Halevy, A. Y., Jeffery, S. R., Ko, D., and Yu, C. 2007. 'Web-Scale Data Integration: You can afford to Pay as You Go". In CIDR. 342–350.
[3]. Bergman, M.K. (2001). "The Deep Web: Surfacing Hidden Value". In The Journal of Electronic Publishing, Vol. 7, No. 1.
[4]. A. Ntoulas, P. Zerfos, and J. Cho, "Downloading Hidden Web Content Through Keyword Queries", ACM transaction on JCDL'05 Denver, Colorado, USA, June.2005.
[5]. Sriram Raghavan and Hector Garcia-Molina."Crawling the hidden web". In Proceedings of the International Conference on Very Large Data Bases, pages 129{138, San Francisco, CA, USA, 2001.
[6]. Kevin Chen-Chuan Chang, Bin He, Chengkai Li, Mitesh Patel, and Zhen Zhang."Structured databases on the web: observations and implications". SIGMOD Record, 33(3):61{70, 2004.
[7]. Bin He, Mitesh Patel, Zhen Zhang, and Kevin ChenChuan Chang. "Accessing the deep web. Communications of the ACM", 50(5):94{101, 2007.
[8]. Barbosa, L., and Freirel, J.(2004)."Siphoning Hidden- Web Data through Keyword-Based Interface"., In Proceedings of SBBD.
[9]. Milan C Pandya, "A domain based approach to crawl the hidden web",Georgia State University, 2006.
[10]. Andreas Paepcke, and Sriram Raghavan, "Searching the Web", ACM Transactions on Internet Technology (TOIT)", 1(1):2–43, August 2001.
[11]. Sherman, Chris, and Gary Price. 2001. "The Invisible Web: Uncovering Information Sources Search Engines Can't See". Medford, NJ: CyberAge Books Ricardo Baeza-Yates and Berthier Ribeirmobileo-Neto, "Modern Information Retrieval", ACM Press/ Addison-Wesley, 1999.
[12]. Ritu Khare Yuan An Il-Yeol Song "Understanding Deep Web Search Interfaces: A Survey" SIGMOD Record, March 2010 (Vol. 39, No. 1).
[13]. Lage, P. et al. " Collecting Hidden Web Pages for Data Extraction". In Proceedings of the 4th international workshop on Web information and data management 2002, PP: 69-75.
[14]. Cope, J., Craswell.N and Hawking, D. Automated Discovery of Search Interfaces on the web. In Proceedings of the Fourteenth Australasian Database Conference (ADC 2003), Adelaide, Australia.
[15]. Bai, P., and Li, J.(2009). The Improved Naive Bayesian WEB Text Classification Algorithm, In International Symposium on Computer Network and Multimedia Technology, (IEEE Explorer).
[16]. Supriya, Meenakshi Sharma, "Deep Web Data Mining", dInternational Journal of IT, Engineering and Applied Sciences Research (IJIEASR) ,Volume 2, No. 3, March 2013