Analysis of DeepWeb Search Interface

Yasha goel[#], Megha Bansal[*]

[#]*Computer Science & Engineering ,*
Maharshi Dayanand University, Rohtak, India

[*]*Assistant Professor, Computer Science & Engineering,*
Maharshi Dayanand University, Rohtak, India

***Abstract*-The large amount of information on web is stored in backend databases which are not indexed by traditional search engines. Such databases are** referred **to as Hidden web databases and extraction of this hidden web content is a potential research area as the pages are dynamically created through search query interfaces. However, direct query through this search interface is laborious way to search. Hence, there has been increased interest in retrieval and integration of hidden web data with a view to give high quality information to the web user. This paper proposes a novel approach that identifies Web page templates and the tag structures of a document in order to extract structured data from hidden web sources as the results returned in response to a user query are typically presented using template generated Web pages.**

***Keywords*: Hidden web, Deep web, Global interface, Hidden web crawlers, Surface web.**

I. INTRODUCTION

Web is a huge hypertext information resource and increases dramatically [3].Web search engines fall into two main categories: general purpose search engines and vertical search engines. A general purpose search engine such as Google provides services for general search. However, a vertical search engine is domain specific information retrieval .vertical search engines have smaller and more manageable indexes .In order to get more accurate topic related and cohesive information, a vertical search engine might embrace data mining techniques to filter, classify, cluster data, find nontrivial knowledge and promote the search quality and relevance [4].Vertical search engine or domain specific search has been key area of research in recent years. The principle of QII is based on hierarchical mapping and DSHWC. As available in the literature [1], the crawling task of domain specific hidden web crawler (DSHWC) has been divided into five phases. Phase1 is concerned with the automatic downloading of the search interfaces. Phase2 employs Domain-specific Interface Mapper [2] that automatically identifies the semantic relationships between attributes of different search interfaces. Using these semantic mappings, the interfaces are then merged to form a Unified Search Interface (USI),which is then filled automatically. The filled USI is then submitted to the Web and the response pages thus generated are collected and analyzed i.e. single unified search interface is presented to the crawler and upon submission of a query via the interface, equivalent queries are submitted to many hidden web databases via front-end query interfaces and then the results are extracted from different web-sources. The advantage of this, for example, in the "airline" domain, is to prevent querying from each airline site among large airline websites which is time consuming; the second advantage is that this integration of several interfaces into a single interface presents a simple and easy query interface that need to be filled. Then data is then downloaded from several hidden airline databases.

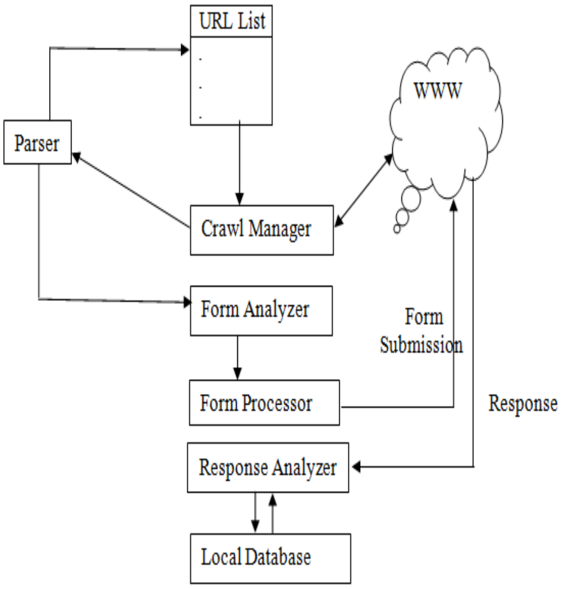

Fig 1: Architecture of deepweb crawler

II. RELATED WORK

One of the prominent works for detection of deep web search interface is done by Leo Breiman (2001)[6] in form of random forest algorithm. A random forest algorithm detects the deep web search interface by using a model, based on decision trees classification. A random forest model can be defined as a collection of decision trees. A decision tree can be generated by bootstrapping processing of the training data. Various classification trees can be generated through random forest algorithm. To classify a new object from its input vector, the sample vector is passed to every tree defined in algorithm. A decision for classification is given by every tree. A decision about most voted classification is done by using all of the classification results of the individual trees. The advantages of random forest algorithm are that it exhibits a substantial performance improvement over single tree classifiers and injecting of the right kind of randomness makes accurate classifiers and regulators. The disadvantage of this algorithm is that it may select unimportant and noisy features in the training data, as a result a bad classification results because of its random selection feature. One of the deep web crawler architecture is proposed by Sriram Raghavan and Hector Garcia-Molina (2001) [7]. In this paper, a task-specific, human-assisted approach is used for crawl the hidden web. There are two basic problems related to deep web search, firstly the volume of the hidden web is very large and secondly there is a need of such type of crawlers which can handle search interfaces efficiently, which are designed mainly for humans. In this paper a model of task specific human assisted web crawler is designed and relized in HiWE (hidden web exposure). The HiWE prototype built at Stanford which crawl the dynamic pages. HiWE is designed to automatically process, analyze, and submit forms, using an internal model of forms and form submissions. HiWE uses a layout-based information extraction (LITE) technique to process and extract useful information. The advantages of HiWE architecture is that its application/ task specific approach allows the crawler to concentrate on relevant pages only and with the human assisted approach automatic form filling can be done. Limitations of this architecture are that it is not precise with response to partially filled forms and it is not able to identify and respond to simple dependency between form elements.

A technique for collecting hidden web pages for data extraction is proposed by Juliano Palmieri Lage et al. (2002) [8] . In this technique the authors have proposed the concept of web wrappers. A web wrapper is programs which extract the unstructured data from web pages. It takes a set of target pages from the web source as an input. These set of target pages are automatically generated by an approach called "Spiders". Spiders automatically traverse the web for web pages. Hidden web agents assist the wrappers to deal with the data available on the hidden web. The advantage of this technique is that it can access a large number of web sites from diverse domains and limitation of this technique is that it can access only that web site that follow common navigation patterns. Further, modification can be done in this technique to cover navigation patterns based on these mechanisms.

A technique for automated discovery of search interface from a set of html forms is proposed by Jared Cope, Nick Craswell and David Hawking (2003) [9]. This paper defined a novel technique to automatically detect search interface from a group of html forms. A decision tree was developed with the C4.5 learning algorithm using automatically generated features from html markup that can give a classification accuracy of about 85% for general web interfaces. Advantage of this technique is that it can automatically discover the search interface. Limitation of this technique is that it is based on single tree classification method and number of feature generation is limited due to use of limited data set. As a future work, modification is suggested that a search engine can be develop using existing methods for other stages along with the proposed one with a technique to eliminate false positives.

A technique for understanding web query interfaces through best effort parsing with hidden syntax is proposed by Zhen Zhang et al. (2004)[10]. This paper addresses the problem of understanding web search interfaces by presenting a best-effort parsing framework. The paper presented a form extractor framework based on 2P grammar and the best effort parses in a language parsing framework. It identifies the search interface by continuously producing fresh instances by applying productions until attaining a fix-point, when no fresh instance can be produced. Best effort parser technique minimizes wrong interpretation as much as possible in a very fast manner. It also understands the interface to a large extent. Advantage of this technique is that it is a very simple and consistent technique with no priority among preferences and it can handle missing elements in form and limitation of this technique is that establishment of single global grammar that can be interacted to the machine globally is a critical issue.

A technique named as "siphoning hidden web data through key word based interface" for retrieval of information from hidden web databases through generation of a small set of representative keywords and build queries is proposed by Luciano Barbosa and Juliana Freire (2004) [11]. This technique is designed to enhance coverage of deep web. Advantage of this technique is that it is a simple and completely automated strategy that can be quite effective in practice, leading to very high coverage of deep web. Limitation of this technique is that it is not able to achieve the coverage for collection whose search interface fixes a number of results. Further the authors have advised that modification can be done in this algorithm to characterize search interfaces techniques in a better way so that different notions and levels of security can be achieved.

An improved version of random forest algorithm is proposed by Deng et al. (2008) [12]. In this improved technique a weighted feature selection algorithm is proposed to generate the decision trees. The advantage of this improved algorithm is that it minimizes the problem of classification of high dimension and sparse search interface using the ensemble of decision trees. Disadvantage of this improved algorithm is that it is highly sensitive towards the changes in training data set.

Further improvement in random forest algorithm is done by Yunming Ye et al. (2009) [13] by using feature weighting random forest algorithm for detection of hidden web search interface. This paper had presented a feature weighting selection process rather than random selection process. Advantage of this technique is that it makes a weighted feature selection process instead of random selection hence reduces the chances of noisy feature selection and limitation of this techniques is that features available only in the search forms were used. Future modification suggested in random forest algorithm to investigate more feature weighting methods for construction of random forests.

III. CONCLUSION

Deep Web Crawling is basically used for searching the hidden web pages which is not searched by traditional search. But sometimes this crawler is also not able to find the deep web pages. In this paper we have studied that how we can make a deep web crawler efficient ,by using dynamic query interface to generate query for searching and extract the data from the web and after searching the page it updates the rank of page and we have seen that weighted page rank formula is much better than original rank formula and which give weightage to the inlinks and outlink and increase the rank of page than a normal formula does.

ACKNOWLEDEMENT

This review paper has been compiled by me under the mentorship of Miss Megha Bansal. Her ideas have been a source of constant inspiration for me and the entire batch of students under her. This assignment given by her has given me an opportunity to search about the deepweb crawler.

REFERENCES

[1]. Bergman, M.K. (2001). *The Deep Web: Surfacing Hidden Value*. In The Journal of Electronic Publishing, Vol. 7, No.

[2]. Peisu, X., Ke, T. and Qinzhen, H.(2008). *A Framework of Deep Web Crawler*. In Proceedings of the 27th Chinese Control Conference, Kunming,Yunnan, China.

[3]. Sharma, D. K., and Sharma, A.K. (2010). *Deep Web Information Retrieval Process*: A Technical Survey. In International Journal of Information Technology & Web Engineering, USA, Vol 5, No. 1.

[4]. Khare, R., An, Y., and Song, Y. (2010). *Understanding Deep Web Search Interfaces*: A Survey. In ACM SIGMOD Record, Volume 39 , Issue 1, PP: 33- 40.

[5]. Sharma D. K., and Sharma A.K. (2009). *Query Intensive Interface Information Extraction Protocol for Deep Web*., In Proceedings of IEEE International Conference on Intelligent Agent & Multi- Agent Systems, PP. 1-5 , IEEE Explorer.

[6]. Breiman, L. (2001*). Random Forests. In Machine Learning,* Vol. 45, No.1, PP: 5-32, Kluwer Academic Publishers.

[7]. Raghavan, S. and Garcia-Molina, H. (2001). *Crawling the Hidden Web.* In Proceedings of the 27th International Conference on Very Large Data Bases, Roma, Italy.

[8]. Lage, P. et al. (2002*). Collecting Hidden Web Pages for Data Extraction*. In Proceedings of the 4th international workshop on Web information and data management , PP: 69-75

[9]. Cope, J., Craswell, N., and Hawking, D. (2003). *Automated Discovery of Search Interfaces on the web*. In

[10]. Proceedings of the Fourteenth Australasian Database Conference (ADC2003), Adelaide, Australi,a.

[11]. Zhang, Z., He, B., and Chang, K. (2004). *Understanding Web Query Interfaces*: Best-Effort Parsing with Hidden Syntax. In Proceedings of ACM International Conference on Management of Data ,PP: 107-118.

[12]. Barbosa, L., and Freirel, J.(2004). *Siphoning Hidden-Web Data through Keyword-Based Interface*., In Proceedings of SBBD.

[13]. Deng, X. B., Ye, Y. M., Li, H. B., & Huang, J. Z. (2008). *An Improved Random Forest Approach* For Detection Of Hidden Web Search Interfaces. In Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, China. IEEE.

[14]. Ye, Y., et al. (2009). *Feature Weighting Random Forest for Detection of Hidden Web Search Interfaces*. In Computational Linguistics and Chinese Language Processing , Vol. 13, No. 4, PP: 387-404.

[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.

AUTHOR

Prof. A. K. Sharma received his M.Tech. (CST) with Hons. from University of Roorkee (Presently I.I.T. Roorkee) and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi and he obtained his second Ph.D. in Information Technology form IIITM, Gwalior in 2004. Presently he is working as Dean, Faculty of Engineering and Technology & Chairman, Dept of Computer Engineering at YMCA University of Science and Technology, Faridabad. His research interest includes Fuzzy Systems, OOPS, Knowledge Representation and Internet Technologies. He has guided 9 Ph.D thesis and 8 more are in progress with about 175 research publications in International and National journals and conferences. The author of 7 books, is actively engaged in research related to Fuzzy logic