International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 |Aug -2016 www.irjet.net p-ISSN: 2395-0072

Implementation of Enhanced Web Crawler for Deep-Web Interfaces

Yugandhara Patil[1], Sonal Patil[2]

[1]*Student, Department of Computer Science & Engineering, G.H.Raisoni Institute of Engineering & Management, Jalgaon, Maharashtra, India.*
[2]*Assistant Professor, Department of Information and Technology , G.H.Raisoni Institute of Engineering & Management, Jalgaon, Maharashtra, India.*

Abstract - *As deep web grows at a very fast pace, there has been increased interest in techniques that help efficiently locate deep-web interfaces. However, due to the large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. Therefore a two-stage enhanced web crawler framework is proposed for efficiently harvesting deep web interfaces. The proposed enhanced web crawler is divided into two stages. In the first stage, site locating is performed by using reverse searching which finds relevant content. In the second stage, enhanced web crawler achieves fast in site searching by excavating most relevant links of site. It uses a novel deep web crawling framework based on reinforcement learning which is effective for crawling the deep web. The experimental results show that the method outperforms the state of art methods in terms of crawling capability and achieves higher harvest rates than other crawlers.*

Key Words: Deep web, Web crawler, Harvest rate, Reinforcement learning etc.

1. INTRODUCTION

Web crawler is a program that traverses the web based on automatic manner to download the web pages or partial pages contents according to the user needs. Web crawling is the means that by which crawler collects pages from the web. The results of crawling may be a collection of web pages at a central or distributed location. Given the continuous growth of the web, this crawled collection guaranteed to be a subset of the web and, indeed, it should be far smaller than the overall size of the web. By design, web crawler aims for a small, manageable collection that's representative of the whole internet.

Deep web or hidden web refers to World Wide Web content that's not a part of the surface web that is directly indexed by search engines. Deep web content is especially vital. Not only its size estimated as many times larger than the so-called surface web, however also it provides users with high quality data. However, to get such content of deep web is difficult and has been acknowledged as a major gap within the coverage of search engines. The deep web might include dynamic, unlinked, restricted access, scripted, or non-HTML/text content residing in domain specific databases, and private or contextual web. This content might exist as structured, unstructured, or semi- structured data in the searchable data sources. The results from these data sources can only be discovered by a direct query. The deep web content is variously distributed across all subject areas from financial information, and shopping catalogs to flight schedules, and medical analysis [3].

There is a requirement for an efficient crawler that's able to quickly and accurately explore the deep web databases [1]. Deep web refers to the hidden part of the web that is still inaccessible for standard web crawlers. To get content of Deep web is difficult and has been acknowledged as a major gap within the coverage of search engines. To this end, Lu Jiang et al. proposes a completely unique deep web crawling framework which is based on reinforcement learning, during which the crawler is regarded as an agent and deep web database because the environment [2]. The agent perceives its current state and selects an action to submit to the environment according to Q-value. The framework not only permits crawlers to find out a promising crawling strategy from its own experience, but additionally permits for utilizing various options of question keywords.

The main objective of this research is to design an appropriate system for efficiently deep web harvesting. It is estimated that deep web contains data in a scale several times bigger than the data accessible through search engines which is referred to as surface web. Although this information on deep web could be accessed through their own interfaces, finding and querying all the interesting sources of information that might be useful could be a difficult, time consuming and tiring task. So to locating the deep web, efficient web crawler is required. So the proposed enhanced web crawler helps in more efficient crawling.

The proposed enhanced web crawler works in two stages. The first stage is site locating stage in which relevant contents are identified by using reverse searching algorithm. Second stage is in-site exploring stage which relevant links are identified. This stage uses reinforcement learning framework which is efficient for deep web interfaces. This proposed solution outperforms the state of art methods in terms of crawling capability, running time and harvest rate.

2. LITERATURE SURVEY

To leverage the massive volume information buried in deep web, previous work has proposed variety of techniques and tools, together with deep web understanding and integration hidden web crawlers and deep web samplers. For all these approaches, the ability to crawl the deep web is a key challenge. Olston and Najork consistently present that crawling deep web has three steps: [5] locating deep website sources, choosing relevant sources and extracting underlying content.

Lu Jiang et. al. proposes a unique deep web crawling framework based on reinforcement learning, within which the crawler is regarded as an agent and deep web database because the environment. The agent perceives its current state and selects an action to submit to the environment according to Q-value [2]. Beyond the agent and the environment, one will identify four main sub-elements of a reinforcement learning system: a policy, a reward function, a value function, and, optionally, a model of the environment [6].

Richard S. Sutton and Andrew G. Barto consider all of the work in best control also to be, in a sense, work in reinforcement learning. They outline reinforcement learning as any effective way of solving reinforcement learning issues, and it's currently clear that these issues are closely related to best control issues, particularly those developed as MDPs. accordingly, they must consider the solution strategies of best control, similar to dynamic programming, also to be reinforcement learning ways [6].

To retrieve the complex query information remains checking for search engine is known as deep web. The deep web is invisible web contains publically accessible pages with data in info similar to Catalogues and reference that a not index by search engine [4]. The Form-Focused Crawler (FFC) [7] and Adaptive Crawler for Hidden web Entries (ACHE) [8] are focused crawlers used for searching interested deep web interfaces. Focused crawler is developed to visit links to pages of interest and avoid links to off-topic region [9].

The Deep web is rapidly growth day over day and to find them with efficiency there's need of effective techniques to achieve best result. Such a system is effectively implemented is smart Crawler, that is a two Stage Crawler efficiently harvesting deep web interfaces. By using some basic idea of search engine strategies they accomplish the great result in searching of most vital data. Those data techniques are as reverse searching and incremental searching [1].

The motivation behind the implementing enhanced web crawler is the need of crawler which can crawl deep websites efficiently because locating deep websites is the challenging issue. Deep web content is particularly important. Not only its size is estimated as hundreds of times larger than the so-called surface Web, but also it provides users with high quality information. However, to obtain such content of deep web is challenging and has been acknowledged as a significant gap in the coverage of search engines. So for accurately and quickly exploring deep databases, the enhanced web crawler is designed by using reinforcement learning framework.

3. PROPOSED WORK

To efficiently and effectively discover deep web data sources, enhanced web crawler is designed with a two-stage architecture, site locating and in-site exploring, as shown in Figure 1. The first site locating stage finds the most relevant contents then the second in-site exploring stage uncovers searchable forms from the site.

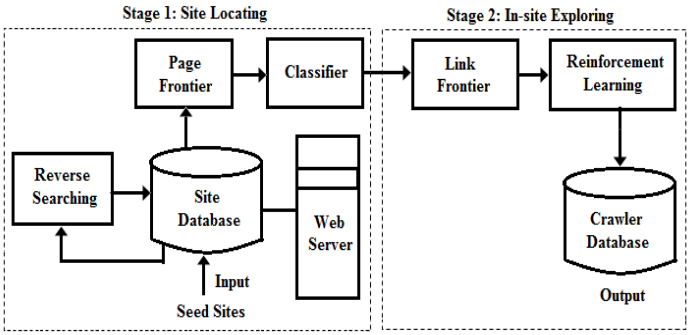

Fig -1: Proposed Architecture of Enhanced Web Crawler

Specifically, the site locating stage starts with a seed site in a site database. Seed site is candidate site given for enhanced web crawler to start crawling, that begins by following URLs from chosen seed web site to explore different pages and different domains. Then it performs reverse searching of known deep web sites for center pages (highly ranked pages that have several links to other domains) and feeds these pages back to the site database. Site Frontier fetches homepage URLs from the site information. To achieve more accurate results for a focused crawl, site classifier categorizes URLs into relevant or irrelevant for a given topic according to the homepage content. Once the most relevant pages are found within the first stage, the second stage performs efficient in-site exploration for excavating searchable forms. Links of a web site are stored in Link Frontier. Afterward reinforcement learning is applied. Lu Jiang et. al says that the reinforcement learning is a novel and efficient framework for deep web crawling.

Reinforcement learning is a computational approach to understanding and automating goal-directed learning and decision-making. It is distinguished from different computational approaches by its emphasis on learning by the individual from direct interaction with its environment, without relying on exemplary direction or completes models of the environment. Reinforcement learning is the initial field

to seriously address the computational issues that arise once learning from interaction with an environment so as to achieve long-term goals. Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards.

Algorithm for Q-Learning
The procedural form of the algorithm is:

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
Initialize s
Repeat (for each step of episode):
Choose a from s using policy derived from Q (e.g. ε -greedy)
Take action a, observe r, s'
$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \, max_a, Q(s', a') - Q(s,a)]$
$s \leftarrow s'$
Until s' is terminal

The parameters used in the Q-value update process are:
α - the learning rate, set between 0 and 1. Setting it to 0 means that the Q-values are never updated, hence nothing is learned. Setting a high value such as 0.9 means that learning can occur quickly.

γ - discount factor, also set between 0 and 1. This models the fact that future rewards are worth less than immediate rewards. Mathematically, the discount factor needs to be set less than 0 for the algorithm to converge.

max_a- the maximum reward that is attainable in the state following the current one. i.e the reward for taking the optimal action thereafter.

This procedural approach can be translated into plain English steps as follows:

1. Initialize the Q-values table, $Q(s, a)$.
2. Observe the current state, s.
3. Choose an action, a, for that state based on one of the action selection policies explained here on the previous page (ε -soft, ε -greedy or softmax).
4. Take the action, and observe the reward, r, as well as the new state, s'.
5. Update the Q-value for the state using the observed reward and the maximum reward possible for the next state. The updating is done according to the formula and parameters described above.
6. Set the state to the new state, and repeat the process until a terminal state is reached.

4. EXPERIMENTAL RESULTS

The proposed enhanced web crawler performs well as compare to SmartCrawler and other standard crawler with respect to three parameters:

1. Running Time
2. Number of searchable forms harvested by crawler
3. Harvest rate

The proposed enhanced web crawler, SmartCrawler and other standard crawler are compared here with respect to above three parameters. These are the parameters which will decide the efficiency and ability of the web crawler. Following graphs shows comparison of enhanced web crawler, SmartCrawler and other general web crawlers with respect to these three parameters.

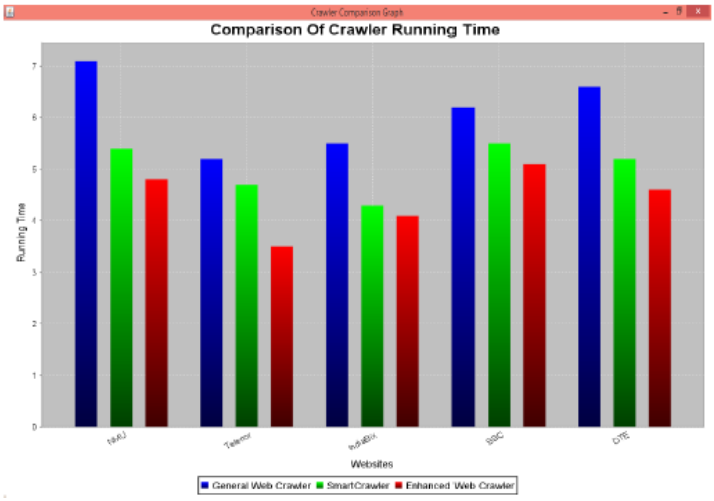

Chart -1: Comparison of Running Time of Crawlers

Chart 1 shows comparison of running times of three crawlers: Enhanced crawler, SmartCrawler and other general crawlers. From which we can see that the proposed enhanced crawler required minimum time to crawl different websites like NMU, Telenor, IndiaBix, SSC, and DTE. These three websites are taken as examples to show that the enhanced crawler has minimum running time as compared to other crawlers.

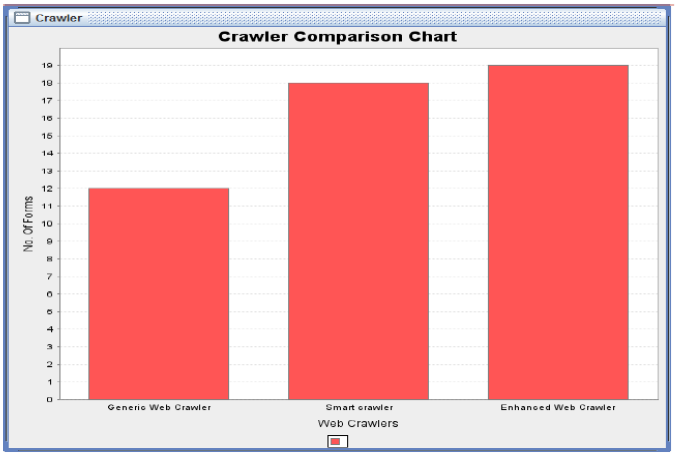

Chart -2: Comparison of Number of Forms Harvested by Crawlers

Chart 2 shows the comparison of number of forms harvested by crawlers. Each website has many searchable forms within it. A good crawler always searches more and more searchable form as possible. Form this graph we can see that the proposed enhanced web crawler harvest more number of forms as compared to SmartCrawler and other generic web crawlers.

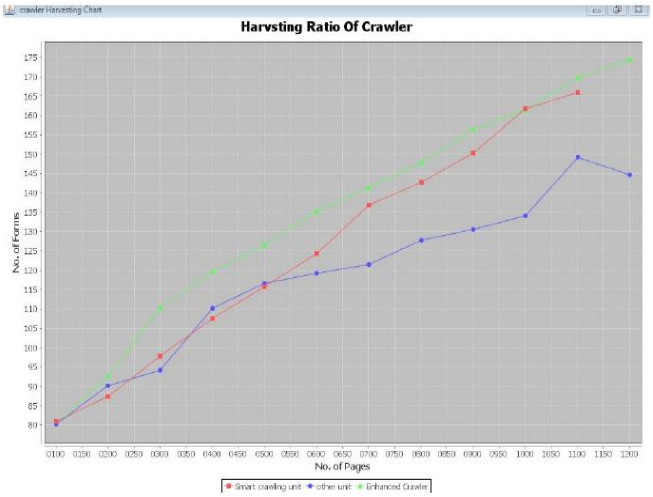

Chart -3: Comparison of Harvest Rate of Crawlers

Chart 3 shows the comparison of harvest rate of crawler. From this graph it can be conclude that the harvest rate of the enhanced web crawler is high than other two crawlers. From overall parameters we can observe that the enhanced web crawler performs well with respect to the three parameters than SmartCrawler and other general crawler.

5. CONCLUSION

Deep web or hidden web refers to World Wide Web content that is not part of the surface web, which is directly indexed by search engines. There is a need for an efficient crawler that is able to accurately and quickly explore the deep web databases. It is challenging to locate the deep web databases, because they are not registered with any search engines, are usually sparsely distributed, and keep constantly changing. To address this problem, the proposed enhanced web crawler works in two stages. First is site locating stage in which reverse searching is used to give more relevant data. The enhanced crawler performs site based locating by reversely searching the known deep web sites for center pages, which can effectively find many data sources for sparse domains. After identifying relevant data, in site exploring is performed in second stage. In this stage more relevant links are extracted from more relevant site and reinforcement framework is applied which is more efficient for deep web crawling. Our experimental results show that the proposed enhanced web crawler outperforms on various parameters so that better crawling is performed. In the future, better crawling method can be developed for deep web interfaces.

REFERENCES

[1] H. H. Feng Zhao, Jingyu Zhou and H. Jin, "Smartcrawler: A two-stage crawler for efficiently harvesting deep-web interfaces," 2015.

[2] J. L. Lu Jiang, Zhaohui Wu and Q. Zheng, "Efficient deep web crawling using reinforcement learning," MOE KLINNS Lab and SKLMS Lab, Xian Jiaotong University,2008.

[3] D. H. Mohamamdreza Khelghati and M. V. Keulen, "Deep web entity monitoring", 2013.

[4] K. Srinivas, P.V. S. Srinivas,A.Goverdhan, "Web Service Architecture for Meta Search Engine", International Journal Of Advanced computer Science And Application, 2011.

[5] O. Christopher and N. Marc, "Web crawling," Foundations and Trends in Information Retrieval, pp. 175–246, 2010.

[6] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction", 2012.

[7] L. Barbosa and J. Freire, "Searching for hidden-web databases," In WebDB, pp. 1–5, 2005.

[8] L. Barbosa and J. Freire, "An adaptive crawler for locating hidden-web entry points," In Proceedings of the 16th international conference on World Wide Web, ACM, pp. 441–450, 2007.

[9] M. V. d. B. Soumen Chakrabarti and B. Dom, "Focused crawling: a new approach to topic-specific web resource discovery", Computer Networks, pp. 1623–1640, 1999.

BIOGRAPHY

Miss.Yugandhara Patil, received degree of Bachelor of Engineering in Computer in 2014 and now pursuing Master of Engineering in Computer Science and Engineering from G. H. Raisoni Institute of Engineering & Management, Jalgaon, Maharashtra, India.

Prof. Sonal Patil, received degree Bachelor of Engineering and Master of Technology in Computer Science and Engineering. She has total 68 publications, out of which 56 are international and remaining are national publications. Now she is working as Head of Department of IT department, G. H. Raisoni Institute of Engineering & Management, Jalgaon, Maharashtra, India.