A Two-Way Smart Web spider

Neha Gadpal[1], Nikita Gajbhiye[2], Prajakta Sable [3], Radhika Thakre[4], Aprajita Gondane[5]

[1,2,3,4,5] *Department of computer science and Engineering, GNI, 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 interface. However, due to the large volume of web resources and dynamic nature of deep achieving wide coverage and high efficiency is a challenging issue. We propose a two-stage framework, namely smart web spider perform site based searching for center pages with the help of search engine, avoiding large number of pages. The smart web spider also efficiently manages the metadata by systematically storing it. In this section we describe characteristic and classification of spiders. We also give some background information about the programming language used to build the spider. The project aims to create a smart web spider for a concept based semantic based search engine. The spider not only aims to crawl the database and World Wide Web but also bring back data but also aims to perform and initial data analysis of unnecessary data before it stores the data. We aim to improve the efficiency of the relevance based searching by using the Smart Spider.*

***Key-words**: hitting ratio, web spider, search engine, database, hidden web, two stage spider, feature selection, deep web.*

1.INTRODUCTION

The internet is a vast collection of billions of web pages containing terabytes of information arranged in thousands of servers using HTML. The size of this collection itself is a formidable obstacle in retrieving information necessary and relevant. This made search engines an important part of our lives. Search engines strive to retrieve information as relevant as possible to the user. One of the building blocks of search engines is the Web spider. A web spider is a bot that goes around the internet collecting and storing it in a database for further analysis and arrangement of the data. The project aims to create a smart web spider for a concept based keyword. based search engine. The spider not only aims to crawl the primary database and World Wide Web and bring back data but also aims to perform an initial data analysis of unnecessary data before it stores the data. We aim to improve the efficiency of the Keyword Based Semantic Search Engine by using the Smart spider.

The *deep* (or *hidden*) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. Based on extrapolations from a study done at University of California, Berkeley, it is estimated that the deep web contains approximately 91,850 terabytes and the surface web is only about 167 terabytes in. More recent studies estimated that 1.9 zetta bytes were reached and 0.3 zettabytes were consumed worldwide in. An IDC report estimates that the total of all digital data created, replicated, and consumed will reach 6 zetta bytes in. A significant portion of this huge amount of data is estimated to be stored as structured or relational data in web databases — deep web makes up about 96% of all the content on the Internet, which is 500-550 times larger than the surface web. These data contain a vast amount of valuable information and entities such as Infomine , Clusty, Books In Print may be interested in building an index of the deep web sources in a given domain (such as book). Because these entities cannot access the proprietary web indices of search engines (e.g., Google and Baidu), there is a need for an efficient.

2. CHARACTERISTICS.

Characteristic features of Web Spiders that crawl the internet must have the following basic features so that they serve their purpose.

2.1 Robustness

The web contains loops called spider traps, which are meant to mislead the spider to recursively crawl a particular domain and get stuck in one single domain. They generate an infinite loop of web pages that lead to nowhere. The spider needs to be resilient to such traps. These traps may not always be designed to mislead the spider but may be a result of faulty website development.

2.2 Politeness

Web servers have policies regulating when a spider can visit them. These politeness policies must be respected. A server is meant to serve other requests that it is originally designed to serve. Hindering the server may lead to blocking of the spider by the server.

2.3 Performance and Efficiency

The use of system resources like processing power, network bandwidth and storage should be judicious. These factors determine how efficient the spider is.

2.4 Quality

The spider should be able to differentiate between information that is useful and information that is not. As servers mainly serve other requests that contain a lot of information that may not be useful. Spider should filter out this content.

2.5 Extensible

The spiders should be able to adapt to the growing number of data formats that it will encounter on web sites. It also needs to cope up with the new protocols that may be used on some servers.

2.6 Traditional spider

A traditional spider periodically crawls the already crawled URLs and replaces the old documents with the newly downloaded documents to refresh its collection. On the contrary, an incremental spider refreshes incrementally the already existing collection of pages by visiting them frequently. This is based upon an estimation of the rate at how often pages change. It also replaces old and less important pages by new and more relevant pages. It resolves the problem of freshness of the data. The advantage of incremental spider is that only valuable data is provided to the user. Thus we save network bandwidth and also achieve data enrichment.

2.7 Distributed spider

Distributed computing technique is the main foundation for distributed web crawling. Many spiders are working at the same time in tandem and distribute the work load of crawling the web in order to have maximum coverage of the internet. A central server manages the communication, synchronization of nodes and communicates between the different bots. It is also geographically distributed. It primarily uses Page rank algorithm to increase efficiency and quality of search. The advantage of distributed web spider is that it is robust. It is resistant to system crashes and other events, and can adopt to various crawling requirements.

3. SYSTEM ARCHITECTURE

To efficiently and effectively discover deep web data sources, Smart web spider is designed with a two stage architecture, site locating and in-site exploring,

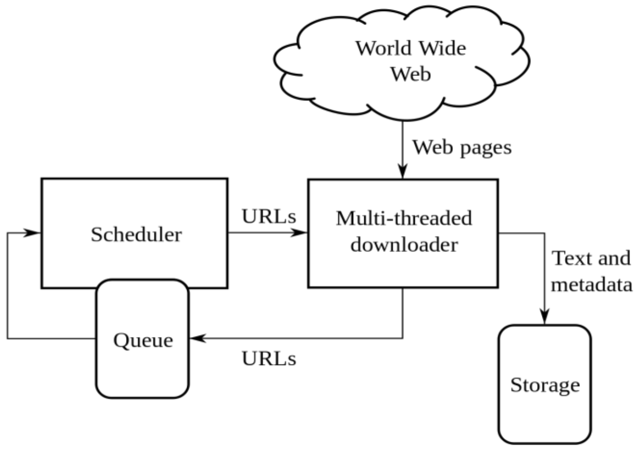

Fig. 1 System Architecture of web spider

As shown in Figure 1. The first site locating stage finds the most relevant site for a given topic, and then the second in-site exploring stage uncovers searchable forms.

4. PROPOSED WORK

We propose a two-stage framework, namely Smart web spider, for harvesting deep web interfaces. In the first stage, web spider performs site-based searching for center pages with the help of search engines, avoiding visiting a large number of pages. To achieve more accurate results for a focused crawl, web spider ranks websites to prioritize highly relevant ones for a given topic. In the second stage, web spider achieves fast in-site searching by excavating most relevant links with an adaptive link-ranking. To eliminate bias on visiting some highly relevant links in hidden web directories, we design a link tree data structure to achieve wider coverage for a website. Our experimental results on a set of representative domains show the agility and accuracy of our proposed crawler framework, which efficiently retrieves deep-web interfaces from large scale sites and achieves higher harvest rates than other crawlers propose an effective harvesting framework for deep-web interfaces, namely Smart-Crawler. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. Web spider is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring. Web spider performs site based locating by reversely searching the known deep web sites for center pages, which can effectively and many data sources for sparse domains. By ranking collected sites and by focusing the crawling on a topic, web spider achieves more accurate results

.

4.1 Scope of Project

The primary purpose of the project is to make a crawler that can provide text files to the search Engine. The text files are meant to be the input for the Search engine which will try to analyze the data and extract meaningful concepts from the data and store them in an SQL Database. The crawler will extract text data from the data obtained from crawling and create text files with the data. The crawler also aims to systematically store metadata in a different set of files for future use. The crawler thus aims to improve the efficiency of the Search engine. In the first stage, web spider performs site-based searching for center pages with the help of search engines, avoiding visiting a large number of pages. To achieve more accurate results for a focused crawl, web spider ranks websites to prioritize highly relevant ones for a given topic. In the second stage, web spider achieves fast in-site searching by excavating most relevant links with an adaptive link-ranking. Benefits of the system is that as in todays era of the Internet, economies have changed the way than they were a few years back. Now everyone have started preferring to go online for completing the process. This not only saves the efforts applied but plays a significant role for profit graph 3 to touch the sky as well. For e.g. as users purchase dresses for festivals or by their need. They spend time to purchase their clothes by their choice like color, size, and designs, rate and so on. But now in this world everyone is busy. They can-not be able to spend time for this. So for these reason we proposed the new system called web crawling. From these we can search a website, and we can get many more pages of that particular website and then by visting each pages we can get their description and their keywords. So by these we can get the chance to check all the information we are interested in. Limitation: • Consuming large amount of datas. Time wasting while crawl in the web.

4.2 Problem Statement

The deep(or hidden) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. An IDC report estimates that the total of all digital data created, replicated, and consumed will reach 6 zettabytes in 2014. A significant portion of this huge amount of data is estimated to be stored as structured or relational data in web databases deep web makes up about 96 percent of all the content on the Internet, which is 500-550 times larger than the surface web These data contain a vast amount of valuable information and entities such as Infomine, Clusty, Books In Print may be interested in building an index of the deep web sources in a given domain (such as book). Because these entities cannot access the proprietary web indices of search engines (e.g., Google and Baidu), 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, previous work has proposed two types of crawlers, generic crawlers and focused crawlers. Generic crawlers fetch all searchable forms and cannot focus on a specific topic. Focused crawlers such as Form-Focused Crawler (FFC) and Adaptive Crawler for Hidden-web Entries can automatically search online databases on a specific topic. FFC is designed with link, page, and form classifiers for focused crawling of web forms, and is extended by ACHE with additional components for form filtering and adaptive link learner. The link classifiers in these crawlers play a pivotal role in achieving higher crawling efficiency than the best-first crawler However, these link classifiers are used to predict the distance to the page containing searchable forms, which is difficult to estimate, especially for the delayed benefit links (links eventually lead to pages with forms). As a result, the crawler can be inefficiently led to pages without targeted forms.

5. PROPOSE METHDOLOGY

RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. The phases in the rapid application development (RAD) model are Business modeling: The information flow is identified between various business functions. Data modeling: Information gathered from business modeling is used to define data objects that are needed for the business. Process modeling: Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects. Application generation: Automated tools are used to convert process models into code and the actual system. Testing and turnover: Test new components and all the interfaces.

6. CONCLUSION

We proposed a two way smart web crawler that will search for content in both primary database and web. It will refine search results using past preferences, and valid site certification. It also features filters that will be used to filter the format of the results displayed. In future, we can incorporate this crawler with machine learning techniques to act and think like humans. This will enable the crawler to give results based upon the context. In addition to this, there is a need to expand the deep web database to crawl more number of websites giving maximum possible number of search results.

REFERANCE

1) Feng Zhao, J. Z. (2015). Smart Crawler: Two stage Crawler For Efficiently Harvesting Deep-Web Interface. IEEE Transactions on Service Computing Volume:pp year 2015

2) IDC worldwide predictions 2014 : Battles for dominance-and survival – on the 3rd platform. http://www.idc.com/ research/predictions14 , 2014.

3) Idc worldwide predictions 2014: Battles for dominance –and survival – on the 3rd platform. http://www.idc.com/ research/Predictions14/index.jsp , 2014.

4) eye He, Dong Xin, Venkatesh Ganti, SriramRajaraman, and Niraj Shah. Crawling deep web entity pages. In Proceedings of the sixth ACM international conference on Web search and datamining, pages 355–364. ACM, 2013.

5) Martin Hilbert. How much information is there in the “information society”? Significance, 9(4):8–12, 2012. “”

6) Dhiraj Khurana, Satish Kumar, “Web Crawler: A Review”, International Journal of Computer Science & Management Studies (IJCSMS), Vol. 12, Issue 01, January 2012, ISSN (Online): 2231 –5268.

7) Debashish, Amritesh, Lizashree “Unvisited URL Relevancy Calculation in Focused Crawling based on Naïve Bayesian Classification”, International Journal of Computer Application, volume 3, July 2010.

8) Qu Cheng, Wang Beizhan, Wei Pianpian, “Efficient Focused Crawling Strategy Using Combination of Link Structure and Content Similarity”, Software School,2009