International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 4
ISSN: 2454-4248
393 – 399

An Efficient Method for Deep Web Crawler based on Accuracy

Pranali Zade[1], Dr. S.W Mohod[2]
Master of Technology,
Dept. of Computer Science and Engg ,
Bapurao Deshmukh College of Engg,Wardha
[1]*pranalizade1234@gmail.com*

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. We propose a three-stage framework, for efficient harvesting deep web interfaces.Project 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 using Naïve Bayes algorithm. In this paper we have made a survey on how web crawler works and what are the methodologies available in existing system from different researchers.

Keywords: *Deep web, web mining, feature selection, ranking*

*****

I. INTRODUCTION

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 2003. More recent studies estimated that 1.9 petabytes were reached and 0.3 petabytes were consumed worldwide in 2007. An IDC report estimates that the total of all digital data created, replicated, and consumed will reach 6 petabytes 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% 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 (ACHE) 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. Besides efficiency, quality and coverage on relevant deep web sources are also challenging.

The propose work, achieve both wide coverage and high efficiency for a focused crawler. Our main contributions are:We propose a novel three-stage framework to address the problem of searching for hidden-web resources. Our site locating technique employs a reverse searching technique (e.g., using Google's"link:" facility to get pages pointing to a given link) and incremental three-level site prioritizing technique for unearthing relevant sites, achieving more data sources. During the in-site exploring stage, design a link tree for balanced link prioritizing, eliminating bias toward web pages in popular directories.

In the propose work an adaptive learning algorithm that performs online feature selection and uses these features to automatically construct link rankers. In the site locating stage, high relevant sites are prioritized and the crawling is focused on a topic using the contents of the root page of sites, achieving more accurate results. During the in-site exploring stage, relevant links are prioritized for fast in-site searching.

 ISSN: 2454-4248
393 – 399

II. LITERATURE SURVEY

There is a rich literature, here we discuss the most related work.

Feng Zhao et al. [1] proposeda two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces. 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.

Jianxiao Liu et al.[2] proposed an Approach of Semantic Web Service Classification Based on Naive Bayes ,proposed a method to classify and organize the semantic Web services to help users find the services to meet their needs quickly and accurately is a key issue to be solved in the era of service-oriented software engineering.

Bo Tang, presents an approach toward Optimal Feature Selection In Naive Bayes For Text Categorization [3].
Author proposed that, automated feature selection is important for text categorization to reduce the feature size and to speed up the learning process of classifiers.

AmrutaPandit and Prof. ManishaNaoghare[4], proposed work for Efficiently Harvesting Deep Web Interface with Reranking and Clustering.The rapid growth of the deep web poses predefine scaling challenges for general purpose crawler and search engines. There are increasing numbers of data sources now become available on the web, but often their contents are only accessible through query interface. Here author proposed a framework to deal with this problem, for harvesting deep web interface.

Anand Kumar et al. [5] presents a two-phase system, to be specific Smart Crawler, for productive gathering profound web interfaces.In this paper, author proposed, web develops at a quick pace, there has been expanded enthusiasm for procedures that assistance effectively find profound web interfaces. Be that as it may, because of the expansive volume of web assets and the dynamic way of profound web, accomplishing wide scope and high proficiency is a testing issue. In the primary stage, Smart Crawler performs site-based hunting down focus pages with the assistance of web crawlers, abstaining from going to a substantial number of pages.

AkshayaKubba[7] mentioned that, Web mining is an important concept of data mining that works on both structured and unstructured data. Search engine initiates a search by starting a crawler to search the World Wide Web (WWW) for documents .Web crawler works in a ordered way to mine the data from the huge repository. The data on which the crawlers were working was written in HTML tags, that data lags the meaning.

Monika Bhide et al. focus on accessing relevant web data and represents significant algorithm i.e. adaptive learning algorithm, reverse searching and classifier[8].the web stores huge amount of data on different topics.The main goal is to locating deep web interfaces. To locating deep web interfaces uses techniques and methods.The locating deep web interfaces system works in two stages. In the first stage apply reverse search engine algorithm and classifies the sites and the second stage ranking mechanism use to rank the relevant sites and display different ranking pages.

RajuBalakrishnan et al.[10] proposed, selecting the most relevant web databases for answering a given query. The existing database selection methods (both text and relational) assess the source quality based on the query-similarity-based relevance assessment.

D. Shestakov, address the accurate estimation of deep web by sampling one national web domain[11].here author report some of their results obtained when surveying Russian web.The Host-IP clustering sampling technique addresses the drawback of previous deep web surveys and allow to characterize a national segment of deep web.Heaiso got an insights on the sight of Russian deep Web by calculating upper bound estimate for the total number of entity in online database.

SuryakantChouthary et al. [12] worked for model-based rich internet applications crawling in which they they design methods based on menu and probability models. Strategies for crawling Web sites efficiently have been described more than a decade ago. Since then, Web applications have come a long way both in terms of adoption to provide information and services and in terms of technologies to develop them.

 ISSN: 2454-4248
393 – 399

III. PROPOSED APPROACH

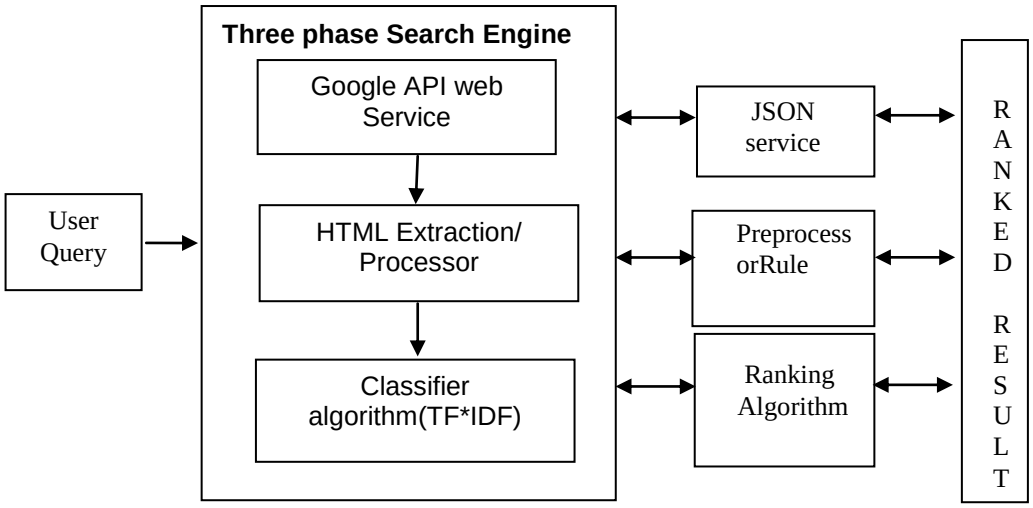

Fig 3: Proposed System Architecture

To efficiently and effectively discover deep web data sources, Crawler is designed with a three-stage architecture, as shown in above Figure. The first site locating stage finds the most relevant site for a given topic, the second in-site exploring stage uncovers searchable forms from the site and then the third stage apply naïve base classification ranked the result.

Specifically, the site locating stage starts with a seed set of sites in a site database. Seeds sites are candidate sites given for Crawler to start crawling, which begins by following URLs from chosen seed sites to explore other pages and other domains. When the number of unvisited URLs in the database is less than a threshold during the crawling process, Crawler performs "reverse searching" of known deep web sites for center pages (highly ranked pages that have many links to other domains) and feeds these pages back to the site database. Site Frontier fetches homepage URLs from the site database, which is ranked by Site Ranker to prioritize highly relevant sites.

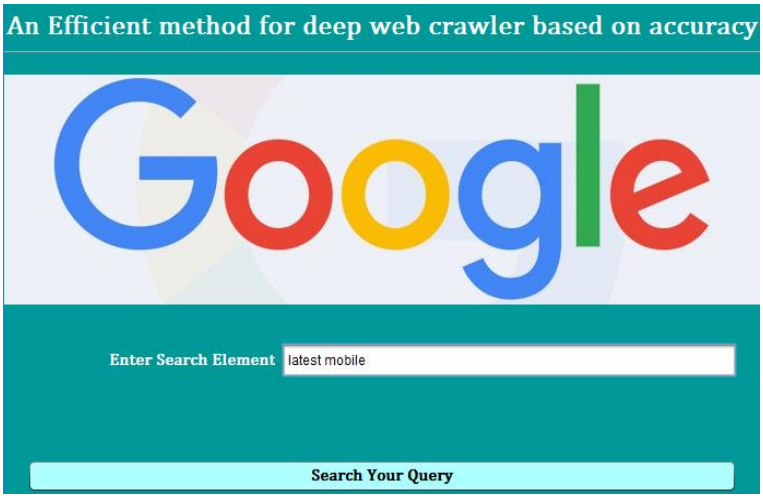

Fig 5: GUI Design for Crawling System

The above figure is the first form of design for proposed system. In this page user needs to enter the query that's gets searched on Google search engine with help of Google API framework. Using this first 5 results of Google search engine can be searched. This is an example for searching this keyword on Google. As the keyword is entered it is send to google search engine using query based JSON API. The nexr form shows the result of searching this keyword on proposed system.

ISSN: 2454-4248
393 – 399

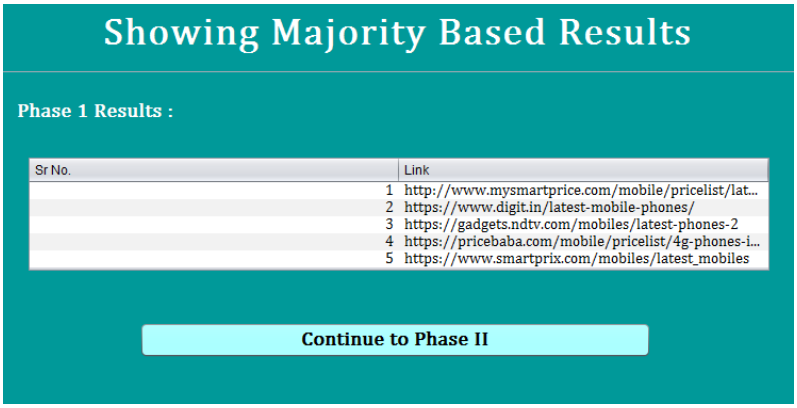

Fig 6: Data access from Google

Figure shows results that are fetched from Google Search Engine. Basic preprocessing is done as Google API return not only link but page details like meta tag titles URL Images and promotional links. All have been removed and original ranked results are showed.

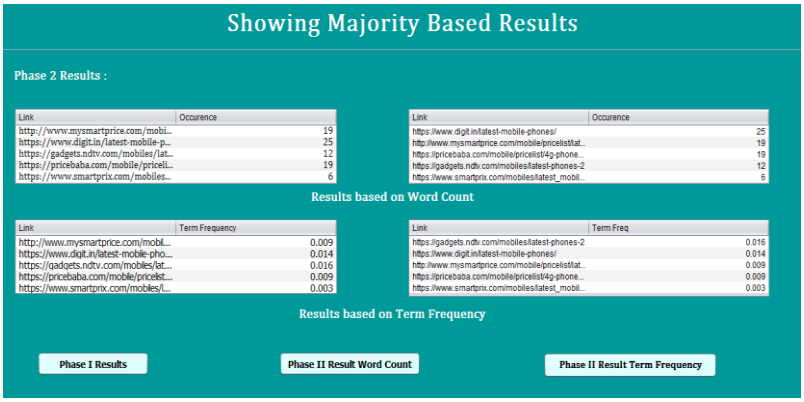

Fig 8: word count on top results and frequency analysis

As we move on to the next phase the proposed algorithm starts re-ranking the results based on k-NN and Naïve Bayes algorithm. Using this algorithm two different frequencies is calculated. First is word-count and second is term frequency.

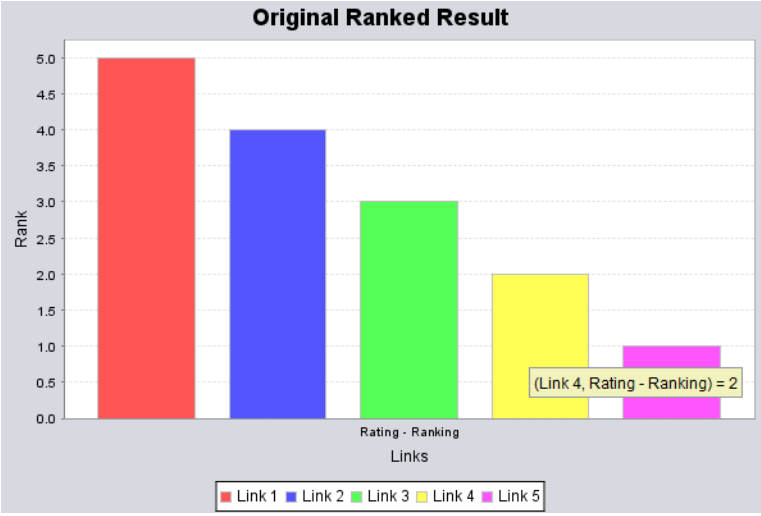

Fig 9: Link Result

 ISSN: 2454-4248
393 – 399

The above graph shows the ranking of links as provided by Google search engine. The bar chart has been drawn using JFreechart library of java and is always changing based on results provided by Google.

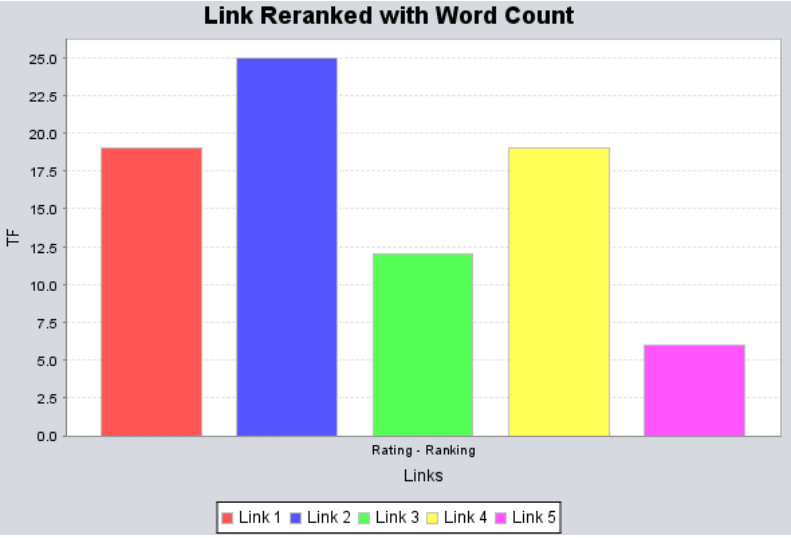

Fig 10 :Link Result With Word Count

The above graph shows the ranking of links based on wordcount based k-NN algorithm in which all the links are searched based on the keyword being searched in the first phase of algorithm. Using this we can verify the results of Google using k-NN algorithm.

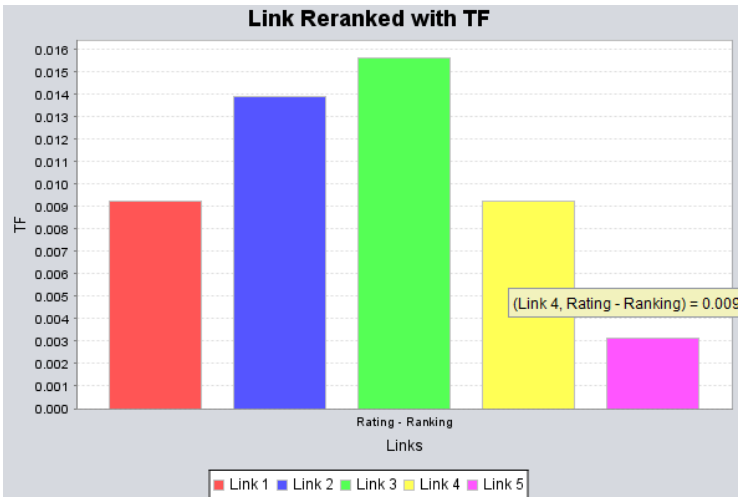

Fig 11: Link Result with Term Frequency

The above graph shows the ranking of links based on TF based NB algorithm in which all the links are searched based on the keyword being searched in the first phase of algorithm. Using this we can verify the results of Google using NB algorithm.

IV. RESULTS AND DISCUSSION

Consider an example, When a search engine returns 30 pages only 20 of which were relevant while failing to return 40 additional relevant pages, its precision is 20/30 = 2/3 while its recall is 20/60 = 1/3. So, in this case, precision is "how useful the search results are", and recall is "how complete the results are".

We consider here 100 queries. For one query the top 5 result will fetch out of 10 results from Google. Thus for 100 queries 500 results will fetch out of 1000 results. The recall, precision and accuracy of a system are calculated from the results taken from the Google and observing results. These experimental results indicate that use of Naïve Bayes Algorithm having better performance than KNN Algorithm for accuracy. Figure 6.1 shows the comparative graph.

 ISSN: 2454-4248
393 – 399

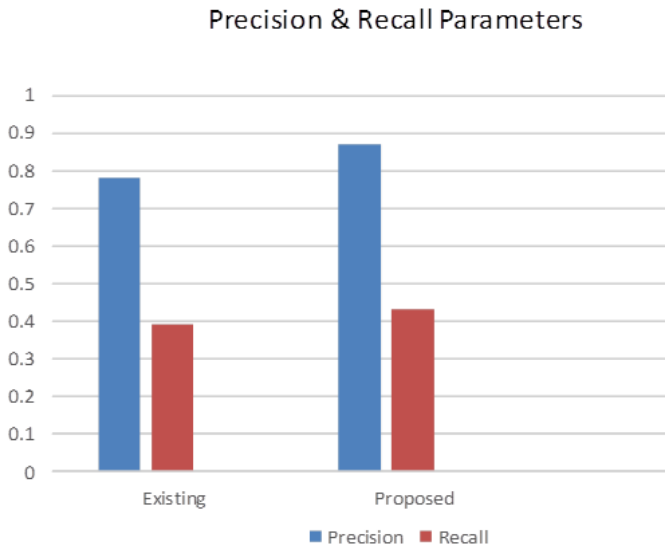

Fig 12: Comparison Graph based on Parameters

From the graph it is clear that value of Precision and Recall for proposed system is greater than existing one. Thus the accuracy of proposed system is more.

V. CONCLUSION AND FUTURE SCOPE

We propose an effective harvesting framework for deep-web interfaces. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. This is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring.Based on the obtained results from study, we conclude that, the approach has better accuracy than the other crawling method. The proposed crawler based on Naïve Bayes classifier. Past frameworks have numerous issues and difficulties. To overcome this Crawler achieves more accurate results and reranks link to prioritize highly relevant ones for a given topic

In future work, we plan to combine pre-query and post-query approaches for classifying deep-web forms to further improve the accuracy of the form classifier.

As the future scope, the following can be done to the algorithm

1) We can further improve this algorithm to include many different types of efficient hybrid page ranking techniques which can further fortify the ranking procedures thereby generating the most accurate crawling results.

2) The algorithm can be improved with respect to do a crawling of the sub-child links also and applying page ranking techniques on same. We can further improve this algorithm to do an intelligent time-based crawling by which the application would fire a search crawl within a specific time and also complete within a specific time thereby making the crawling process more efficient.

REFERENCES

[1] Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin "Smart Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces" in IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 9, NO. 4, JULY/AUGUST 2016.

[2] Jianxiao Liu, ZonglinTian, Panbiao Liu, Jiawei Jiang, "An Approach of Semantic Web Service Classification Based on Naive Bayes" in 2016 IEEE International Conference on Services Computing, SEPTEMBER 2016.

[3] Bo Tang, Student Member, IEEE, Steven Kay, Fellow, IEEE, and Haibo He, Senior Member, IEEE "Toward Optimal Feature Selection in Naive Bayes for Text Categorization" in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 9 Feb 2016.

[4] AmrutaPandit ,Prof. ManishaNaoghare, "Efficiently Harvesting Deep Web Interface with Reranking and Clustering", in International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016.

[5] Anand Kumar , Rahul Kumar, SachinNigle, MinalShahakar, "Review on Extracting the Web Data through Deep Web Interfaces, Mechanism", in International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 1, January 2016

[6] Sayali D. Jadhav, H. P. Channe "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques" in International Journal of Science and Research, Volume 5 Issue 1, January 2016.

[7] AkshayaKubba, "Web Crawlers for Semantic Web" in International Journal of Advanced Research in Computer

 ISSN: 2454-4248
393 – 399

Science and Software Engineering, Volume 5, Issue 5, May 2015.

[8] Monika Bhide, M. A. Shaikh, AmrutaPatil, SunitaKerure,"Extracting the Web Data Through Deep Web Interfaces" in INCIEST-2015.

[9] Y. He, D. Xin, V. Ganti, S. Rajaraman, and N. Shah, "Crawling deep web entity pages," in Proc. 6th ACM Int. Conf. Web Search Data Mining, 2013, pp. 355–364.

[10] RajuBalakrishnan, SubbaraoKambhampati, "SourceRank: Relevance and Trust Assessment for Deep Web Sources Based on Inter-Source Agreement" in WWW 2011, March 28–April 1, 2011.

[11] D. Shestakov, "Databases on the web: National web domain survey," in Proc. 15th Symp. Int. Database Eng. Appl., 2011, pp. 179–184. [12] D. Shestakov and T. Salakoski, "Host-ip clustering technique for deep web characterization," in Proc. 12th Int. Asia-Pacific Web Conf., 2010, pp. 378–380.

[12] SuryakantChouthary, EmreDincturk, SeyedMirtaheri, Ggregor V. Bochmann, Guy-Vincent Jourdan and IosifViorelonut"model-based rich internet applications crawling: ―menu‖ and ―probability‖ models "in journal of Web Engineering, Vol.0, No.2003.