ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 7.327

Volume 6, Issue 11, November 2018

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Deep Web Harvesting Using SmartCrawler

Anita Vittal Kodam[1]
Student of Computer Science & Engineering
N.B.Navale Sinhgad College of Engineering
Kegaon, Solapur-413255, Maharashtra – India.

Prof. V. V. Pottigar[2]
Assistant Professor of Computer Science & Engineering
N.B.Navale Sinhgad College of Engineering
Kegaon, Solapur-413255, Maharashtra – India.

Prof. A. V. Mophare[3]
Assistant Professor of Computer Science & Engineering
N.B.Navale Sinhgad College of Engineering
Kegaon, Solapur-413255, Maharashtra – India.

Abstract: ***The Heavy usage of internet, large amount of data is spread over the network, which provide access to particular data or to search most relevant data is a very challenging for search engines to fetch relevant data as per user's need and which consume more time. So, to reduce large amount of time spend on searching most relevant data we provide two stage framework of search engine called SmartCrawler.***

In first stage SmartCrawler performs website based searching for center pages with the help of search engine by avoiding visiting huge number of unwanted pages To get more precise and useful results SmartCrawler ranks website by using ranking mechanism to prioritize extremely relevant for a given topic. SmartCrawler performs Reverse Searching to discover more searchable web forms. In the second stage, SmartCrawler performs fast in-site searching and website ranking for finding closely relevant websites. Adaptive Learning Algorithm performs Online Feature Selection and automatic construction of Link Rankers.

Keywords: Deep-Web, Search-Key, Smart Crawler, Rank, Search, Graph.

I. INTRODUCTION

One of the main components of web search engines which can assemble a corpus of web pages, index them, and allow users to issue queries against the index and find the web pages that match the queries are called as a Web Crawler or Spiders or robot[1][2][3].Deep web which is also called as a invisible web, which can't be get with the single search because they are not registered by search engines[1][2][3].To overcome such type of problem we propose the SmartCrawler, which is having two stage framework for efficient harvesting deep web[1].

SmartCrawler focus on a specific topic and it fetches only the most relevant searchable forms to a given topic or search queriy. Thus It performs a superior level of data examination and data mining from the deep web.

SmartCrawler has two stages: In the first stage, SmartCrawler performs website based searching for center pages with the help of search engine by avoiding visiting huge number of unwanted pages [1]. To get more precise and useful results SmartCrawler ranks website by using ranking mechanism to prioritize extremely relevant for a given topic [1]. SmartCrawler performs Reverse Searching to discover more searchable web forms [1].

In the second stage, SmartCrawler performs fast in-site searching and website ranking for finding closely relevant websites. Adaptive Learning Algorithm performs Online Feature Selection and automatic construction of Link Rankers [1].

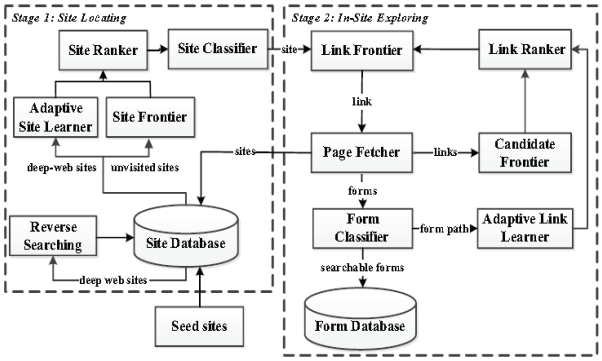


Fig: Workflow of System Architecture.

II. OVERVIEW OF SYSTEM

In this Project, I propose an effective deep web harvesting framework, namely SmartCrawler, for achieving both wide coverage and high efficiency for a focused crawler. Based on the observation that deep websites usually contain a few searchable forms and most of them are within a depth of three our crawler is divided into two stages:

Site locating and insite exploring. The site locating stage helps achieve wide coverage of sites for a focused crawler, and the in-site exploring stage can efficiently perform searches for web forms within a site.

Our main contributions are:

1. I proposed a novel two-stage framework to address the problem of searching for hidden-web resources. Our site locating technique employs a reverse searching technique (e.g., using Googles link: facility to get pages pointing to a given link) and incremental two-level site prioritizing technique for unearthing relevant sites, achieving more data sources. During the in-site exploring stage, we design a link tree for balanced link prioritizing, eliminating bias toward webpages in popular directories.

2. I proposed 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 insite exploring stage, relevant links are prioritized for fast in-site searching.

A. *SITE LOCATING*

The site locating stage finds relevant sites for a given topic, consisting of site collecting, site ranking, and site classification.

A.1 Site Collecting

The traditional crawler follows all newly found links.In contrast, our SmartCrawler strives to minimize the number of visited URLs, and at the same time maximizes the number of deep websites.[1] To achieve these goals, using the links in downloaded webpages is not enough. This is because a website usually contains a small number of links to other sites, even for some large sites.

Reverse Searching

The idea is to exploit existing search engines, such as Google, Baidu, Bing etc., to find center pages of unvisited sites. This is possible because search engines rank pages of a site and center pages tend to have high ranking values.Algorithm 1 describes the process of reverse searching. pre-defined threshold. We randomly pick a known deep website or a seed site and use general search engines facility to find centre pages and other relevant sites, Such as Googles link: , java site:, c,c++ domain:

Algorithms Used:

Reverse Searching Algorithm:
Input: Seed sites harvested deep web sites.
Output: Relevant sites.
While of Candidate sites less than a threshold do
// Pick a deep website
Site = get Deep website (Site Database, Seed Sites)
Result Page = ReverseSearch(Site)
Links= Extract Links (Result Page)
Foreach link In Links do
Page = DownloadPage (Link)
Relevant= Classify (Page)
If relevant then
Relevant Sites=Extract Un Visited Site(Page)
Output relevant Sites
End
End
End

Incremental site Prioritizing Algorithm : **for Site Ranking Mechanism**
Input: SiteFrotntier
Output: Searchable forms
Hqueue= SiteFrontier. CreateQueue(High Priority)
Lqueue=Sitefrontier.CreateQueue(Low Priority)
While siteFrontire is not empty do
if Hqueue is empty then
Hqueue.addAll(Lqueue)
Lqueue .clear()
end
Site= Hqueue .Poll()
Relevant =ClassifySite(Site)
If releveant then
performInSiteExploring(Site)
Output forms and OutOfSiteLinks
SiteRanke.rank(OutOfSiteLinks)
If forms is not empty then
Hqueue.add(OutOfSiteLinks)
end
else
Lqueue.add(OutOfSiteLinks)
end
end
end

III. Adaptive Learning Process

SmartCrawler has an adaptive learning strategy that updates and leverages information collected successfully during crawling As shown in Figure , both Site Ranker and Link Ranker are controlled by adaptive learners. Periodically, FSS and FSL are adaptively updated to reflect new patterns found during crawling. As a result, Site Ranker and Link Ranker are updated. Finally, Site Ranker re-ranks sites in Site Frontier and Link Ranker updates the relevance of links in Link Frontier.

IV. Advantages of Proposed System

1. It gives a specific output to the user.

2. An adaptive learning algorithm that performs on line feature selection and uses these features to automatically construct link rankers

3. Our site locating technique employs a reverse searching technique (e.g., using Googles link: facility to get pages pointing to a given link) and incremental two level site prioritizing technique for unearthing relevant sites, achieving more data sources.

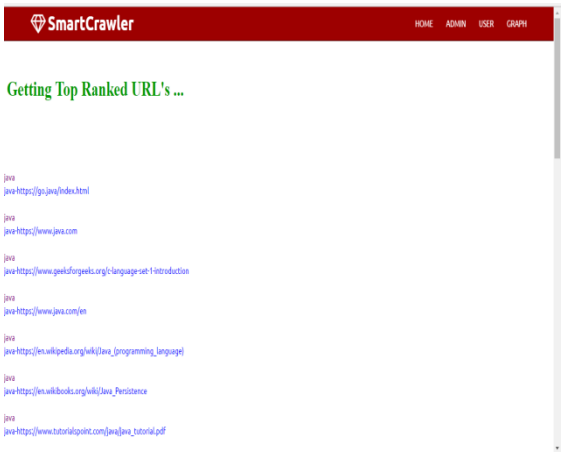
V. Execution Report

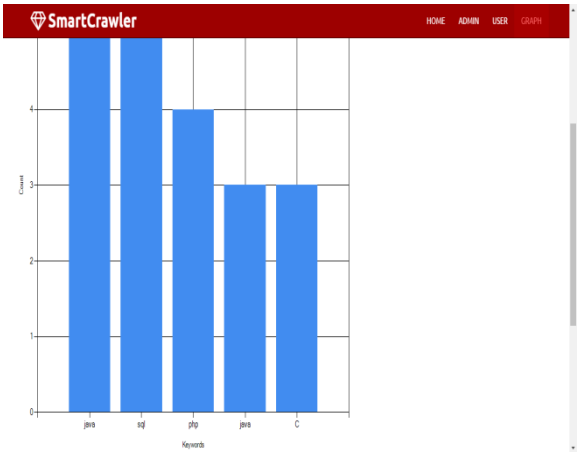
Evaluating the efficiency of SmartCrawler in

-obtaining relevant deep websites and searchable forms,

-analyzing the effectiveness of site collecting, and

-assessing the performance of adaptive learning





VI. Conclusion

It can propose an effective harvesting framework for deep-web interfaces, namely SmartCrawler. This can prove that the proposed methodology achieves both wide coverage for deep web interfaces and maintains highly efficient crawling.

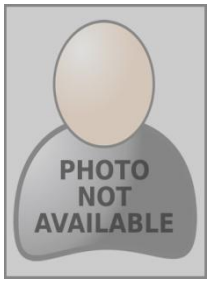
Acknowledgement

I would like to take this opportunity to express my gratitude and deep regard to Prof.V.V.Pottigar and Prof.A.V.Mophare, for the guidance, valuable feedback and constant encouragement throughout writing this paper without which this paper would be incomplete.

References

1. Paper No.[1] Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin. "SmartCrawler:Two Stage crawler for efficient harvesting Deep-web interface vol(99)-2015
2. Paper No. [19] Olston Christopher and Najork Marc. Web crawling. Foundationsand Trends in Information Retrieval, 4(3):175–246, 2010.
3. Paper No. [12] Denis Shestakov and Tapio Salakoski. Host-ip clusteringtechnique for deep web characterization. In Proceedings of the12th International Asia-Pacific Web Conference (APWEB), pages378–380. IEEE, 2010.
4. Paper No. [28] Andr´e Bergholz and Boris Childlovskii. Crawling for domainspecifichidden web resources. In Web Information SystemsEngineering, 2003. WISE 2003. Proceedings of the Fourth InternationalConference on, pages 125–133. IEEE, 2003.
5. Paper No. [10] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Towardlarge scale integration: Building a metaquerier over databaseson the web. In CIDR, pages 44–55, 2005.

Author(s) Profile



Anita Vittal Kodam, received B.E in Information Technology From Walchand Institute of Technology, Solapur affliated to Solapur University and pursuing M.E in Computer Science and Engineering from N.B.Navale Sinhgad College Of Engineering, Solapur affiliated to Solapur University.

