KEYWORD SEARCH TO LOCATE THE DEEP WEB DATABASES WITH CRAWLER

Mr. Y. Poojitha[1]
3rd Year Student,
Department of Computer Science,
SV U CM & CS, Tirupati.

Prof. S. Rama Krishna[2],
Professor,
Department of Computer Science,
SV U CM & CS,, Tirupati.

Abstract: Due to extensive usage of Internet, substantial amount of data has extended widely over web, which serve access to particular data or to fetch more relevant data. It would be challenging to the search engine to provide quick results that is most relevant to the users. To search the relevant data and to reduce amount of time in fetching data, here propose the ―Smart Crawler. This returns most relevant data from the popular and most specific websites. It uses multiple search engines that processes the query provided by the user, cluster the results collected in a single platform and performs two stage crawling on data and URLs. In which in-site map generation is done to obtain relevant site with techniques such as reverse searching and page ranking.

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 zetta bytes were reached and 0.3 zetta bytes were consumed worldwide in 2007 . An IDC report estimates that the total of all digital data created, replicated, and consumed will reach 6 zetta bytes 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 In fomine , Clusty [8], Books In Print [9] 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.

A main feature of our method is the representation of query interfaces in a hierarchical format. We provide concrete examples of applications that utilize query interfaces and we show how these applications would benefit from a hierarchical representation of query interfaces.

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

METHODOLOGY

We 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 in-site 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.

System Model

To efficiently and effectively discover deep web data sources, SmartCrawler is designed with a two stage architecture, site locating and in-site exploring. 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 from the site. Specifically, the site locating stage starts with a seed set of sites in a site database. Seeds sites are candidate sites given for SmartCrawler 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, SmartCrawler 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 databases, which are ranked by Site Ranker to prioritize highly relevant sites. The Site Ranker is improved during crawling by an Adaptive Site Learner, which adaptively learns from features of deep-web sites (web sites containing one or more searchable forms) found. 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. After the most relevant site is found in the first stage, the second stage performs efficient in-site exploration for excavating searchable forms. Links of a site are stored in Link Frontier and corresponding pages are fetched and embedded forms are classified by Form Classifier to find searchable forms. Additionally, the links in these pages are extracted into Candidate Frontier. To prioritize links in Candidate Frontier, Smart Crawler ranks them with Link Ranker. Note that site locating stage and in-site exploring stage are mutually intertwined. When the crawler discovers a new site, the site’s URL is inserted into the Site Database. The Link Ranker is adaptively improved by an Adaptive Link Learner, which learns from the URL path leading to relevant forms.

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. To achieve these goals, using the links in downloaded web pages is not enough. This is because a website usually contains a small number of links to other sites, even for some large sites. For instance, only 11 out of 259 links from web pages of aaronbooks.com pointing to other sites; amazon.com contains 54 such links out of a total of 500 links (many of them are different language versions, e.g., amazon.de). Thus, finding out-of-site links from visited web pages may not be enough for the Site Frontier. In fact, our experiment in Section 5.3 shows that the size of Site Frontier may decrease to zero for some sparse domains. To address the above problem, we propose two crawling strategies, reverse searching and incremental two-level site prioritizing, to find more sites

Site Ranker:

Once the Site Frontier has enough sites, the challenge is how to select the most relevant one for crawling. In SmartCrawler, Site Ranker assigns a score for each unvisited site that corresponds to its relevance to the already discovered deep web sites.

In Site Exploring:

Once a site is regarded as topic relevant, in-site exploring is performed to find searchable forms. The goals are to quickly harvest searchable forms and to cover web directories of the site as much as possible. To achieve these goals, in-site exploring adopts two crawling strategies for high efficiency and coverage. Links within a site are prioritized with Link Ranker and Form Classifier classifies searchable forms.

Link Ranker: Link Ranker prioritizes links so that Smart Crawler can quickly discover searchable forms. A high relevance score is given to a link that is most similar to links that directly point to pages with searchable Forms.

Feature Selection And Ranking:

Smart Crawler encounters a variety of web pages during a crawling process and the key to efficiently crawling and wide coverage is ranking different sites and prioritizing links within a site. This section first discusses the online feature construction of feature space and adaptive learning process of SmartCrawler, and then describes the ranking mechanism.

CONCLUSION

In this paper, we propose an effective harvesting framework for deep-web interfaces, namely SmartCrawler. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. Smart Crawler is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring. Smart 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.

Applications present convenient means for hackers to spread malicious content on Facebook. However, little is understood about the characteristics of malicious apps and how they operate. In this paper, using a large corpus of malicious Facebook apps observed over a 9-month period, we showed thatmalicious apps differ significantly from benign apps with respect to several features. For example, malicious apps aremuchmore likely to share names with other apps, and they typically request fewer permissions than benign apps. Leveraging our observations, we developed FRAppE, an accurate classifier for detecting malicious Facebook applications.Most interestingly, we highlighted the emergence of app-nets—large groups of tightly connected applications that promote each other. We will continue to dig deeper into this ecosystem of malicious apps on Facebook, and we hope that Facebook will benefit from our recommendations for reducing the menace of hackers on their platform.

REFERENCES

[1] C. Pring, —100 social media statistics for 2012,‖ 2012 [Online]. Available:

http://thesocialskinny.com/100-social-media-statistics-for-2012/

[2] Facebook, Palo Alto, CA, USA, —Facebook Opengraph API,‖ [Online]. Available:http://developers.facebook.com/docs/reference/api/

[3] —Wiki: Facebook platform,‖ 2014 [Online]. Available: http://en. wikipedia.org/wiki/Facebook_Platform

[4] —Pr0file stalker: Rogue Facebook application,‖ 2012 [Online]. Available: https://apps.facebook.com/mypagekeeper/?status=scam_report-_fb_survey_scam_pr0file_viewer_2012_4_4

[5] —Whiich cartoon character are you—Facebook survey scam,‖ 2012 [Online].

https://apps.facebook.com/mypagekeeper/?status=scam_report_fb_survey_scam_whiich_cartoon_character_are_you_2012_03_30

[6] G. Cluley, —The Pink Facebook rogue application and survey scam,‖ 2012 [Online].Available:http://nakedsecurity.sophos.com/2012/02/27/pink-facebook-survey-scam/

[7] D. Goldman, —Facebook tops 900 million users,‖ 2012 [Online]. Available: http://money.cnn.com/2012/04/23/technology/facebookq1/index.htm

[8] HackTrix, —Stay away from malicious Facebook apps,‖ 2013 [Online]. Available: http://bit.ly/b6gWn5

Authors Profile

Y POOJITHA, received Bachelor of Computer Science degree from Sri Venkateswara University, Tirupathi in the year of 2013-2016. Pursuing Master of Computer Applications from Sri Venkateswara University, Tirupati in the year of 2016-2019. Research interest in the field of Computer Science in the area of Big Data Analytics , Cloud Computing andData mining

Prof Dr S. Ramakrishna, working as a Professor in Dept of Computer Science, Sri Venkateswara University College of Commerce Management and Computer Science, Tirupati, (AP)-India. Received M.Sc, M.Phil, M.Tech (IT) and Doctorate in Computer Science from S.V University, Tirupati, having 27 years experience in teaching field. Additional Assignments Working as Dean of Examinations for S.V University, Worked as Additional Convener for S.V University RESET Examinations, Worked as Coordinator for M.Sc Computer Science, Worked as BoS Chairman in Computer Science. Research Papers Published in National & International Journals :99, Total Number of Conferences participated :33, Total number of Books Published:7, Total number of Training Programs Attended : 3, Total number of Orientation & Refresher Courses Attended : 4.Number of research degrees awarded under my guidance :- M.Phil: 20,Ph.D:20.