Web Forum Crawling

R.Priya, Ms.S.Dhanalakshmi, S.Priyadharshini

Computer Science and Engineering, Arunai Engineering College, Thiruvannamalai.

Abstract- The supervised web-scale forum crawler is to crawl relevant forum content from the web with minimum overhead. Forum threads contain information content that is the target of forum crawlers. each forums have different layouts or styles and have different forum software packages, they always have similar constant navigation paths connected by specific URL types to direct users from entry pages to thread page. we reduce the web forum crawling problem to a URL-type recognition problem. And shows how to learn accurate and effective regular expression patterns of constant navigation paths from automatically created training sets using aggregated results from weak page type classifiers. Robust page type classifiers can be experienced from as few as five annotated forums and applied to a large set of unseen forums.

Index Terms- EIT path, forum crawling, ITF regex, page classification(PC), page type, URL pattern learning, URL type, present Forum Crawler Under Supervision (FoCUS).

I. INTRODUCTION

The World-Wide-Web (WWW) is growing exponentially and has become increasingly difficult to retrieve relevant information on the web. The rapid growth of the WWW poses unprecedented scaling challenges for general purpose crawlers and search engines. In this paper, we present Forum Crawler Under Supervision (FoCUS), a supervised web-scale forum crawler. The goal of FoCUS is to crawl relevant forum content from the web with minimal overhead, this crawler is to selectively seek out pages that are relevant to a predefined set of topics, rather than collecting and indexing all accessible web documents to be able to answer all possible ad-hoc queries. FoCUS continuously keeps on crawling the web and finds any new web pages that have been added to the web, pages that have been removed from the web. Due to growing and dynamic nature of the web; it has become a challenge to traverse all URLs in the web documents and to handle these URLs. We will take one seed URL as input and search with a keyword, the searching result is based on keyword and it will fetch the web pages where it will find that keyword.

All the major search engines have highly optimized crawling system, although working and details of documentation of this system are usually with their owner. It is easy to build a crawler that would work slowly and download few pages per second for a short period of time. In contrast, it's a big challenge to build the same system design, I/O, network efficiency, robustness and manageability. Every search engine is divided into different modules among those modules crawler module is the module on which search engine relies the most because it helps to provide the best possible results.

II. LITERATURE SURVEY

A.WEB CRAWLING**:**

Automated traversal of web to collect all the useful informative pages, effectively and efficiently Gather information about link structure interconnecting the informative pages. Web application designed to manage user created content. Online discussion area where anyone can discuss their favorite topics.

1.How it works**:**

Pre-samples few pages to discover the repetitive regions. Group pre-sampled pages into clusters based on their repetitive regions where each cluster can be considered a vertex in the sitemap.

B.Irobot

Tool to crawl through Web Forums Intelligent enough to understand structure of forums before selecting traversal path. It work towards two issues are Important page and Important links.

3.Read URL

We are concentrating on focus ontology which search for the relevant web pages based on the keyword we give. Actually it forms a hierarchy of links. The web information on the particular web page for a particular keyword, which we give as, input. It will search for the link on that seed URL and after that switch to that link and find another link on that web page but it should match with the keyword, it will do like that until it reach the limit that we set. But it may be possible that it will not found the number of links that we set before. Then it shows that the web page is not having any further link for that particular keyword. While fetching the links the user profiles also make sure that it should fetch only the unique links, mean s that it should not revisit the same link again and again. Finally , when we finished with the links, we will give one txt file as input and run the three pattern matching algorithm.

III. STUDIES AND FINDINGS

a. READ URL:

We are concentrating on focus ontology which search for the relevant web pages based on the keyword we give. Actually it forms a hierarchy of links. The web information on the particular web page for a particular keyword, which we give as, input. It will search for the link on that seed URL and after that switch to that link and find another link on that web page but it ,should match with the keyword, it will do like that until it reach the limit that we set. But it may be possible that it will not found the number of links that we set before. Then it shows that the web

page is not having any further link for that particular keyword. While fetching the links the user profiles also make sure that it should fetch only the unique links, mean s that it should not revisit the same link again and again. Finally , when we finished with the links, we will give one txt file as input and run the three pattern matching algorithm.

b. PATTERN RECOGNITION:

Here with pattern we mean only text. Pattern matching is used for syntax analysis. When we compare pattern matching with regular expressions then we will find that patterns are more powerful, but slower in matching. A pattern is a character string. All keywords can be written in both the upper and lower cases. A pattern expression consists of atoms bound by unary and binary operators. Spaces and tabs can be used to separate keywords. Text mining is an important step of knowledge discovery process. It is used to extract hidden information from not-structured or semi-structured data. This aspect is fundamental because much of the web information is semi-structured due to the nested structure of HTML code, much of the web information is linked, and much of the web information is redundant. Web text mining helps whole knowledge mining process of mining, extraction and integration of useful data, information and knowledge from the web page content. Pattern recognition is applied on the web information like this , When we start the retrieval it will give me the links related to the keyword. It will then read the web pages that are extracted from the links and while it will read the web page it will extract only the content. Here content means only the text that is available on the web page. It should not include images, tags, and buttons. The extracted content should be stored in some file. But it should not include any HTML tags.

c. IDENTIFICATION PROCESS:

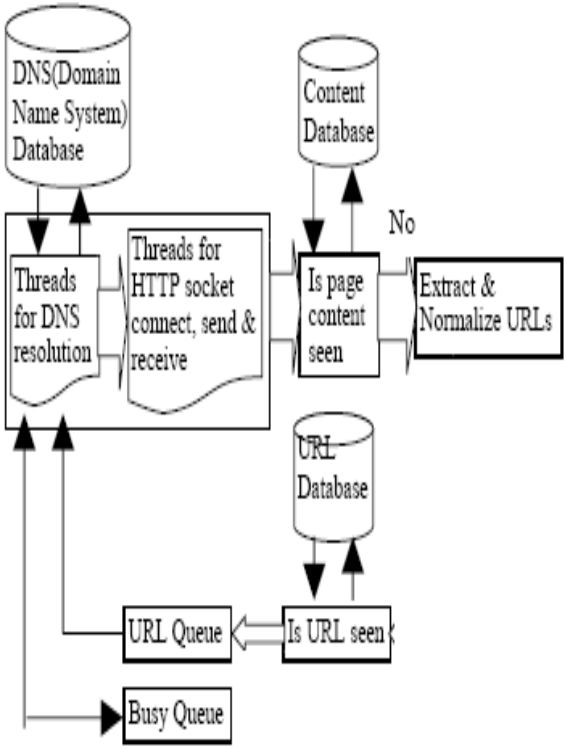
This process will identify the required urn is whether right kind of link or wrong kind link. It will identify the url, protocol link also for retrieve the relevant web page for user requesting. It's used to omit bad urls while user requesting web pages. Bad urls are identified by pattern of protocol occur on the relevant web pages on the server side.

d. DOWNLOADING PROCESS:

After completion of all process the downloading will started. It will start to downloading requesting url link of users need. After three checking process only it will downloaded the relevant link for users request. It will working efficiently to users, the requested link will retrieve.

e. INDEX URL AND THREAD URL TRAINING SETS:

Recall that an index URL is a URL that is on an entry or index page; its destination page is another index page; its anchor text is the board title of its destination page. A thread URL is a URL that is on an index page; its destination page is a thread page; its anchor text is the thread title of its destination page. We also note that the only way to distinguish index URLs from thread URLs is the type of their destination pages. Therefore, we need a method to decide the page type of a destination page.The index pages and thread pages each have their own typical layouts. Usually, an index page has many narrow records, relatively long anchor text, and short plain text; while a thread page has a few large records Each post has a very long text block and relatively short anchor text. An index page or a thread page always has a timestamp field in each record, but the timestamp order in the two types of pages are reversed: the timestamps are typically in descending order in an index page while they are in ascending order in a thread page.



IV. RELATED WORK

A.Evaluations of FoCUS Modules

Evaluation of Index/Thread URL Detection

To build page classifiers, we manual selected five index pages, five thread pages, and five other pages from each of the 40 forums and extracted the features. Results of Entry URL Discovery manually selected 10 index pages, 10 thread pages, and 10 other pages from each of the 160 forums. This is called 10-Page/160 test set. We then ran Index/Thread URL Detection module described "Index URL and Thread URL Training Sets" in Section 4.3.1 on the 10-Page/160 test set and manually checked the detected URLs. Note that we computed the results at page level not at individual URL level since we applied a majority voting procedure.

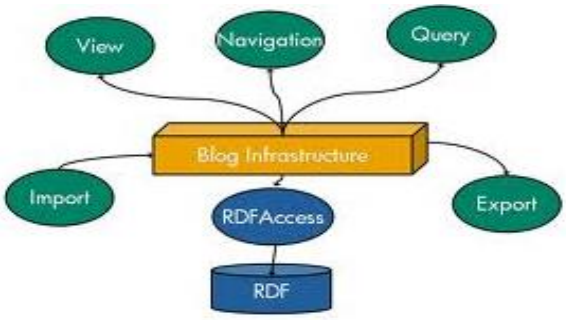


Fig.3.rdf

B.Evaluation of Page-Flipping URL Detection

To test page-flipping URL detection, we applied the module described "Page-Flipping URL Training Set"and manually checked whether it found the correct URLs. The method achieved 99 percent precision and 95 percent recall. It is not successfuls mainly due to JavaScript-based page-flipping URLs or HTML DOM tree alignment error.

C.Evaluation of Entry URL Discovery

As far as we know, all prior works in forum crawling assume that an entry URL is given. But finding forum entry URL is not trivial. For each forum in the test set, we randomly sampled a page and fed it to this module. Then, we manually checked if the output was indeed its entry page. In order to see whether FoCUS and the baseline were robust, we repeated this procedure 10 times with different sample pages. The baseline had 76 percent precision and recall. On the contrary, FoCUS achieved 99 percent precision and 99 percent recall. The low standard deviation also indicates that it is not sensitive to sample pages. There are two main failure cases: 1) forums are no longer in operation and 2) JavaScript generated URLs which we do not handle currently.

D.Evaluation of Online Crawling

We have shown in the previous sections that FoCUS is efficient in learning ITF regexes and is effective in detection of index URL, thread URL, page-flipping URL, and forum entry URL. In this section, we compare FoCUS with other existing methods in terms of effectiveness and coverage .

V. CONCLUSION

We are concentrating on focus crawler which search for the relevant web pages based on the keyword we give. Actually it forms a hierarchy of links. The crawler on the particular web page for a particular keyword, which we give as, input. It will search for the link on that seed URL and after that switch to that link and find another link on that web page but it should match with the keyword, it will do like that until it reach the limit that we set.Knutt-Morris-Pratt method takes advantage of the partial-match, Identify the bad URL in a website. No. of character present in a web page. Identify type of protocol used for the web page.Retrieve the web pages.we apply pattern recognition over text.Pattern symbolizes check text only.Check how much text is available on web page.

REFERENCES

[1] C. Gao, L. Wang, C.-Y. Lin, and Y.-I. Song, "Finding Question- Answer Pairs from Online Forums," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 467-474, 2008.

[2] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, "Deriving Marketing Intelligence from Online Discus- sion," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 419-428, 2005.

[3] Y. Guo, K. Li, K. Zhang, and G. Zhang, "Board Forum Crawling: A Web Crawling Method for Web Forum," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence, pp. 475-478, 2006.

[4] M. Henzinger, "Finding Near-Duplicate Web Pages: A Large- Scale Evaluation of Algorithms," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 284-291, 2006.

[5] H.S. Koppula, K.P. Leela, A. Agarwal, K.P. Chitrapura, S. Garg, and A. Sasturkar, "Learning URL Patterns for Webpage De- Duplication," Proc. Third ACM Conf. Web Search and Data Mining, pp. 381-390, 2010.

[6] K. Li, X.Q. Cheng, Y. Guo, and K. Zhang, "Crawling Dynamic Web Pages in WWW Forums," Computer Eng., vol. 33, no. 6, pp. 80-82, 2007.

[7] G.S. Manku, A. Jain, and A.D. Sarma, "Detecting Near-Duplicates for Web Crawling," Proc. 16th Int'l Conf. World Wide Web, pp. 141- 150, 2007.

[8] U. Schonfeld and N. Shivakumar, "Sitemaps: Above and Beyond the Crawl of Duty," Proc. 18th Int'l Conf. World Wide Web, pp. 991- 1000, 2009.

[9] X.Y. Song, J. Liu, Y.B. Cao, and C.-Y. Lin, "Automatic Extraction of Web Data Records Containing User-Generated Content," Proc. 19th Int'l Conf Information and Knowledge Management, pp. 39-48, 2010.

[10] "WeblogMatrix," http://www.weblogmatrix.org/, 2012.

[11] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine." Computer Networks and ISDN Systems, vol. 30, nos. 1-7, pp. 107-117, 1998.

AUTHORS

First Author: R.PRIYA,ME(final yr),ARUNAI ENGINEERING COLLEGE,rpriyajoseph77@gmail.com
Second Author: –Ms.S.DHANALAKSHMI,Asst Prof(CSE),ARUNAI ENGINEERING COLLEGE,dhanalakshmi1984@gmail.com
Third author: S.PRIYADHARSHINI, ME(final yr),ARUNAI ENGINEERING COLLEGE,priyainfotech26@gmail.com

Correspondence Author – R.PRIYA,ME(final yr),ARUNAI ENGINEERING COLLEGE,rpriyajoseph77@gmail.com