FORUM CRAWLER UNDER SUPERVISION METHODOLOGY FOR AUTHENTICATION PATH IN WEB FORUMS

A Naveen Kumar [1], T Lavanya[2]

[1]M.Tech (CS) Scholar, [2]Assistant Professor

Nalanda Institute of Engg & Tech. (NIET), Siddharth Nagar, Guntur, (India)

ABSTRACT

In this paper we propose a Focus Crawler Under Supervision method which will serve as a supervised web-scale forum crawler. The goal of this method is only to crawl relevant content in the forum from web with a reduced overhead. The threads of forum will contain the information which is the target for the web crawlers. We know that the forums will have different styles or layouts and are sometimes powered by various software but they are said to be having similar impact for navigation paths which will be based or connected by specific URL types that will lead the user from the entry page to various thread pages. By observing this we reduce the problem of web crawling into a URL type recognition problem and by this we can show how the efficient and accurate regular expressions patterns of the essentially connected navigation paths from an training set which is created automatically using the aggregation of results from classifiers of weak page type. A few robust page type classifiers can be trained as forums that are annotated and can be applied to a large set of forums that are unseen. The test results of this have shown that this method shown 98% effectiveness and is able to cover 97% on a large set of test forums powered by nearly 150 different software packages used for forums. This method is very much useful because Internet is emerging exponentially and has become more progressive. But because of this the data retrieval has become complicated over the internet when the user tries to retrieve any important information. This rapid growth of the internet has unpredicted challenges for the Search Engines and General purpose crawlers. But the method proposed in this paper is of intention to crawl forum of relevant information from the internet with minimal overhead. This method since it keeps continuous crawling it will be able to say which the new pages are and which the removed pages are.

I. INTRODUCTION

Internet has evolved as the most powerful source of information. Users from various places use it for various types of data. There comes the need to make the search process of internet for information more efficient. As the size of the internet grows the volume of the information to be crawled also grows in direct proportional to each other. This will make the search engine engineering a very challengeable task. Here in this method we perform three tasks basically they are: Search the Internet or pages based on important words or keywords. They follow indexing mechanism for the words they used and where they used those word because this makes crawling a bit simpler. Now they will allow the users to view the words individually or in combinations where all these words can be seen in the index. WebCrawler is basically a computer program that is able of browsing the Internet following some methodologies given. The main functionality of the crawler is to go through or crawl through

the links provided getting information from them and adding that information to the index of search engines as specified in methodology. Although the internet is a vast source of information of almost all type, this information will be often scattered among different servers and hosts. So here our aim is to design a most excellent and promising search manner in less time. Here in this paper we are proposing a crawler under supervision, supervised forum-crawler of web-scale merged with progression to find copyright infraction. There are two issues for any crawler: First it should have a capability to plan. Second it needs exceedingly optimized and vigorous architecture for its system so that it will be able of downloading number of pages per second though there are crashes and it will be manageable and considerate of resources and web servers.

II. RELATED WORK

A tale approach exists for learning the patterns of regular expression of URLs that can lead the crawler from the entry page to the target page. These target pages were found through comparing the DOM trees of pages where a trial of target page is preselected. Though it is very effective it only works for a particular site where the preselected sample target page is drawn. When we need to crawl the new site it is very much essential to repeat the same process every time. But this is not much useful phenomenon for large scale crawling. So in contrast, in our proposed approach we make the crawler to learn the URL patterns across multiple sites so that it can automatically find forum's entry page where a page is given from the forum. The main problem with the existing approach is it did not explain procedure for discovering and traversing the URLs. The rules are very specific which says that this can be applied to specific forums only which are powered by the particular software package in which the heuristics are conceived. But to our unfortunate the forum matrix states that there is lot of incomparable forum software packages used on Internet. We also have a presented algorithm to address the problem of traverse path selection. This has a scheme of page-flipping link and skeleton link. Skeleton links are explained as the most important links which support the structure of the forum site. Here the importance is determined by how much information it gives and its coverage. The page-flipping links are determined by their connectivity. By following these two i.e. Skeleton and Page-Flipping links it is demonstrated that iRobot is able of achieving effectiveness and courage. Another related work for our proposed system is work proposed to avoid duplicate detection. That is because the forum crawling also desires to duplicates removal since it will increase the data and decreases the understandability of it. If the duplicate detection is content based then it can be carried out only when the pages are downloaded. But URL based duplicate detection is not supportive. In the forums which are to be crawled there will be Index URLs and thread URLs along with page-flipping URLs and a simple URL string de-duplication technique is adopted. To avoid the unnecessary crawling some techniques are included they are industry standards such as "no follow", Robots Exclusion standard and sitemap protocol.

III. PROPOSED SCHEME

In this we discuss about the proposed scheme and how we have to implement it for crawling.

3.1 Overview of Proposed Scheme

Here we present the architectural diagram which consists of two major parts: Firstly the learning part which will learn the ITF regular expressions of given forum. Secondly the crawling part will apply these learned ITF regular expressions to crawl all the threads efficiently.

a) **Page Type**: Here we classify the forum pages into following page types.

Entry Page: This will be the home page of forum. It will contain a list of boards about the forum and is the least familiar ancestor of all the thread pages in the forum.

Index Page: This is a page of board in forum. This one typically will contain a table-like structure which contains information of a board or thread.

Thread Page: This is a page of thread in a forum and it contains a list of posts where this post will have user generated content which belongs to comparable division.

b) **URL type**: Here we discuss about types of URL in this module

Index URL: It is the URL which will be on an entry page or index page and will be pointing towards index page. It will be having the anchor text that shows the title of Destination Board.

Thread URL: It is the URL that will be on an index page and will be pointing to the thread page. It will be having the anchor test which is the title of the destination thread.

3.2 ITF Regular Expressions Learning

Here we learn about the ITF regular expressions, our proposed mechanism which adopts a supervision training procedure contains two steps they are:

A) Constructing URL training Sets:

Goal of this URL training sets construction is constructing the sets of highly precise index URL, page-flipping URL strings and Thread URL automatically for ITF regular expression learning. Here we make use of a comparable process that can construct Index and Thread URL training sets since they have the properties that are very much comparable with only exception of their destination pages.

B) Learning ITF Regular Expressions:

This sub module deals with the process which says how to construct index URL, page-flipping URL and Thread URL training set. We here also explain how to learn the ITF regular expressions from these sets.

3.3 Online Crawling

Here we perform the online crawling using a breadth-first strategy. As per the proposed mechanism the entry URL will be pushed into a URL queue first. Next when it has to download the page it will fetch the URLs from the URL queue and then it will download the page corresponding to that URL. Then all these outgoing URLs which by now will be coordinated with any one of the learnt regular expression into the URL queue. This process will be repeated in the proposed mechanism until either the URL queue is empty or conditions are satisfied. Our proposed mechanism only needs to apply the learned ITF regular expressions in newly downloaded pages on innovative outgoing URLs to make the online crawling more proficient. This mechanism does not need to group classifying pages, outgoing URLs, learn regular expressions and recognize page-flipping URLs again for that forum.

3.4 Entry URL Discovery

Here we discuss about the entry URL. Entry URL is explained as the one which needs to be precise for starting the crawling process. When we discuss particularly about the web-scale crawling then the manual forum for entry URL bad notation is not practical. The discovery of forum entry URL is not a trivial task because the

URLs vary from forums to forums. That's the reason why we include a Heuristic rule for stumbling on the entry URL as a base line. This is because the Heuristic base line will try to stumble the URL with the help of following URL which are ending with '/' in a URL. The key words are:

1. Forum
2. Board
3. Community
4. Bbs and
5. Discuss.

If any of the above keyword is found then the host of URL to this keyword will be extracted and is made as it entry URL. If it did not happen then the URL host itself is made as the entry URL. To make our proposed system more practical and scalable we here use some techniques for designing a very simple and effective discovery method for forum entry URL.

IV. RESULT

In previous section we have seen and discussed about various proposed things. And in this section we are going to see the results of implementations for the proposed techniques of the previous section. Here we are going to explain about the results with the help of a table and a graph and then in a separate module we explain with a detailed description such as overview, Entry URL discovery and Online Crawling.

4.1 Entry URL Discovery

Here in this module we are going to discuss the forum crawling as assumed in URL entry. We had discussed previously that finding a forum Entry URL is not trivial. To show this we included Heuristic baseline into our URL entry discovery method. Here for each and every forum in the sample set we fed into this module a randomly sampled page. And a check is done to confirm whether the output of that is an entry page or not. In order to check whether our proposed mechanism and the baseline given for finding the entry URL are robust or not, we repeated the process for 10 times with a set of unusual sample pages. The results that are originated from these checks are given in table 1. The baseline given for the Entry URL discovery had 76% precision and recall. On the other hand our proposed mechanism for web crawling achieved 99% of recall and 99% of precision. It has a very low standard deviation which also designates that it will not be sensitive to sample pages. There exist two main failure situations for this. They are

1) If the forums are no longer in operation and
2) The URLs generated by JavaScript which are not handled currently.

Here we balanced different types of URLs for finding the efficiency of URL discovery and thread URL in terms of generic crawler which you will find in figure-02.

Method	Precision %		Recall %	
	Average	Std.Dev	average	Std.Dev
BaseLine	76.38	1.74	76.38	1.74
	99.31	0.20	99.13	0.32

Table 1: Results of Entry URL Discovery

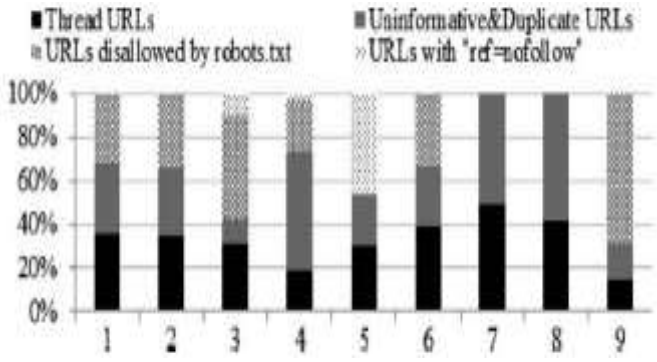

Fig 1: Ratio of different URLs discovered by a generic crawler

4.2 Evaluation of Online Crawling

Here we evaluate our proposed mechanism with the existing methods find the efficiency of the result.

Below there is a table where we preferred nine forums among a great collection of test forums for the investigation of this assignment. Here eight among nine forums belongs to popular software packages which are used by many forum sites. This says that about 53 percent of forums are powered by 200 packages as deliberated in this paper, and are about 15 percent of all the forums we found for testing.

ID	Forum	Forum Name	Software	Threads
1	forums.afterdawn.com	AfterDawn Forums	Customized	535,383
2	forums.asp.net	ASP.NET forums	Community Server	66,966
3	forums.xda-developer.com	Andriod Forums	vBulletin	299,073
4	Bbs.cqzg.in	Chinese Forums	Discuz	428,555
5	Forums.crackberry.com	BlackBerry Forums	vBulletin V2	525,381
6	Jkcn.net/bbs	Japanese Forums	IP.Board	180,692
7	Forums.gentoo.org	Gentoo Forums	phpBB V2	681,813
8	Techreport.com/forums	TechReport	phpBB	65,083
9	www.redanwhitekop.com/forum	Liverpool IC Forum	SMF	138,963

Table 2: Forums Used In Online Crawling Evaluation

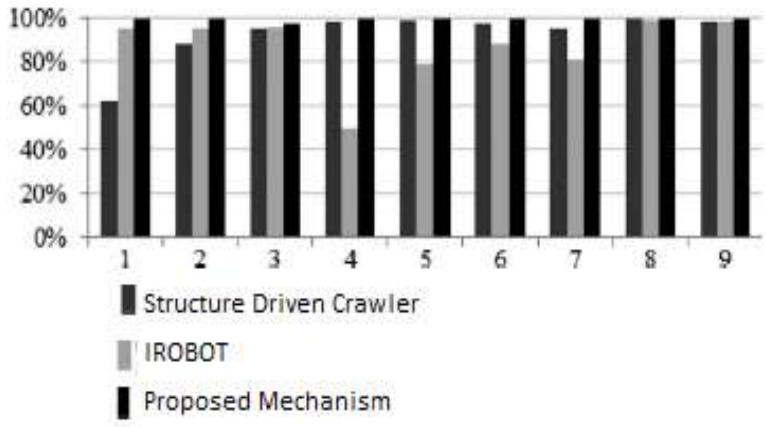

Fig 2. Coverage Comparison between the Structure-Driven Crawler, Irobot and Proposed Mechanism

Here in the above figure comparison of coverage between the Structure-driven crawler, iRobot and Proposed Mechanism. Among them the Structure-driven crawler is not a forum crawler but it can be used for the forums. To make this Structure-driven crawler more useful for comparison we used it for finding the page-flipping URL patterns so that its coverage can be increased. On the other hand coming to the iRobot we re-implemented it. Here we have permitted the Structure-driven crawler, iRobot and the proposed method to crawl each forum up to the situation where there is no page left for retrieval. After doing so we made the count of the threads and other pages that were crawled, correnpondingly.

V. CONCLUSION

A tale method crawler which is able of downloading and storing the web pages frequently for a search engine is being presented in this paper. A need for such a mechanism aroused since the rapid growth of internet made searching a suitable link nearly impossible. We also symbolized the technique in our proposed system which is actually developed exactly for extracting only the relevant web pages that is of interested topic from the internet. The design of our proposed system is capable for evaluation of the text which can be found on a link with given input as a text file. Crawler will make use of the pattern recognition technique to generate the count of how many times does the input text exists in the text establish on a link. Information that is generated by this will give overhanging in the efficiency of the pattern matching. Proposed system will keeps on crawling the web continuously to find whether any new page is added or an existing page is removed from the web. Since there is a lot of vibrant activity growth of the internet, traversal of the URLs in the web documents and handling these URLs has become confront. Here we are taking a seed URL as input and search with a keyword, the result of the search is based on the keyword and so will obtain the pages in which that keyword is present.

REFERENCES

[1] 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.

[2] R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang, “iRobot: An Intelligent Crawler for Web Forums,” Proc. 17th Int‟l Conf. World Wide Web,pp. 447-456, 2008.

[3] Dasgupta, R. Kumar, and A. Sasturkar, "De-Duping URLs via Rewrite Rules," Proc. 14th ACM SIGKDD Int"l Conf. Knowledge Discovery and Data Mining, pp. 186-194, 2008.

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

[5] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T.Tomokiyo, "Deriving Marketing Intelligence from Online Discussion," Proc. 11th ACM SIGKDD Int"l Conf. Knowledge Discovery and Data Mining, pp. 419-428, 2005.

[6] 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.

AUTHOR PROFILE

	**A Naveen Kumar** is currently pursuing M.Tech in the Department of Computer Science, from Nalanda Institute of Engineering & Technology (NIET), siddharth Nagar, Kantepudi(V), Sattenapalli (M), Guntur (D), Andhra Pradesh , Affiliated to JNTU-KAKINADA.
	**T Lavanya** working as Assistant Professor at Nalanda Institute of Engineering & Technology (NIET), siddharth Nagar, Kantepudi(V), Sattenapalli (M), Guntur (D), Andhra Pradesh , Affiliated to JNTU-KAKINADA.