Automated Path Ascend Crawl Method for Web Forums

Mr. Prabakaran V,
Department of CSE,
Velammal Institute of Technology,Thiruvallur,
Chennai-601204
vpraba1991@gmail.com

Mr. Thirunavukkarasu M,
Department of CSE,
Velammal Institute of Technology,Thiruvallur,
Chennai-601204
thiruarasu1991@gmail.com

***Abstract*—FoCUS (Forum Crawler Under Supervision), is a supervised web-scale forum crawler. The goal of FoCUS is to crawl relevant forum content from the web with minimal overhead. FoCUS is an automation engine that will dynamically crawl the relevant content in a forum . Forum threads contain information content that is the target of forum crawler. Cleanup of data and moving the contents to the appropriate web pages is the major scope of the project. The content of forum may be the queries asked by the users. After crawling the content, FoCUS will dynamically move the queries in the related forum, which will deal the particular query. Then FoCUS cleanup the unrelated query from the particular forum, and that free space is allocated to new queries posted by user. FoCUS take six path from entry page to thread page. It helps the frequent thread updation in forum. FoCUS makes use the technique called differential content extraction, which helps to maintain a record for already crawled data. In each time FoCUS will not crawl the forum data from the beginning, it will maintain a record of already crawled data and manipulates only the newly posted queries.**

***Keywords*— EIT Path, Forum Crawling, ITF Regex, URL Type.**

I. INTRODUCTION

Internet forums [12] (also called web forums) are important services where users can request and exchange information with others. It helps to know user's opinion about a product and understand what are their expectations. To harvest knowledge from forums, their content must be downloaded first. A web forum crawler which can collect the forum data automatically according to scheduled time such as once in a week. The collected data will be stored in the database. The data can be used for data mining or social network analysis.

In the existing system iRobot forum crawler is used, which crawl the forum content. It does not deal with the frequent thread updation in forum.iRobots tree like traversal didn't allow more than one path from starting page node to same ending page node. So it takes only one path (first path that is entry-board-thread) from entry to thread page. Here sampling strategy and in formativeness estimation is not robust. Existing system doesn't follow the differential content extraction. That is it doesn't maintain a record of previously stored data. When new queries are posted by user, the crawler can start the crawling process from the beginning in every time. So it become a time consuming process.The main drawbacks of the existing system are:

- No clear segregation of page identification is carried out
- It takes only one path from entry to thread page.
- It doesn't make use of differential content extraction technique.

FoCUS tried to create an automation engine which will take care of traversing the contents dynamically. Moving towards the hyperlinks related to the forum and cleanup the related links. Integrating the missed out data pages in future were considered as the core proposed approaches included in the system. In our proposed system, we are utilizing the features of differential content extraction instead of an inefficient entire system scanning. This option will enhance the performance of the system very much. The option of differential content is done with the help of page indexes and number of links options or link value. In addition, amend and building the knowledge database enable the system a very efficient one in a longer vision. Scanning the entire web pages through Key match cum Knuth–Morris–Pratt algorithm is used. The proposed system maintains a record of already crawled data. The six paths from entry to thread page are given below:

1. entry→ board →thread
2. entry →list-of-board→ board→ thread
3. entry→ list-of-board & thread→ thread
4. entry→ list-of-board & thread →board→ thread
5. entry→ list-of-board→ list-of-board & thread→ thread
6. entry→ list-of-board→ list-of-board & thread→board →thread

The main advantages of Focus are given below:

- Automation web crawling is done with this application.
- FOCUS takes six paths from entry to thread page.
- Differential content extraction is used.

The major contributions of this paper are as follows:
1. We create an automatic engine which will crawl the forum pages automatically..
2. Focus makes use of the technique called differential content extraction which helps to maintain the record of already crawled data. So the effectiveness becomes increased.
3. Cleanup of data and moving the contents to the appropriate web pages is the major scope of FoCUS.
4. After remove the unrelated links, FoCUS allocates that space to the newly posted queries.

II. RELATEDWORK

Vidal Caj.R, Yang. J.M, Lai.W, Wang.Y, and Zhang.L [2] , iRobot has an intelligence to understand the content and the structure of a forum site, and then decide how to choose traversal paths among different kinds of pages. Furthermore, it also achieves the following advantages: (1) significantly decreases the duplicate and invalid pages ;(2) saves substantial
network bandwidth and storage as it only fetches informative pages from a forum site;(3) It provides a great help for further indexing and data mining;(4) Effectiveness: it intelligently skip most invalid and duplicate pages, while keep informative and unique ones;(5) Efficiency: iRobot only need a few pages to rebuild the sitemap. It is also have some disadvantages such as; It follow a tree like traversal, so it didn't allow more than one path from starting page to ending page. It doesn't deal how to design a repository for forum archiving.

Wang.Y, Yang's.-M, Lai.W, Cai.R, Zhang.L, and Ma.W.-Y [5], Exploring Traversal Strategy is a traversal strategy consists of the identification of the skeleton links and the detection of the page-flipping links. Furthermore, it achieve the following advantages: (1) The skeleton links instruct the crawler to only crawl valuable pages and meanwhile avoid duplicate and uninformative ones;(2) page-flipping links tell the crawler how to completely download a long discussion thread which is usually shown in multiple pages in Web forums. It has some demerits such as it doesn't deal with how to optimize the crawling schedule to incrementally update the archived forum content. And also it doesn't deal how to parse the crawled forum pages to separate replies in each post thread**.**

Brin.S and Page.L [1], Web Search Engine" it is a large-scale search engine which makes heavy use of the structure present in hypertext, for example is Google. Google is designed to crawl and index the web efficiently and produce much more satisfying search results than existing systems. It

Answers tens of millions of queries every day. This paper provides an in-depth description of large-scale web search engine. It makes heavy use of the structure present in hypertext. But it doesn't deal how to effectively deal with uncontrolled hypertext collections, where anyone can publish anything they want. The technical challenges involved with using the additional information present in hypertext to produce better search results.

Guo.Y, Li. K, and Zhang.K [3], Board Forum Crawling is a web crawling method for web forum. This method exploits the organized characteristics of the Web forum sites and simulates human behavior of visiting Web Forums. Board Forum Crawling can crawl most meaningful information of a Web forum site efficiently and simply. Experiments have shown BFC is an efficient and economical method and has been used in a real project. Limiting to the space, the details of the method, such as link clustering based on URL is the main demerit of this paper.

III. FOCUS–A SUPERVISED FORUM CRAWLER

A. System Overview

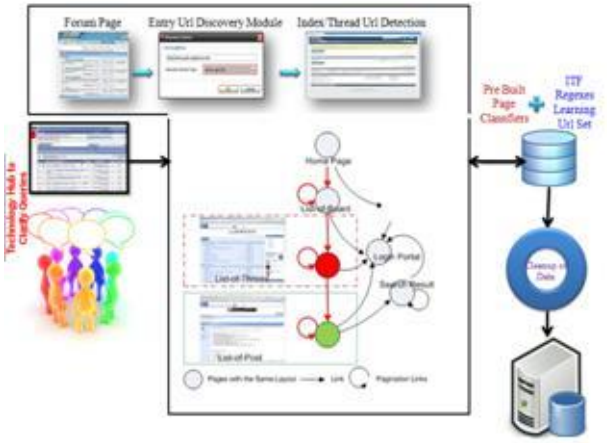

Fig. 1. The overall architecture of FoCUS

Fig. 1 shows the overall architecture of FoCUS. The user comes with a query, first point to the forum page. Given any page of a forum, FoCUS first finds its entry URL using the *Entry URL Discovery* module. Then, it uses the *Index/Thread URL Detection* module to detect index URLs and thread URLs on the entry page; the detected index URLs and thread URLs are saved to the URL training sets. It detects the keyword. Pre-Built Page classifier makes the record of already crawled data. The *ITF Regexes Learning* module compare the

new keywords with the keywords that stored in the database. If any mismatch occurs it means that the data is irrelevant and Focus clean that data and the remaining data stored on the system.

B. FoCUS Modules

1. Main Forum
This module, act as an integral portal of querying according to the basic doubts in the technology. Logged user can be able to post their queries by selecting the options of technology. A usual application with an authentication page with user creation can be done in the application.

2. Forum Thread
An individual thread will be created by the users based on the queries raised by the users. The threads will be segregated based on the technology of the project. Once the user wants to check an individual thread, open the website and move the appropriate technology and click the relative words to check it out.

3. Authentication
In this module, an authentication page is created which will enable the user to login into the system. The option of creating the registered user is provided to the system.

4. Forum crawling
The users were permitted to crawl the web pages automatically, once the user provides the necessary options of website details. An automatic recognition mechanism with an underlying Top down keyword based search algorithm is implemented to identify the exact URL's and the navigation of the web pages will happen automatically. The next page in the forum is moved and the thread in each individual category is scanned.

Top down key word based search algorithm

- Keyword search algorithm is an algorithm for finding an item with specified properties among a collection of items.
- The items may be stored individually as records in a database. A Keyword search looks for words anywhere in the record.
- The key words are searched top to down in the database.

5. Keyword integrate engine
Once the crawling engine has entered into the individual thread, the keywords were scanned through KMP Algorithm and the keywords were identified. The keywords were compared with the existing datasets. Once the keywords are matched. Immediately, the system will automatically move the web pages to the appropriate technology.

Knuth–Morris–Pratt algorithm

- Knuth–Morris–Pratt string searching algorithm (or KMP algorithm) searches for occurrences of a "word" W within a main "text string" S by employing the observation that when a mismatch occurs.
- The word itself embodies sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters.

```
              1         2
m: 012345678901234567890 12
S: ABC ABCDAB ABCDABCDABDE
W: ABCDABD
i: 0123456
```

- We proceed by comparing successive characters of W to "parallel" characters of S, moving from one to the next if they match. However, in the fourth step, we get S [3] is a space and W [3] = 'D', a mismatch.
- Rather than beginning to search again at S[1], we note that no 'A' occurs between positions 0 and 3 in S except at 0; hence, having checked all those characters previously.
- We know there is no chance of finding the beginning of a match if we check them again. Therefore we move on to the next character, setting m = 4 and i = 0.

6. Forum manual cleanup
Cleanup of data and moving the contents to the appropriate web pages is the major scope of the project. In this module, the unwanted data will be cleaned up and the forum data will be moved to the web pages according to the technology. The data will be cleaned up and the forums will be moved to the technology in turn, the display of the particular thread should be moved to the appropriate forum. In addition, the knowledge database will get accumulated with lots of knowledgeable information which will be used in future cases of getting more streamlined data processing.

IV. PROPOSED ALGORITHMS

Algorithm kmp_search: input:

Associate array of characters, S

Associate array of characters, W

output: associate whole number (the zero-based position in S at that W is found) outline variables: associate whole number, m ← zero (the starting of the present match in S) an integer, i ← zero (the position of the present character in W) associate array of integers, T (the table, computed elsewhere) whereas m+i is a smaller amount than the length of S, do:

if W[i] = S [m + i],

if i equals the (length of W)-1,

return m

let i ← i + one

otherwise, let m ← m + i - T[i],

if T[i] is bigger than -1,

let i ← T[i] else

let i ← 0

Assuming the previous existence of the table T, the search portion of the Knuth–Morris–Pratt formula has quality O(k), wherever k is that the length of S and also the O is big-O notation. As apart from the fastened overhead incurred in getting into and exiting the perform, all the computations area unit performed within the whereas loop, we'll calculate a certain on the quantity of iterations of this loop; so as to try to to this we tend to initial build a key observation concerning the character of T. By definition it's made in order that if a match that had begun at S[m] fails whereas scrutiny S[m + i] to W[i], then consequent attainable match should begin at S[m + (i - T[i])]. Particularly consequent attainable match should occur at the next index than m, in order that T[i] < i. Using this truth, we'll show that the loop will execute at the most 2k times. For in every iteration, it executes one amongst the 2 branches within the loop. the primary branch invariably will increase i and doesn't modification m, in order that the index m + i of the presently scrutinized character of S is increased .The second branch adds i - T[i] to m, and as we've got seen ,this is often continually a positive range. So the situation m of the start of the present potential match is increased. Now, the loop ends if m + i = k; so every branch of the loop are often reached at the most k times, since they severally increase either m + i or m, and m ≤ m + i: if m = k, then actually m + i ≥ k, in order that since it will increase by unit increments at the most, we tend to should have had m + i = k at some purpose within the past, and so either means we'd be done. Thus the loop executes at the most 2k times, showing that the time quality of the search formula is O (k).Here is otherwise to admit the runtime: allow us to say we start to match W and S at position i and p, if W exists as a substring of S at p, then W [0 through m] == S [p through p+m]. Upon success, that is, the word and also the text matched at the positions(W[i] == S[p+i]), we tend to increase i by one (i++).Upon failure, that is, the word and also the text doesn't match at the positions(W[i]!= S[p+i]), the text pointer is unbroken still, whereas the word pointer roll-back a particular amount(i = T[i],wherever T is that the jump table) and that we decide to match W[T[i]] with S[p+i]. the utmost range of roll-back of i is finite by i, that's to mention, for any failure, we are able to solely roll-back the maximum amount as we've got progressed up to the failure. Then it's clear the runtime is 2k.

A. Algorithm kmp_table:

Input: associate array of characters, W
Associate array of integers, T
Output: nothing outline variables:
An integer, pos ← 2
An integer, cnd ← 0
Let T [0] ← -1, T [1] ← 0
Whereas pos is a smaller amount than the length of W,
do: if W [pos - 1] = W[cnd],
let cnd ← cnd + one, T[pos] ← cnd, pos ← pos + one
otherwise, if cnd > 0,
let cnd ← T[cnd]

The quality of the table formula is O (n), wherever n is that the length of W. As apart from some data format all the work is completed within the whereas loop, it's enough to indicate that this loop executes inO(n) time, which can be done by at the same time examining the quantities pos and pos - cnd. Within the initial branch, pos - cnd is preserved, as each pos and cnd area unit incremented at the same time, however naturally, pos is increased. Within the second branch, cnd is replaced by T[cnd], that we tend to saw higher than is often strictly but cnd, so increasing pos - cnd. Within the third branch, pos is incremented and cnd isn't, therefore each pos and pos - cnd increase. Since pos ≥ pos - cnd, this suggests that at every stage either pos or a edge for pos increases; so since the formula terminates once pos = n, it should terminate once at the most 2n iterations of the loop, since pos - cnd begins at one. So the quality of the table formula is O (n)

IV. CONCLUSION AND FUTURE ENHANCEMENT

We proposed and implemented FoCUS, a supervised forum crawler. We reduced the forum crawling problem to a URL type recognition problem and showed how to leverage implicit navigation paths of forums, i.e. EIT path, and designed methods to learn ITF regexes explicitly.We also showed that FoCUS can effectively apply learnt forum crawling knowledge on 160 unseen forums to automatically collect index URL, thread URL, and page-flipping URL training sets and learn ITF regexes from the training sets. These learnt regexes can be applied directly in online crawling. Training and testing on the basis of the forum package makes our experiments manageable and our results applicable to many forum sites. And also the proposed method is more accuracy than existing method .Due to this better accuracy, we achieved very good time consumption. In future, we would like to discover new threads and refresh crawled threads in a timely manner. The initial results of applying a FoCUS-like crawler to other social media are very promising. We are planning to conduct more comprehensive experiments to further verify our approach and improve upon it.

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]Y.Guo, K.Li, K.Zhang, and G.Zhang.
Board Forum Crawling: a Web Crawling
Method for Web Forum.*Proc.2006*
IEEE/WIC/ACM Int'l Conf.Web Intelligence,
Pp.475-478, 2006.

[4]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.

[5] Wang.Y, Yang.J.-M, Lai.W, Cai.R, Zhang.L, and Ma.W.-Y.,'Exploring
Traversal Strategy for Web Forum Crawling'. *Proc. 31st Ann. Int'l ACM*
SIGIR Conf. Research and Development in Information Retrieval, pp. 459-
466, 2008.

[6]A. Dasgupta, R.Kumar, and A.Sasturkar.De-duping URLs via rewrite rules. *Proc.14thACM SIGKDD Int'l Conf.Knowledge Discovery. and Data Mining, pp.186 - 194,2008.*

[7]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