A Novel Method for Crawl Forum Threads

1. CH. V. N. Krishna Murthy, 2. V. Chandra Sekhara Rao
1, 2 Dept., of CSE, Sri Aditya Engineering College, Surampalem, EG.dist, AP, India.

Abstract: In internet crawler it is program that visits websites and reads their pages and other information in order to create entries for a search engine index. Now we introducing forum crawler which traverses the web forums. But present crawlers are not providing relevant content as well as URL type reorganization problem to forum users. In order to overcome this problem we present Forum Crawler under Supervision which crawl relevant forum content from the web with less overhead.

Index Terms: URL, forum crawling, URL patterns

1. Introduction

Internet forums (also called web forums) are important services where users can request and exchange information with others. For example, the Trip Advisor Travel Board is a place where people can ask and share travel tips. Due to the richness of information in forums, researchers are increasingly interested in mining knowledge from them. Zhai and Liu Yang et al. and Song et al. extracted structured data from forums. Gao et al. identified question and answer pairs in forum threads. Zhang et al. proposed methods to extract and rank product features for opinion mining from forum posts. Glance et al. tried to mine business intelligence from forum data. Zhang et al. proposed algorithms to extract expertise network in forums.

In this paper, we present FoCUS (Forum Crawler under Supervision), a supervised web-scale forum crawler, to address these challenges. The goal of FoCUS is to crawl relevant content, i.e. user posts, from forums with minimal overhead. Forums exist in many different layouts or styles and powered by a variety of forum software packages, but they always have implicit navigation paths to lead users from entry pages to thread pages.

2. Related Work

Guo et al. did not mention how to discover and traverse URLs. Li et al. developed some heuristic rules to discover URLs. However, their rules are too specific and can only be applied to specific forums powered by the particular software package in which the heuristics were conceived. Unfortunately, according to Forum Matrix, there is hundreds of different forum software packages used on the Internet. A recent and more comprehensive work on forum crawling is iRobot by Cai et al. [13]. IRobot aims to automatically learn a forum crawler with minimum human intervention by sampling pages, clustering them, selecting informative clusters via an informativeness measure, and finding a traversal path by a spanning tree algorithm. But iRobot would only take the first path (entry! board! thread). iRobot learns URL location information to discover new URLs in crawling, but a URL location might become invalid when the page structure changes. As opposed to iRobot, we explicitly define entry-index-thread paths and leverage page layouts to identify index pages and thread pages. FoCUS also
learns URL patterns instead of URL locations to discover new URLs. Thus, it does not need to classify new pages in crawling and would not be affected by a change in page structures. The respective results from iRobot and FoCUS demonstrated that the EIT paths and URL patterns are more robust than the traversal path and URL location feature in iRobot. Another related work is near-duplicate detection. Forum crawling also needs to remove duplicates. But content based duplicate detection is not bandwidth efficient, because it can only be carried out when pages have been downloaded. URL-based duplicate detection is not helpful. It tries to mine rules of different URLs with similar text.

To alleviate unnecessary crawling, industry standards such as Robots Exclusion Standard, and Sitemap Protocol have been introduced. By specifying the “rel” attribute with the value, page authors can inform a crawler that the destination content is not endorsed. However, it is intended to reduce the effectiveness of search engine spams, but not meant for blocking access to pages. It is designed to specify what pages a crawler is allowed to visit or not. Sitemap is an XML file that lists URLs along with additional metadata including update time, change frequency etc. Generally speaking, the purpose of robots.txt and Sitemap is to enable the site to be crawled intelligently. So they may be useful to forum crawling. However, it is difficult to maintain such files for forums as their content continually changes.

2. Literature Survey

Besides duplicate links and uninformative pages, a long forum board or thread is usually divided into multiple pages which are linked by page-flipping links. These relationships should be preserved while crawling to facilitate downstream tasks such as page wrapping and content indexing. We propose a new system for web crawl as FoCUS: Learning to Crawl Web Forums. It is a system overcome by existing crawl systems. In this method for learning regular expression patterns of URLs that lead a crawler from an entry page to target pages. Target pages were found through comparing DOM trees of pages with a pre-selected sample target page. It is very effective but it only works for the specific site from which the sample page is drawn. The same process has to be repeated every time for a new site. Therefore, it is not suitable to large-scale crawling. In contrast, FoOCUS learns URL patterns across multiple sites and automatically finds forum entry page given a page from a forum. Experimental results show that FoCUS is effective in large scale forum crawling by leveraging crawling knowledge learned from a few annotated forum sites. A recent and more comprehensive work on forum crawling is iRobot. iRobot aims to automatically learn a forum crawler with minimum human intervention by sampling forum pages, clustering them, selecting informative clusters via an in formativeness measure, and finding a traversal path by a spanning tree algorithm. However, the traversal path selection procedure requires human inspection.

To download forum data effectively and efficiently, we should first understand the characteristics of most forum sites. In general, content of a forum is stored in a database. When a Web forum service receives a user request, it dynamically generates a response page based on some pre-defined templates. The whole forum site is usually connected as a very complex graph by many links among various pages. Due to these reasons, forum sites generally have the following common characteristics. First, duplicate pages (or content) with different Uniform Resource Locators (URLs) will be generated by the
service for different requests such as "view by date" or "view by title." Second, a long post divided into multiple pages usually results in a very deep navigation. Sometimes a user has to do tens of navigations if he/she wants to read the whole thread, and so does a crawler. Finally, due to privacy issue, some content such as user profiles is only available for registered users. Generic Web crawlers, which adopt the breadth-first strategy are usually inefficient in crawling Web forums. A Web crawler must make a tradeoff between "performance and cost" to balance the content quality and the costs of bandwidth and storage.

3. Existing System

The existing system is a manual or semi automated system, i.e. The Textile Management System is the system that can directly sent to the shop and will purchase clothes whatever you wanted.

The users are purchase dresses for festivals or by their need. They can spend time to purchase this by their choice like color, size, and designs, rate and so on.

They But now in the world everyone is busy. They don’t need time to spend for this. Because they can spend whole the day to purchase for their whole family. So we proposed the new system for web crawling.

Disadvantages

1. Consuming large amount of data’s.

2. Time wasting while crawl in the web.

4. Proposed System

We propose a new system for web crawl as FoCUS: Learning to Crawl Web Forums. It is a system overcome by existing crawl systems. In this method for learning regular expression patterns of URLs that lead a crawler from an entry page to target pages. Target pages were found through comparing DOM trees of pages with a pre-selected sample target page. It is very effective but it only works for the specific site from which the sample page is drawn. The same process has to be repeated every time for a new site. Therefore, it is not suitable to large-scale crawling. In contrast, FoCUS learns URL patterns across multiple sites and automatically finds forum entry page given a page from a forum. Experimental results show that FoCUS is effective in large scale forum crawling by leveraging crawling knowledge learned from a few annotated forum sites. A recent and more comprehensive work on forum crawling is iRobot. iRobot aims to automatically learn a forum crawler with minimum human intervention by sampling forum pages, clustering them, selecting informative clusters via an in formativeness measure, and finding a traversal path by a spanning tree algorithm. However, the traversal path selection procedure requires human inspection.

ADVANTAGES OF PROPOSED SYSTEM

We show how to automatically learn regular expression patterns (ITF regexes) that recognize the index URL, thread URL, and page-flipping URL using the page classifiers built from as few as five annotated forums.

5. System Architecture
ALGORITHM: Index URL and Thread URL Detection

Input: sp: an entry or index page
Output: it_group: a group of index/thread URLs

1. let it_group be p: data
2. url groups = Collect URL groups by aligning HTML DOM tree of sp;
3. foreach ug in url_groups do
4. uanchor_len = Total anchor text length in ug;
5. end foreach
6. if group = argmax(uanchor_len) in url_groups;
7. if_group.DstPageType = Majority page type of the destination pages of URLs in ug;
8. if_group.DstPageType is INDEX_PAGE
9. if_group.Urltype = INDEX_URL;
10. else if if_group.DstPageType is THREAD_PAGE
11. if_group.Urltype = Thread_URL;
12. else
13. if group =
14. end if
15. return if_group;

6. Modules

1. Signup & Login:

In this module, we have two sub-modules. They are,

- **User signup & login:** In this module user can create account with our site by filling details. And then they can login with our site using this user name and password

- **Admin login:** The owner of this system have an own user name and password for login with the page.

2. Upload File:

In this module the owner of the site have to upload a new file for crawl in this site. The user of the page wants to crawl in the site. So the admin should upload a maximum of files for the users need.

Also the admin can view the user details those are having account in his page. And they can view files which they are already uploaded in database.

3. Crawl in Web:

The goal of this paper is crawl on the web. So the user can view files in this site which they are uploaded by admin. The users can search a files what they need to know about that.

Also they can view the related searches based on their search. The search contains additional links of its contents also. This web crawling proposed like tree search.

And then user can view their own details which they already gave while signup with this site. They also can change / modify the details.

4. Similarity Measure
The algorithm mainly use a variation vector space model TF-IDF weighting scheme.

The slightly varied weight computation is realized as follows:

Lastly it uses, Jaccard similarity measure formula to compute the similarity between documents.

$$w_{ij} = \left( 0.5 + \frac{0.5 \times f_{ij}}{\max\{f_{ij}, f_{2j}, \ldots, f_{|V_j|}\}} \right) \times \log \frac{N + 1}{d_i}$$

**SimRank Algorithm:**

Use an improved PageRank algorithm in which two distinct weight values are assigned to the title and body of a page, respectively

- It has the ability of eliminating useless documents such as advertising web pages that resulting in both disk and time saving.

**ALGORITHM EVALUATION STEPS:**

- Homogeneous: consisting of 1 type of objects

- Similarity btw. \(a\&b\) denoted by:
  - if \(a = b\), \(s(a, b) = 1\), \(s(a, a) = s(b, b) = 1\)
  - otherwise:
    $$s(a, b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$
  - \(C\) is called as “confidence level” or “decay factor”, a constant btw. 0 & 1

  - if \(|I(a)|\) or \(|I(b)|\) is 0, \(s(a, b) = 0\)

  - symmetric : \(s(a, b) = s(b, a)\)

- Similarity btw. \(a\&b\) is the average similarity btw. in-neighbors of \(a\) and in-neighbors of \(b\)

7. Conclusion

In this paper, 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. entry-index-thread (EIT) path, and designed methods to learn ITF regexes explicitly. Experimental results on 160 forum sites each powered by a different forum software package confirm that FoCUS could effectively learn knowledge of EIT path and ITF regexes from as few as 5 annotated forums. We also showed that FoCUS can effectively apply learned forum crawling knowledge on 160 unseen forums to automatically collect index URL, thread URL, and page-flipping URL string training sets and learn the ITF regexes from the training sets. These learned regexes could be applied directly in online crawling. Training and testing on the basis of forum package makes our experiments manageable and our results applicable to many forum sites. Moreover, FoCUS can start from any page of a forum, while all previous works expect an entry page is given. Our test results on 9 unseen forums show that FoCUS is indeed very effective and efficient and outperforms the state-of-the-art forum crawler, iRobot. The results on 160 forums show that FoCUS can apply the learned
knowledge to a large set of unseen forums and still achieve a very good performance. Though, the method introduced in this paper is targeted at forum crawling, the implicit EIT-like path also apply to other sites, such as community Q&A sites, blog sites, and so on.

In the future, we would like to handle forums which use JavaScript, include incremental crawling, and discover new threads and refresh crawled threads in a timely manner. The initial results of applying FoCUS-like crawler to other social media are very promising. We would like to conduct more comprehensive experiments to further verify our approach and improve upon it.

References


Mr. CH. V. N. Krishna Murthy is a student of Sri Aditya Engineering College, Surampalem. Presently he is pursuing M.Tech [CSE] from this college and he received his B.Tech degree in Computer Science & Engineering from Chaitanya Institute of Science & Technology, affiliated to JNTUK Kakinada in the year 2011. His area of interest includes Object Oriented languages, Web Technologies and current trends in Computer Science.

Mr. V. Chandra Sekhara Rao obtained his B.Tech degree in Computer Science & Engineering from GMR Institute of Technology, Rajam, affiliated to Jawaharlal Nehru Technological University Kakinada and M.Tech degree in Computer Science & Engineering from Jawaharlal Nehru Technological University Kakinada, India. He is working as a Assistant Professor of Computer Science & Engineering at Sri Aditya Engineering College, Surampalem, India. His area of interests includes Network Security, Cloud Computing and Data mining.