RESEARCH ARTICLE OPEN ACCESS

Forum Crawler Under Supervision

Mrs.C. Navamani MCA., M.Phil., M.E., K.Bhuvaneswari,
Assistant Professor[1], Final MCA[2],
Department of Computer Applications,
Nandha Engineering College,
Erode-52.

--**************************---

Abstract:

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. Forum threads contain information content that is the target of forum crawlers. Although forums have different layouts or styles and are powered by different forum software packages, they always have similar implicit navigation paths connected by specific URL types to lead users from entry pages to thread pages. Based on this observation, we reduce the web forum crawling problem to a URL-type recognition problem. And we show how to learn accurate and effective regular expression patterns of implicit navigation paths from automatically created training sets using aggregated results from weak page type classifiers. Robust page type classifiers can be trained from as few as five annotated forums and applied to a large set of unseen forums. Our test results show that FoCUS achieved over 99 percent effectiveness and 97 percent coverage on a large set of test forums powered by over 150 different forum software packages. In addition, the results of applying FoCUS on more than 100 community Question and Answer sites and Blog sites demonstrated that the concept of implicit navigation path could apply to other social media sites.

Keywords— **Forum crawling, ITF regex, page classification, page type, URL pattern learning**

--**************************---

1 INTRODUCTION

INTERNET forums [4] (also called web forums) are im- portant services where users can request and exchange information with others. For example, the TripAdvisor 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 knowl- edge from them. Zhai and Liu [28], Yang et al. [27], and Song et al. [23] extracted structured data from forums. Gao et al. [15] identified question and answer pairs in forum threads. Zhang et al. [30] proposed methods to extract and rank product features for opinion mining from forum posts. Glance et al. [16] tried to mine business intelligence from forum data. Zhang et al. [29] proposed algorithms to extract expertise network in forums. To harvest knowledge from forums, their content must be downloaded first. However, forum crawling is not a trivial problem. Generic crawlers [12], which adopt a breadth-first traversal strategy, are usually ineffective and inefficient for forum crawling. This is mainly due to two noncrawler- friendly characteristics of forums [13], [26]: 1) duplicate links and uninformative pages and 2) page-flipping links. A forum typically has many duplicate links that point to a common page but with different URLs [7], e.g., shortcut links pointing to the latest posts or URLs for user experience functions such as “view by date” or “view by title.” A generic crawler that blindly follows these links will crawl many duplicate

pages, making it inefficient. A forum also has many uninformative pages such as login control to protect user privacy or forum software specific FAQs. Following these links, a crawler will crawl many unin- formative pages. Though there are standard-based methods such as specifying the "rel" attribute with the "nofollow" value (i.e., "rel ¼ nofollow") [6], Robots Exclusion Standard (robots.txt) [10], and Sitemap [9] [22] for forum operators to instruct web crawlers on how to crawl a site effectively, we found that over a set of nine test forums more than 47 percent of the pages crawled by a breadth-first crawler following these protocols were duplicates or uninformative. This number is a little higher than the 40 percent that Cai et al. [13] reported but both show the inefficiency of generic crawlers. More information about this testing can be found in Section 5.2.1. 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, for example, see Figs. 2, 3b, and 3c. Generic crawlers process each page individually and ignore the relationships between such pages. These relationships should be preserved while crawling to facilitate downstream tasks such as page wrapping and content indexing [27]. For example, multiple pages belonging to a thread should be concatenated together in order to extract all the posts in the thread as well as the reply-relationships between posts. In addition to the above two challenges, there is also a problem of entry URL discovery. The entry URL of a forum points to its homepage, which is the lowest common ancestor page of all its threads. Our experiment "Evaluation of Starting from Non-Entry URLs" in Section 5.2.1 shows that a crawler starting from an entry URL can achieve a much higher performance than starting from no entry URLs. Previous works by Vidal et al. [25] and Cai et al. [13] assumed that an entry URL is given. But entry URL
[4] IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 6, JUNE 2013 1293
[5] . J. Jiang is with the Information Processing Center, University of Science and Technology of China, Microsoft Building 2, No. 5 Danling Street, Haidian District, Beijing, P.R. China. E-mail: silyt@mail.ustc.edu.cn. . X. Song is with the Machine Intelligence and Translation Laboratory, Harbin Institute of Technology, Harbin 150001, P.R. China, and Microsoft Corporation, One Microsoft Way, Redmond, WA 98052. E-mail: xysong@mtlab.hit.edu.cn. . N. Yu is with the Information Processing Center, University of Science and Technology of China, Huangshan Road, Hefei, Anhui Province, P.R. China 230026. E-mail: ynh@ustc.edu.cn. . C.-Y. Lin is with the Microsoft Research Asia, Microsoft Building 2, No. 5 Danling Street, Haidian Distrct, Beijing, P.R. China 100190. E-mail: cyl@microsoft.com. Manuscript received 28 Sept. 2011; revised 12 Jan. 2012; accepted 22 Feb. 2012; published online 5 Mar. 2012. Recommended for acceptance by R. Kumar. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-09-0585. Digital Object Identifier no. 10.1109/TKDE.2012.56.1041-4347/13/$31.00 2013 IEEE Published by the IEEE Computer Society discovery is not a trivial problem. An entry URL is not necessarily at the root URL level of a forum hosting site and its form varies from site to site. Without an entry URL, existing crawling methods such as Vidal et al. [25] and Cai et al. [13] are less effective. In this paper, we present Forum Crawler Under Super- vision (FoCUS), 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 are powered by a variety of forum software packages, but they always have implicit navigation paths to lead users from entry pages to thread pages. Fig. 1 illustrates a typical page and link structure in a forum. For example, a user can navigate from the entry page to a thread page through the following paths: 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 ! thread6. entry ! list-of-board ! list-of-board & thread ! board ! thread We call pages between the entry page and thread page which are on a breadth-first navigation path the index pages. We represent these implicit paths as the following naviga- tion path (entry-index-thread (EIT) path): entry page!index page!thread page Links between an entry page and an index page or between two index pages are referred as index URLs. Links between an index page and a thread page are referred as thread URLs. Links connecting multiple pages of a board and multiple pages of a thread are referred as page-flipping URLs. A crawler starting from the entry URL only needs to follow index URL, thread URL, and page-flipping URL to traverse EIT paths that lead to all thread pages. The challenge of forum crawling is then reduced to a URL type recognition problem. In this paper, we show how to learn URL patterns, i.e., Index-Thread-page-Flipping (ITF) regexes, recognizing these three types of URLs from as few as five annotated forum packages and apply them to a large set of 160 unseen forums packages. Note that we specifically refer to "forum package" rather than "forum site." A forum package such as vBulletin1 can be deployed by many forum sites. The major contributions of this paper are as follows:

[6] 1. We reduce the forum crawling problem to a URL type recognition problem and implement a crawler, FoCUS, to demonstrate its applicability. 2. 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. 3. We evaluate FoCUS on a large set of 160 unseen forum packages that cover 668,683 forum sites. To the best of our knowledge, this is the largest evaluation of this type. In addition, we show that the learned patterns are effective and the resulting crawler is efficient. 4. We compare FoCUS with a baseline generic breadth- first crawler, a structure-driven crawler, and a state- of-the-art crawler iRobot and show that FoCUS outperforms these crawlers in terms of effectiveness and coverage. 5. We design an effective forum entry URL discovery method. To ensure high coverage, we show that a forum crawler should start crawling forum pages from forum entry URLs. Our evaluation shows that a naı ̈ve entry link discovery baseline can achieve only 76 percent recall and precision; while our method can achieve over 99 percent recall and precision. 6. We show that, though the proposed approach is targeted at forum crawling, the implicit EIT-like path also apply to other User Generated Content (UGC) sites, such as community Q&A sites and blog sites. The rest of this paper is organized as follows. Section 2 is a brief review of related work. In Section 3, we define terms used in this paper. We give our observations on forums and describe the detail of the proposed approach in Section 4. In Section 5, we report the results of our experiments. In the last section, we draw conclusions and point out future directions of research.

2 RELATED WORK

Target pages were found through comparing DOM trees of pages with a preselected 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 for large-scale crawling. In contrast, FoCUS learns URL patterns across multiple sites and automatically finds a forum's entry page given a page from the forum. Experimental results show that FoCUS is effective at large-scale forum crawling by leveraging crawl- ing knowledge learned from a few annotated forum sites. Guo et al. [17] and Li et al. [20] are similar to our work. However, Guo et al. did not mention how to discover and traverse URLs. Li et al. developed some heuristic rules to 1294 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 6, JUNE 2013

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 ForumMatrix [2], there is hun dreds of different forum software packages used on the Internet. Please refer to [2], for more information about forum software packages. In addition, many forums use their own customized software. 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. However, the traversal path selection procedure requires human inspection. Follow up work by Wang et al. [26] proposed an algorithm to address the traversal path selection problem. They introduced the concept of skeleton link and page-flipping link. Skeleton links are "the most important links supporting the structure of a forum site." Importance is determined by informativeness and coverage metrics. Page-flipping links are determined using connec- tivity metric. By identifying and only following skeleton links and page-flipping links, they showed that iRobot can achieve effectiveness and coverage. According to our evaluation, its sampling strategy and informativeness estimation is not robust and its tree-like traversal path does not allow more than one path from a starting page node to a same ending page node. For example, as shown in Fig. 1, there are six paths from entry to threads. 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 [18], [21] is not bandwidth- efficient, because it can only be carried out when pages have been downloaded. URL-based duplicate detection [14], [19] is not helpful. It tries to mine rules of different JIANG ET AL.: FOCUS: LEARNING TO CRAWL WEB FORUMS 1295 URLs with similar text. However, such methods still need to analyze logs from sites or results of a previous crawl. In forums, index URLs, thread URLs, and page-flipping URLs have specific URL patterns. Thus, in this paper, by learning patterns of index URLs, thread URLs, and page-flipping URLs and adopting a simple URL string de-duplication technique (e.g., a string hashset), FoCUS can avoid duplicates without duplicate detection. To alleviate unnecessary crawling, industry standards such as "nofollow" [6], Robots Exclusion Standard (ro- bots.txt) [10], and Sitemap Protocol [9], [22] have been introduced. By specifying the "rel" attribute with the "nofollow" value (i.e., "rel ¼ nofollow"), page authors can inform a crawler that the destination content is not endorsed. However, it is intended to reduce the effective- ness of search engine spams, but not meant for blocking access to pages. A proper way is robots.txt [10]. It is designed to specify what pages a crawler is allowed to visit or not. Sitemap [9] is an XML file that lists URLs along with additional metadata including update time, change fre- quency 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. In our experiment in Section 5.2.1, more than 47 percent of the pages crawled by a generic crawler which can properly understand these industry standards are uninformative or duplicates.

3 TEMINOLOGY

To facilitate presentation in the following sections, we first define some terms used in this paper.

PageType:We classified forum pages into page types.

Entry Page: The homepage of a forum, which contains a list of boards and is also the lowest common ancestor of all threads

Index URL: A URL that is on an entry page or index page and points to an index page.

Thread URL: A URL that is on an index page and points to a thread page. Its anchor text is the title of its destination thread. Figs. 3b and 3c show an example.

Page-flipping URL: A URL that leads users to another page of the same board or the same thread. Correctly dealing with page-flipping URLs enables a crawler to download all threads in a large board or all posts in a long thread. See Figs. 2, 3b, and 3c for examples.

Other URL: A URL that is not an index URL, thread URL, or page-flipping URL.

EIT Path: An entry-index-thread path is a navigation path from an entry page through a sequence of index pages (via index URLs and index page-flipping URLs) to thread pages (via thread URLs and thread page-flipping URLs).

ITF Regex: An index-thread-page-flipping regex is a regular expression that can be used to recognize index, thread, or page-flipping URLs. ITF regex is what FoCUS aims to learn and applies directly in online crawling. The learned ITF regexes are site specific, and there are four ITF regexes in a site: one for recognizing index URLs, one for thread URLs, one for index page-flipping URLs, and one for thread page-flipping URLs. Fig. 9 gives an example. A perfect crawler starts from a forum entry URL and only follows URLs that match ITF regexes to crawl all forum threads. The paths that it traverses are EIT paths.

4 FoCUS

A SUPERVISED FORUM CRAWLER In this section, we first give our observations and an overview. The remaining sections go into greater depth for each module.

4.1 Observations In order to crawl forum threads effectively and efficiently, we investigated about 40 forums (not used in testing) and found the following characteristics in almost all of them.

Navigation path is Despite differences in layout and style, forums always have implicit navigation paths leading users from their entry pages to thread pages. In general crawling, Vidal et al. [25] learned “navigation patterns” leading to target pages (thread pages in our case). iRobot also adopted a similar idea but applied page sampling and clustering techniques to find target pages (Cai et al. [13]). It used in formativeness and coverage metrics to find traver-sal paths (Wang et al. [26]). We explicitly defined the EIT path that specifies what types of links and pages that a crawler should follow to reach thread pages. 2. URL layout. URL layout information such as the location of a URL on a page and its anchor text length is an important indicator of its function. URLs of the same function usually appear at the same location. For example, in Fig. 3a, index URLs appear in the left rectangles. In addition, index URLs and thread URLs usually have longer anchor texts that provide board or thread titles (see Figs. 3a and 3b for an example). 3. Page layout. Index pages from different forums share a similar layout. The same applies to thread pages. For example, in Fig. 2, the index pages from two different forums have the similar page layout. However, an index page usually has a very different page layout from a thread page. As shown in Fig. 2, an index page tends to have many narrow records giving information of boards or threads; a thread page typically has a few large records that contain forum posts FoCUS learns page type classifiers directly from a set of annotated pages based on this characteristic. This is the only step where manual annotation is required for FoCUS. Inspired by these observations, we developed FoCUS. The main idea behind FoCUS is that index URL, thread URL, and page-flipping URL can be detected based on their layout characteristics and destination pages;v and forum pages can be classified by their layouts. This knowledge about URLs and pages and forum structures can be learned from a few annotated forums and then applied to unseen forums. Our experimental results in Section 5 confirm the effectiveness of our approach.

4.2 System Overview

It consists of two major parts: the learning part and the online crawling part. The learning part first learns ITF regexes of a given forum from automatically constructed URL training exam- ples. The online crawling part then applies learned ITF regexes to crawl all threads efficiently. 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. Next, the destination pages of the detected index URLs are fed into this module again to detect more index and thread URLs until no more index URL is detected. After that, the Page-Flipping URL Detection module tries to find page- flipping URLs from both index pages and thread pages and saves them to the training sets. Finally, the ITF Regexes Learning module learns a set of ITF regexes from the URL training sets.

Once the learning is finished, FoCUS performs online crawling as follows: starting from the entry URL, FoCUS follows all URLs matched with any learned ITF regex. FoCUS continues to crawl until no page could be retrieved or other condition is satisfied.

4.3 ITF Regexes

Learning To learn ITF regexes, FoCUS adopts a two-step supervised training procedure. The first step is training sets construc- tion. The second step is regexes learning.

4.3.1 **Constructing URL Training Sets**

The goal of URL training sets construction is to automatically create sets of highly precise index URL, thread URL, and page-flipping URL strings for ITF regexes learning. We use a similar procedure to construct index URL and thread URL training sets since they have very similar properties except for the types of their destination pages; we present this part first. Page-flipping URLs have their own specific properties that are different from index URLs and thread URLs; we present this part later. 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. As we mentioned in Section 4.1, 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 (user posts). 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. In addition, each record in an index page or a thread page usually has a link pointing to a user profile page (see Figs. 2 and 3 for example). Inspired by such characteristic, we propose features based on page layouts and build page classifiers using Support Vector Machine (SVM) [24] to decide page type. All the features are extracted based on the page layout, outgoing links, and metadata and DOM tree structures of the records. Page content is not used in the features as FoCUS does not care about the page content in crawling. The most features are shown in Table 1. The feature “Record Text Similarity” just computes the similarity of texts among all records, but does not care what the text was saying. Note that the feature “Number of Groups” means the number of aligned groups after HTML DOM tree alignment which will be explained later. SVMlight Version 6.022 with a default linear kernel setting is used. One index page classifier and one thread page classifier are built using the same feature set. FoCUS does not need strong page type classifiers which we will explain later.

In our experiment, we tested over 160 forums (10 pages each of index, thread, and other page), with each powered by a different software package. Our classifiers achieved 96 percent recall and 97 percent precision for index pages and 97 percent recall and 98 percent precision for thread pages with different amounts of training data (see Table 2). As we discussed earlier, index pages and thread pages have typical and representative layout structures, our classifiers achieved high performance with a few training forums. After showing how to detect index and thread pages, we next describe how to find index and thread URLs. Recall again from the definition of index (or thread) URL that its anchor text is the board (or thread) title of its destination page. As illustrated in Figs. 3a and 3b, index (or thread) URLs usually have relatively long anchor text, are grouped according to their functions and placed in the same "table" column position. Leveraging such structured layout infor- mation, we group URLs based on their locations and treat URLs as a group instead of as individual URLs. We then assume that the URL group with the longest total anchor text is a candidate group of index (or thread) URLs. This simple group anchor-text-length-based discriminative method does not need a length threshold to decide the type of a URL and can take care of index (or thread) URLs with short anchor text. According to [23], [28], URLs that appear in an HTML table-like structure can be extracted by aligning DOM trees and stored in a link-table. This technique has been well studied in recent years, so we will not discuss it here. In this paper, we adopted the partial tree alignment method in [23]. Fig. 5 shows an example of the table-like structure of an index page and its corresponding link-table. Fig. 5a is a screenshot of an index page, in which each row contains a thread title, the most recent poster, a shortcut to the last post, number of posts, and number of views. After aligning the DOM tree in Fig. 5a, a link-table is generated as shown in Fig. 5b. We can see that, thread titles, thread starters, last update times, the most recent posters, numbers of posts, and numbers of views are all aligned into the corresponding columns in the table. In Fig. 5, thread titles with their URLs are aligned to group 1, the most recent posters with their URLs are aligned to group 4, and the shortcuts are aligned to group 5. Based on our assumption of index or thread URL groups, we will select group 1 as the candidate for the index or thread URL group because it has the longest anchor text. Note that we cannot determine type of the candidate group so far. We need to check the destination page type of the URLs in the candidate group. A majority voting method is adopted to determine the URL type since classification results on individual destination page might be erroneous. By utilizing aggregated classification results, FoCUS does not need very strong page classifiers. In summary, we create index and thread URL training sets using the algorithm shown in Fig. 6. Lines 2-5 collects all URL groups and computes their total anchor text length; line 6 selects the URL group with longest anchor text length as the index/thread URL group; and lines 7-14 determines its URL type. If the pages are not index or thread page, the URL group is discarded. Page-flipping URL training set. Page-flipping URLs point to index pages or thread pages but they are very different from index URLs or thread URLs. Wang et al. [26] proposed "connectivity" metric to distinguish page-flipping URLs from other loop-back URLs. However, the metric only works well on the "grouped" page-flipping URLs. The one page-flipping URL in one page. For example, URLs in the rounded rectangles in the top-right corner of Fig. 2 are grouped page-flipping URLs. But in many forums, there is only one page-flipping URL in one page, which we called single page-flipping URL. See Fig. 7 for example (this case is more common in blog sites). Such URLs cannot be detected using the "connectivity" metric. To address this short-coming, we observed some special properties of page- flipping URLs and proposed an algorithm to detect page- flipping URLs based on these properties. In particular, the grouped page-flipping URLs have the following properties:

Their anchor text is either a sequence of digits such as 1, 2, 3, or special text such as "last." 2. They appear at the same location on the DOM tree of their source page and the DOM trees of their destination pages. 3. Their destination pages have similar layout with their source pages. We use tree similarity to determine whether the layouts of two pages are similar or not. As to single page-flipping URLs, they do not have the property 1, but they have another special property. 4. The single page-flipping URLs appearing in their source pages and their destination pages have the same anchor text but different URL strings.

URL type to page-flipping URL. In our experiment over 160 forum sites (10 pages each of index and thread page), our method achieved 95 percent recall and 99 percent precision. We apply this method to both index pages and thread pages; the found page-flipping URLs are saved as training examples.

4.3.2 Learning ITF Regexes We have shown how to create index URL, thread URL, and page-

flipping URL string training sets; next we explain how to learn ITF regexes from these training sets. Vidal et al. [25] applied URL string generalization, but we do not use their method because it is too strict and easily affected by negative URL examples. It requires very clean, precise URL examples. However, FoCUS cannot guarantee this since its training sets are all created automatically. For example, given URLs as follows (the top four URLs are positive while the bottom two URLs are negative):
http://www.gardenstew.com/about20152.html
http://www.gardenstew.com/about18382.html
http://www.gardenstew.com/about19741.html
http://www.gardenstew.com/about20142.html
http://www.gardenstew.com/user-34.html
http://www.gardenstew.com/post-180803.html It We briefly describe their technique below. For more details, please refer to their paper. Starting with the generic pattern "*," their algorithm finds more specific patterns matching a set of URLs. Then each specific pattern is further refined into more specific patterns. Patterns are refined recursively until no more patterns can be refined. Given the previous example, "*" is

1.http://www.gardenstew.com/about\d+.html.
2.http://www.gardenstew.com/user-\d+.html.
3.http://www.gardenstew.com/post-\d+.html. Each pattern matches a subset of URLs. These patterns are refined recursively until no more specific patterns can be generated. These three patterns are the final output as they cannot be refined further. We made one modification of the technique: a refined pattern is retained only if the number of its matching URLs is greater than an empirically determined threshold. This is designed to reduce patterns with low coverage since we expect a correct pattern should cover many URLs. The threshold is set to 0.2 times the total count of URLs. It was determined based on five training forums. For the above example, only the first pattern is retained. In practice, this method might include session ID [8] in learned URL patterns. We issue multiple requests of a forum URL over a span of time to detect any embedded session IDs and exclude them from URL matching. At last, FoCUS learned a set of ITF regexes for a given forum. Each ITF regex contains three elements: page type of destination pages, URL type, and the URL pattern. Fig. 9 shows the learned ITF regexes from forum http:// www.gardenstew.com/.

4.4 Online Crawling Given a forum, FoCUS first learns a set of ITF regexes following the procedure scribed in previous sections. Then it performs online crawling using a breadth-first strategy

What makes FoCUS efficient in online crawling is that it only needs to apply the learned ITF regexes on new outgoing URLs in newly downloaded pages. FoCUS does not need to group outgoing URLs, classify pages, detect page-flipping URLs, or learn regexes again for that forum. Such time consuming operations are only performed during the learning phase.

4.5 Entry URL Discovery In previous sections, we explained how FoCUS learns ITF regexes that can be used in online crawling. However, an entry URL needs to be specified to start the crawling process. To the best of our knowledge, all previous methods assumed that a forum entry URL is given. In practice, especially in web-scale crawling, manual forum entry URL annotation is not practical. Forum entry URL discovery is not a trivial task since entry URLs vary from forums to forums. To demonstrate this, we developed a heuristic rule to find entry URL as a baseline. The heuristic baseline tries to find the following keywords ending with "/" in a URL: forum, board, community, bbs, and discus. If a keyword is found, the path from the URL host to this keyword is extracted as its entry URL; if not, the URL host is extracted as its entry URL. Our experiment shows that this naı ̈ve baseline method can achieve about 76 percent recall and precision. To make FoCUS more practical and scalable, we design a simple yet effective forum entry URL discovery method based on some techniques introduced in previous sections.

5 EXPERIMENTS

To carry out meaningful evaluations that are good indicators of web-scale forum crawling, we selected 200 different forum software packages from ForumMatrix [2], Hot Script [3], and Big-Boards [5]. For each software package, we found a forum powered by it. In total, we have 200 forums powered by 200 different software packages. Among them, we selected 40 forums as our training set and leave the remaining 160 for testing. These 200 packages cover a large number of forums. The 40 training packages are deployed by 59,432 forums and the 160 test packages are deployed by 668,683 forums. To the best of our knowledge, this is the most comprehensive investigation of forum crawling in terms of forum site coverage to date. In addition, we wrote scripts to find out how many threads and users are in these forums. In total, we estimated that these packages cover about 2.7 billion threads generated by over 986 million users. It should be noted that, on all forums, the top 10 most frequent packages are deployed by 17 percent of all forums and cover about 9 percent of all threads.

To further check how many annotated pages FoCUS needs to achieve good performance. We conducted similar experiments but with more training forums (10, 20, 30, and 40) and applied cross validation. The results are shown in Table 2. We find that our page classifiers achieved over 96 percent recall and precision at all cases with tight standard deviation. It is particularly encoura- ging to see that FoCUS can achieve over 98 percent precision and recall in index/thread URL detection with only as few as five annotated forums.

5.1.2 Evaluation of Page-Flipping URL Detection To test page-flipping URL detection, we applied the module described "Page-Flipping URL Training Set" in Section 4.3.1 on the 10-Page/160 test set and manually checked whether it found the correct URLs. The method achieved 99 percent precision and 95 percent recall. The failure is mainly due to JavaScript-based page-flipping URLs or HTML DOM tree alignment error.

5.1.3 Evaluation of Entry URL Discovery As far as we know, all prior works in forum crawling assume that an entry URL is given. However, finding forum entry URL is not trivial. To demonstrate this, we compare our entry URL discovery method with a heuristic baseline discussed in Section 4.5. 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 results are shown in Table 3. 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.

5.2 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 (defined later). We selected nine forums (Table 4) among the 160 test forums for this comparison study. Eight of the nine forums are popular software packages used by many forum sites (except one customized package used by afterdawn.com). These packages cover 388,245 forums. This is about 53 percent of forums powered by the 200 packages studied in this paper, and about 15 percent of all forums we have found. We now define the metrics: effectiveness and coverage. Effectiveness measures the percentage of thread pages among all page crawled of a forum; coverage measures the percentage of crawled thread pages to all retrievable thread pages of the forum.

we would like to have 100 percent effectiveness and 100 percent coverage when all retrievable threads of a forum are crawled and only thread pages are crawled. A crawler can have high effectiveness but low coverage and low effectiveness and high coverage. For example, a crawler can only crawl 10 percent of all retrievable pages, i.e., 10 percent coverage, with 100 percent effectiveness; or a crawler needs to crawl 10 times of retrievable thread pages, i.e., 10 percent effectiveness to reach 100 percent coverage. In order to make a fair comparison, we have mirrored the 160 test forums by a brute-force crawler. All the following crawling experiments were simulated on the mirrored data set.

5.2.1 Evaluation of a Generic Crawler To show the challenges in forum crawling, we implemented a generic breadth-first crawler following the protocols of "nofollow" [6] and robots.txt [10]. This crawler also recorded the URLs with attribute "rel ¼ nofollow" or disallowed by robots.txt but did not visit them. Fig. 11 shows the ratio of thread URLs, uninformative and duplicate URLs, URLs disallowed by robots.txt and URLs with "rel ¼ nofollow." We can see that nofollow is only effective on three forums while robots.txt is effective on six forums. Neither nofollow nor robots.txt is effective on two forums as they are not used on the two forums. Even though they help a generic crawler avoid a lot of pages on seven forums, the crawler still visited many uninformative and duplicate pages. The effectiveness and coverage of this crawler is shown in Fig. 12. The coverage on all forums is almost 100 percent, but the average effectiveness is around 50 percent. The best effectiveness is about 74 percent on "xda-developers (3)." This forum maintained robots.txt better than the other forums. These results showed that "nofollow" and robots.txt did help forum crawling, but not enough. Therefore, we can see a generic crawler is less effective and not scalable for forum crawling, and its performance depends on how well the "no follow" and robots.txt is maintained. Evaluation of starting from nonentry URLs. As we discussed earlier, a crawler starting from the entry URL can achieve higher coverage than starting from other URLs. We used the generic crawler in Section 5.2.1, but set it to only follow index URLs, thread URLs, and page-flipping URLs. According to the definitions of URL types in Section 3, through simple URL string deduplication (e.g., a string hash set), a crawler following only index URLs, thread URLs, and page-flipping URLs will achieve almost 100 percent effectiveness. The generic crawler started from the entry URL and a randomly selected nonentry URL, respectively. It stopped when no more pages could be retrieved. We repeated this experiment with different nonentry URLs. The results are shown in Fig. 13 (we did not show the effectiveness as it was 100 percent). When starting from entry URL, all coverages are very close to 100 percent. When starting from a nonentry URL, coverages decreased significantly. The average coverage was about 52 percent. The coverage on "cqzg (4)" was very high. This forum has many cross-board URLs that help a crawler reach different boards and threads. But in other forums, there are fewer cross-board URLs. This experiment showed that an entry URL is crucial for forum crawling.

Although the structure-driven crawler [25] is not a forum crawler, it could be applied to forums. To make a more meaningful comparison, we adapted it to find page-flipping URL patterns in order to increase its coverage. As to iRobot, we reimplemented it. We let the adapted structure-driven crawler, iRobot, and FoCUS crawl each forum until no more pages could be retrieved. After that we counted how many threads and other pages were crawled, respectively. As showed earlier, starting from nonentry URLs will yield lower coverage. In online crawling, we let crawlers start from nonentry URLs and entry URLs, respectively. The results of iRobot and structure-driven crawler starting from nonentry URLs are much worse than starting from entry URLs. The coverage is lower than 30 percent in average. Since iRobot and structure-driven crawler expect entry URLs as they assumed, we only reported the results when starting from entry URLs.

5.3 Evaluatioin of Large Scale Crawling All previous works evaluated their methods on only a few forums. In this paper, to find out how FoCUS would perform in real online crawling, we evaluated FoCUS on 160 test forums that represented 160 different forum software packages. After learning ITF regexes, we found that FoCUS failed on two forums. One was no longer in operation and the other used JavaScript to generate index URLs. We tested FoCUS on the remaining 158 forums. The effectiveness in 156 out of the 158 forums was greater than 98 percent. The effectiveness of the remaining two forums was about 91 percent. The coverage of 154 out of the 158 forums was greater than 97 percent. The coverage on the remaining four forums ranged from 4 to 58 percent. For these four forums, FoCUS failed to find the page-flipping URLs since they either used JavaScript or they were too small. Without page-flipping URLs, a crawler could only get the first page of each board, and misses all threads in the following pages. Thus, the coverage might be very low. The smallest forum in the 158 test forums had only 261 threads and the largest one had over 2 billion threads.

5.4 Evaluation on cQA Sites and Blog Sites We proposed solving large scale forum crawling problem as a URL type recognition problem by recognizing the EIT path through learning the ITF regexes. In this section, we would like to demonstrate that similar concept can be applied to sites with similar organization structure such as such as community Question and Answer sites (cQA) and blog [1] sites. In cQA sites, the question metadata, e.g., question title, asked time, number of answers, are listed in question index pages structurally similar to forum index pages. By clicking on the title, we can reach a question thread page which contains details of a question, e.g., question content, all answers, etc. In addition, page-flipping URLs are used to link multiple question index pages. Similarly, in blog sites, the metadata of blog posts are listed in blog index pages and the link behind the blog post title leads users to the blog post page, which contains the full content of the blog post and user comments. These blog index pages also contain page-flipping URLs. However, the page-flipping URLs in blog index pages are usually single page-flipping URLs (refer to "Page-Flipping URL Training Set" in Section 4.3.1 for detail). Recall that in forums, the page-flipping URLs are usually grouped page-flipping URLs. This makes it harder to detect page-flipping URLs in blog sites. For example, the "connectivety" metric proposed by Wang et al. [26] does not work in most blog sites in our experiment. Note that question thread pages or blog post pages do not contain page-flipping URLs. Thus, we do not need to detect page-flipping URLs in these pages. In order to show that similar crawling strategy used by FoCUS can be applied to cQA sites and blog sites, we selected more than 100 popular cQA sites and blog sites as evaluation corpus. Table 6 lists 10 of these selected sites.

5.4.1 Evaluation on Blog Sites Following a similar procedure to forum site list preparation, we collect a list of blog software packages or hosting services from WeblogMatrix [11] and "40+ Free Blog Hosts";7 then we manually found at least one instance blog site for each blog software package or host service. Many popular packages and hosting services (e.g., WordPress8) are included in the list. In addition, we also collected a few well-known blog sites which are not powered or hosted by these software packages or services. In total, we collected 59 blog sites. FoCUS succeeded to learn URL patterns of 55 sites. Two failed blog sites required login and the remaining two used JavaScript. The numbers of blog posts in these 55 sites vary from 20 to about 20,475. There are 63,859 blog posts in total. As shown in Table 7, FoCUS achieved 97 percent effectiveness and nearly 100 percent coverage on all blog sites. The lowest effectiveness is 97.27 percent and the lowest coverage is 99.76 percent; the microaverage and macro- average results are nearly100 percent. Note that46 out of the 55 blog sites used single page-flipping URL in their blog index pages. This demonstrated that our page-flipping URL detection module is very effective. Compare to FoCUS, iRobot performs much worse. The average effectiveness is about 90 percent but the coverage is less than 23 percent. Blog sites usually have fewer noisy pages than cQA sites, so iRobot achieves a higher effectiveness on blog sites than on cQA sites. However, iRobot used "connectivity" metric which aims to detect grouped page-flipping URLs but would miss the single page-flipping URLs on the 46 sites. Therefore, the coverage of iRobot on blog sites is rather low. This indicates iRobot's "connectivity" metric is not scalable. The results on cQA sites and blog sites both show that FoCUS is very effective on other social sites besides forums. This also demonstrated that the concept of EIT path can apply to cQA sites, blog sites and other similar social sites and can achieve the same performance as on forums.

6 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., 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 can effectively learn knowledge of EIT path from as few as five 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 training sets and learn ITF regexes from the training sets. These learned 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. Moreover, FoCUS can start from any page of a forum, while all previous works expected an entry URL. Our test results on nine unseen forums show that FoCUS is indeed very effective and efficient and outperforms the state-of-the-art forum crawler, iRobot. 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 applies to other sites, such as community Q&A sites and blog sites. Our experiment results over 100 cQA sites and blog sites have demonstrated this. 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.

REFERENCES

[1] Blog, http://en.wikipedia.org/wiki/Blog, 2012.

[2]"ForumMatrix," http://www.forummatrix.org/index.php, 2012.

[3] Hot Scripts, http://www.hotscripts. wiki com/index.php, 2012.

[4]Internet Forum, http://en.wikipedia.org/ /Internet_forum, 2012.

[5] "Message Boards Statistics," http://www.big-boards.com/ statistics/, 2012.

[6] nofollow, http://en.wikipedia.org/wiki/Nofollow, 2012.

[7] "RFC 1738—Uniform Resource Locators (URL)," http://www. ietf.org/rfc/rfc1738.txt, 2012.

[8] Session ID, http://en.wikipedia.org/wiki/Session_ID, 2012.

[9] "The Sitemap Protocol," http://sitemaps.org/protocol.php, 2012.

[10] "The Web Robots Pages," http://www.robotstxt.org/, 2012.

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

[12] 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.

[13] 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. [14] A. 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] V.N. Vapnik, The Nature of Statistical Learning Theory. Springer, 1995.

[25] M.L.A. Vidal, A.S. Silva, E.S. Moura, and J.M.B. Cavalcanti, "Structure-Driven Crawler Generation by Example," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 292-299, 2006.

[26] Y. Wang, J.-M. Yang, W. Lai, R. Cai, L. Zhang, and W.-Y. Ma, "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. [27] J.-M. Yang, R. Cai, Y. Wang, J. Zhu, L. Zhang, and W.-Y. Ma, "Incorporating Site-Level Knowledge to Extract Structured Data from Web Forums," Proc. 18th Int'l Conf. World Wide Web, pp. 181- 190, 2009.