ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (285-296), Month: October - December 2014, Available at: www.researchpublish.com

FoCUS: Forum Crawler Under Supervision

[1]G.Hema Kumar, [2]G.K.Venkata Narasimha Reddy

[1]PG Scholar, St. Johns College of Engineering & Technology, Yerakota, AP, India
[2]Associate Professor, St.Johns College of Engineering & Technology, Yerakota, AP, India

Abstract: **In this paper, we present FoCUS (Forum Crawler under Supervision), a supervised web-scale forum crawler. The aim of FoCUS is to only probe 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 troubles and show how to learn accurate and effective regular expression patterns of implicit steering paths from an automatically created training set using aggregated results from weak page type classifiers. Robust page type classifiers can be trained from as few as 5 annotated forums and applied to a large set of unseen forums. Our test results show that FoCUS achieved over 98% effectiveness and 97% coverage on a large set of test forums powered by over 150 different forum software packages.**

Keywords**: Navigation Paths, Thread Pages, Robust page.**

1. INTRODUCTION

Internet forums are consequential platforms where users can request and exchange information with others. For example, the Trip Advisor Peregrinate Board is places where people can ask and apportion peregrinate tips. Due to the richness of information in forums, researchers are increasingly fascinated with mining cognizance from them. Zhai et al., Yang et al. and Song et al. extracted structured data from forums. Glance et al. endeavored to mine business astuteness from forum data. Zhang et al. planned algorithms to extract proficiency network in forums. Gao et al. identified question and answer pairs in forum threads. According to an article from eMarketer - Where Are Convivial Media Optically discerning the Most Prosperity? - Forums are still part of the global gregarious media strategy of the Top 500 Companies, and they are still getting marketing prosperity with forums.

To harvest cognizance from forums, their contents have to be downloaded first. Generic crawlers, which adopt a breadth first traversal strategy, are conventionally ineffective and inefficient for forum crawling. This is mainly due to two non-crawler-cordial characteristics of forums: (1) duplicate links & uninformative pages and (2) page-flipping links. A forum customarily has many duplicate links which point to page but with different URLs, e.g., shortcut links pointing to latest posts or URLs for utilizer experience functions such as "view by title". A Generic crawler that blindly follows these links will trawl many duplicate pages that make it inefficient. A Forum typically has many uninformative pages such as authenticate control to bulwark' privacy. Following these links, a crawler will trawl many uninformative pages. Though there are standard predicated methods such as designating the "rel" attribute with "nofollow" value (i.e. "rel=nofollow")2, Robots Exclusion Standard (robots.txt)3, and Sitemap4, for forum operators to injuctively authorize web crawlers on how to crawl a site efficaciously, we found that over a set of 9 test forums more than the pages trawled by a generic crawler following these protocols are duplicate or uninformative. This number is a little higher than the 40% that Cai et al. reported but both show the inefficiency of generic crawlers.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (285-296), Month: October - December 2014, Available at: www.researchpublish.com

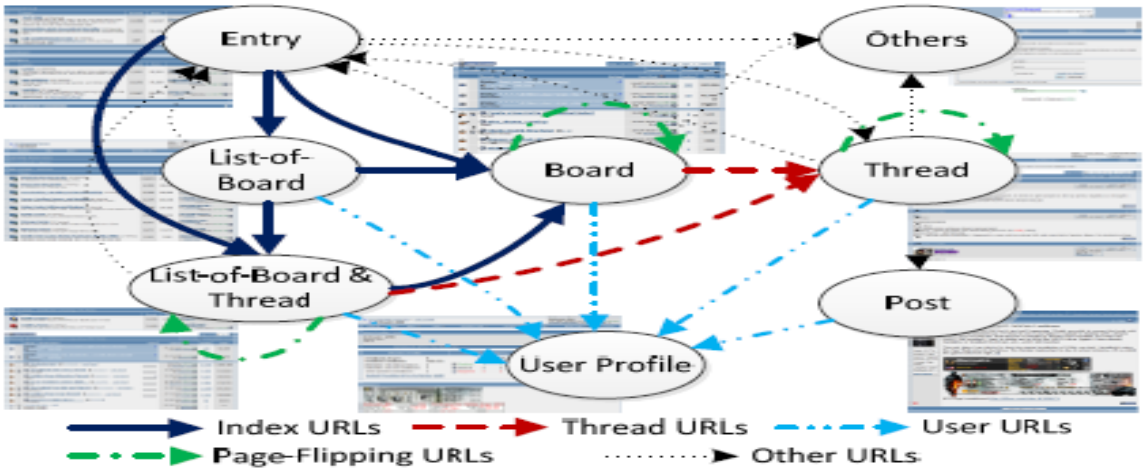

Fig. 1 Typical link structure in Forums

Besides duplicate links & uninformative pages, a long forum board or thread conventionally divided into multiple pages which are linked by page-flipping links. Generic crawlers process each page individually and ignore the relationship between such pages. These relationships should be preserved while crawling to facilitate downstream tasks such as page wrapping and content indexing. For example, multiple pages belonging to a thread should be concatenated together in order to extract all posts of this thread as well as the replication relationships between posts.

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 trawl pertinent content, i.e. user posts, from forums with minimal overhead. Forums subsist 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 ingression pages to thread pages. Figure 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 → board → thread
6. Entry → list-of-board → list-of-board & thread → thread

We call pages between the entry page and thread page which are on a breadth-first navigation path the *index page*. We represent these implicit paths as the following navigation path (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 page of a forum only needs to follow index URLs, thread URLs, and page-flipping URLs to traverse EIT path and achieve 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 regular expression patterns, i.e. ITF regexes, recognizing these three types of URLs from as few as 5 annotated forum packages and apply them to a large set of 160 nseen forums packages. Note that we specifically refer to "forum package" rather than "forum site". A forum software package such as vBulletin5 can be deployed by many forum sites

Major Contributions:

- We reduce the forum crawling quandary to a URL type apperception quandary and implement a crawler, FoCUS, to demonstrate its applicability.
- We show how to automatically learn regular expression patterns (ITF regexes) that recognize the index URL, thread URL, and page-flipping URL utilizing the page classifiers built from as few as 5 annotated forums.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (285-296), Month: October - December 2014, Available at: www.researchpublish.com

- 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 scale evaluation of this type. In addition, we show that the patterns are effective and the resulting crawler is efficient.
- 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.
- We design an effective forum entry URL discovery method. Entry URLs need to be specified to start crawling to get higher recall. But entry page discovery is not a trivial task since entry pages vary from forums to forums. Our evaluation shows that a naïve baseline can achieve only 76% recall and precision; while our method can achieve over 95% recall and precision.

2. TERMINOLOGY

Page Type: We classified forum pages into four page types:

Entry Page: A page that is the lowest common ancestor of all thread pages in a forum. See Figure 2 (a).

Index Page: A page that contains a table-like structure; each row in it contains information on URLs pointing to a board or a thread. See Figure 2 (b). In Figure 1, list-of-board page, list-of board & thread page, board page are all index pages.

Thread Page: A page that contains a list of posts with user generated content (UGC). See Figure 2 (c).

Other Page: A page that is not an entry, index, or thread page.

URL Type: There are four types of URLs:

Index URL: A URL that is on an entry page or index page and points to an index page. Its anchor text shows the title of its destination board. Figure 2 (a) and (b) show an example.

Thread URL: A URL that is on an index page and points to a thread page. Its anchor text shows the title of its destination thread. Figure 2 (b) and (c) show an example.

Page-flipping URL: A URL that leads users to another page of a same board or a 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 Figure 2 (b) and (c) for examples.

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

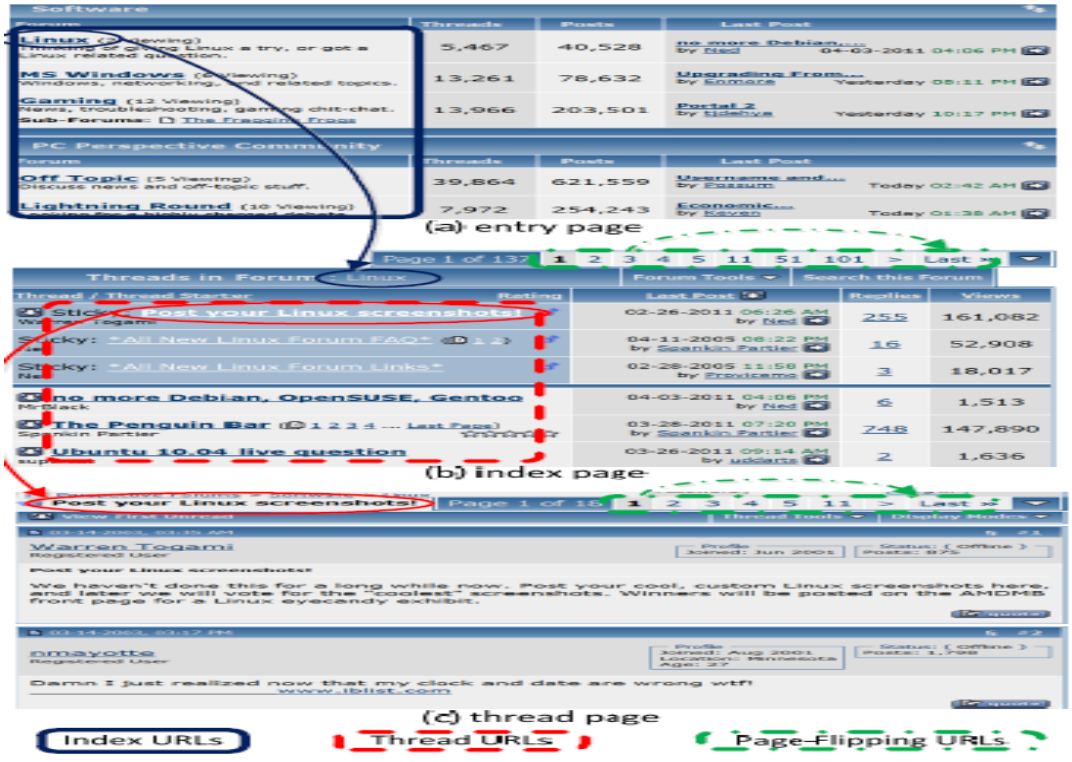

Fig. 2 An instance of EIT path: entry → board → thread

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

ITF Regex. An ITF (index-thread-page-flipping) regex is a regular expression that used to recognize index, thread, or page flipping URLs on EIT path. ITF regexes is what FoCUS aims to

learn and applies directly in online crawling.

3. FoCUS – A SUPERVISED FORUM CRAWLER

In this section, we first motivate and give an overview of our approach. The remaining sections deep dive into each module.

3.1 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:

a. Navigation Path: Despite differences in layout and style, forums always have similar implicit navigation paths leading users from their entry pages to thread pages. In general web crawling, Vidal et al. learned "navigation patterns" leading to target pages (thread pages in our case). iRobot also adopted similar idea but applied page sampling and clustering techniques to find target pages (Cai et al). It used informativeness and coverage metrics to find navigation paths (or traversal paths in Wang et al.). We explicitly defined EIT path that specifies what types of URL and page that a crawler should follow to reach thread pages.

b. 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 Figure 2 (a) and (b), 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 Figure 2 (a) and (b)).

c. Page Layout: Index pages from different forums share similar layout. The same applies to thread pages. However, an index page usually has very different page layout from a thread page. An index page tends to have many narrow records giving information about boards or threads. A thread page typically has a few large records that contain user posts. iRobot used this feature to cluster similar pages together and apply its informativeness metric to decide whether a set of pages should be crawled. FoCUS learns page type classifiers directly from a set of annotated pages based on this characteristic. This is the only part 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 pageflipping URLs can be detected based on their layout characteristics and destination pages; 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.

3.2 System Overview

Figure 3 shows the overall architecture of FoCUS. It consists of two major parts: the learning part and the online crawling part. The learning part learns ITF regexes of a given forum from automatically constructed URL examples. The online crawling part applies learned ITF regexes to crawl all threads efficiently. Given any page of a forum, FoCUS first finds its entry URL using *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 set. Next, the destination pages of the detected index URLs are feed to this module again to detect more index URLs and thread URLs until no more index URL detected. After that, the *Page-Flipping URL Detection* module tries to find page-flipping URLs in both index pages and thread pages and saves them to the training set. Finally, the *ITF Regexes Learning* module learns a set of ITF regexes from the URL training set. FoCUS performs online crawling as follows: it first pushes the entry URL into a URL queue; next it fetches a URL from the queue and downloads its page, and then pushes the outgoing URLs that are matched with any learned ITF regex into the URL queue. This step is repeated until the URL queue is empty.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (285-296), Month: October - December 2014, Available at: www.researchpublish.com

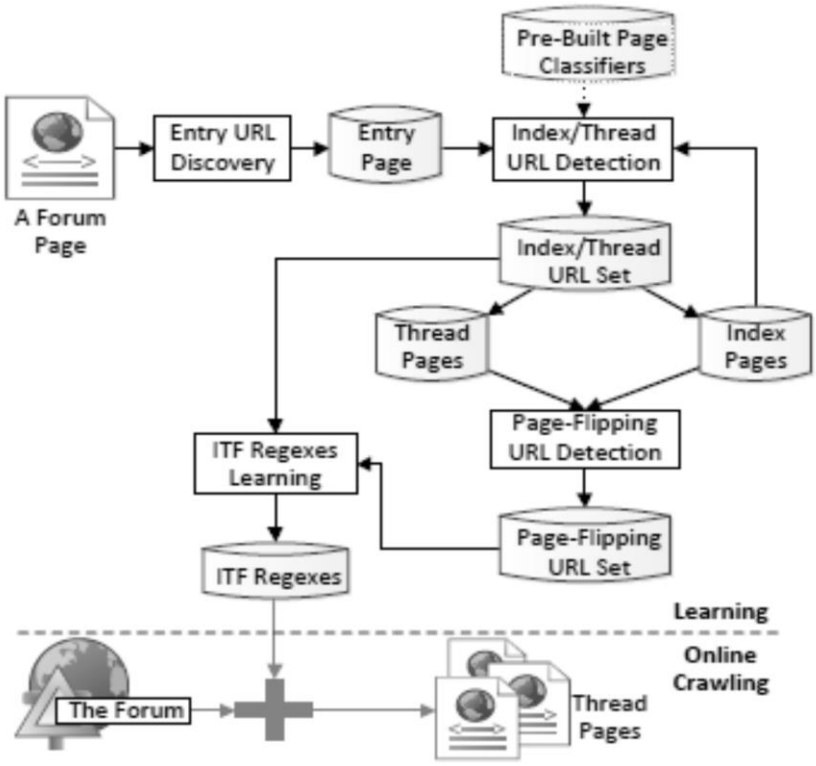

Fig. 3 Overall architecture of FoCUS

3.3 Learning ITF Regexes

To learn ITF regexes, FoCUS adopts a two-step supervised training procedure. The first step is training set construction. The second step is regex learning.

3.3.1 Constructing URL Training Set

The goal of training set construction is to automatically create sets of highly precise index URL, thread URL, and page-flipping URL string samples for regex learning. We use a similar procedure to construct index URL and thread URL training sets since they have very similar properties except the types of their destination pages; we present this part first. Page-flipping URL strings have their own specific properties which are different from properties of index URL and thread URL strings; we present this part later.

3.3.1.1 Index and Thread URL String Training Sets

Recall that an index URL is a URL that is on an entry page or index page; and its destination page is another index page; while a thread URL is a URL that is on an index page; and its destination page is a thread 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 page type of a destination page. As we mentioned in Section 3.1, the index page and thread page 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, usually user posts or merchant advertisements. 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 Figure 2 for example). Inspired by such characteristic, we propose features representing page layouts as shown in Table I and build page classifiers using Support Vector Machine (SVM) to decide page type. SVM*light* Version 6.026 with a default linear kernel setting is used to build our classifiers. One index page classifier and one thread page classifier are built using the same feature set. FoCUS does not need strong page type classifiers.

Table I. Main features used in index/thread page classification

Feature Name	Value Type	Short Description
Record Count	Float	Number of records
Max Anchor Text Length	Float	The max length in characters of anchor text per record
Max Height	Float	The max value of height of each record
Max Text Length	Float	The max length in characters of plain text per record
Average Text Length	Float	The average length in characters of plain text among all records
Has Timestamp	Boolean	Whether each record has a timestamp
Time Order	Float	The order of timestamps in the records if the timestamps exist
Has User Link	Boolean	Whether each record has a linkpointing to a user profile page

3.4 Online Crawling

Given a forum, FoCUS first learns a set of ITF regexes following the procedure described in the previous sections. Then it performs online crawling using a breadth-first strategy. It first pushes the entry URL into a URL queue; next it fetches a URL from the URL queue and downloads its page; and then it pushes the outgoing URLs that are matched with any learned regex into the URL queue. FoCUS repeats this step until the URL queue is empty or other conditions are satisfied. What makes FoCUS efficient in online crawling is that it only needs to apply the learned ITF regexes on 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 its learning phase.

3.5 Entry URL Discovery

In the previous sections, we explained how FoCUS learns ITF regexes that can be used in online crawling to determine what URLs to follow and what URLs to ignore. However, an entry page needs to be specified to start the crawling process. To the best of our knowledge, all previous methods assumed a forum entry page is given. In practice, especially in web-scale crawling, manual forum entry page annotation is not practical. Forum entry page discovery is not a trivial task since entry pages vary from forums to forums. Our experiment shows that a naïve baseline method can achieve only about 76% recall and precision. To make FoCUS more practical and scalable in web-scale crawling, we design a simple yet effective forum entry URL discovery method based on some techniques introduced in previous sections. We observe that (1) almost every page contains a link to lead users back to the entry page of a forum; (2) an entry page has most index URLs since it leads users to all forum thread pages. Based on the index URL detection module and the index page classifier described the Section 4.3.1.1, we proposed a method for finding the entry page for a forum given a URL pointing to any page of the forum. Our evaluation on 160 forums shows that this method can achieve 98% precision and 95% recall

4. EXPERIMENTS

To carry out meaningful evaluations that is good indicators of web-scale forum crawling, we selected 200 different forum software packages from Forum Matrix [1], Forum Software [2], and Big-Board*s [3]. For each software package, we found a forum site 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 software packages cover a large number of forum sites. The 40 training packages are deployed by 59,432 forum sites and the 160 test packages are deployed by 668,683 forum sites. 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, posts, 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 according to our statistics, on all forum sites that we have found, the top 10 most frequent software packages are deployed by 17% of all forum sites and cover about 9% of all threads. In the evaluation of page type classification, index/thread URL detection, and entry page discovery, we selected the training/test pages and checked results manually as the data set is not very large (no more than 5,000 pages). But in the evaluation of online crawling, it's impossible to manually verify whether all the crawled pages are thread page or not. Instead, we wrote a set of URL based rules to detect index and thread pages for each forum. We used these rule sets to check the page type of crawled pages. To ensure that our rule sets can be used as an alternative to manual evaluation, we used each rule set to

annotate 200 pages randomly selected from its corresponding forum. We found that these rule sets achieved at least 99% precision and 100% recall.

4.1 Efficiency of FoCUS

We evaluated the efficiency of FoCUS in terms of the number of pages crawled and the time spent during its learning phase. Similar to structure-driven crawler and iRobot, FoCUS fetches some pages during its learning phase. We evaluated how many pages FoCUS needed to visit to get satisfactory learning results. Among the 160 test forums used in our evaluation, the maximum number of pages FoCUS visited is 2,663, the minimum number is 252, and the average number is 442. We found that the structuredriven crawler needed to visit at least 3,000 pages to find navigation patterns, which is larger than Vidal et al. reported; iRobot needed to sample at least 2,000 pages to find complete traversal paths. To estimate the time spent on a forum during the learning phase, we ran FoCUS on a machine with two 4-core 2.20 GHz CPUs, 32 GB memory, and 64-bit Windows Server 2008 SP1 OS. The maximum time spent on a forum is 3,416 seconds, the minimum time was 25 seconds, and the average was 466 seconds. According to our experience, even a skilled human would spend about 1,200 seconds on average to write URL rules for a forum.

4.2 Evaluations of FoCUS Modules

4.2.1 Evaluation of Index/Thread URL Detection

To build page classifiers, we manually selected 5 index pages, 5 thread pages, and 5 other pages from each of the 40 forums and extracted the features listed in Table I. For testing, we manually each of the 160 forums. This is called 10-Page/160 test set. Note that we computed the results at the page level not at the individual URL level since we applied a majority voting procedure. To further examine how many annotated pages FoCUS needs to achieve a good performance, we conducted similar experiments but with more training forums (10, 20, 30, and 40) and applied cross validation. We find that our classifiers achieved over 96% recall and precision at all cases with tight standard deviation. It is particularly encouraging to see that FoCUS can achieve over 98% precision and recall in index/thread URL detection with only as few as 5 annotated forums.

4.2.2 Evaluation of Page-Flipping URL Detection

We applied the module on the 10- Page/160 test set and manually checked whether it found the correct page-flipping URLs. The method achieved 99% precision and 93% recall. The failure is mainly due to JavaScript-based page-flipping URLs or HTML DOM tree alignment error.

4.2.3 Evaluation of Entry URL Discovery

As far as we know, all prior works in forum crawling assume that an entry page is given. However, finding forum entry page is not trivial. To demonstrate this, we compare our entry page detection method with a heuristic 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 page URL; if not, the URL host is extracted as its entry page URL. For each forum in the test set, we randomly sampled a page and fed it to this module. Then we manually checked if its output was indeed its entry page. In order to see whether FoCUS and the baseline are robust, we repeated this procedure 10 times with different sample pages. The baseline had 76% precision and recall. On the contrary, FoCUS achieved 98% precision and 95% 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.

4.3 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 9 forums (Table II) from among the 160 test forums for this comparison study. 8 of the 9 forums are popular software packages used by many forum sites (plus one customized package used by afterdawn.com). These software packages cover 388,245 forums. This is about 53% of forums powered by the 200 packages studied in this paper, and about 15% of all forums we have found.

We now define effectiveness and coverage metric. Effectiveness measures the percentage of thread pages among all pages crawled:

$$\text{Effectiveness} = \frac{\#\text{Crawled threads}}{\#\text{Crawled pages}} \times 100\%$$

Coverage measures the percentage of crawled thread pages of a forum crawled to all retrievable thread pages of the forum:

$$\text{Coverage} = \frac{\#\text{Crawled threads}}{\#\text{Threads in all}} \times 100\%$$

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

Table II. Forums used in online crawling evaluation

ID	Forum	Software	#Threads
1	http://forums.afterdawn.com/	Customized	535,383
2	http://forums.asp.net/	CommunityServer	66,966
3	http://forum.xda-developers.com/	vBulletin	299,073
4	http://bbs.cqzg.cn/	Discuz!	428,555
5	http://forums.crackberry.com/	vBulletin V2	525,381
6	http://forums.gentoo.org/	phpBB V2	681,813
7	http://lkcn.net/bbs/	IP.Board	180,692
8	http://www.techreport.com/forums/	phpBB	65,083

4.3.1 Evaluation of a Generic Crawler

To show the hardness of the challenges in forum crawling, we implemented a generic breadth-first crawler following the protocols of "nofollow" and robots.txt. This crawler also recorded the URLs with the attribute "rel=nofollow" or which were disallowed by robots.txt but did not visit them.

Figure 4 shows the ratio of thread URLs, uninformative & duplicate URLs, URLs disallowed by robots.txt and URLs with "rel=nofollow". We can see that "nofollow" is only effective on 3 forums while robots.txt is effective on 6 forums. Neither nofollow nor robots.txt is effective on 2 forums as they are not used on the 2 forums. Even though they help a generic crawler avoid a lot of pages on 7 forums, the crawler still visited many uninformative & duplicate pages. To show that clearly, the effectiveness of this crawler is shown in Figure 7 and coverage is shown in Figure 8.

The coverage on all forums is almost 100%, but the average effectiveness is around 53%. The best effectiveness is about 74% on "xda-developers (3)". It's because this forum better maintained robots.txt than other forums. This showed that "nofollow" and robots.txt did help forum crawling, but not enough. We can conclude that a generic crawler is less effective and not scalable for forum crawling, and its performance depends on how well the "nofollow" and robots.txt is maintained.

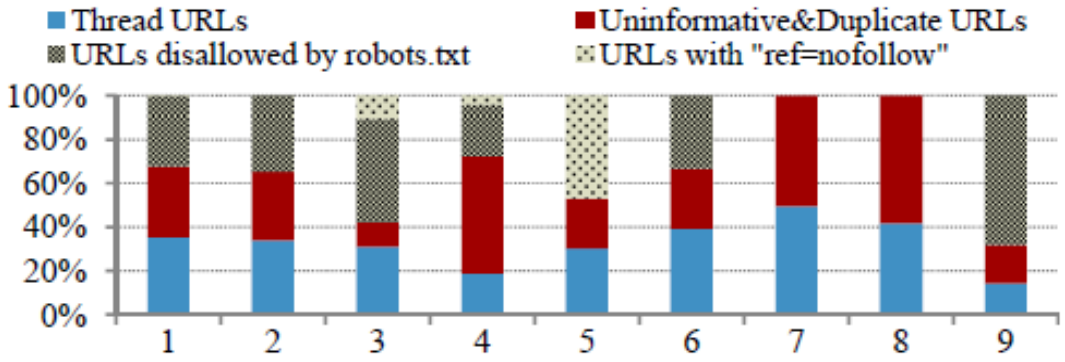

Fig. 4 Ratio of different URLs discovered by a generic crawler

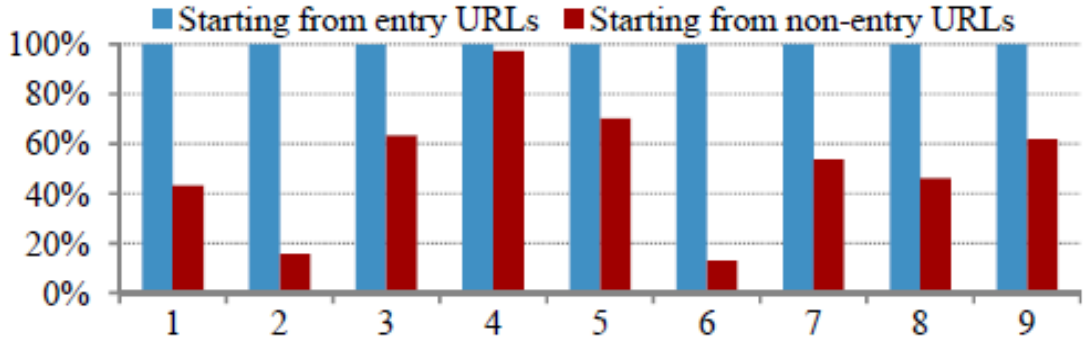

Fig. 5 Coverage comparison between starting from entry URL and non-entry URL

4.3.2 Evaluation of Starting from Non-Entry URLs

As we discussed earlier, a crawler starting from the entry URL cab achieve higher coverage than when starting from other URLs. We also did an experiment to verify this. We used the generic crawler in Section 4.3.1, but set it only follow the index URL, thread URL and page-flipping URL. So it is an ideal forum crawler with 100% effectiveness. It starts from the entry URL and a randomly selected non-entry URL respectively. The crawler stopped when no more pages could be retrieved. We also repeat this experiment with different non-entry URLs.

The results are shown in Figure 6 (we did not show the effectiveness as it was 100%). When starting from entry URL, the coverage is all close to 100%. But when starting from a non-entry URL, the coverage decreased significantly. The average coverage is about 52%. The coverage on "cqzg (4)" is very high. This is because there are many cross-board URLs in that forum. That is, a page of one board contains URLs pointing to other boards. But in other forums, there are fewer cross-board URLs. Thus the crawler could only find the threads in the board to which the starting URL belongs. This experiment showed that an entry URL is critical for forum crawling, and automatic entry URL discovery is necessary to make a crawler scalable.

4.3.3 Evaluation of Structure-Driven Crawler

Although the structure-driven crawler is not a forum crawler, it could be applied to forums. To make it a more meaningful comparison, we adapted it to find page-flipping URL patterns in order to increase its coverage.

The results of it are shown in Figure 6. We observe that it performed well on "cqzg (4)" and "techreport (8)". On these 2 forums, it found the exact patterns of index URL, thread URL, and page-flipping URL. However, it did not perform well on other forums, and for example, suffered from low effectiveness. This is primarily due to the absence of good and specific URL similarity functions in structure-driven crawler. Thus it did not find precise URL patterns. Therefore, it performed similarly to the behavior of a generic crawler and got good coverage on the forums except "afterdawn (1)". This is because it does not find the pattern of page-flipping URL on this forum. From the results, we can conclude that a structure-driven crawler with small domain adaptation is not enough to be used as an effective forum crawler. We compare FoCUS with a more competent forum crawler, iRobot, and a generic crawler in the next section.

4.3.4 Online Crawling Comparison

In this section, we report the result of comparison between a generic crawler described in Section 4.3.1, iRobot (we reimplemented iRobot for our evaluations) and FoCUS. We let iRobot and FoCUS crawl each forum until no more pages could be retrieved. After that we count how many threads and other pages were crawled, respectively.

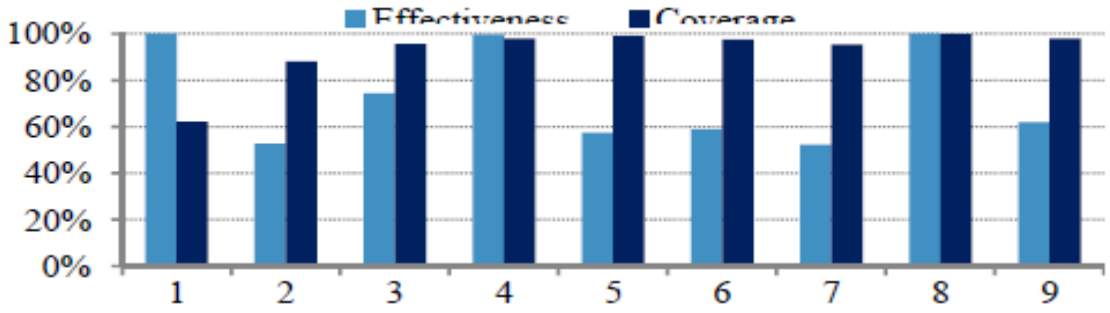

Fig. 6 Effectiveness and coverage of structure-driven crawler

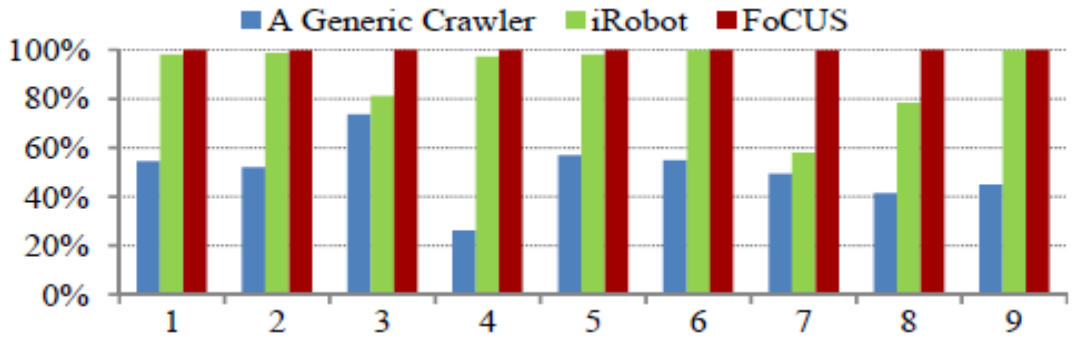

Fig. 7 Effectiveness comparison between a generic crawler iRobot and FoCUS

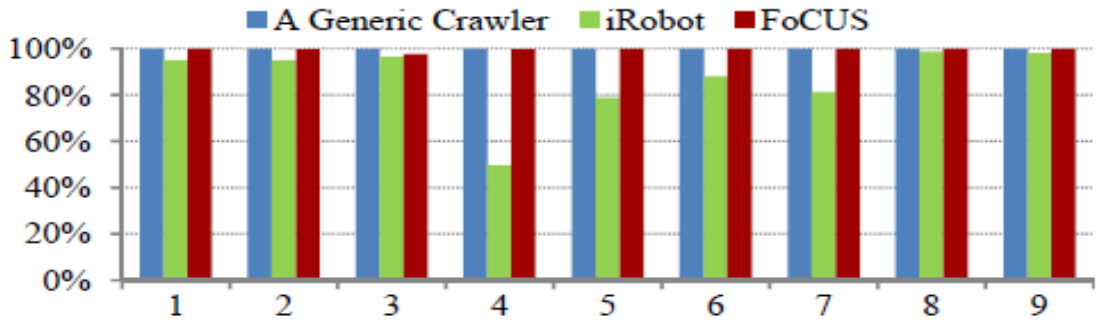

Fig. 8 Coverage comparison between a generic crawler, iRobot and FoCUS

4.3.4.1 Crawling Effectiveness Comparison

Figure 7 shows the effectiveness comparison. FoCUS achieved almost 100% effectiveness on all forums. This confirms the effectiveness and robustness of FoCUS. The generic crawler is much less effective as we discussed in Section 4.3.1. As a comparison forum crawler, iRobot's effectiveness was about 90%. It obtained high effectiveness on 6 out of 9 forums. But, it did not perform well on "xda-developers (3)", "lkcn (7)", and "techreport (8)". After an error analysis, we found iRobot's ineffectiveness was mainly due to that its random sampling strategy sampled many useless and noisy pages. For "xda-developers (3)" and "techreport (8)", they have many redundant URLs, which made it hard to select the best traversal path. As to "lkcn (7)", iRobot crawled many user profiles. Because this forum did not impose login control or robots.txt, many user profile pages were sampled and retained by iRobot's informativeness estimation.

Comparing to iRobot, FoCUS learns the EIT path in forums and ITF regexes directly. Therefore, FoCUS was not affected by noisy pages and performed better. This indicates that given the fixed bandwidth and storage, FoCUS could crawl much more valuable content than iRobot.

4.3.4.2 Crawling Coverage Comparison

According to Figure 8, FoCUS had better coverage than iRobot. The average coverage of FoCUS was 99%, which is very close to a generic crawler, comparing to iRobot's 86%. After an error analysis, we found that iRobot's low coverage on "cqzg (4)" was due to the fact its forum pages had two layout structures. iRobot learned only one structure from sampled pages. This led to iRobot's loss of many thread pages. For "crackberry (5)" and "Gentoo (6)", iRobot suffered from changed page-flipping URL locations across pages. iRobot's tree-like traversal path also deceased its coverage on "lkcn (7)" and "cqzg (4)". In these 2 forums, some boards have many sub-boards. Recall that iRobot's tree-like traversal path selection does not allow more than one path from an entry page node to a thread page node. Thus iRobot missed the thread URLs in these sub-boards.

In contrast, FoCUS crawled almost all the threads in forums since it learned EIT path and ITF regexes directly. In online crawling, FoCUS found all boards and threads whether they appeared in repetitive regions or not. The results on coverage also demonstrated our methods of index/thread URL and page-flipping URL detection are very effective.

4.3.5 Large Scale Online 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 which represents 160 different forum software packages. After learning ITF regexes, we found that FoCUS failed on 2 forums. One forum was no longer in operation and the other used JavaScript to generate index URLs. We tested FoCUS on the remaining 158 forums.

The effectiveness of 156 out of the 158 forums was greater than 98%. The effectiveness of the remaining 2 forums was about 91%. The coverage of 154 out of the 158 forums was greater than 97%. The coverage of the remaining 4 forums ranged from 4% to 58%. For these forums, FoCUS failed to find the page-flipping URLs since they either used JavaScript or they were too small.

The smallest forum in the 158 test forums had only 261 threads and the largest one had over 2 billion threads. To verify that FoCUS performs well across forums of different sizes, we calculated its micro-average and macro-average of effectiveness and coverage. As shown in Table III, the micro-average and macroaverage are both high and they are very close. This indicates that FoCUS performed well on small forums and large forums and is practical in web-scale crawling. To the best of our knowledge, it's the first time such a large-scale test has been reported.

Table III. Micro/Macro-average of effectiveness and coverage

	Effectiveness %	**Coverage %**
Micro-Average	99.96	99.16
Macro-Average	99.85	97.46

5. CONCLUSION

In this paper, we proposed and implemented FoCUS, a supervised forum crawler. We reduced the forum crawling quandary to a URL type apperception quandary and showed how to leverage implicit navigation paths of forums, i.e. ingression-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 attest that FoCUS could efficaciously cognizance of EIT path and ITF regexes from as few as 5 annotated forums. We additionally showed that FoCUS can efficaciously apply learned forum erudition on 160 unseen forums to automatically accumulate 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 substratum of forum package makes our experiments manageable and our results applicable to many forum sites. Moreover, FoCUS can commence from any page of a forum, while all anterior works expect an ingress page is given. Our test results on 9 unseen forums show that FoCUS is indeed very efficacious

and efficient and outperforms state-of-the-art forum crawler, iRobot. The results on 160 forums show that FoCUS can apply the learned cognizance to an immensely colossal 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 withal apply to other sites, such as community Q&A sites, blog sites, and so on.

REFERENCES

[1] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016," Tech. Rep., 2012.

[2] Y. Li, Y. Zhang, and R. Yuan, "Measurement and Analysis of a Large Scale Commercial Mobile Internet TV System," in ACM IMC, pp. 209–224, 2011.

[3] T. Taleb and K. Hashimoto, "MS2: A Novel Multi-Source Mobile-Streaming Architecture," in IEEE Transaction on Broadcasting, vol. 57, no. 3, pp. 662–673, 2011.

[4] X. Wang, S. Kim, T. Kwon, H. Kim, Y. Choi, "Unveiling the BitTorrent Performance in Mobile WiMAX Networks," in Passive and Active Measurement Conference, 2011.

[5] Nafaa, T. Taleb, and L. Murphy, "Forward Error Correction Adaptation Strategies for Media Streaming over Wireless Networks," in IEEE Communications Magazine, vol. 46, no. 1, pp. 72–79, 2008.

[6] J. Fernandez, T. Taleb, M. Guizani, and N. Kato, "Bandwidth Aggregation-aware Dynamic QoS Negotiation for Real-Time Video Applications in Next-Generation Wireless Networks," in IEEE Transaction on Multimedia, vol. 11, no. 6, pp. 1082–1093, 2009.

[7] T. Taleb, K. Kashibuchi, A. Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto, "A Cross-layer Approach for An Efficient Delivery of TCP/RTP-based Multimedia Applications in Heterogeneous Wireless Networks," in IEEE Transaction on Vehicular Technology, vol. 57, no. 6, pp. 3801–3814, 2008.

[8] K. Zhang, J. Kong, M. Qiu, and G.L Song, "Multimedia Layout Adaptation Through Grammatical Specifications," in ACM/Springer Multimedia Systems, vol. 10, no. 3, pp.245–260, 2005.

[9] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-Time System for Adaptive Video Streaming Based on SVC," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1227–1237, Sep. 2007.

[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.