Exploring Traversal Strategy for Web Forum Crawling

Yida Wang[†]*, Jiang-Ming Yang[‡], Wei Lai[‡], Rui Cai[‡], Lei Zhang[‡], and Wei-Ying Ma[‡]

[†] CSSAR, Chinese Academy of Sciences, Beijing 100190, P.R. China

[‡] Microsoft Research Asia, Beijing 100190, P.R. China

[†] wangyida@cssar.ac.cn, [‡] {jmyang, weilai, ruicai, leizhang, wyma}@microsoft.com

ABSTRACT

In this paper, we study the problem of Web forum crawling. Web forum has now become an important data source of many Web applications; while forum crawling is still a challenging task due to complex in-site link structures and login controls of most forum sites. Without carefully selecting the traversal path, a generic crawler usually downloads many duplicate and invalid pages from forums, and thus wastes both the precious bandwidth and the limited storage space. To crawl forum data more effectively and efficiently, in this paper, we propose an automatic approach to exploring an appropriate traversal strategy to direct the crawling of a given target forum. In detail, the traversal strategy consists of the identification of the *skeleton links* and the detection of the *page-flipping links*. The skeleton links instruct the crawler to only crawl valuable pages and meanwhile avoid duplicate and uninformative ones; and the page-flipping links tell the crawler how to completely download a long discussion thread which is usually shown in multiple pages in Web forums. The extensive experimental results on several forums show encouraging performance of our approach. Following the discovered traversal strategy, our forum crawler can archive more informative pages in comparison with previous related work and a commercial generic crawler.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering, information filtering*.

General Terms

Algorithms, Performance, Experimentation

Keywords

Forum crawler, sitemap, traversal strategy.

1. INTRODUCTION

Web forum (also named bulletin or discussion board) is a Web application for holding discussions and posting user-created content (UCC). Web forum may refer to either the entire community (e.g. MSDN) or a specific sub-forum within a large Web site dealing with a distinct topic (e.g. Amazon) [1]. With millions of users' contribution, forum data usually has plenty of highly valuable knowledge and information, and becomes an important resource on the Web. The strong driving force behind the highly valuable forum data is the power of users and communities. For example, consumers more like to share their comments and experiences to some electronic products on Web forums. In comparison with the reviews from official websites, comments from forums are relatively unbiased and with more perspectives. Most commercial search engines such as Google, Yahoo!, and Live have begun to leverage information from Web forums to improve their search result qualities. To do this, search engines have to first download pages from various forums and build a locally indexed repository [5]. High quality search results highly depend on a high quality repository. This requires the crawler to fetch as much as possible valuable data from the infinite number of Web forum pages [3], using the limited bandwidth and storage space .

However, traditional generic crawlers [5] which basically adopt the breadth-first traversal strategy [4] are usually inefficient in crawling forum sites. This is because forum sites have some different characteristics from other general websites. First, to help users conveniently browsing, a Web forum usually has many shortcut links which point to the same content but often have different URLs. Indiscriminately following all these shortcuts will lead to many duplicate pages in forum crawling. Second, most Web forums have access control to protect privacy, and some links are forbidden for unregistered users (such as a crawler). Rudely following these links in crawling will result in many uninformative pages such as a login portal. According to the statistics in [7], using a breadth-first and depth-unlimited crawler, averagely there are more than 40% useless (uninformative or duplicate) forum pages among all the crawled pages. These pages seriously degrade the repository quality. Moreover, generic crawlers totally ignore the problem of page-flipping in Web forums. That is, a long discussion thread is usually divided and shown in multiple pages. Such a relationship among these pages should be preserved in crawling to facilitate further indexing and storage. At last, as content of Web forums usually changes more frequently, a crawler needs a high frequent re-crawling schedule to keep refreshing the repository, which makes the above three problems even worse. This motivates us to explore a better traversal strategy for Web forum crawling.

In general, for forum crawling, an ideal traversal strategy needs to answer two questions: 1) *what kinds of links should be followed*? To save bandwidth and reduce redundancy, only the links that are linked to valuable pages should be kept whereas the links to uninformative and duplicate pages should be dropped; and 2) *how to follow these links*? To better schedule the crawling queue and to build more efficient index structures, the flipping relationships among pages should be discovered and preserved.

To the best of our knowledge, little existing work in literatures has systemically investigated the traversal strategy for forum crawl-

SIGIR'08, July 20–24, 2008, Singapore.

Copyright 2008 ACM 978-1-60558-164-4/08/07…$5.00.

*This work was performed when the first author was an intern in Microsoft Research Asia.

ing. Most existing research works on Web crawling only utilize out-link related information such as URL pattern and anchor text [8][12] to design traversal strategies, which ignore the relationships among various pages but just judge each out-link on different pages independently. However, considering the complex in-site link structures and access controls of Web forums, such works cannot completely meet the requirements of forum crawling. We will provide a more detailed review to discuss these related research efforts in Section 2. Therefore, to well answer the aforementioned two questions, a crawler needs to utilize more information than a link itself, and should think about the traversal strategy globally.

In this paper, we propose a novel solution for efficient forum crawling. To understand the global organization of a forum, we first re-construct its *sitemap* based on a few thousands pages randomly sampled from the target site [7]. The sitemap tells us how many kinds of pages are there in the target forum, and how they are linked together. In this way, we can leverage more global knowledge to explore the traversal strategy. For example, besides the URL pattern of a link, we also know the characteristics of both its source and target pages. Moreover, as such information of the source/target pages is summarized based on the statistics of a group of pages, it is more robust than only considering a single page. Given the sitemap, the traversal problem becomes to investigate *whether to follow* and *how to follow* every link in the sitemap. More specifically, in this paper we focus on the identification of two kinds of links, *skeleton link* and *page-flipping link*, from the sitemap, as described in the following.

- **SKELETON LINK IDENTIFICATION**. To answer the first question, in this module we try to discover all principal links pointing to valuable and informative pages (called skeleton links) in the target forum site. We introduce two robust criteria to evaluate the importance of each link, and also propose a search-based algorithm which is inspired by the browsing behavior of forum users.
- **PAGE-FLIPPING LINK DETECTION**. To answer the second question, in this module we try to identify all links corresponding to page-flipping. That is, a crawler must correctly follow these links one-by-one to completely download a discussion thread. After investigating a substantial number of examples from various Web forums, we propose a novel criterion according to the navigation connectivity of a group of pages. Experimental results show that this criterion can significantly distinguish page-flipping links from others.

The rest of this paper is organized as follows. First, we briefly review some related works in Section 2, and define the problem settings in Section 3. The system overview of the proposed approach is introduced in Section 4, and the algorithm details are described in Section 5. Experiment evaluations are reported in Section 6. And in the last section, we draw conclusions and point out some future research directions.

2. RELATED WORK

To make a tradeoff between the "performance and cost", most generic Web crawlers adopt the breadth-first strategy and limit the crawling depth. However, in practice it is hard to select a proper crawling depth for each site. A shallow crawling strategy cannot ensure to access all valuable content, whereas a deep crawling strategy may result in too many duplicate and invalid pages. To improve this, some research works tried to find more effective crawling strategy than the breadth-first one [2][4]. For example, the on-line page importance computation (OPIC) was proposed in [2] which utilized the partial PageRank to estimate the importance of a page, to help a crawler skip valueless pages. However, we argue that this approach is not suitable for forum sites, because it is hard to appropriately estimate the PageRank scores for pages in Web forums. As content in Web forums changes frequently every day, new generated pages usually have very low PageRank scores.

Deep Web (or hidden Web) crawling [14] is also a related research topic. This is because forums are also a kind of deep Web which consists of dynamic pages generated from a database. However, the focuses of a forum crawler and a deep Web crawler are different. A deep Web crawler focuses on how to prepare appropriate queries to probe and retrieve hidden pages; while a forum crawler is interested in how to identify valuable links to follow, given that most forum pages have explicit in-links.

The most related work is focused crawling [8][12][13][15], which attempts to retrieve Web pages that are relevant to some pre-defined topics or labeled examples. The target descriptions in focused crawling are quite different in various applications. In [8][12][13], the targets are usually described with a series terms or topics; while in [15] the target was described by the DOM tree of a manually selected sample page. Forum crawler is also a kind of targeted crawler as it selectively downloads informative pages containing user-created content. However, the existing methods are not suitable for forum crawling. First, semantic topics in forums are too diverse to be simply characterized with a list of terms. The structure driven-based approach in [15] seems promising since it only cares about the layout structure of the target pages. This is suitable for forum crawling as most pages in forums are generated by some pre-defined templates; and different templates have different layout structures to present different content such as *list-of-board*, *list-of-thread*, *post-of-thread*, *user profile*, etc. However, the strategy in [15] only focuses on finding a possible path to the target pages, but cannot guarantee that it is an optimal one. Actually, in forums there are usually multiple paths to visit a page. Most of these paths are just shortcuts and cannot promise to cover all the targets. Moreover, the problem of detecting page-flipping links was not discussed in [15]. We will compare our approach with the structure-driven based approach in the experiments part.

There is also one recent study investigating the problem of forum crawling [9]. Unfortunately, it mainly adopted some heuristic rules as traversal strategies, and can only deal with forums with some specific organization structures. While in reality there are hundreds of forum structures implemented by either Website developers or different Internet forum software. Thus it is impractical to define universal heuristics for general forum crawling.

3. PROBLEM SETTING

To make a clear presentation and to facilitate the following discussions, we first explain some concepts used in this paper.

SITEMAP. A sitemap is a directed graph consisting of a set of *vertices* and the corresponding *links*. Each *vertex* represents a group of forum pages which have similar page layout structure; and each *link* denotes a kind of linking relationship between two vertices. Fig. 1 (a) provides an illustration of the sitemap for the ASP.NET forum (http://forums.asp.net). For vertices, we can find that each vertex is related to one kind of pages in the forum, as shown in

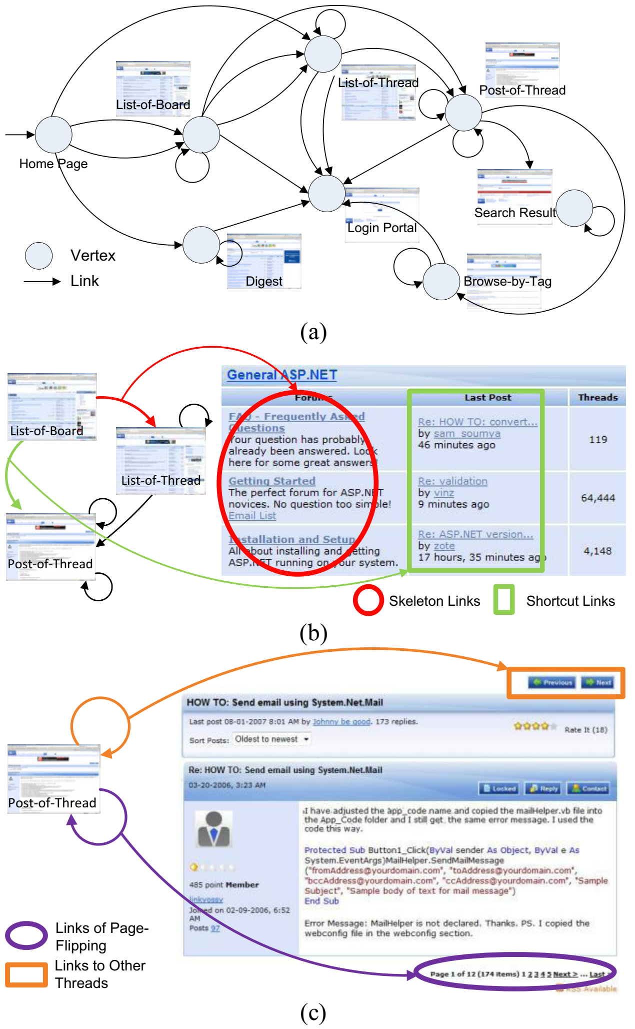
List-of-Board
List-of-Thread
Post-of-Thread
Home Page
Search Result
Login Portal
Digest
Browse-by-Tag
Vertex
Link
(a)
General ASP.NET
Forums
Last Post
Threads
FAQ - Frequently Asked Questions
Your question has probably already been answered. Look here for some great answers
Re: HOW TO: convert...
by sam_soumva
46 minutes ago
119
Getting Started
The perfect forum for ASP.NET novices. No question too simple!
Email List
Re: validation
by vinz
9 minutes ago
64,444
Installation and Setup
All about installing and setting ASP.NET running on your system.
Re: ASP.NET version...
by zote
17 hours, 35 minutes ago
4,148
Skeleton Links
Shortcut Links
(b)
Previous
Next
HOW TO: Send email using System.Net.Mail
Last post 08-01-2007 8:01 AM by Johnny be good. 173 replies.
Sort Posts: Oldest to newest
Rate It (18)
Re: HOW TO: Send email using System.Net.Mail
03-20-2006, 3:23 AM
Locked
Reply
Contact
I have adjusted the app_code name and copied the mailHelper.vb file into the App_Code folder and I still get the same error message. I used the code this way.
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs)MailHelper.SendMailMessage ("fromAddress@yourdomain.com", "toAddress@yourdomain.com", "bccAddress@yourdomain.com", "ccAddress@yourdomain.com", "Sample Subject", "Sample body of text for mail message")
End Sub
485 point Member
Joined on 02-09-2006, 6:52 AM
Posts 97
Error Message: MailHelper is not declared. Thanks. PS. I copied the webconfig file in the webconfig section.
Page 1 of 12 (174 items) 1 2 3 4 5 Next > ... Last
Links of Page-Flipping
Links to Other Threads
(c)

Fig. 1. An illustration of the (a) Sitemap, (b) Skeleton Link, and (c) Page-Flipping Link. The graph in (a) and the screen shots in (b) and (c) have been simplified for a clear view.

Fig. 1 (a) with the typical pages and labels. We can also find that pages in some vertices such as the "login portal" are of little information. For links, we should address that in this paper each link is characterized by both the *pattern* and the *location* of where the corresponding URLs are located in a page. Location information can help distinguish different links when there is no distinct URL pattern in a forum; for more details please refer to [7]. Therefore, in Fig. 1 (a), we can find there may be multiple links from one vertex to another.

SKELETON LINK. Skeleton links, as the name shows, are the most important links supporting the structure of a forum site. As aforementioned, to facilitate users' browsing, in Web forums there are usually more than one path to visit a page. For example, as shown in Fig. 1 (b), to browse a *post-of-thread* page, we can either directly click on the links in the green square in a *list-of-board* page; or follow the links in the red ellipse in a *list-of-board* page to reach a *list-of-thread* page first, and then follow links in the *list-of-thread* page to read each post. Actually, links in the green square are a kind of shortcut links which are designed to help users read the newest posts more conveniently; while links in the red ellipse reflect the real organizational structure of the forum. That is, browsing the *list-of-board* pages, *list-of-thread* pages, and *post-of-thread* pages in a sequential manner. Although a crawler can download some post pages following the shortcut links, it takes a huge risk that most post pages will be missed in the crawling. In fact, only the newest post pages have shortcuts in the *list-of-board* page. Moreover, there are often multiple shortcuts pointing to a page, which may lead to duplicate copies in crawling. In contrast, skeleton links ensure to reach all the valuable pages with few redundant.

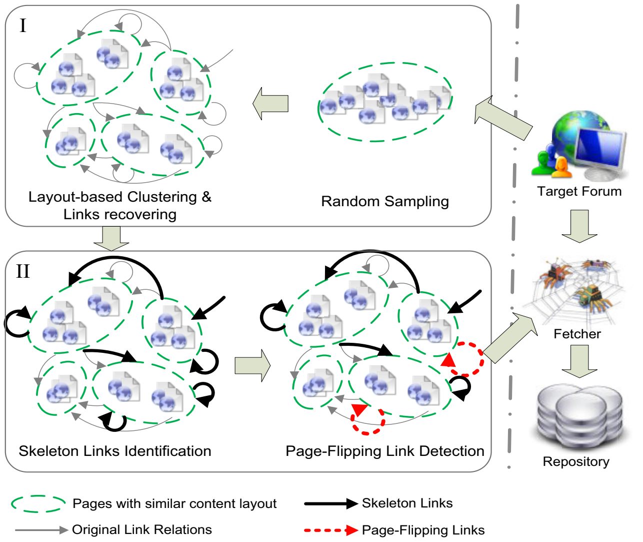
I
Layout-based Clustering & Links recovering
Random Sampling
Target Forum
II
Skeleton Links Identification
Page-Flipping Link Detection
Fetcher
Repository
Pages with similar content layout
Skeleton Links
Original Link Relations
Page-Flipping Links

Fig. 2. The flowchart to the proposed approach, which consists of two parts: (I) sitemap recovering; and (II) traversal strategy exploring

PAGE-FLIPPING LINK. Page-flipping links are a kind of *loop-back* links in the sitemap. That is, following such kind of link we can reach another page in the same vertex in the sitemap graph. Correctly dealing with page-flipping links has the following advantages which were ignored in previous works: 1) it enables the crawler to completely download a discussion thread. A long thread may consists of tens or even hundreds of pages, most of which are missed in a generic crawling as their link depths are too deep; and 2) it also provides useful information to rank forum data. The number of flipping pages can be used to approximate the popularity (how many replies) of a discussion thread. However, it should be noticed that not all loop-back links are page-flipping links. Fig. 1 (c) shows such an example. In Fig. 1 (c) there are two kinds of loop-back links: one is the "previous" and "next" buttons marked with the orange square, which actually are shortcuts pointing to the posts of another two discussion threads; the other is the digital sequence marked with the purple ellipse, which are real page-flipping links pointing to the rest of posts in the same thread. Therefore, we need a method to detect page-flipping links from other loop-back links in the sitemap.

4. SYSTEM OVERVIEW

The flowchart of our method is illustrated in Fig. 2, which mainly consist of two parts: (I) sitemap recovering and (II) traversal strategy exploring.

The goal of the first step is to estimate the sitemap structure of the target forum using a few sampled pages. Therefore, the sampling quality is the foundation of the whole mining process. To keep the sampled pages as diverse as possible in terms of page layout and to retrieve pages at deep levels, we adopted a combined strategy of breadth-first and depth-first using a double-ended queue. In the implementation, we tried to push as many as possible unseen URLs from each crawled page to the queue, and then randomly pop a URL from the front or the end of the queue for a new sampling. In practice, it was found that sampling a few thousands pages is enough to re-construct the sitemap for most forum sites. After that, pages with similar layout structures are further clustered into groups (*i.e.* vertices) using the single linkage algorithm, as marked with green dashed ellipses in Fig. 2. In our approach, we utilized the *repetitive regions* to characterize the content layout of each page. Repetitive regions are very popular in forum pages to present data records stored in a database. Considering that two similar pages may still have different numbers of advertisements, images, and even some complex sub-structure embedded in user posts, the repetitive region-based representation is more robust than the whole DOM tree [15]. Finally, all possible links among various vertices are established, if in the source vertex there is a page having an out-link pointing to another page in the target vertex. As aforementioned, each link is described by both the URL pattern and the location (the region where the corresponding out-link is located). For more details of this step, please refer to our previous work in [7].

The second part is in charge of exploring an optimal traversal strategy on the constructed sitemap. First, through the module of *skeleton link identification*, the links pointing to redundant and uninformative pages are discarded and only the links pointing to valuable pages are preserved, as marked with the dark arrows in Fig. 2. Then, all the possible loop-back links of page-flipping, as marked with the red dashed arrows in Fig. 2, are detected from the skeleton links through the module of *page-flipping link detection*. This indicates that a page-flipping link must be a skeleton link first. The methods in this step are discussed in detail in the following section.

5. EXPLORING TRAVERSAL STRATEGY

In this section, we present the algorithm details for exploring the optimized traversal strategy in the sitemap. First, the skeleton links are identified from the sitemap to ensure an efficient crawling. Second, the page-flipping links are detected to restore a long discussion thread which is shown with multiple pages.

5.1 Skeleton Link Identification

To indentify skeleton links, first we summarize some common characteristics of skeleton links, and then propose an appropriate strategy to distinguish skeleton links from others.

5.1.1 Criteria of Skeleton Links

After investigating the organization structures of a substantial number of various Web forums, we found that most skeleton links have the following two characteristics.

- Skeleton links should point to those vertices containing valuable and informational pages, such as the *list-of-thread* pages and *post-of-thread* pages. If one skeleton link is missed, the crawler will miss a number of valuable pages. In other words, a complete set of skeleton links is the necessary condition to retrieve all the valuable pages in a forum.
- Skeleton links should not introduce any redundant crawling or valueless information. In other words, a link is not a skeleton link if it points to many duplicate pages or valueless pages (such as a login failure prompt page).

Therefore, the characteristics of a skeleton link actually depend on the behaviors of its target vertex. To quantitatively measure the above two characteristics, in this work, we propose two criteria based on the results of near-duplicate detection. This is because duplicate means redundant information in crawling. Moreover, valueless pages in forum sites are usually a group of pages which are completely duplicated with each others. For example, all the login prompt pages returned by a forum are exactly the same page. In our approach, the content-based near-duplicated detection algorithm [10][11] is employed. In the implementation, each Web page is first cleaned by removing all the HTML and script tags and is then characterized with fingerprints such as Shingles [6] or SimHash [11]. At last, any two pages with a small L_2 distance in the fingerprint space are considered as duplicates.

For each vertex in the sitemap, assuming that the number of sampled pages in this vertex is *N*, and we finally get *K* unique pages after the near-duplicate detection. The two criteria, *coverage* and *informativeness*, are defined as follows.

1) Retrieving all the *K* unique pages is the ideal result we want to achieve. However in real crawling, usually the crawler can only download some of them. Assuming there are K' unique pages retrieved ($K' \leq K$), the *coverage* is defined as:

$$Cov = \frac{K'}{K} \times 100\% \qquad (1)$$

The higher Cov is, the more valuable pages are crawled.

2) Suppose D_i is the set of all the duplicates of the i^{th} ($i = \{1,2,\ldots,K\}$) unique page in the vertex (including the i^{th} unique page itself), and $\|D_i\|$ denotes the number of pages in D_i. Then, the *informativeness* can be defined as:

$$Info = -\frac{1}{log(N)}\sum_{i=1}^{K}\frac{\|D_i\|}{N}log\left(\frac{\|D_i\|}{N}\right) \qquad (2)$$

For the case that all the pages are duplicated with each other, *K*=1, $\|D_i\| = N$, and *Info* = 0, which means there is no information in these pages. In contrast, *Info* will reach the maximum value 1 when there are no duplicates.

With these two criteria, we can evaluate each link independently. However, considering that the choosing of a link (or a set of links) may affect the evaluation of other links, it still cannot guarantee a global optimization only with the above criteria. We still need a strategy to search the skeleton links from a global perspective.

5.1.2 Search for Skeleton Links

Typically, a user usually starts browsing a forum from its portal, follows one link on the portal to visit a page in the second level, and then navigates to pages in deeper levels. Inspired by this observation, the search process is carried out according to the depths of vertices in the sitemap, as shown in Fig. 3. That is, vertices are investigated one by one from vertices in top levels to vertices in deeper levels. The processed vertices are put into the set ***PV***, and the unprocessed vertices are in the set ***UV***. For each vertex in ***UV***, all the links from the vertices in ***PV*** pointing to this vertex (including its loop-back links) are the candidates of skeleton links for this vertex, denoted as $\{L_1,\ldots,L_m\}$, which is also the input of the procedure *of GetSkeletonLinks*.

The goal of *GetSkeletonLinks* is to find out a combination of links from $\{L_1,\ldots,L_m\}$, which maximizes the performance given by the above two criteria. Considering each L_i may be selected as a ske-

```
Identify the skeleton links in the sitemap
input: an unprocessed vertices set $UV$={all the vertices in the sitemap}, a processed vertices set $PV=\phi$;
output: a set of skeleton links $SL$;
begin
    $SL=\phi$;
    while ($UV \neq \phi$) do
        $V \leftarrow$ the top most level vertex in $UV$;
        $UV \leftarrow UV - \{V\}$ and $PV \leftarrow PV \cup \{V\}$
        Let $\{L_1, \ldots, L_m\}$ be the links from a vertex in $PV$ pointing to $V$;
        $SL \leftarrow SL \cup$ GetSkeletonLinks($\{L_1, \ldots, L_m\}$);
    end for while
end
```

Fig. 3. The procedure of skeleton link identification.

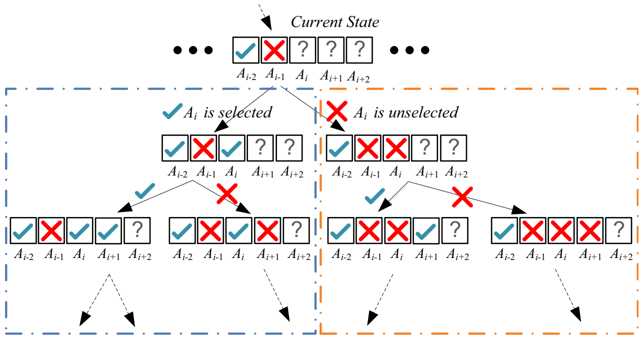

Fig. 4. An illustration of the search process of skeleton links.

leton link or not, the search space consists of 2^m states. Actually, the search space can be represented as an m-level binary tree, in which the two sub-trees under a node denote the two decisions (selected or not), as the two dashed boxes shown in Fig. 4. To accelerate the search process, in practice a pruning is adopted with the following two rules: (1) if a link is selected but causes a significant drop of informativeness, it shouldn't be a skeleton link. The sub-tree in which the link is selected can be pruned; (2) if a link is rejected but causes a significant drop of coverage, it should be a skeleton link. The sub-tree in which the link is rejected can be pruned. If neither (1) nor (2) are satisfied, the algorithm just keeps both of the two decisions and investigates the rest search space recursively. The detailed algorithm of *GetSkeletonLinks* is shown in Fig. 5.

5.2 Page-Flipping Link Detection

Page-flipping link is a kind of loop-back links of a vertex. But not all the loop-back links are page-flipping links. For example, in the vertex of *post-of-thread*, there are usually two kinds of loop-back links. One is the page-flipping link which connects multiple pages belonging to one thread, as the purple ellipse shown in Fig. 1 (c). The other is the "Previous/Next" navigation link which points to the first page of other threads, as the orange rectangle shown in Fig. 1 (c). The goal of page-flipping link detection is to distinguish the page-flipping link from other loop-back links.

In this paper, we found a special characteristic for page-flipping links. That is, for page-flipping links, if there is a path from a page A to another page B, there must be a path from B to A. While for other loop-back links, they cannot guarantee such a characteristic. For example, we can browse a thread from its first page A_1 to its n^{th} page A_n, and can also go back from A_n to A_1 only through the page-flipping link. However, for "Previous/Next" navigation links,

```
GetSkeletonLinks($\{L_1, \ldots, L_m\}$)
    input: links set $\{L_1, \ldots, L_m\}$
    $CS = \{CS_1, CS_2, \ldots, CS_m\}$ is the current state of the links
    output: a set of skeleton links;
    Begin
        $Cov_{Max} = 0$; $Info_{Max} = 0$; // to record the coverage and informativeness of $State_{best}$
        Define $State_{best}$ to record the best state of links in the search history;
        $CS_i \leftarrow$ selected, for all $1 \le i \le m$;
        JudgeSkeletonLink(1);
        return the selected links in $State_{best}$;
    End

JudgeSkeletonLink($i$)
    input: $i$ is the subscript of the links in $\{L_1, \ldots, L_m\}$;
    $TH_{Cov}$ is the coverage threshold and $TH_{Info}$ is the informativeness threshold;
    Begin
        if ($i > m$)
            if ( $CS$ is better than $State_{best}$ )
                $State_{best} \leftarrow CS$;
                $Cov_{Max} \leftarrow Cov_{cur}$;
                $Info_{Max} \leftarrow Info_{cur}$
            exit;
        Endif
        for ($CS_i$ in {unselected, selected})
            $Cov_{cur} \leftarrow$ coverage of $CS$;
            $Info_{cur} \leftarrow$ informativeness of $CS$;
            if ($Cov_{cur} < TH_{Cov} * Cov_{Max}$
                or $Info_{cur} < TH_{Info} * Info_{Max}$) continue;
            JudgeSkeletonLink($i$+1);
        Endfor
    End
```

Fig. 5. The pseudo code of *GetSkeletonLinks.*

if we jump from A_n to another page B_1 in another thread, we cannot go back from B_1 to A_n only with navigation links–we can only first jump to A_1 and follow the page-flipping link to go back to A_n. In other words, using the navigation link there is only a path from A_n to B_1, but no path from B_1 to A_n.

Based on this observation, for each kind of loop-back link, we define a measurement called "connectivity", as:

$$Connectivity = \frac{\sum_{\{A,B\}} Path(A,B) \cdot Path(B,A)}{\sum_{\{A,B\}} Path(A,B)} \qquad (3)$$

where A and B are any two pages in the vertex. If there is a path from A to B following only this kind of link, $Path(A,B) = 1$; otherwise $Path(A,B) = 0$. With this definition, page-flipping link will receive evidently larger "connectivity" score than other loop-back links. In the experiments, we will exhaustively compare this characteristic between page-flipping links and other loop-back links on various forum sites.

In practice, for each vertex, we first estimate the connectivity of each loop-back link as defined in Eq. 3. Meanwhile, the average connectivity score of all the loop-back links on this vertex is also estimated. Finally, the loop-back links whose connectivity is higher than the average score are selected as the page-flipping links for this vertex.

Table 1. Web Forums in the Experiments

Web Forums	Description
http://www.biketo.com/bbs/	A bike forum (in Chinese)
http://forums.asp.net/	A commercial technical forum (in English)
http://post.baidu.com/	The largest Chinese community (in Chinese)
http://bbs.cqzg.cn/	A general forum (in Chinese)
http://forums.gentoo.org/	Gentoo Linux forum (in English)
http://forums.microsoft.com/MSDN/	Forum of Microsoft software developer network (in English)
http://www.photozo.com/forum/	Digital photography forum (in English)
http://forums.afterdawn.com/	CD, DVD and video forum (in English)

6. EXPERIMENTS AND RESULTS

In this Section, we present the experimental results of the proposed system, including the performance analysis of our system, and some comparisons with a generic crawler, the structure-driven crawler in [15], in terms of both the effectiveness and efficiency.

6.1 Experiment Setup

To evaluate the performance of our system on various situations, eight different forums were selected in diverse categories (including bike, photography, travel, computer technique, and some general forums) in the experiments, as listed in Table 1.

Since the structure-driven-based approach in [15] seems promising and is more relevant to our work, we also include it in the experiments. The original structure-driven-based approach only crawl one kind of target pages, and therefore requires one sample page of the target pages. But there are usually multiple kinds of valuable pages in a forum site. We have tried our best to adapt the structure-driven-based approach to forum crawling by: 1) manually choosing one sample page from each vertex since pages in the same vertex have similar layout; and 2) adjusting the threshold of URL pattern generation in the structure-driven-based approach.

To set up a consistent data collection for further evaluation and comparison, we first mirrored the above eight sites using a commercial search engine crawler. The crawler was adjusted to be domain-limited, depth-unlimited and never dropping pages. For each site, the crawler starts from the homepage and follows any links belonging to that domain; and a unique URL address will be followed only once. For each forum we kept the first 30,000 correctly crawled pages, which is large enough to cover all kinds of pages in that forum and can be considered as a sketch of the original site. The following crawling experiments, including our method, the structure-driven crawler, and the generic crawler, were all simulated on this data collection.

Moreover, for a quantitative evaluation, we also manually labeled the traversal strategy as the baseline. Since there are quite few performance differences between the baseline and the strategy by our automated exploring method, we will not compare them further in the later part.

6.2 Evaluation of Skeleton Link Identification

The performance analysis here includes: 1) the crawling quality; and 2) the crawling effectiveness and efficiency.

6.2.1 Crawling Quality

First, as introduced in Section 4 and Section 5, for each site we randomly sampled 5,000 pages to build the sitemap and generate the traversal strategy. Then we used our method and the structure-driven crawler to simulate the crawling of 30,000 pages on the mirrored dataset which is treated as a sketch of the original sites.

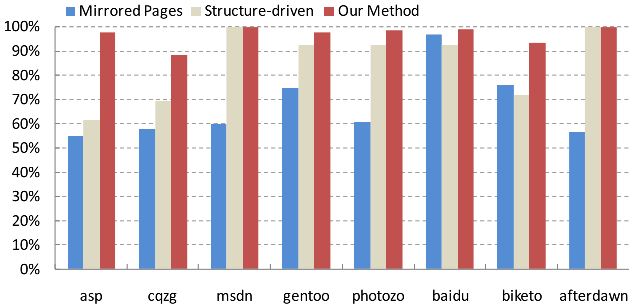

Fig. 6. The comparison of informativeness for the mirrored pages and, the pages crawled by our method and the structure-driven crawler, on various forums, respectively.

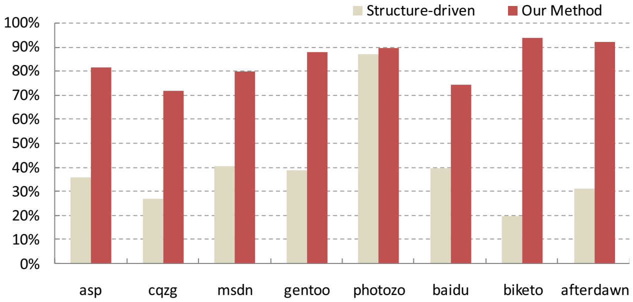

Fig. 7. The coverage of valuable pages retrieved by the structure-driven crawler and our method.

We first evaluated the informativeness of results using different methods. Fig. 6 illustrates the comparison of our method, the structure-driven crawler and the mirrored sites crawled by the general crawler. Informativeness is defined in Eq. (2) in Section 5.1.1. It is clear that our method can effectively increase the informativeness by avoiding duplicate pages in the original mirrors. On average, the informativeness score by our method can reach 0.97; for the structure-driven crawler the number is 0.85; while for mirrored pages the number is 0.67. Moreover, it is worth noting that our method can significantly increase informativeness while keeping a high coverage ratio. As shown in Fig. 7, more than 83% valuable pages can be visited by our method. But there are only 40% valuable pages that can be visited by the structure-driven crawler.

The structure-driven crawler drops both the coverage and informativeness because the traversal strategy for a forum site is generally much more complex than that for a traditional web site. There are several reasons. First, even to crawl one kind of target pages, multiple traversal paths are required in forum sites. For example in the "Asp.Net" forum, the *list-of-thread* pages can be crawled from *list-of-board* pages. But there are a lot of successive *thread list pages* can be only crawled from previous *list-of-thread* pages by page-flipping links. But the structure-driven crawler only considers one traversal path for each target. Second, the structure-driven crawler describes a traversal path based on URL patterns which are usually not robust for forum sites. Such as in the forums "CQZG" and "Biketo", the structure-driven crawler cannot generate reasonable URL patterns by its automatic method to distinguish valuable pages from duplicate pages and will drop the informativeness of results.

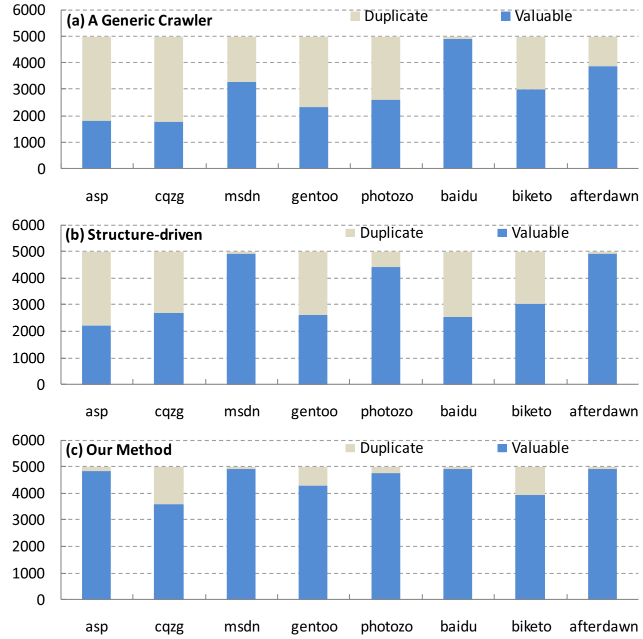

Fig. 8. The comparison of effectiveness between (a) the generic crawler, (b) the structure-driven crawler and (c) our method, on the top 5,000 retrieved pages.

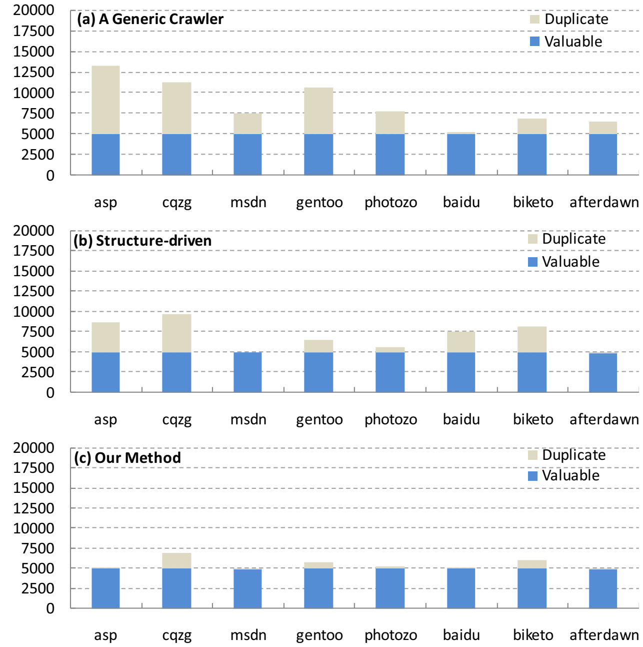

Fig. 9. The comparison of efficiency between (a) the generic crawler, (b) the structure-driven and (c) our method, to retrieve 5,000 valuable pages.

From Fig. 6 and Fig. 7, we can find that the informativeness scores of various forums are quite different by the generic crawler. Some forums supported by commercial companies are well designed and have less valueless pages, such as Baidu. Most forums developed by commercial organizations are of this style. In contrast, some forums like CQZG are of lower informativeness due to many duplicate pages. Also, forums with restricted access controls are often of lower informativeness, such as the Asp.Net forum.

In contrast to the generic crawler and the structure-driven crawler, our method can achieve promising performance on all these forums. We also noticed that there are still some problems that should be improved. First, we should better balance the tradeoff between guaranteeing coverage and removing useless pages. Taking CQZG as an example, its coverage is somewhat harmed although its informativeness significantly increases. Second, we need more evidences to remove duplicates. In the current implementation, the duplicate detection is mainly based on the algorithm introduced in Section 5.1, which may be too simple to handle very complex situations.

6.2.2 *Crawling Effectiveness & Efficiency*

Effectiveness and efficiency are two important criteria to evaluate a Web crawler. Effectiveness means that given a number of retrieved pages, how many of them are valuable and informative. Effectiveness is very important for saving network bandwidth and storage space. Efficiency means how fast a crawler can retrieve a given number of valuable pages. Efficiency is important as it determines how quickly a crawler can update its repository and indexing. In this subsection, we compare our method with the generic crawler and the structure-driven crawler, in terms of both effectiveness and efficiency.

To evaluate the effectiveness, we retrieved additional 5,000 pages from the mirrored sites using the generic crawler, the structure-driven crawler and our method, respectively. We can find out how many valuable and duplicate pages were visited by the three crawlers, as shown in Fig. 8 (a), (b) and (c). In the top 5,000 pages crawled by the generic crawler, the ratio of valuable pages is around 59%, averaged across the eight forums. The average ratio is 69% for the structure-driven crawler. In comparison, the average ratio of valuable pages crawled by our method is 91%, which is a significant improvement over the other two methods. In other words, given a fixed bandwidth, our method can crawl almost 1.53 times valuable pages than a generic crawler and 1.32 times valuable pages than the structure-driven crawler.

To evaluate the efficiency, we continually crawled each mirrored site until additional 5,000 valuable pages are retrieved. Then we investigate how many pages have to be downloaded respectively using the generic crawler, the structure-driven crawler and our method. The results are shown in Fig. 9 (a), (b) and (c). From Fig. 9, it can be seen that to archive 5,000 valuable pages, a generic crawler averagely needs to crawl about 8700 pages; the structure-driven crawler needs to crawl about 7100 pages. In contrast, our method only needs to fetch about 5500 pages. Consequently, supposing a constant downloading time for each page, to archive the same number of valuable pages, a generic crawler will require 1.57 times crawling time than our method and the structure-driven crawler will require 1.40 times crawling time than our method.

6.3 Evaluation of Page-Flipping Detection

In this section, we evaluate the performance of the page-flipping link detection. First of all, we consider all the loop-back links in a site as the candidates for page-flipping links. Then we detect page-flipping links among all the candidates by measuring their connectivity scores according to the algorithm presented in Section 5.2.

To evaluate the distinguish ability of our method on different size of dataset and find an appropriate size; we repeated the measuring

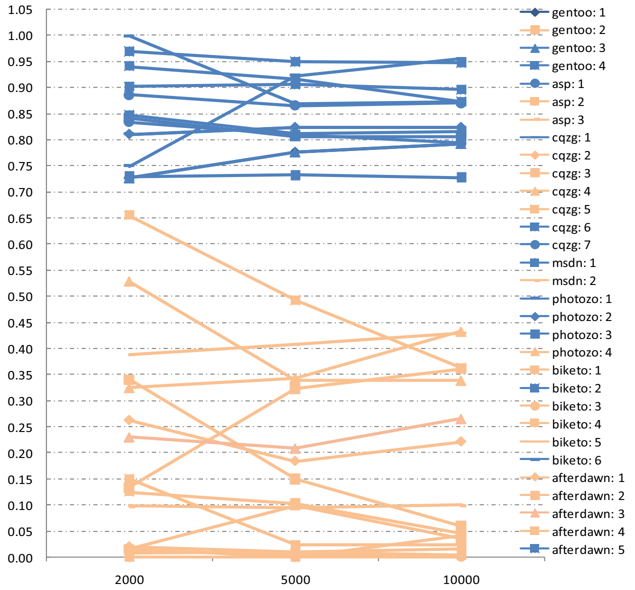

Fig. 10. The connectivity scores of all the loop-back links with different numbers of sampled pages. Page-flipping links are marked in blue and non page-flipping links are in orange.

process for each link by varying the number of sampled pages. There are 31 candidate loop-back links in all forum sites. The connectivity scores of these loop-back links are shown in Fig. 10. We manually labeled all candidates and mark in Fig. 10 the 14 page-flipping links in blue and the other 17 non page-flipping links in orange. It can be seen that the difference between page-flipping links and non page-flipping links are distinct using 5,000 pages and the difference becomes more apparent using larger datasets. With 5,000 sampled pages for loop-back links of all sites, the minimal connectivity score for page-flipping links is higher than 70% while the maximal score for non page-flipping links is lower than 50%. The results show that the proposed criterion can accurately detect the page-flipping links from other loop-back links.

7. CONCLUSION AND FUTURE WORK

In this paper, we have presented a complete solution to automatically explore an appropriate traversal strategy to direct the crawling of a given target forum. Traditional generic crawlers which simply adopt the breadth-first traversal strategy are usually inefficient due to the complex in-site link structures and login controls in most forum sites. To understand the global organization of a forum, we proposed to first re-construct its *sitemap* based on a few thousands pages randomly sampled from the target site. In this way, we can leverage more global knowledge to explore the traversal strategy. The proposed solution mainly consists of the identification of *skeleton* links and the detection of *page-flipping* links. The skeleton links instruct the crawler to only crawl valuable pages and meanwhile avoid duplicate and uninformative ones; and the page-flipping links tell the crawler how to completely download a long discussion thread which is usually shown in multiple pages in Web forums. To evaluate the proposed traversal strategy, we conducted extensive experiments on several forums. The experimental results demonstrated promising performance in terms of both efficiency and effectiveness. Following the discovered traversal strategy, our forum crawler can efficiently archive more informative pages and restore a long discussion thread which is divided into multiple pages, which is impossible in most previous works.

In the future, we will study how to optimize the crawling schedule to incrementally update the archived forum content, and how to parse the crawled forum pages to separate replies in each post thread. This will further structuralize forum pages, and enable many interesting and promising applications based on forum content mining.

8. REFERENCES

[1] Internet Forum. http://en.wikipedia.org/wiki/Internet_forum.

[2] S. Abiteboul, M. Preda, and G. Cobena. Adaptive on-line page importance computation. In *Proc. 12th WWW*, pages 280-290, Budapest, Hungary, May 20-24, 2003.

[3] R. Baeza-Yates and C. Castillo. Crawling the infinite Web: five levels are enough. In *Proc. 3rd Workshop on Algorithms and Models for the Web-Graph*, *LNCS*, volume 3243, pages 156-167, Rome, Italy, Oct. 16, 2004.

[4] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez. Crawling a Country: better strategies than breadth-first for Web page ordering. In *Proc. 14th WWW*, pages 864-872, Chiba, Japan, May 10-14, 2005.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107-117, 1998.

[6] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. In *Proc. 6th WWW*, pages 1157–1166, Santa Clara, California, USA, Apr. 1997.

[7] R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang. iRobot: An Intelligent Crawler for Web Forums. In *Proc. 17th WWW*, pages 447–456, Beijing, P.R. China, April 21-25, 2008.

[8] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11-16):1623−1640, 1999.

[9] Y. Guo, K. Li, K. Zhang, and G. Zhang. Board forum crawling: a Web crawling method for Web forum. In *Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intelligence*, pages 745−748, Hong Kong, Dec. 2006.

[10] M. Henzinger. Finding near-duplicate Web pages: a large-scale evaluation of algorithms. In *Proc. 29th SIGIR*, pages 284−291, Seattle, Washington, USA, Aug. 2006.

[11] G. S. Manku, A. Jain, and A. D. Sarma. Detecting near-duplicates for web crawling. In *Proc. 16th WWW*, pages 141−150, Banff, Canada, May 8-12, 2007.

[12] F. Menczer, G. Pant, P. Srinivasan, and M. E. Ruiz. Evaluating topic-driven Web crawlers. In *Proc. 24th SIGIR*, pages 241-249, New Orleans, LA, USA, Sept. 9-12, 2001.

[13] S. Pandey and C. Olston. User-centric Web crawling. In *Proc. 14th WWW*, pages 401-411, Chiba, May 10-14, 2005.

[14] S. Raghavan and H. Garcia-Molina. Crawling the hidden Web. In *Proc. 27th VLDB*, pages 129-138, San Francisco, CA, USA, Sept. 11-14, 2001.

[15] M. L.A. Vidal, A. S. da Siva, E. S. de Moura, and J. M. B. Cavalcanti. Structure-driven crawler generation by example. In *Proc. 29th SIGIR*, pages 292-299, Seattle, Washington, USA, Aug. 6-11, 2006.