ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, Supplement, December 2009, pp.176−183

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种高效的动态脚本网站有效页面获取方法*

夏 冰, 高 军+, 王腾蛟, 杨冬青

(北京大学 信息科学技术学院,北京 100871)

An Efficient Valid Page Crawling Approach for Websites with Dynamic Scripts

XIA Bing, GAO Jun+, WANG Teng-Jiao, YANG Dong-Qing

(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

+ Corresponding author: E-mail: gaojun@pku.edu.cn

Xia B, Gao J, Wang TJ, Yang DQ. An efficient valid page crawling approach for websites with dynamic scripts. *Journal of Software*, 2009,20(Suppl.):176−183. http://www.jos.org.cn/1000-9825/09021.htm

Abstract: In times of Web 2.0, more and more websites adopt dynamic scripts for user interaction, and the switches between pages are no longer all based on the "<a>" tags and the URL is no longer the unique identification of a Web page. Traditional Web crawlers can't deal with Web pages containing dynamic scripts, as a result, search engines, such as Google, give up these Web pages. The research on crawling website with dynamic scripts is still in the early stage. This paper proposes an efficient valid page crawling approach for websites with dynamic scripts. Firstly, by training the paper can get the events and the Web elements that triggered the events, which would lead the people to desired Web pages. Then, the paper generates the XPath patterns of these elements and record the events the people need to trigger. During crawling, the paper only considers these event and element combinations for accelerating the crawling. Additionally, the paper demonstrates the efficiency and the effectiveness of the approach by extensive experimental evaluation.

Key words: dynamic scripts; AJAX; page similarity; XPath; Web crawler

摘 要: 随着 Web2.0 时代的到来,越来越多的网站采用了动态脚本的方式与用户进行交互.页面的转换不再仅仅通过点击"<a>"标签进行,URL 也不再是页面的唯一标识.传统网络爬虫无法应对含动态脚本的网页,如 Google 等搜索引擎即对这些网页采取回避的态度.对这些网页的抓取方法的研究仍处在起步阶段,提出了一种高效的动态脚本网站有效页面的获取方法,首先通过训练获得哪些页面元素触发的哪些事件将引向我们所需的页面,并总结出这些页面元素的 XPath 特征及触发的事件类型.在以后的抓取中,只触发这些页面元素上的特定事件,从而提升抓取效率.此外,通过实验证明了我们方法的效率和性能.

关键词: 动态脚本;AJAX;页面相似性;XPath;网络爬虫

随着 Web 2.0 时代的到来,越来越多的网站采用了动态脚本的方式与用户进行交互,其中就包括了

* Supported by the National Natural Science Foundation of China under Grant No.60873062(国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2009AA01Z150, 2007AA01Z191, 2006AA01Z230 (国家高技术研究发展计划(863)); the Peking University-Morgan Stanley Research Aid Program (北京大学-摩根士丹利研究资助项目)

Received 2009-05-01; Accepted 2009-07-20

AJAX(Asynchronous JavaScript and XML)技术.使用这样的方式,一方面服务器并不需要存储大量的静态网页,而是在用户请求时直接从数据库里取出数据动态生成网页返回给用户,节约了存储空间,并增强了灵活性;另一方面,大量 AJAX 技术的运用使得在浏览器与服务端交互时,用户仍然可以对浏览的页面进行操作,增强了用户体验.

网络爬虫是搜索引擎,数据分析等工作的基础.但是,上述技术的运用使传统的网络爬虫失去了作用.传统的网络爬虫的工作原理是获取一个页面的 HTML 源码,抽取出其中的"<a>"标签里的"href"属性指向的 URL 链接并跟踪链接.然而,在 Web 2.0 环境下,动态脚本网站的抓取面临着 3 个新的挑战:

(1) 一个页面的 HTML 源码并不就是最终的页面内容,浏览器可能还需要额外执行一些 JavaScript,这时页面内容会发生改变;

(2) "<a>"标签的"href"属性不再重要甚至无效,新的链接经常是通过 JavaScript 控制打开,而且" ",""等其他标签也能够有以前"<a>"标签的链接效果;

(3) URL 不再是一个页面的唯一标识,对 AJAX 网站而言,多页面可以共享一个 URL.

目前,Google,Yahoo!,百度等知名搜索引擎对动态脚本网站均无法正常获取,除非这些网站给它们预留传统爬虫的接口.而对于 AJAX 网站,由于这些搜索引擎都是按 URL 来组织的,更是无法处理.随着这一类型网站的逐渐增多,越来越多的有效信息涵盖在这些网站中,如新浪,搜狐,腾讯等主流门户网站的论坛,博客等均为此类网站.我们迫切需要一种有效的抓取动态脚本网站的方法.

通常来说,动态脚本网站的网页是从数据库里面读取数据然后嵌入特定的模板生成的,因此同一网站同一类型的网页具有很高的结构相似性.图 1 所示为网易三篇不同新闻的评论页面.网易新闻的评论使用了 AJAX 技术,评论的内容是页面 Load 过程中动态载入的,评论翻页时 URL 也不会发生改变.从图中我们可以看到它们的结构极为相似.通过页面相似性,我们便可以判定哪些页面是我们需要抓取的页面.

Fig.1 News comment pages of NetEase
图 1 网易新闻评论页面

本文针对动态脚本网站带来的挑战,做出了如下一些贡献:

(1) 为动态脚本网站建立了一个状态转换图模型,不再将 URL 视为页面的唯一标识.相应的,页面之间的转

换也不是跟踪“<a>”标签的“href”属性,而是采用了模拟用户行为触发事件的办法来改变.这样“ ”,“”等各种标签上 “onclick”,“onmouseover”等事件绑定的 JavaScript 函数均能够触发,从而获取到我们所需的数据;

(2) 提出了利用 XPath 归约有效页面元素的方法,所谓有效页面元素就是其上的事件能够将我们引向需要页面的页面元素.由于 URL 不能作为特征,本文在页面元素的级别上进行训练,归约出有效页面元素的 XPath 特征,使得在以后的抓取过程中只需触发特定页面元素上的特定事件,极大地减少了抓取过程中需要触发的事件数,提高了抓取效率.这一贡献是本文的独创性工作.

(3) 提出了基于结构相似性的有效页面判定,使用树的编辑距离来测量页面的相似性,从而判定出哪些页面是有效的页面.在训练过程中,将对所有页面元素上所有可能事件进行触发,对改变后的页面进行有效页面判定,并归约出引向有效页面的页面元素的 XPath 特征.

本文第 1 节介绍动态脚本网站抓取和页面相似性判定方面的相关工作.第 2 节阐述本文给出的方法的基本原理.第 3 节对应用本文给出的方法的具体算法进行描述.第 4 节通过实验验证本文给出的方法的优越性.最后进行总结并指出下一步的研究工作.

1 相关工作

动态脚本网站抓取是一个比较新的领域,文献[2]提出了可以分析所有页面元素,看它们哪些事件绑定了哪些 JavaScript 函数,然后依次执行这些 JavaScript 函数来进行抓取的方法.这需要我们对 JavaScript 有很深的理解,并对其调用过程进行细致的分析.然而,分析页面元素的事件绑定函数是一件很困难的事情,并且事件是可以动态绑定函数的.此外,他们提出的主要优化是 cache 直接向服务器请求数据的 JavaScript 函数及其参数,若再次以相同参数调用此函数则返回 cache 的数据.且不说全局变量,this 指针和 cookie 等会对 JavaScript 函数的执行效果产生影响,它也没有考虑如何避免获取不相关数据这一问题.

文献[1]提出了模拟用户行为对页面元素进行操作,本文也采用了这样的方式来触发事件.但是他们并没有考虑如何智能地避免获取不相关数据这一问题,他们定义了一种语言让用户可以指定触发哪些页面元素的哪些事件,这需要用户自己对页面源码进行深入分析.若使用全自动的方式,抓取的时间代价是不可接受的.

文献[1,2]均针对 AJAX 网站建立了自己的模型,文献[1]的模型仍需要对 JavaScript 深入分析;文献[2]的模型采用了流图,从后面的状态无法返回到前面的状态,使得从一个状态转换到另一个状态的路径偏长.本文中提出的状态转换图模型对所有动态脚本网站均适用,且缩短了状态之间转换的距离.

利用页面相似性抓取关注的页面在文献[3]里被提出,但其未涉及到页面元素的级别,方法只是抽取出相似页面的 URL 特征,根据抽取出的 URL 特征去抓取页面.而首先,网站的 URL 并不一定具有那么强的特征;另外,对动态脚本网站而言,URL 不一定是唯一的标识,我们不能根据 URL 特征进行抓取.

本文利用页面相似性训练得到哪些页面元素的哪些事件可以引向我们需要的页面,然后归约出这些页面元素的 XPath 特征.在以后的抓取过程中将只选取这些 XPath 的页面元素,触发其上的特定事件,以加快抓取速度.就对页面元素进行总结以及归约出 XPath 作为特征描述来加速抓取而言,还没有发现有类似的工作.

2 基本原理

本文针对动态脚本网站,首先为其建立了一个状态转换图模型来进行描述;然后利用页面相似性进行训练,归约出需触发事件的页面元素 XPath 特征及需触发的事件类型;最后将训练的结果应用于动态脚本网站的抓取,只触发特定页面元素上的特定事件,从而加快抓取速度.

2.1 动态脚本网站

动态脚本网站是应用了 JavaScript,VBScript 等脚本技术的网站.如前面动态脚本网站带来的挑战性里提出,动态脚本网站通常具有 3 个特征:

(1) 页面初始载入后还需要跟服务器多次交互才能得到最终页面.以网易新闻评论页面为例,载入时并不

含评论的内容,而是后来由 JavaScript 控制向服务器再次请求得到评论内容.

(2) 页面的转换不仅是由“<a>”标签控制,任意页面元素上的“onclick”,“onmouseover”等各种事件均可以引起页面的改变.仍以网易新闻评论页面为例,评论的翻页即是由“onclick”事件绑定的 JavaScript 函数来控制.

(3) 页面改变,URL 不一定发生变化.在网易新闻评论页面中,评论的翻页均不改变页面的 URL.

2.2 状态转换图模型

由于对动态脚本网站而言,URL 不能作为一个页面的唯一标识,本文建立了一个状态转换图模型来描述动态脚本网站.浏览过程中,用户的每个行为都可能对 DOM 树产生影响,我们将用户浏览过程中的每棵 DOM 树视为一个状态.

定义 1. 一个动态脚本网站的状态转换图为一个二元组$\langle \boldsymbol{V},\boldsymbol{E}\rangle$,其中$\boldsymbol{V}$是表示状态的节点的集合,每个节点$v\in\boldsymbol{V}$表示该网站抓取过程中的一个状态;$\boldsymbol{E}$是节点间的边的集合,每条边是一个二元组$\langle event,xpath\rangle$,$v_1$到$v_2$的有向边$\langle event,xpath\rangle$存在当且仅当$v_1$可以通过触发xpath代表的页面元素上的事件event转换到状态v_2.

图 2 是一个这样的状态转换图的例子.状态 S1 下,点击“/html/body/div[1]/a[1]”代表的页面元素可以转换到状态 S2,鼠标移到“/html/body/div[2]/div[1]”代表的页面元素上可以转换到状态 S3;状态 S2 下,点击“/html/body/div[1]/a[2]”代表的页面元素可以回到状态S1,鼠标移到“/html/body/span[1]”代表的页面元素上可以转换到状态 S3.

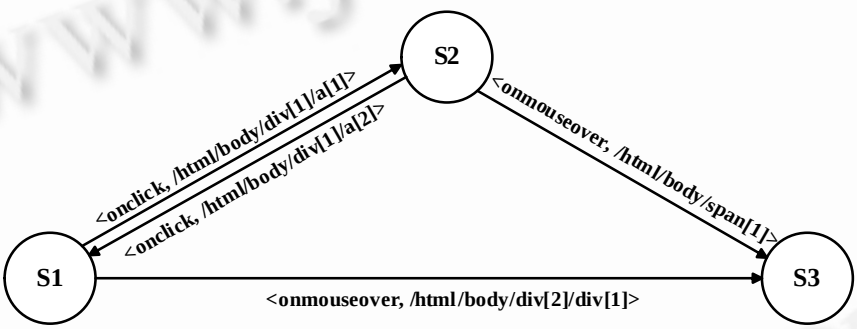

Fig.2 A state transition diagram example

图 2 状态转换图示例

2.3 相似性判定

一个页面可以用一棵 DOM 树来表示,我们使用树的编辑距离作为比较两个页面相似的度量.直观地说,两棵树的编辑距离是将一棵树变换到另一棵树的最小编辑操作的代价.显然,两棵树的编辑距离越小,它们越相似.

通常来说,树之间的编辑操作考虑 3 种情况:(a) 删除一个节点;(b) 插入一个节点;(c) 替换一个节点.每种操作都会有一个代价.将一棵树变换为另一棵树需要一组操作,其中代价最小的一组操作就是这两棵树的编辑距离.

本文采用了一种叫做 RTDM 的算法来计算两个页面的 DOM 树之间的编辑距离.RTDM 算法被证明是计算 DOM 树之间的编辑距离的有效的算法,它引入了受限的自顶向下映射的概念,将求两棵树之间的编辑距离转为求两棵树之间的受限的自顶向下映射的代价.此算法详见文献[4].

相似性判定在本文的算法里有两个地方被用到,一是判定是否是有效的页面,这时只考虑结构的相似;二是判定页面是否发生改变,这时考虑包括文本节点在内的整棵树的相似.

传统的页面相似性判定只用于判定页面是否是有效的页面.因为在 Web 2.0 以前,URL 是一个页面的唯一标志,只需要根据 URL 便可以判定是否为不同的页面.而现在 URL 不能作为唯一的标志,只有比较页面的具体内容才能够判定是否为不同的页面.对一个页面而言,其上面可以触发大量的事件.不引起页面变化的事件我们

固然认为页面没有发生改变,但也有不少事件只是使页面发生了轻微的改变,如改变了文字的颜色,将隐藏的内容显示出来等等.这样的页面对我们后面的分析等工作往往不会有新的价值,反而会提高工作量.因此对于轻微的页面变化,我们同样认为页面没有发生改变.这一点也是通过相似性判定来进行的,若变化程度不超过一定的阈值便认为没有发生变化.

2.4 XPath

XPath 是 XML 文档中进行查询的语言,它用路径表达式来选取 XML 文档中的节点或节点集.XPath 同样可以应用在 HTML 中.由于在动态脚本网站中 URL 不能作为页面的唯一标识,通过相似性判定不能对 URL 的特征进行归约.本文方法以页面元素为中心,使用了 XPath 作为页面元素的描述,并对所有引向我们需要页面的页面元素的 XPath 进行了归约,同时记录需触发的事件.

本文对 XPath 的归约采用了如下的归约方法:被归约的 XPath 路径经过的页面元素名称必须相同,对页面元素的序号进行归约.如对"/html/body/div[4]/li[1]/a[1]"和"/html/body/div[4]/li[2]/a[1]"这两个 XPath,我们归约为 "/html/body/div[4]/li[*]/a[1]"; 如还存在"/html/body/div[3]/li[1]/a[1]"这一 XPath, 我们将归约为"/html/body/div[*]/li[*]/a[1]".但"/html/body/div[1]","/html/body/span[1]"和"/html/body/div[1]/span[1]"中的任何两个均不被归约.本文后续部分提到的两个实验及我们选取的其他一些网站的例子中均发现,这样归约出的 XPath 结果不再含有无效的页面元素,可以作为抓取过程中的特征.

3 算法描述

本文给出的方法分为两个阶段:训练阶段和应用阶段.在进行抓取前,我们首先进行训练.给定各种类型页面的一些样本,训练部分将对这些页面依次触发所有可能的事件,记录页面的变化情况.完成后,将对每种类型页面有效页面元素的 XPath 和事件类型进行归约,总结出特征.在应用阶段,我们抓取网页过程中只触发训练部分总结出的具有特定 XPath 的页面元素上的特定事件.这样一来,需要触发的事件数大为降低.

3.1 训练阶段

训练过程将接受各类型页面的一些样本作为输入,输出有效页面元素的 XPath 特征和需触发的事件类型.对每个样本,将依次触发其上的各种可能事件,将事件类型,页面元素的 XPath 和状态是否变化以及若变化变为何种类型页面记录下来,并回退到触发事件前的状态.所有样本均处理完成后,再根据记录总结出有效页面元素的 XPath 特征及需要触发的事件类型.算法描述如下:

算法 1. 训练算法.

```
1.  Procedure Training(Queue trainingQueue)
2.  while trainingQueue 不为空
3.      page←trainingQueue.pop()
4.      while page 还有未触发的事件
5.          触发此事件
6.          记录事件类型,页面元素的 XPath,页面是否变化及若变化后为何种类型
7.          回退到触发事件前状态
8.      end while
9.  end while
10. 根据记录归约出有效页面元素的 XPath 特征及触发事件类型
```

3.2 应用阶段

应用阶段将接受训练阶段总结出的各类型页面有效页面元素的 XPath 特征及需要触发的事件类型,和一个初始的 URL 作为输入,输出为下载后的页面.对于每个需处理的页面,将根据其页面类型触发特定 XPath 代表的页面元素上的特定事件,并将变化后的新页面加入待处理队列.可以根据需要选择宽度优先或深度优先等各

种抓取策略.下面给出的抓取算法是采取宽度优先的抓取策略:

算法 2. 宽度优先的抓取算法.

```
1.  Procedure BreadthFirstCrawl (Url index)
2.  抓取 index 页面并保存
3.  将此页面加入 crawlQueue
4.  while crawlQueue 不为空
5.      page←crawlQueue.pop()
6.      根据页面类型确定需触发事件的页面元素的 XPath 及事件类型
7.      while page 还有待触发的事件
8.          触发此事件
9.          if 页面变化且为未处理过的页面 then
10.             保存此页面并将其加入 crawlQueue
11.         end if
12.         回退到触发事件前状态
13.     end while
14. end while
```

4 实验结果

我们在单机上进行了模拟实验,机器的参数为:Intel CPU E2160 1.80GHz,内存 2GB,网卡 100Mbps,操作系统为 Windows XP,程序使用 Java 开发.本文通过两个实验来比较利用本文给出的方法进行抓取与通用抓取相比带来的效率提升.通用抓取为文献[1]里提出的“Full Auto Scan”方法,它将尝试触发每个可能的事件,并将每个不同的页面都加入待处理队列.

第 1 个实验是抓取 QQ 论坛,它的文章列表页面的翻页是用 JavaScript 来控制的.我们从 http://bbs.news.qq.com/b-1001024024 出发,触发所有“<a>”标签的“onclick”事件.文章列表页面和文章页面是我们所需要的页面.作为对比的通用抓取算法也做了优化——如果触发事件后改变的页面不是文章列表页面或者文章页面,则不将其加入待处理队列.抓取一定数量的有效页面需要触发的事件数如图 3 所示.

从实验结果可以看出,利用本文给出的方法抓取需要触发的事件数要比优化的通用抓取方法少,且随有效页面数的提高增长稳定;而优化的通用爬虫在抓取 750 个有效页面后需要触发的事件数快速增长,有明显的发散趋势.这是因为抓取开始阶段将很多文章列表页面加入到待处理队列里,这些文章列表页面的包含的“<a>”标签质量较高,多数将导向新的文章列表页面或文章页面.伴随着抓取的过程,程序处理的文章页面占了多数,它们包含的“<a>”标签质量较低,大多导向我们不需要的页面.从而需要触发的事件数在抓取一段时间后会快速增长.

第 2 个实验是比较两种方法对 AJAX 网站的抓取效率.网易新闻的评论页面是 AJAX 的,我们进行评论翻页时 URL 不会发生变化.实验随机选取了 3 篇网易新闻的评论作为待抓取的内容,触发所有“<a>”标签的“onclick”事件.与第 1 个实验类似,作为对比的通用抓取算法同样做了优化,只将 URL 不发生改变的新页面加入待处理队列.实验结果如表 1 所示.

Table 1 Experiment result on crawling news comment pages of NetEase

表 1 网易新闻评论抓取实验结果

News	Number of comment pages	Number of events triggered		Recall percentage of our approach (%)
		Optimized generic crawling method	Our approach	
News 1	7	1 114	78	100%
News 2	8	1 698	90	100%
News 3	11	1 609	156	100%

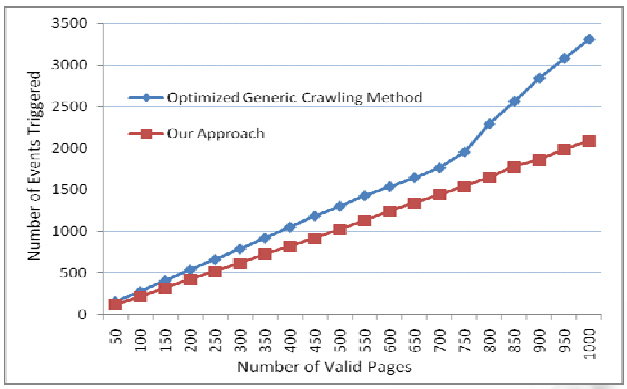

Fig.3 Experiment result on crawling QQ forum

图 3 QQ 论坛抓取实验结果

从实验结果可以明显看出利用本文给出的方法进行抓取的优势.对于抓取这 3 篇新闻的评论页面,本文给出的方法需要触发的事件数远少于优化的通用抓取算法.而且,随着评论页数的增加,优化的通用抓取算法需触发事件数变化不稳定,而本文给出的方法则相对稳定增加.这也是因为本文给出的方法只关注有效页面元素的缘故.另外值得注意的是,利用本文给出的方法抓取,召回率为 100%,即抓回了我们所需的全部数据.

在上面的两个实验中,我们仅关注了“<a>”标签上的“onclick”事件,而“ ”,“”等其他页面元素以及“onmouseover”等其他事件我们没有考虑.我们可以推测,如果考虑了这些,通用抓取算法的效率将更加低下,而本文给出的方法的效率却不会有什么影响.从上面的两个实验的结果我们可以看出,利用本文给出的方法进行抓取在效率方面具有明显的优势,且抓取需触发的事件数随需要抓取的有效页面数增长稳定.这证明了本文给出的方法的有效性.

5 总结和进一步研究

本文为动态脚本网站建立了一个通用的状态转换图模型,可以有效地应用于动态脚本网站的抓取.本文提出了一种高效抓取动态脚本网站有效页面的方法,首先进行训练,给出各种类型网页的一些样本,利用页面相似性总结触发哪些页面元素上的哪些事件将引向我们所需的页面,页面元素的特征通过对其 XPath 归约得到;以后在抓取过程中我们只触发这些 XPath 代表的页面元素上的特定事件,以减少触发事件数.

实验证明,与通用抓取方法相比,本文给出的方法具有明显的优势,大幅减少了需触发事件数.然而,要实现工业级应用的动态脚本网站抓取,还有很多工作要做.首先,多个页面都可以触发事件转换到同一个页面,我们如何在已经得到此页面的情况下避免这样的事件触发,例如在网易新闻的评论页面里,第 1 页可以跳转到第 5 页,第 2 页也可以跳转到第 5 页,在第 1 页里我们已经触发过跳转到第 5 页的事件,如何能够避免在第 2 页时不触发跳转到第 5 页的事件.此外,动态脚本网站更新频繁,未来我们还应建立一个针对动态脚本网站的有效的重爬机制.

References:

[1] Mesbah A, Bozdag E, van Deursen A. Crawling ajax by inferring user interface state changes. In: Proc. of the 8th Int'l Conf. on Web Engineering (ICWE 2008). New York: IEEE Computer Society, 2008. 122−134. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4577876

[2] Duda C, Frey G, Kossmann D, Matter R, Zhou C. AJAX Crawl: Making AJAX applications searchable. In: Proc. of the 25th Int'l Conf. on Data Engineering (ICDE 2009). Shanghai: IEEE Computer Society, 2009. 78−89. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4812393

[3] Vidal MLA, Silva AS, Moura ES, Cavalcanti JMB. Structure-Driven crawler generation by example. In: Proc. of the 29th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 2006). Seattle: ACM, 2006. 292−299. http://doi.acm.org/10.1145/1148170.1148223

[4] Reis DC, Golgher PB, Silva AS, Laender AF. Automatic Web news extraction using tree edit distance. In: Proc. of the 13th Int'l Conf. on World Wide Web (WWW 2004). New York: ACM, 2004. 502−511. http://doi.acm.org/10.1145/988672.988740

[5] Vieira K, Silva AS, Pinto N, Moura ES, Cavalcanti JMB, Freire J. A fast and robust method for Web page template detection and removal. In: Proc. of the 15th ACM Int'l Conf. on Information and Knowledge Management (CIKM 2006). Arlington: ACM, 2006. 258−267. http://doi.acm.org/10.1145/1183614.1183654

[6] Cai R, Yang JM, Lai W, Wang YD, Zhang L. iRobot: An intelligent crawler for Web forums. In: Proc. of the 17th Int'l Conf. on World Wide Web (WWW 2008). Beijing: ACM, 2008. 447−456. http://doi.acm.org/10.1145/1367497.1367558

[7] Pandey S, Olston C. User-Centric Web crawling. In: Proc. of the 14th Int'l Conf. on World Wide Web (WWW 2005). Chiba: ACM, 2005. 401−411. http://doi.acm.org/10.1145/1060745.1060805

[8] Chawathe SS, Rajaraman A, Garcia-Molina H, Widom J. Change detection in hierarchically structured information. In: Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 1996). Montreal: ACM, 2006. 493−504. http://doi.acm.org/10.1145/233269.233366

[9] Barbosa L, Freire J. An adaptive crawler for locating hidden-Web entry points. In: Proc. of the 16th Int'l Conf. on World Wide Web (WWW 2007). Banff: ACM, 2007. 441−450. http://doi.acm.org/10.1145/1242572.1242632

[10] Mesbah A, van Deursen A. Migrating multi-page Web applications to single-page ajax interfaces. In: Proc. of the 11th European Conf. on Software Maintenance and Reengineering (CSMR 2007). Amsterdam: IEEE Computer Society, 2007. 181−190. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4145036

[11] Marchetto A, Tonella P, Ricca F. State-Based testing of ajax Web applications. In: Proc. of the 1st Int'l Conf. on Software Testing Verification and Validation (ICST 2008). Lillehammer: IEEE Computer Society, 2008. 121−130. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4539539

[12] Li XM, Yan HF, Wang JM. Search Engine: Principle, Technology and Systems. Beijing: Science Press, 2005. 29−54(in Chinese)

附中文参考文献:

[12] 李晓明,闫宏飞,王继民.搜索引擎——原理、技术与系统.北京:科学出版社,2005.29−54.

夏冰(1985－),男,江苏兴化人,硕士生,主要研究领域为 Web 环境下的数据库,信息集成.

王腾蛟(1973－),男,博士,副教授,主要研究领域为数据库,信息系统.

高军(1975－),男,博士,副教授,主要研究领域为数据库,信息系统.

杨冬青(1945－),女,教授,博士生导师,主要研究领域为数据库,信息系统.