

基于 Rhino 的 JavaScript 动态页面解析研究与实现

金晓鸥, 钟宝燕, 李 翔

(上海交通大学 信息安全工程学院, 上海 200240)

摘 要: 面对互联网上占据全国页面总数 50% 以上的动态页面, 当前网络舆情管控工作中的信息采集环节对以动态页面为主要发布形态的互联网媒体无法实现信息获取。鉴于此, 文中提出了基于 Rhino 实现 JavaScript 动态页面解析的整体方案。实验结果表明该方案充分丰富了互联网舆情管控工作的数据源对象, 是实现动态页面内超链接网络地址递归获取和网页主体内容提取行之有效的解决方案。

关键词: 脚本解释引擎 Rhino; JavaScript 动态页面; 动态页面解析

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2008)02-0001-04

Research and Implementation of Interpreting JavaScript Dynamic Web Page Based on Rhino Engine

JIN Xiao-ou, ZHONG Bao-yan, LI Xiang

(Information Security Engineering School of Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: Dynamic Web page holds more than 50% of the total Web pages in countywide; however, the information collector of current network public opinion monitoring system can not get the information of Internet medium which uses dynamic Web page as its main content distribution form. Thereby, there is a scheme for interpreting JavaScript dynamic Web page by using Rhino engine presented in this paper. Proved by the experiments, this scheme is an effective one for extracting the hyperlink network addresses and content of dynamic Web page and it has enriched the work data set of network public opinion monitoring.

Key words: Rhino script engine; JavaScript dynamic Web page; interpret dynamic Web page

0 引 言

中国互联网络信息中心(CNNIC)2007年1月23日发布的《第19次中国互联网络发展状况统计报告》^[1]显示,截至2006年底我国网民人数达到了1.37亿,占全国人口总数的10.5%。在充分享受信息时代给予学习工作、休闲娱乐带来巨大便利的同时,还要充分认识到互联网对于社会舆论和大众文化潜在的影响与威胁。如何对互联网发布、传输和浏览的内容进行有效的监督、管理和引导是我国推进现代化建设,营造和谐社会的重要课题之一。

目前,互联网舆情管控工作旨在通过采集、分析和表达互联网媒体发布内容,为网络监管部门提供舆情管控参考依据。舆情管控工作主要由信息采集、内容

分析和结果呈现三大环节共同组成,其中信息采集环节为后续的分析、表达提供必要的原始数据材料,其工作机理类似于传统的“网络机器人”,以某一个页面为起始页,递归获取网页主体内容,及其内嵌超链接所指向的网络文件数据,文中主要研究其中的JavaScript动态页面解析技术。

1 论文工作对象与研究现状

1.1 动态页面

根据网页中是否含有浏览器执行脚本,将网页分为静态页面和动态页面。静态页面的主体内容及其内部包含的超链接网络地址分别以文本信息和唯一资源标识符(URL)的方式直接嵌入页面源文件的HTML标记(Tag)中。可以使用传统的HTML标记识别的方法^[2],实现页面主体内容与其所含超链接网络地址的提取功能。

然而,动态页面中除了包含少量静态URL外,还含有大量必须通过浏览器执行脚本才能得到的超链接网络地址和网页主体内容^[3]。举例如下:

收稿日期:2007-05-03

基金项目:国家自然科学基金项目(60502032, 60402019);上海市科委项目(065115020);教育部新世纪优秀人才支持计划项目(NCET-06-0393)

作者简介:金晓鸥(1983-),女,浙江永嘉人,硕士研究生,研究方向为互联网内容安全;李 翔,副教授,研究方向为网络内容安全。

例 1, 在浏览器中打开网页 <http://work.cat898.com/list.asp?boardid=1> (凯迪社区 - 猫眼看人 - 帖子列表), 右键查看源代码, 得到其中某个帖子的源代码, 如下:

```
<Script Language=JavaScript>document.write(dvbbs.topic_list(TempStr,'1590974','1','储户在银行被运钞警卫开枪打死。','Susan* * *','gun $ 17167040 $ 2007-4-12 20:23:16 $ 必须枪毙! $ $ 53304 $ 1590974 $ 1','face1.gif','68004','2007-4-9 13:04:17','1252','88272','0','2007-4-12 20:21:15','0','0','0','0','0','0','5','1'));hiddentr('follow1733888');</Script>
```

例 2, 在浏览器中打开网页 <http://club.book.sohu.com/r-zz0090-59022-0-5-0.html>, 右键查看源代码, 在源代码中查询帖子中的某个句子, 比如“青春不再, 红颜不再”, 将发现其在源代码中不存在。

由以上的例子可知, 传统的 HTML 标记识别方法无法完成动态页面主体内容及其所含超链接网络地址的提取工作。然而在全国页面总数中, 动态网页占据 50% 以上^[1], 并以 JavaScript 语言编写的动态页面最为流行。因而, 当前舆情管控工作中的信息采集环节无法对以动态页面为主要发布形态的互联网媒体实现信息获取, 这势必影响管控工作的顺利进行。鉴于此, 文中重点研究关于 JavaScript 动态页面内嵌超链接以及页面主体内容的解析与提取技术, 研究工作旨在丰富舆情管控系统所处理的数据源对象, 切实提高管控工作的功能能级。

1.2 动态页面解析研究现状

关于动态页面解析, 即内嵌超链接与页面主体内容的提取, 主要存在如下两种解决方案^[4]: 一是使用完整的开源浏览器 (例如早期的 Netscape Navigator, 以及 Mozilla) 渲染整个动态页面, 从浏览器的输出结果中提取页面主体内容与超链接网络地址; 二是自行构建脚本解析环境, 利用开源浏览器项目中的脚本解释引擎, 实现相关脚本片段的解析, 从而获得动态页面主体内容与超链接网络地址, 参见文献[3]。较之前一种解决办法, 后者工作针对性更强, 执行效率更快, 实现过程中使用的资源空间更少。正因如此, 选取合适的脚本解释引擎, 构建动态页面核心脚本所需要的解析环境, 已经成为当前动态页面解析工作的主流技术。

2 JavaScript 动态页面解析知识基础

2.1 HTML 文档对象模型

HTML 文档对象模型 HTML DOM 是 W3C^[5,6] (World Wide Web Consortium) 所定义的一套访问和操作 HTML 文档的标准^[5], 它将网页源文件映射成一系

列 HTML DOM 对象, 每个 HTML DOM 对象同时包含实现网页动态显示的属性和操作网页源文件的方法, 主要对象有 Window (代表浏览器窗口), Document (代表当前网页源文件), Location (包含当前 URL 信息) 等。这些对象中, 只在 Window 和 Document 对象的方法参数中包含超链接网络地址和页面主体内容信息, 而 Location 对象只是存储当前网页 URL 信息, 其他 HTML DOM 对象分别存储浏览器显示状态和历史信息等。

2.2 脚本解释引擎 Rhino

除了 SpiderMonkey^[3], 开源浏览器 Mozilla 还发布了纯 Java 语言编写的 JavaScript 脚本解释引擎 Rhino^[7]。与 SpiderMonkey 相比, Rhino 的优势在于可以实现从 Java 对象到动态页面脚本片段常用语言——JavaScript 对象的直接映射, 这有利于简化脚本解析环境的构建工作, 减少脚本解释引擎与脚本片段在实现语言方面的差异。同时, 在脚本解析的过程中, Rhino 对于 JavaScript 对象的操作结果可以通过访问已经本地创建的、与其一一对应的 Java 对象直接获得。正因如此, 选择 Rhino 作为文中的脚本解释引擎。

Rhino 的主要功能是脚本执行时的运行环境管理。运行环境是指用来保存所要执行的脚本中的变量、对象和执行上下文的空间。运行环境中的变量和对象由运行环境内所有的执行上下文共享, 即一个执行上下文创建的变量或对象其他上下文也可以访问, 运行环境负责处理变量或对象访问时的同步和互斥问题。运行环境和执行上下文是执行脚本语句的场所, 因此在应用程序中应首先调用 Rhino 提供的 API (应用程序接口) 建立一个运行环境和若干个执行上下文, 然后调用相应的 API 建立脚本语言的内置对象。

3 实现 JavaScript 动态页面解析

由于脚本解释引擎 Rhino 无法识别 JavaScript 脚本片段中包含的 HTML DOM, 在把动态页面脚本片段传递给 Rhino 前, 需要先对脚本片段中的 HTML DOM 实现本地创建, 给出每个 HTML DOM 的方法和属性描述, 这是构建脚本运行环境最为核心的工作。

3.1 JavaScript 脚本片段提取

JavaScript 动态页面脚本片段主要包含以下三种存储方式:

- 1) 最为常见的位于 `<script>` 和 `</script>` 标签之间;
- 2) 使用 “JavaScript:” 脚本描述方式, 置于某个 HTML 标记中;
- 3) 位于 `<script>` 标签的 SRC/ARCHIVE 属性中

指向的外部js文件中。对于前两种可以使用传统的HTML标记识别方法实现脚本片段提取,分别匹配<script>和</script>标记对和“JavaScript:”字符串。对于后一种,先获得网页源文件内SRC/ARCHIVE属性中以.js结束的字符串,再结合当前网页的基地址构造该js文件的网络绝对地址,最后对其单独实现网络获取,即从动态页面所属主机上获取该js文件。

3.2 HTML DOM本地创建

正如前文所述,HTML DOM中只有Window和Document对象的方法参数中含有超链接网络地址信息和页面主体内容。因此在进行HTML DOM对象本地创建时,将其余对象的属性和方法简单地设置为空(NULL)。

在Window和Document对象的方法参数中,与超链接网络地址、页面主体内容相关的函数可以分为两类:第一类以Window对象的open方法为代表,open方法的参数是动态页面中的超链接网络地址,参数类型是JavaScript语言内置String类型。在引擎外创建该类方法时,声明该方法的行为是把参数,即超链接网络地址送入信息采集环节的待获取URL队列中;第二类以Document对象的write方法为代表,write方法的参数(同为JavaScript语言内置String类型)是一段表达脚本片段最终在浏览器中呈现内容的静态网页源文件。类似于常见的静态网页,在作为write方法参数的网页源文件中,超链接网络地址和页面主体内容被分别以URL和文本信息置方式直接嵌入HTML标记中。在引擎外创建该类方法时,声明该方法的行为是把参数,即静态网页源文件写入位于本地特定的文件中。

由于Rhino能够自动在Java对象和JavaScript对象之间根据“对象名称一致性”的原则实现一一对应。因此,当Rhino在执行脚本片段中的“Window.open()”与“Document.write()”时,实际上是分别调用在脚本解

送引擎外,Java语言作用域中定义的,与上述两方法同名的Java函数,执行函数体中关于函数行为的描述。

3.3 调用Rhino实现JavaScript动态页面解析

在完成脚本片段提取和HTML DOM本地创建后,就可以调用Rhino提取JavaScript动态页面中的超链接网络地址及页面主体内容。当遇到脚本片段中的HTML DOM时,Rhino根据引擎外创建的同名函数体中的行为描述执行相应动作。

根据HTML DOM本地创建结果,Rhino将脚本片段中Window对象方法open参数体现的超链接网络地址直接送入信息采集环节的待获取URL队列中,实现动态页面内含超链接的递归获取功能。与其类似,把脚本片段中Document对象方法write参数所指向的静态网页源文件写入本地特定的文件中。在此基础上,使用传统的HTML标记识别方法,提取得到的静态网页源文件中的超链接网络地址与页面主体内容,将前者送入信息采集环节的URL队列,把后者交信息采集环节统一实现数据存储,如图1所示。

4 JavaScript动态页面解析实验结果

基于Rhino能够实现JavaScript动态页面所含超链接网络地址,以及页面主体内容的提取功能,为舆情管控工作中信息采集环节存在的JavaScript动态页面解析问题提供了行之有效的解决方案,如下所示。

文中1.1节例1的实验结果:

解析前帖子在源代码中的存在形式:

```
<Script Language=JavaScript>document.write(dvbbs_topic_list(TempStr,'1590974','1','储户在银行被运钞警卫开枪打死。','Susan***','gun $ 17167040 $ 2007-4-12 20:23:16 $ 必须枪毙! $ $ 53304 $ 1590974 $ 1','facel.gif','68004','2007-4-9 13:04:17','1252','88272','0','2007-4-12 20:21:15','0','0','0','0','0','0','5','1'));hiddentr('follow1733888');</Script>
```

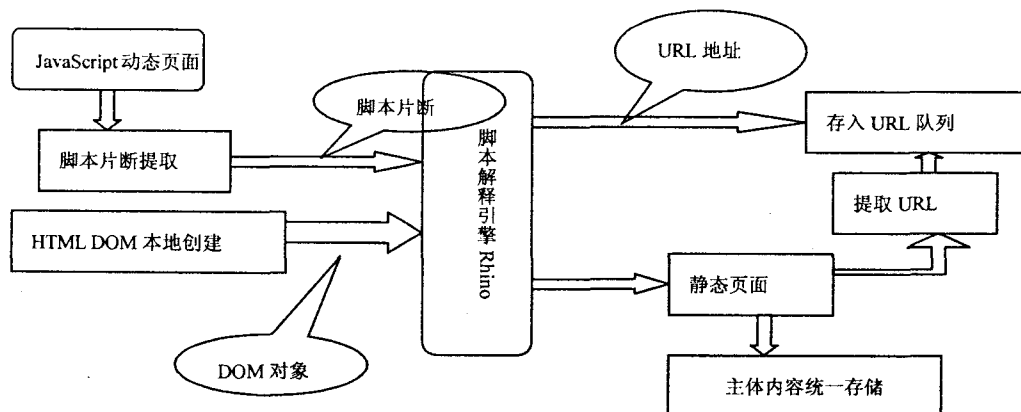


图1 基于Rhino实现JavaScript动态页面解析的具体过程

比率 P 对于算法性能的影响如图 3 所示。

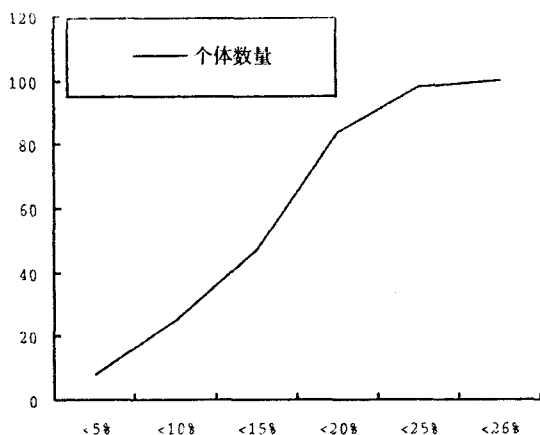


图 2 解的分布图

表 2 阈值 D 对求解结果的影响

D	0	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2
最优值	26474	25907	25135	25034	25474	24402	24961	24720	26189
平均值	33745	31835	31890	31850	31319	31760	31377	31870	32327
平均收敛代数	125.2	152.5	154.0	156.4	167.4	163.64	167.2	158.6	151.0

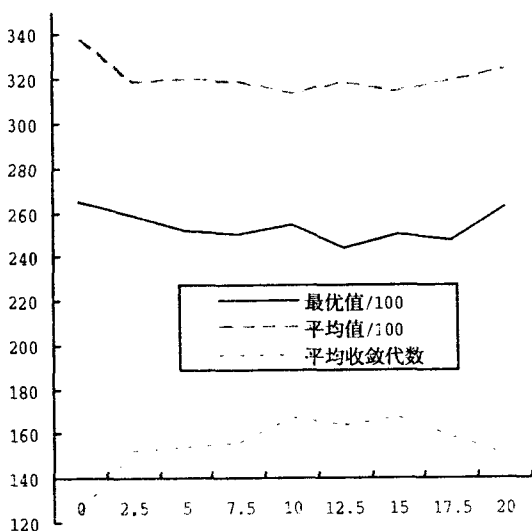


图 3 P 值对算法性能的影响

4 结束语

建立了一个多物流中心配送模型,并基于多物流中心配送的特点,设计了求解算法。基于在多物流中心配送情况下,通常每个物流中心只负责满足一定范围的需求点的需求量的事实,算法首先使用扩大的集

合覆盖的方法对所有需求点进行了预处理,确定了每个配送中心可能提供服务的需求点范围。这样既可以克服将多中心配送问题转化为单中心配送问题时,由于需求点分派不合理所造成的求解误差较大的问题,又解决了随机分派需求点所产生的运算量较大的问题。采用两重遗传算法求解模型,第一重算法采用单亲遗传算法,为算法设计了新的编码方案和交叉规则。对每个物流中心所服务的需求点的路径安排,采用简单遗传算法。设计了新的交叉规则和优良个体保存方案,较好地解决了单遗传算法随机交叉操作产生的早熟现象,同时也使算法具有一定的收敛速度。数据实验表明,算法是有效的。

在一个物流配送中心所负责的需求点的路径安排上,笔者只考虑了各顶点间的坐标关系,没有考虑实际存在的路径关系。同时也没有考虑多车辆配送的车辆调度和路径安排问题。在实际应用中,可用文献[6]中建立的车辆调度模型和算法代替(5)式及其求解算法,来求解基于实际交通网络的多配送中心的车辆调度和路径安排问题。

参考文献:

- [1] 张俊伟,王 勃,马范援.多仓库多配送点的物流配送算法[J].计算机工程,2005,31(21):192-194.
- [2] Skok M, Skrlac D, Krajcar S. The genetic algorithm method for multiple depot capacitated vehicle routing problem solving [C]//The Fourth International Conference on Knowledge-based Intelligent Engineering Systems & Allied Technologies. Brighton, UK: UK Press, 2000:520-526.
- [3] Filipec M, Skrlac D, Krajcar S. Genetic algorithm approach for multiple depot capacitated vehicle routing problem solving with heuristic improvements built-in[J]. International Journal of Modeling and Simulation, 2000, 20(4):320-328.
- [4] 魏百鑫,史海波.基于整车配送的多仓库开路 VRPTW 问题的研究与实现[J].信息与控制,2005,34(3):350-355.
- [5] 王小平,曹立明.遗传算法[M].西安:西安交通大学出版社,2002.
- [6] 戴树贵,潘荫荣,胡幼华.基于最小费用的物流配送模型及其混合单亲遗传算法[J].计算机应用,2005,25(11):2681-2684.

(上接第 4 页)

- 集技术中的应用[J].计算机应用,2004,24(2):33-36.
- [4] 关于搜索引擎页面分析中的 javascript 处理的 2 个思路 [EB/OL]. 2006-09-03. <http://blog.csdn.net/DanceFire/archive/2006/09/03/1163683.aspx>.
 - [5] W3C Recommendation. Document Object Model [DB/OL]. 1998-10. <http://www.w3.org/TR/REC-DOM-Level>

-1/.

- [6] 柳正青,刘怀亮,李振坤,等.XML 编程接口的研究与一个应用模型[J].微机发展,2003,13(6):61-64.
- [7] Individual Mozilla. org contributors, Rhino Documentation [DB/OL]. 2006-12. <http://www.mozilla.org/rhino/doc.html>.