文章编号:1005－0523(2007)01－0067－04

面向 URL 的网络机器人软件模型的研究与实现

李广丽,刘觉夫

(华东交通大学 信息工程学院,江西 南昌 330013)

摘要:WEB 数据挖掘的关键是设计智能、高效的网络机器人.详细分析了面向 URL 的网络机器人的工作流程及实现它的关键技术,提出用多个队列管理 URL 列表,且队列元素按文档相关性高低排序,并行高速地下载网页.此外,在文档相关性计算中设计了一个可收敛的迭代阈值算法,有效地解决了相关度阈值设定的随意性.

关 键 词:网络机器人;URL 种子;广度优先;文档相关性;阈值

中图分类号:TP393.08　　**文献标识码**:A

0 概述

WEB 站点包含了大量纷繁复杂的信息,如何对它们进行挖掘以获取有价值的参考数据已经成为当今数据挖掘研究的热点问题.WEB 数据挖掘的关键是设计网络机器人,因此,本文将从网络机器人的工作流程、关键实现技术、软件算法三个方面对其展开深入的研究,旨在设计一个适应能力强,功能完善的网络机器人的软件模型.软件开发环境选择 Sun JDK＋Borland Jbuilder＋SQL Server＋IIS＋Bot 包.

1 网络机器人工作流程

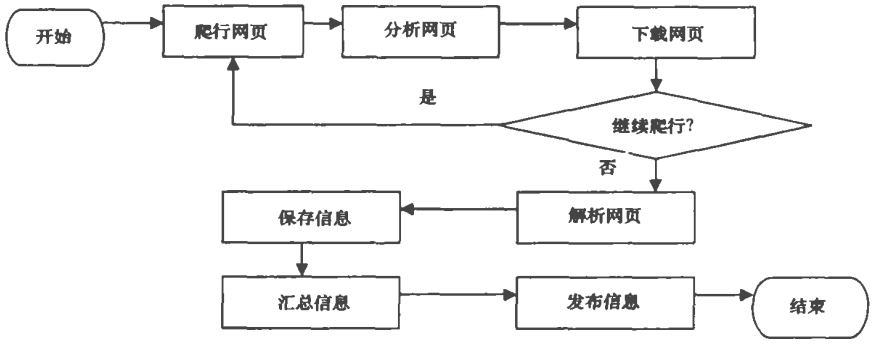

图 1　网络机器人工作流程图

在给定主题或 URL 后网络机器人开始爬行网页.如“主题型网络机器人”根据主题爬行网页,它建立在完备索引库的基础之上,有很高的“召回率”和“准确率”,但开发难度大,成本高,适用于高端搜索引擎系统.而“面向 URL 的网络机器人”在给定一个 URL 后开始爬行,这个 URL 也称种子,它适合对一个整体网站进行信息提取,但它缺乏对 URL 重要性的衡量,容易返回大量无用网页.因此,从提高“召回

收稿日期:2006－10－11

基金项目:赣教技字[2006]177 号,华东交通大学校立基金 01305120

作者简介:李广丽(1978－),女,广西博白人,华东交通大学信息学院,讲师.

率”及降低成本的角度出发，笔者提出“URL作为种子的网络机器人”模型.该模型兼顾URL种子和主题，由URL种子出发爬行网页，并通过文档相关性来衡量网页的重要性.

如果把爬行到的网页全部下载下来，则可为进一步的数据挖掘提供大量信息.但这会降低搜索的“精确度”，因此，必须对网页的重要性展开分析.笔者根据文档相关性对网页重要性进行分析，对相关性计算值低于某阈值的网页不予下载.

WEB是一张“图”，理论上给定一个URL种子，网络机器人可以爬行完“图”中的所有结点.但爬行网页不是任由网络机器人在WEB页面之间不断迁移，当爬行信息获取充分之后应中断其爬行，笔者采用设置中断爬行时间和特定爬行范围的方法控制网络机器人的爬行.如规定爬行在1天之内完成，或规定URL范围和爬行深度进行爬行.“下载网页”即把网页以HTML的形式保存到硬盘上.HTML是一种标签语言，网络机器人应该能够“认识”这些标签，从中挖掘重要数据，以完成“解析网页”的工作.

信息被解析之后采用数据库、HTML和XML三种方式扩展数据应用.数据库方式能较安全地保存数据；HTML方式中，解析的信息在汇总后被格式化地写入HTML文件，并由WEB服务器发布；而XML方式中，解析的信息被写入XML文件，由客户端负责对其解释，它降低了WEB服务器的负载并统一了数据表示，若结合XSLT对XML进行DOM编程，还可完成各种高级的“数据查询”功能.

2 网络机器人的设计

2.1 爬行算法应用

网络机器人的爬行算法包括深度优先、广度优先、Fish—Search等.深度优先(广度优先)算法对目标站点进行深度(广度)遍历，获取网页.Fish—Search算法模拟海中的鱼群觅食现象，当食物找到时，鱼被继续繁殖，否则鱼死掉.它的缺陷是文档相关性计算简单，Shark—Search算法是对Fish—Search算法的改进，它扩展了相关性的度量方法.由于站点的重要数据一般位于“树”中的较高层次，所以，广度优先算法更利于在较短时间内获取重要数据，并且深度优先算法很容易陷入一旦进去再也出不来的情况，故本文讨论的“URL作为种子的网络机器人”模型中选择广度优先算法.爬行算法应用如图2.

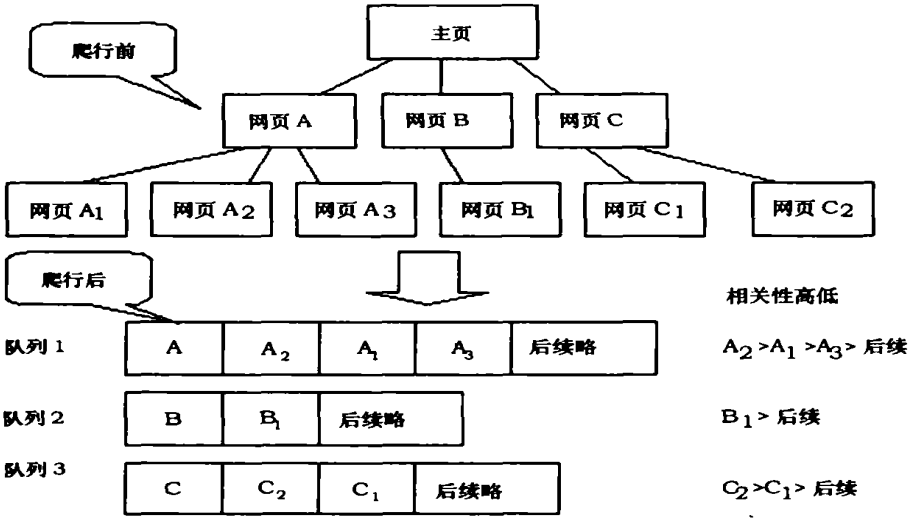

图2 爬行算法应用

广度优先又分为递归实现和非递归实现，在爬行网页过程中每次都会获取几十或上百个URL，并且URL数量随着爬行深度的下降有剧烈跳变的可能，若采用递归算法，那么频繁的压栈出栈操作会使系统的性能受到极大的影响.故选择非递归实现，并且采用队列管理URL列表.由于文档相关性决定了URL价值的高低，笔者考虑队列元素按相关性高低排序.文献[3]中采用单个队列管理URL列表，笔者认为此做法受限于队列长度，会影响对队列中重要URL的响应，故改用多个队列管理URL列表，通过多线程的方式对每个队头分别响应，完成网页爬行.

2.2 文档相关性计算

为保证网络机器人在爬行网页时靠近主题，必须定量衡量网页的文档相关性，笔者采用向量空间模型(VSM)计算文档相关性.设关键词n个，关键词权值ω_i，主题向量表示为：

$\alpha=(\alpha_1, \alpha_2, \cdots, \alpha_i, \cdots \alpha_n)$　$i=1,2,3\cdots n$　$\alpha_i=\omega_i$

统计 n 个关键词的出现频率，用 x_i 表示，故页面主题向量表示为：

$\beta=(x_1\omega_1, x_2\omega_2, \cdots, x_i\omega_i, \cdots x_n\omega_n)$　$i=1,2,3\cdots n$

而页面的文档相关性表示为：

$Cos<\alpha,\beta> = (\alpha,\beta)/|\alpha||\beta|$

指定阈值 t，当 $t<= Cos<\alpha,\beta>$ 时认为当前页面和主题相关，将其 URL 加入到下载队列中．t 值的设定需要对未知网页的文档相关性有较正确的估计，文献 3 中提出如果需要较多的网页可以把 t 设小一些，相反 t 可以大一些．这种方法固定了文档相关性阈值，一旦真实阈值偏离该值，很容易陷入无用网页太多或有用网页太少两个极端情况．所以，笔者借鉴数字图像处理中迭代法分割灰度图像的思想，设计了下面的算法，动态地更新阈值 t．

Step1：设定初始阈值 $ti=t0$，该值很小；

Step2：计算当前文档相关性 $t'=Cos<\alpha,\beta>$，如果 $t'>=ti$，主题相关，否则，主题偏离；

Step3：在已下载 URL 队列中找到相关性最高的数值 t_{max} 和最低的数值 t_{min}，更新阈值 $t_{i+1}=(t_{max}+t_{min})/2$，此时按下面的公式累加

$$T_{mean1}=\sum_{j=t_{min}}^{t_{i+1}} jh(j)/\sum_{j=tmin}^{t_{i+1}} h(j), T_{mean2}=\sum_{j=t_{i+1}}^{t_{max}} jh(j)/\sum_{j=t_{i+1}}^{t_{max}} h(j)$$

计算 $t_{i+1}=(T_{mean1}+T_{mean2})/2$，式中 h(j) 表示对应文档相关性数值出现的频率．因此，设计了一个队列，队列元素中存放其文档相关性数值和出现的频率．

Step4：应用新阈值下载网页，然后令 $t_i=t_{i+1}$ 并重复 Step2 和 Step3 计算阈值．当阈值有收敛倾向，如 $|t_{i+1}-t_i|<\varepsilon$（$\varepsilon$ 表示无穷小正值），认定当前阈值为合适阈值，后续网页的文档相关性计算遵循该值，迭代终止．

2.3　关键技术应用

1) Socket 技术：网络机器人爬行时需要保持和站点的 HTTP 连接，它是典型的 HTTP 应用．HTTP 建立在 TCP/IP 协议之上，它也是一种 Socket 协议，因此，网络机器人本质上是架构在 Socket 之上的网络通信程序．Java 中定义了两个类 Socket 和 ServerSocket 完成套接字编程，考虑到 Socket 类是高层父类，其扩展性更优，故软件中基于 Socket 类进行网络通信程序设计．

2) 多线程技术：网络机器人在下载网页时经常会遇到多个下载任务之间的串行等待，这不利于系统性能的改善，故采用多任务并行调度方法，将多个 HTML 网页的下载分配到独立线程中，最大限度地利用计算机资源提高下载速度．

3) 标签解析技术：要解析网页中的信息就必须让网络机器人能够“认识”网页中的各种格式化设置．网页的基本形式是 HTML，所以网络机器人必须理解 HTML 标签的含义．Java 中的 Swing 类和 Bot 包中的 HTMLPage 类都可以解析网页，Swing 类为底层解析包，解析原理复杂，而 HTMLPage 类则是针对不同标签解析的高层 API，编程更方便．笔者选择后者．

4) 流技术：Socket 成功建立之后，网络机器人与站点之间的通信以流的方式进行，此外，在扩展数据应用时需要将数据以流的方式记录到 HTML 或 XML 文件中．程序设计中主要运用到了 Java 的文件输出流/输入流操作．

3　部分核心代码

本文讨论的网络机器人模型的应用环境：URL 种子是 www．51job．com，笔者为此设计了 14 个解析函数，分别是 getZipcode()、getCompanyinfo()、getHR()、getPhone()、getFax()、getJobInfo()、getSalary()、getResumelang()、getLang()、getYear()、getPlace()、getDate()、getMail()、getNum() 和 getSite()．现以 getJobInfo() 为例说明程序实现，其它同理．

```
Protected String getJobInfo(String filename){
try {String url = "http://127.0.0.1/job/"+filename; //向 WEB 服务器发套接字连接请求
HTTPSocket http = new HTTPSocket(); //创建套接字连接
http.send (url, null);
int i = http.getBody().indexOf("职位描述");//在 HTML 中定位"职位描述"
int j = http.getBody().indexOf("申请这个职位");//在 HTML 中定位字符串
String info = http.getBody ().substring (i,j);//提取 i 字符与 j 字符之间的信息
return info;//返回解析结果
}catch (Exception e) {…//信息未能正确提取提示，省略 }}
```

扩展数据应用是把数据以流的方式写入到文件中,笔者为此设计了 WriteIntoHTML()和 WriteIntoXML()两个格式化函数,现以 WriteIntoHTML()为例说明程序实现.

```
Protected void WriteIntoHTML(){
…//创建 HTML 文件,打开文件流 ps,省略
ps.println("<html><head><title>网络机器人运行结果</title></head>");//文件标题
ps.println ("<body> <h1>网络机器人的运行结果</h1>");//解析结果标题
ps.println ("<table width='75%' border='1' align='center'>");//以表格形式汇总信息
ps.println("<tr> <td align='center'>电邮</td>");//表头开始,绘制表项
ps.println("<td align='center'>发布日期</td>"); //绘制表项
…//其他表项的绘制同上,省略
ps.println("</tr>"); //表头结束
for(int i=1; i<=numofinfo; i++)//numofinfo 表示已经提取的信息的条数
{ ps.println("<tr>"); //解析的数据单元开始
String mail = getMail(); //解析"公司邮件地址"
ps.print("<td align='center'>+mail+</td>");//输出"公司邮件地址"解析结果
…//其他各项的编码同上,省略
ps.println("<tr>");//当前一条记录输出完毕
}…//关闭文件流 ps}
```

4 结语

在给定 URL 种子:www.51job.com,设定主题为"程序员",爬行深度为 4,爬行范围不超出 51job 网站后,笔者进行了测试,当线程数目达到 100 左右时,系统性能最佳.相关度阈值可以收敛,此外,管理 URL 列表的队列数目随着爬行深度的下降受到一定限制,该队列数目不能无限增加,故根据较高层次中 URL 的数量确定其队列数目,效果较好.

参考文献:

[1]Jeff Heaton,童兆丰等译. 网络机器人 Java 编程指南[M]. 北京:电子工业出版社,2002.

[2]张红斌. 网上求职机器人的软件设计[J].华东交通大学学报,2006,23(1):113-116.

[3]汪涛,樊孝忠. 主题爬虫的设计与实现[J].计算机应用,2004,24(6):270-272

[4]龙腾芳.数据挖掘技术在农业领域中的应用研究[J].微计算机信息,2005,21(8):42-44

Research and Realization of a Spider Model Facing URL

LI Guang-li, LIU Jue-fu

(School of Information Engineer, East China Jiaotong Univ, Nanchang 330013, China)

Abstracts: The key issue of mining data on WEB is how to design an intelligent and effective spider. The paper analyzes the work flow and key technologies of the spider facing URL in details. It also brings forward the mind that adopting several queues to manage the URL list, in order to download HTML files in high speed we sort the URLs by document correlativity. Moreover, we import the idea of iterative threshold into computing document correlativity, which resolve the random modification of threshold.

Key words: spider; URL seed; scope first; document correlativity; threshold