基于 WEB 挖掘的网络爬虫设计与实现①

肖 毅[1], 张 林[2], 聂笑一[1]

[1](湖南农业大学 信息科学技术学院, 长沙 410128)
[2](湖南农业大学 东方科技学院, 长沙 410128)

摘 要: 从介绍 Web 挖掘与数据挖掘的差异入手, 分析 Web 挖掘中 Web 爬虫的必要性和现代 Web 挖掘技术的发展方向, 在深入了解 Web 爬虫的原理及其功能的基础上, 提出一个现代网站通用的挖掘模型, 并利用该模型设计一种网络爬虫. 经实例证明, 该爬虫能高效爬取更多的各种页面数据.

关键词: 数据挖掘; Web 爬虫; 挖掘技术

Design and Realization of Web Crawlwer Based on Web Minning

XIAO Yi[1], ZHANG Lin[2], NIE Xiao-Yi[1]

[1](Information Science and Technology College, Hunan Agricultural University, Changsha 410128, China)
[2](Orient Science& Technology College, Hunan Agricultural University, Changsha 410128, China)

Abstract: The diffeences between web-minning and data-mining were introduced in this paper firstly, then the necessity of Web crawler during web-minning and the development of modern web-minning technology were analysed. Based on the deep understanding of the principle and its function of Web crawler, a minning model popular in modern website was put forward, and a web crawler was designed by the use of this model. Tested by several examples, this kind of crawler can get more diversified pagedata efficiently.

Key words: data-mining; Web crawler; Web-minning technology

随着互联网的不断发展和普及, web 成为人们不可缺少的一部分, 同时也是人们获得信息的重要途径. 如何充分有效地利用 web 中数量庞大的信息成为一个不可回避的问题, web 数据挖掘技术也逐渐成为 web 技术中的重要部分. Web 挖掘是指综合利用数据挖掘技术对 Web 内容、Web 结构及 Web 日志等进行分析处理, 从中获得对决策制定有价值的各种信息的过程. Web 挖掘技术与传统的数据挖掘的比较, 主要区别在于数据收集. 对于 web 挖掘而言, 数据收集是一个具有挑战性的任务, 在对 Web 结构和内容挖掘时, 需要爬取大量的网页, 如何高效地爬取网页并处理获取的网页信息是数据收集的关键也是其难点. Web 爬虫是 Web 挖掘中重要技术之一, 是爬取页面的重要手段, 通过爬虫的构建达到 Web 信息搜索的目的[1].

1 数据挖掘与Web挖掘

传统的数据挖掘又称为数据库知识发现. 是指从数据源(如数据库、文本、图片、万维网等)中探寻有用的模式或知识的过程. 对于数据挖掘模式来说必须是有用, 有潜在价值的[2]. 它主要基于人工智能、机器学习、模式识别、统计学、数据库、可视化技术等, 高度自动化地分析企业的数据, 做出归纳性的推理, 从中挖掘出潜在的模式, 帮助决策者调整市场策略, 减少风险, 做出正确的决策. 其知识的获取主要分为三个步骤: 预处理、数据挖掘和后续处理, 同时整个过程可以进行迭代, 通过多次迭代获得最终结果[2].

随着万维网和文本文件的规模扩大, 文本挖掘和 Web 挖掘也越来越流行. 文本挖掘是从文本文件中抽取到有效且有价值的信息, 进行整合和分类后获得更

① 收稿时间:2013-03-04;收到修改稿时间:2013-04-07

高的价值. 而 Web 挖掘是指在万维网上挖掘潜在的、有用的信息. Web 挖掘的数据与传统数据不同, 既存在结构化数据, 也存在半结构化数据, 如: 图片, 文本, 异构数据等. 正因为数据不同, 所以比传统的单个数据库挖掘要复杂的多. 通过 Web 挖掘出来后的数据, 可以通过分析为用户提供更好的服务, 也能从中获得潜在的客户、用户和市场. 虽然 Web 挖掘使用许多数据挖掘技术, 但是又不仅仅是数据挖掘的一个运用. Web 挖掘任务主要被分为三种类型: Web 结构挖掘、Web 内容挖掘和 Web 使用挖掘[3]. Web 爬虫是 Web 挖掘的一种较为常用的实现方式, 也是搜索引擎的重要组成. Web 爬虫是一个获取网页信息的程序, 它通过在 Web 上下载网页来获取数据. 传统的 Web 爬虫从一个或若干初始网页的 URL 开始, 获得初始网页上的 URL, 在抓取网页的过程中, 不断从当前页面上抽取新的 URL 放入队列, 直到满足系统的一定停止条件而终止数据爬取行为. 正因为 Web 爬虫具有自动而高效地获取信息的能力, 常常用来建立数据仓库, 达到 Web 信息搜索的目的[4].

2 信息搜索与数据挖掘

对于 Web 网络中的数据, 是复杂、异质和庞大的, 对待如此具有针对性和多变性的海量信息进行挖掘时, 数据挖掘的技术也要随之而加以改进. 因此要为它搭建新的挖掘模型, 提出新的挖掘算法和体系结构[5]. 对此我们提出一个较为简单, 通用的数据搜索模型, 如图 1 所示.

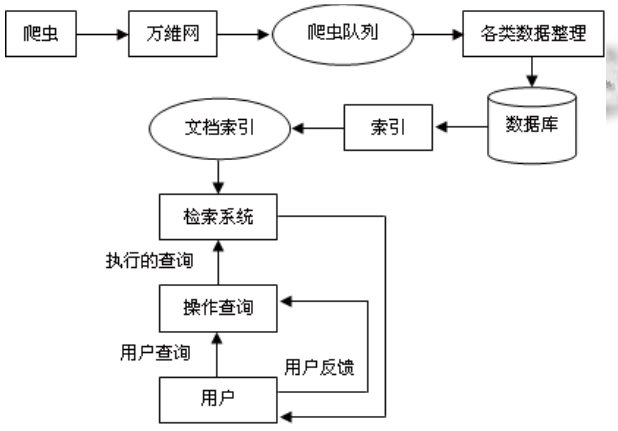

图 1 数据搜索引擎模型

在整个模式中, 爬虫对于数据获取是不可或缺的. 它对互联网中一些相关有用的信息, 进行挖掘, 充实数据库内容并及时让信息得到更新.

现今网络资源以文本资源为主, 爬虫则是从一系列种子网页开始, 结合内容和链接进行信息采集, 可以采集到图片, 音频, 甚至包括多媒体在内的多媒体信息资源, 不仅仅获得了信息资源的完整性, 同时还高效的获得数据资源, 为模型打下牢固的基础.

采集到的信息对象首先要进行预处理, 对一些不符合要求的信息和坏死链接进行剔除, 再将处理后的信息存入相应的数据库中. 数据库由文本库和其他媒体库组成, 结合用户人工输入查询的特点, 进行规范的排列. 最后导出为实际数据仓库, 为以后数据分析和决策支持系统提供数据支持[6].

3 Web爬虫模型设计与实现

互联网由数以 10 亿计的网页构成, 对于其中的静态网页来说, 采集信息时只要将所有网页取回, 并存放到网页库内即可. 但事实上, 绝大多数网页是一个动态的实体, 因此也要进行网页的同步更新抓取, 即进行增加, 删除, 移动, 修改链接等操作[7].

3.1 爬虫原理

在了解网页的基本结构后, 我们针对通用型网页, 构建了一个简单的顺序爬虫, 该爬虫主要使用广度优先算法, 通过添加对 URL 相应内容进行分析, 将有用的 URL 加入队列. 爬虫流程图如图 2 所示.

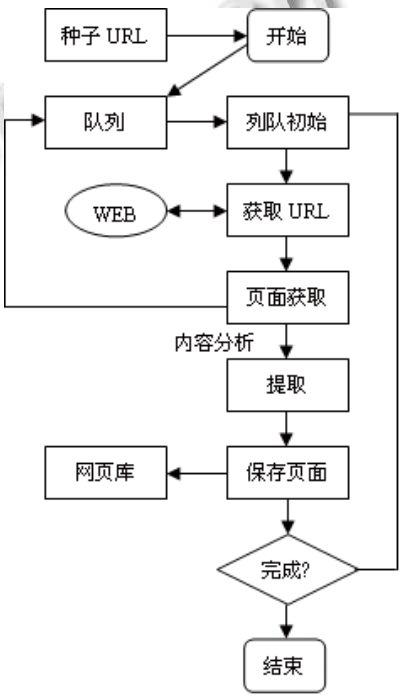

图 2 顺序爬虫流程图

当一张网页爬取下来后，就可以对其进行内容解析. 通过获取页面的 DOM 树结构，再用程序语言将其通过分析出来有用的 URL 获取下来存入数据库中，最后记录该 URL 的层级. 以下代码段是利用正则匹配方法获取有用的 URL.

```
$str = '<a href="url.html"><span>text1</span></a>';
preg_match_all('%<a[\s\S]*?href="([^"]+)[^>]+>([\s\S]*?)</a>%', $ str, $result, PREG_PATTERN_ORDER);
var_dump($result);
```

爬虫程序每次抓取一张页面内容，不对其他资源进行抓取. 对第一次获得的 URL 进行记录，当程序第一次执行完成后，将数据表里分析出来 URL 取出存入 URL 列表队中. 在进行初始化时，仅存入数据表里提供的 URL. 之后，通过 URL 列表队中的 URL 再对列表队中每一个 URL 页面进行逐个抓取，将新获得的 URL 追加到列表队中并存入数据表中，进行层级记录.

当主程序判断 URL 列表队为空时，停止程序. 如主程序遇到报错或一些其他原因，可以利用预设条件或跳过该 URL，直接进行下个 URL 页面抓取，并在数据库错误队列中存入数据段以记录报错 URL，待后面进行分析原因.

每次从 URL 列表队中抓取完一个 URL 页面后，进行列表队内存释放以提高效率. 程序中要设定 URL 优先级，当 URL 列表队成员过多时，进行优先级较低级的 URL 踢出队列，并在数据库中记录未抓取再将其加入备用队列中，可以进一步优化爬取的效率.

3.2 初始化 URL

URL 列表队的初始化是先将列表队伍清空，并使其成为先进先出列表队. 每次先入队列的 URL 地址，位于队列队首，队首 URL 地址先进行爬取. 当页面信息获取完后，清除该 URL 地址，并将下个 URL 地址提取到队首来，新加入队列的 URL 地址添置到队列尾部. 每一次都是从队首中获取 URL 地址，一直到整个队列中没有 URL 地址，则停止程序的运行.

3.3 解析网页

对于每个运用需求的不同，爬虫也需要进行进一步的内容判断. 对于类似图片、音乐、pdf 等不同的文件类型，先进行图片地址获得，但是不加入 URL 列表队，程序段如下:

```
$conn = file_get_contents($url);
$preg = "# <div><a target=\"_blank\" href=\"(.*)\"><img src=\"(.*)\" alt=\"(.*)\"/></a></div>#iUs";
preg_match_all($preg,$conn,$result);
var_dump($result);
```

先通过 file_get_contents()函数获取整个页面信息，其中运用了正则匹配的方法. 然后，将页面信息直接下载存入数据仓库并记录数据的位置，命名一定要规范，以便于后面的使用. 对于 TEXT 等文本类型的数据，我们将要进行下部操作以获取内容，用相同的正则表达式的方式获取并存入数据表中. 而对于一些获取超时并且报错的页面，加入错误队列中，进行后续分析.

3.4 程序实现

整个程序实现依赖于数据库，因为对于一些有用的信息要进行保存，方便下一次操作和结果分析. 整个程序运行时拥有 3 个队列: 备用队列、运行队列、错误队列. 其中，备用队列是用于当运行队列满员时，爬虫程序新发现的 URL 的集合和从爬虫运行队列中提出来的 URL 地址; 运行队列是程序正在处理的 URL 列表队; 错误队列是程序在爬取页面出错或读取数据超时等情况的 URL 地址集.

在整个程序运行时，进行爬取的 URL 页面地址只在 URL 列表队中获得，并且每次仅有一个. 每次完成的 URL 地址，进行将其从列表队中剔除，并在数据库中记录该 URL 已进行数据爬取.

3.5 算法比较

通用爬虫的爬取算法是从一个根节点的 URL 开始，获取该 URL 网页信息后将提取到的相关 URL 放入爬取队列中，当获得到一定数量的 URL 或者发现目标 URL 时停止程序，此类算法属于盲目搜寻法，即对所有相关节点进行遍历，直到找到目标或者返回空值[8]. 因此，对一些像图片、音频等信息含量密集度高且具有一定结构的数据常常无法有效获取，且常返回大量与目标数据不符的数据，为后面去除数据工作加大难度. 经过实验证明，通用爬虫算法对那些用 AJAX 加载的页面不能较好的支持. 为此本爬虫算法中添加了对 URL 的合理分析和过滤，有效地改良了爬虫的性能.

4 爬取实验数据及结果

经过实验验证，本爬虫对于静态页面均能进行有

效爬取和解析, 对于一些运用 AJAX 技术加载的页面则要进一步添加触发器对地址进行请求, 触发器程序段如下:

```
function curlRequest($url)
    {
      $ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER,
1);
curl_setopt($ch, CURLOPT_HEADER, 0);
      $response = curl_exec($ch);
      curl_close($ch);
      return $response;
    }
```

本爬虫在 12396 湖南农业信息服务网、议论纷纷、佐名片等多个网站进行了爬取实验, 爬取不同的网页所得到的效率有所不同. 而且, 爬取的效率也与对方服务器性能、宽带、用户访问量以及爬虫程序所在主机的硬件配置等因素相关. 最终平均所有实验结果得出平均值: 在带宽 4M, 内存 2G 的主机上解析页面并且下载页面中图片的效率为: 每分钟 12 张质量较高的图片. 通过添加对 URL 分析过滤功能后, 对于一些没有用的 URL 地址进行剔除, 能更好的提高爬虫的效率. 在整个爬取过程中, 有大约 80%的时间处在网页的下载过程中, 其中包括, 网页 HTML 节点获取, 图片下载, 数据整合, 数据库连接等, 大约 20%时间用于数据分析、数据分类、与网络通信和内存释放等计算机处理. 通过爬取比对实验得出: 获取数据正确率与通用爬虫相比提高约 32%, 错误 URL 减少约 55%, 数据整合时间减少了近 37%.

5 结语

在网络数据量急剧膨胀的今天, 如何利用 Web 数据挖掘的技术高效获取所需信息成为一种必需. 而对于 Web 挖掘来说 Web 爬虫运用最广泛, 本文中主要运用 Web 结构挖掘, 通过对链接中寻找有用的信息的方法, 对爬取出来的信息进行抽取, 然后进行分类或聚合后最终获得有用的信息. 再对这些信息做进一步分析, 还可得到用户搜索方式和行为等其他信息资源, 为 SEO 等相关操作提供数据支持.

参考文献

1 苏新宁,杨建林.数据挖掘理论与技术.北京:科学技术文献出版社,2003.15–18.

2 俞勇,薛桂荣,韩定一.Web 数据挖掘.北京:清华大学出版社,2012.42–49.

3 罗刚,王振东.自己动手写网络爬虫.北京:清华大学出版社,2010.158–160.

4 王常红要,等.基于 HTML 标记用途分析的网页正文提取技术.计算机工程与设计,2010,31(24):5187–5191.

5 于海涛.Web 挖掘技术在搜索引擎中的应用.齐齐哈尔师范高等专科学校学报,2009,(6):32–37.

6 Tsuda K, Kurihara K. Graph mining with variational Dirichlet process mixture models. Proc. of the 8th SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics. New York: ACM Press,2008:332–342.

7 朱丽红,赵燕平.Web 挖掘研究综述.情报杂志,2004(7):358–341.

8 徐远超,刘江华,刘丽珍.基于 Web 的网络爬虫的设计与实现.微计算机信息,2007,23(7):119–121.