基于 Python 的简单网络爬虫的实现

戚利娜，刘建东
（吉首大学张家界学院，湖南 张家界 427000）

摘　要：从网络爬虫技术的定义出发，介绍了爬虫架构以及与爬虫架构相关模块的功能和实现方法，并给出了以案例的形式实现了简单网络爬虫算法过程。

关键词：Python 技术；网络爬虫

1　引言

随着互联网技术的发展，网络信息过载已经成为不争的事实。因此，如何有效提取互联网信息，并将这些信息充分利用已经成为一个巨大的挑战。搜索引擎成为用户检索信息的工具。而所谓搜索引擎的重要组成部分，网络爬虫是一种按照一定的规则，自动地抓取万维网信息的程序和脚本。

网络爬虫技术的应用范围较广。例如，可将爬虫获取的有价值数据资源进行整合，实现不同类型的垂直领域的应用，图书价格比对，新闻主题聚合网等。特别要提到的是，现今大数据时代，机器学习算法需要大量的网络数据作为训练数据，一定程度上说，训练数据的质量高低决定了机器学习算法效果的差异。而获取训练数据的方法除了其他典型的统计数据外，网络爬虫提取数据也是其中主要的方法。网络爬虫技术是目前大数据时代的重要基础应用。

网络爬虫技术分为以下几类：

（1） 通用网络爬虫；

（2） 聚焦网络爬虫；

（3） 增量网络爬虫；

（4） Deep Web 爬虫。

上述爬虫技术的关键技术是类似的，就简单网络爬虫进行讨论。首先介绍简单网络爬虫的架构，然后分别介绍架构的 3 个模块以及实现方法，最后以具体案例谈谈如何利用 Pathon 实现简单网络爬虫。

2　爬虫架构和流程

网络爬虫的架构分 3 部分：（1）爬虫调度端；（2）网络爬虫主程序；（3）目标数据。而爬虫主程序又由 3 个模块协作完成，它们分别是：1）Url 管理器，主要负责提取待爬虫的 Url 地址，并且删除已经爬虫的 Url 地址；2）网页下载器，主要根据 Url 地址，从万维网上下载指定地址的网页内容；3）网页解析器，根据网页内容将有价值的信息解析。具体的架构图如图 1 所示：

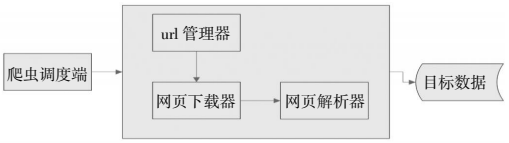

图 1　爬虫架构图

大致流程是爬虫调度端调用爬虫主程序来获取目标数据。具体如图 2 所示，首先调度器询问 URL 管理器，是否存在待爬取的 URL，如果返回肯定的结果，则调度器从 URL 管理器得到第一个待爬取的 URL 地址，调度器通过 URL 地址利用下载器下载网页内容，然后将网页内容发送到解析器，解析有价值数据和新的 URL，调度器获取有价值数据后传送到应用程序，将新的 URL 增加到 URL 管理器。上述过程无限循环。

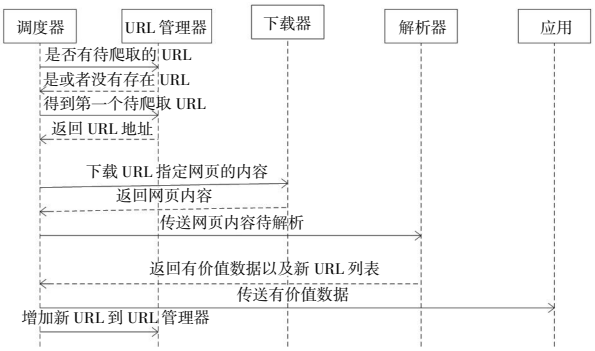

图 2　爬虫时序图

3　爬虫模块以及实现方法

根据上节架构图，分别讨论爬虫主程序的 3 个模块以及实现方法。

3.1　Url 管理器模块功能以及实现

Url 管理器的主要作用是管理待抓取的 Url 集合和已被

作者简介：戚利娜（1981–），女，讲师，研究方向：计算机应用技术；刘建东（1978–），男，讲师，研究方向：计算机应用技术。

收稿日期：2017–01–23

抓取的 Url 集合。最终目的是为了防止重复抓取和循环抓取。例如，假设有两个网页 Page_A 和 Page_B，这两个网页的网址分别是 url_A 和 url_B，并且两个网页相互链接，即 Page_A 链接了 Page_B，而 Page_B 也链接了 Page_A。当通过网页下载器，网页解析器将 Page_A 的内容解析后，获取了 Page_A 的链接页面 Page_B，爬虫程序响应下载和解析 Page_B 内容，而 Page_B 又链接了 Page_A。如果没有 Url 管理器，爬虫程序会继续下载和解析 Page_A 内容，重复上次过程，不断循环，无法终止。

Url 管理器的功能需要满足：（1）判断新的 Url 是否存在待爬 Url 集合中；（2）添加新的 Url 到待爬 Url 集合中；（3）判断是否还有待爬 Url；（4）获取待爬 Url；（5）将已爬 Url 移入已爬 Url 集合中。

实现 Url 管理器的方法可以分为 3 类：（1）用内存存储 Url 地址，该方法适合 Url 数据量较少的情况。具体而言，将 Url 存入内存的两个集合，分别表示待爬集合和已爬集合，在 Python 中用 Set() 实现。因为 Set() 集合本身具有清除重复值的效果；（2）可以使用关系数据库来实现，该方法适合永久存储。例如，建立表名为 Urls 的表，其中有两个字段，分别表示 Url 地址以及是否被爬取；（3）存储在缓存数据库 redis 中，适合存储大数据量的 Url 地址，同样的用 set 集合表示，因为 redis 支持 set 集合。文中实验的 Url 数据不多，因此采用内存存储的方法实现。

3.2 网页下载器实现

网页下载器是爬虫程序的主要核心模块，因为只有将 Url 对应网页下载到本地才能继续后续的分析与处理。网页的内容一般是 HTML 格式，而下载到本地时以字符串的格式存储。目前 Python 支持的网页下载工具有两类：（1）urllib2，Python 官方支持的基础模块；（2）request，第三方包，具有非常强大的功能。因为文中针对的是简单的网络爬虫，所以选择 urllib2 来实现。具体的实现代码如下：

```
 import urlib2
reponse = urllib2.urlopen(url)
Print response.read()
```

3.3 网页解析器实现

网页解析器即从网页内容中提取目标数据的一种工具。解析器的输入参数是下载的 HTML 格式的网页字符串内容，输出有两个部分：（1）被解析网页链接 Url；（2）目标数据。

Python 支持的网页解析器有以下几类：（1）正则表达式；（2）html.parse；（3）BeautifulSoup；（4）lxml。正则表达式是将网页内容当作字符处理，解析复杂内容的网页具有较高的失误率。其余 3 种解析器是基于网页的 BOM 结构进行解析，具有简单、易学等特点。此外，BeautifulSoup 结合了其他两个解析器，功能强大。本文选择使用 BeautifulSoup 解析器来处理。而 BeautifulSoup 属于第三方插件，需要安装。

利用 BeautifulSoup 获取网页内容的主要方法如表 1 所示：

表 1 BeautiulSoup 解析网页的方法

方法名	说明
BeautifulSoup（wordName,method,charset) l	构造对象，参数分别是被解析网页文档；解析规则，默认 html.parse；字符集,默认 utf-8。
find_all (name,attrs,string)	根据节点名称、属性、内容搜索所有节点
find (name,attrs,string)	根据节点名称、属性、内容搜索第一个节点

4 具体案例

以“百度百科爬取爬虫词条相关的 1000 个页面”为案例，说明简单爬虫的实现。为了能进行爬虫，需要以下准备工作：（1）选择第一个 Url 地址，在本节案例中选择第一个 Url 地址为 http://baike.baidu.com/item/网络爬虫?fr=aladdin，（2）了解到网页中链接的 Url 的格式为“/item/网络爬虫?”相对路径，因此解析的 Url 地址需要加上“http://baike.baidu.com”前缀以补充完整，（3）发现网页使用的字符集为 utf-8。做好上述准备工作后，利用上节提到的模块方法进行解析。具体的算法如表 2 所示：

表 2 爬虫核心算法步骤

本算法步骤是爬取百度百科爬虫词条相关的 1000 个页面，具体参数是：root_url 的参数值为” http://baike.baidu.com/item/网络爬虫?fr=aladdin”，表示爬虫初始 url	
1	Url 管理器待爬集合添加 root_url
2	初始爬虫次数为 1
3	WHILE url 管理器还存在待爬 url
4	new_url = Url 管理器得到待爬 url
5	html_content = 网页下载器下载页面 (new_url)
6	新的 url,新数据=网页解析器解析内容（new_url,html_content）
7	Url 管理器添加新的 url 到待爬集合
8	收集数据
9	爬虫次数递增 1
10	If 爬虫次数达到 1000 那么
11	break;
12	END

5 结语

网络爬虫技术具有较高的应用价值，从应用角度，可以对收集的数据做进一步处理；从算法效率角度，存在改进空间。

参考文献

[1] 罗刚．自己动手写网络爬虫 [M] ．清华大学出版社，2010.