基于豆瓣电影的数据采集的设计与实现

黄泽辉 张桂花
四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘 要】在这个大数据时代，豆瓣网已经成为最重要的社交网站之一，笔者针对豆瓣电影网的特性设计并实现了对网页内容的信息采集。通过 python 进行爬虫获取豆瓣电影中的排名，电影，时间，导演，评分版块，对电影进行降序排名，并将目标数据存储到电子表格中，获得豆瓣电影排行榜单，可用于后续的数据研究。

【关键词】python；豆瓣电影网；流程设计；网络爬虫

引言

随着互联网时代的快速发展，很多时候我们不用再自己去过滤和筛选信息，通过一些特定的点位我们就能获取到自己想要的数据，例如通过分析一篇博客我们从作者，标题，就能够预测出整篇文章的写作内容。在国内有许多对社交网站进行数据采集和分析的例子，比如常常听到的新浪微博。而豆瓣网，则是一个以书影音为起点的社交网站，截止到二零一二年八月份，豆瓣就已经有超过一亿的月度覆盖独立用户数，一点六亿日均页浏览量。本次设计基于 Python 的库方法，通过对网页数据进行采集分析等流程，最后将结果存储在表格中。

1 数据采集

数据采集是一个由浅入深的过程，其作用是通过网络爬虫获取页面内需要的内容，用于后续进行数据地分析。关于网络爬虫的实现方法，可以分为以下几种主要类型：(1) 基于套接字的爬虫程序编写：它是效率最高的底层方法，但开发效率最低。(2) 基于请求 – 响应协议编写爬虫：主要使用与协议相关的高开发性操作。效率高，但受到各种限制。(3) 无接口浏览器爬虫：它是一个基于 WebKit 的服务器端，它实现对 Web 的支持，而不需要浏览器的支持。其开发效率高，但执行速度慢。（4）基于 Selenium 的界面浏览器爬虫：它是用于网络应用程序测试的工具。试验直接在浏览器中运行，就像真正的用户正在操作一样。支持的浏览器包括 IE（7、8、9）套件等，其开发效率较高，但执行速度最慢，因为每次对网页内容进行爬网时都应该打开浏览器，导致执行速度最慢 [1]。综上，此次实验采用基于请求 – 响应协议编写爬虫。而爬虫语言的选择并不是一成不变的，无论是 JAVA，PHP 或是其他语言，都可以实现网络爬虫，前者生态圈庞大但是本身比较笨重，后者本身没有多线程概念，异步支持少，并发不足，而 Python 本身语言简洁，拥有丰富的库，所以在爬虫时它成为实验首选。

2 豆瓣网数据采集设计

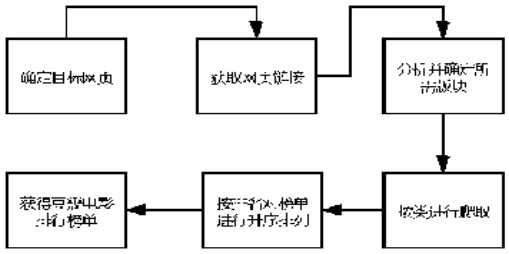

图 1 数据采集流程设计图

Fig.1 Data collection flow chart

在网上进行查找后，不难发现要查找豆瓣网电影的相关资料有 3 种网页地址：（1）豆瓣网首页；（2）豆瓣电影分区页面；（3）豆瓣电影排行榜。在这之中，第一个豆瓣网页首页面，虽然在网页中能够通过链接转到豆瓣电影的页面，但是网页爬取是采集链接中的网页界面，通过该链接只能索引到豆瓣主页但无法爬取到电影版块的信息，所以并不适合这次采集。第二个豆瓣电影的主页，该页面包含了豆瓣电影的所有内容，以主题模块进行分割，包含排行，口碑，年度榜单，最新最热等榜单，但是该页面版块分类过于繁杂，很难找到一个固定的指标对所有内容顺序爬取，并不适合此次实验要求，第三个豆瓣电影排行前二百五十的榜单页面，该页面通过排名将所有电影串联起来，每一条排名都带有对应的标签，内容简洁。所以综上，此次实验选择使用第三个网址作为爬取数据的链接。确定采集页面网址之后，则需要获取网页的用户代理和请求头，通过打开浏览器按“F12”，可以查看到网页源码，找到网络下的文档

选项，在请求头一栏可以在底部看到主机和用户代理，将页面中的服务器域名和用户代理添加到代码的请求头中即可实现网页的选取。除此之外，在浏览器中此次采集数据确定页面的代码如下：headers = {

‘user-agent’：‘Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML， like Gecko) Chrome/59.0.3071.115 Safari/537.36’，

‘Host’：‘movie.douban.com’

}

爬取用户代理过程中容易遇到基于它的反爬，服务器后台会针对访问的用户代理进行统计，在一定时间内同一用户代理的访问次数超过特定值时，会被封禁 IP，从而造成无法对网页进行爬取数据的情况，在采集网站数据时，频繁更换用户代理可以有效避免对应的反扒机制，库中 fake-useragent 包对更换用户代理提供很好的支持，在库中引入该包即可使用，通过每次发送请求时依靠 random 获取随机用户代理，即可实现用户代理的不停更换。在使用此包时需要注意生成随机用户代理后该方法中存储的用户代理列表也会发生变动，所以需要对本地的用户代理列表进行更新，否则会在程序运行的过程中报错。

在爬取数据前首先要确定爬取的内容以及爬取下来的数据是否对后续的研究有用，此次研究主要是对豆瓣 TOP250 电影进行信息采集，所以此次设计摒弃了主题，影评这些内容繁多的部分，选择从电影，发布时间，导演，评分，排名五个方面进行爬取，通过排名将所有电影按照升序的方式进行排列，因为排名和评分往往是观众在选择观看的电影时最重要的一个指标。在爬取的过程中，通过类来进行数据的采集，比如当爬取内容途中需要采集电影名字时，在网页源码中找到对应的部分，即可找到需要选择的类为“hd”，通过对爬取部分类的选取，即可获得每个类中的页面内容，此次爬取实现的具体代码为：soup = BeautifulSoup(res.text， “lxml”) div_list = soup.find_all(‘div’， class_=’hd’) 此处运用到了 beautifulSoup 这个库，该库是一种灵活方便，高效处理，支持多类解析器的网页解析库，使用此库可以不用正则表达式就完成对网页信息内容的提取，在按类爬取完成后，采用循环遍历网页中查找到的类函数，再在所需元素后面添加 <> 中的字母即可提取其中的元素，如爬取过程中需要提取 ... 中的元素时，在元素后添加 .p 即可，此次实现提取元素的代码如下（以电影名为例）：

```
for each in div_list:
movie = each.a.span.text.strip();
movie_list.append(movie);
```

对豆瓣电影网页进行数据采集和分析后，需要将爬取下来的数据进行存储，所以这里引入了 xlwt 包，此包非常实用，是进行爬取成功后写入表格时不可或缺的工具，其对应的还有一个 xlrd 的包，该包主要实现表单的读取，xlwt 既可以实现指定表单的写入，还可以实现指定单元格的写入，进行数据保存首先要导入该包，导入后利用 file = xlwt.Workbook() 新建一个表单文件，然后利用该包中的其他方法实现表单头的列名设置，五列列名分别为排名、电影、时间、导演、评分，再通过循环将每一列对应的数据填写进去，这样，我们就通过循环完成电子表格的填写，并将数据填写在 data.xls 中，完成豆瓣电影网数据的爬取分析和存储。通过这样的方法所爬取下来的数据，可以运用到大数据分析或挖掘中，是数据处理中很好的素材。

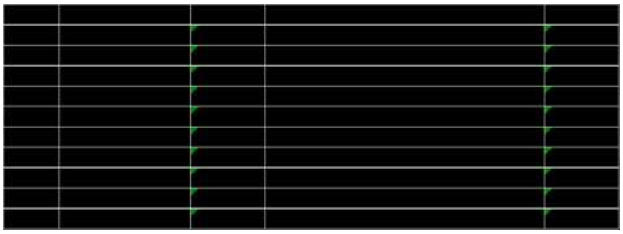

图 2 爬取结果图

Fig.2 Crawling result chart

这一步需要注意的是通过 xlwt 只能写入复合类型表格无法写入 XML 类型表格，虽然运行过程中 XML 类型表格能够生成但无法用 excel 打开（测试环境为 2013 版本）。此外，在运用 xlwt 中的公式包时，其生成的复合类型表格虽然能够打开但是公式并不稳定，在查看完进行关闭时会有弹窗提示是否保存修改，后续若要读取此类文件时，则必须对复合类型表格进行保存，否则其中涉及公式的单元格无法被读取（无论是数值还是公式）。

3 实验总结与体会

此次实验主要有 2 个爬取豆瓣电影网页的体会：（1）在爬取前要先查看库中是否有自己所需的包，如果没有对应的包，在导入时会报错，并且后续在使用包中的方法也会失败，比如要实现访问网络，就要先在库中导入 requests 包，在使用 requests.get 时，该方法就会将键和值放入一个字典中，通过 params 参数来传递，其作用相当于 urllib.urlencode。而如果我们要实现数据的可视化时，则需要引入 Matplotlib，它既是强大的可视化工具，也是一个作图库。（2）在爬取页面内容时，先要对爬取页面进行查找，确认爬取的内容是合法的且没有恶意占用资源，分布式拒绝服务（DDoS）是当今危害最大的网络安全攻击之一。最近，恶意网络爬虫已被用于对万维网上的网站执行自动 DDoS 攻击 [2]。爬虫不能涉及个

人隐私，如果爬虫程序采集到公民的姓名、联系方式、身份证号、家庭住址、行踪、财产状况等个人信息，并将其用于非法途径，则构成违法行为。虽然爬虫协议不是法规、标准及合约，但与爬虫协议相关的国内外八个判决[3]表明，中外法院都认同他是网络环境下著作权人可以采取的有效控制访问的技术措施，构成搜索行业的惯例。爬虫应该建立在合法的基础上，爬取所获得的信息才有利用价值。

4 结语

本文以豆瓣电影网为例，设计并实现了对特定数据的分析和采集。该数据可用作后续对电影影响及受众的数据挖掘，具有一定的研究价值。未来优化会考虑使用分布式或多线程以提高数据采集的效率。

【参考文献】

[1] Tian Fang,Tan Han,Cheng Zhang,Ya Juan Yao. Research and Construction of the Online Pesticide Information Center and Discovery Platform Based on Web Crawler[J]. Procedia Computer Science,2020,166.

[2] Dusan Stevanovic,Aijun An,Natalija Vlajic. Feature evaluation for web crawler detection with data mining techniques[J]. Expert Systems With Applications,2012,39(10).

[3] 范长军 . 行业惯例与不正当竞争 [J]. 法学家 ,2015(05):84-94+178.