# 计算机科学与工程学院

# （网络空间安全学院）

# 实习实训报告书

学生学号： 2019081310017

学生班号： 2019080105

学生姓名： 万理

学生专业： 计算机科学与技术

实习单位：四川华迪信息技术有限公司

指导教师： 曹旻涵

填表说明

1、实习实训旨在让学生将在校期间学习的理论知识与工程实践相结合，提升学生的实践能力、掌握工程开发的过程和方法，并提高学生应用理论解决实际工程问题的能力。

2、本表中的专业实践总结报告请用A4纸撰写，正文小四号字，宋体1.5倍行距；字数在5000字以上。

一、专业实习总结报告

|  |
| --- |
| **一、实习项目名称：**imdb电影网站数据分析  **二、实习时间、地点：**2021.7.3-2021.7.12**线上**  **三、实习目的和任务:**   1. 熟悉各类主流软件开发技术的配置、开发、部署应用﹔ 2. 理解项目管理知识与应用，培养知识产权及其管理意识﹔ 3. 能够通过集成软件开发主流技术构建完整的企业级应用解决方案; 4. 能够熟练使用主流设计工具完成应用系统模块设计与建模工作﹔   5)掌握大数据关键技术﹐如Hadoop生态系列技术:HDFS MapRedce等; Spark生态系列技术RDD 、Spark SQL & Datakame等;  6)能够熟练使用Pythn 、Java语言和各种开发技术实现大数据应用的解决方案;  7)掌握Python网络数据采集技术，大数据分析及可视化技术;  **四、实习内容和要求：**  **1.第一阶段：大数据核心开发技术**  学习python的基本语法，了解爬虫的原理与实现，flask框架的实现，并进行简单实践。随后是hadoop与spark的环境安装与配置。最后学习了如何用spark来分析处理数据。  **2.第二阶段：大数据综合应用**  进行了组内的分工，决定进行电影网站数据分析的项目，同时制定计划，分块分期完成项目。   1. **实习工作成果：**   我在项目中担任爬虫工程师的角色。在编写代码前，向数据分析师和项目经理确定了爬取的需求：网页为imdb的电影详情页面。  每个电影详情页面需要提取的内容如下：  电影名称，时长，类型，发行时间，简介，导演，编剧，演员，评分，评分人数，短评数，影评数，Metascore，语言，原产国，制作公司，估计预算，全球票房，获奖及提名数。  每个电影详情页面的url结构为https://www.imdb.com/title/imdbid。其中，imdbid是每部电影对应的七位数字。  link.csv文件包含62000余部电影imdb id，依据这些id生成url进行爬取。link.csv来自于MovieLens发布于2019年12月的ml-25m数据集。ml-25m数据集网址：<https://grouplens.org/datasets/movielens/25m/>    通过requests库依据这些id生成的url获取每部电影详情页面的HTML。  通过logging库对HTML的获取情况进行日志记录，便于debug和查看爬取进度。    借助bs4库的BeautifulSoup模块解析和提取 HTML数据。使用第三方的解析器lxml，比Python默认的解析器速度更快。主要用find（），find\_all（），find\_next()等函数在解析生成的文档树中提取感兴趣的信息。  提取时还用到了unicodedata库对一些信息进行了规范化处理。页面的缺失信息则通过异常捕获进行置空。  每个详情页的信息保存在一个list中，作为一行写入csv文件，所以要用到csv库。  为了爬取中断后从断点继续爬取，引入了white\_list.txt文件保存已经处理过的id及其imdbid，重新运行程序会跳过这些id，不用从头来过。  相应的，也有black\_list.txt文件。用来保存获取HTML失败的电影id及其imdbid，可以后面单独处理。一般是两类：一是网页已经无法访问，即404；二是imdbid不足七位且未补0到七位，这种情况将black\_list.txt另存为csv文件，在csv文件里用公式TEXT(B2：,REPT("0"，8))即可补0。  简单来说，black\_list.txt文件中的电影可以不处理。处理的话，只需要转csv进行补0，然后作为给程序提供imdbid的文件，再运行一下程序即可。  考虑到耗时和够用就行，最终只爬取了26000余部电影的信息，没有爬完。    **六、总结及心得体会：**  由于我在组内负责爬虫工作，所以我先就爬虫的学习与实践进行一下总结，然后再谈谈其他方面的心得体会。  对于我们实训的这个项目，为了和老师的例子区别开，就放弃了豆瓣。起初我尝试爬取猫眼网站的电影信息，但由于反爬机制会跳转到美团验证中心，就打算换个网站。比较权威的电影网站让我想到了imdb，虽然需要翻下墙，但试了下，发现没有任何反爬机制，甚至爬取时都不需要休眠，就很棒。  在编写代码前，向数据分析师和项目经理确定了每个电影详情页面需要提取的内容如下：  电影名称，时长，类型，发行时间，简介，导演，编剧，演员，评分，评分人数，短评数，影评数，Metascore，语言，原产国，制作公司，估计预算，全球票房，获奖及提名数。  能够正常获取imdb的HTML文件后，发现有些信息用正则表达式不好定位。参考了Beautiful Soup的官方文档和一些博客文章，再借用第三方库lxml用于解析HTML，最后成功通过find（），find\_all（）等函数爬取了所有目标数据。  有些遗憾的是，分布式爬虫没来得及弄。所以数据爬取耗时比较长，大概花了8个小时爬了26000余部电影的目标信息。  有空我打算自己做一个实时分析B站数据的项目，一方面，把分布式爬虫等技术实践一下；另一方面，把其他同学负责的数据分析与网页可视化也自己做一下。最后部署到自己的服务器上，美滋滋。  一.什么是爬虫  在学习爬虫之前我只对爬虫有个概念性的认识。通过向服务器发送请求获取服务器传回信息，再根据其提取所需的信息。原理虽然简单，但是涉及的细节非常多，从一个坑爬出来又掉进另一个坑。  二.post和get  post和get是两种向服务器发送请求的方式，有些http基础的同学应该都清楚他们的用处，在写爬虫时，涉及到同服务器交互肯定是离不了这两种方法的。其中get请求用于一系列对服务器数据没有影响的操作，如获取html，检索等，而post一般是用于对服务器数据有影响的功能，如提交表单等。如果只是简单爬取信息，一般用get方法。除此之前，get方法的安全性较Post方式要差些，如果需要提交包含机密信息的话，建议用Post提交方式，但get的执行效率比post方法好。  三.编码方式  这是跌进的第一个坑。源于对一个网页爬取时，不管怎么样都失败，浪费了将近一天时间才发现是编码混乱的原因造成。目前大多数中文网站采用GBK，GB2312，UTF-8这三种编码方式进行编码，详细的编码原理可查阅其他资料。通常在获取一个网页的html代码后都会先根据其编码方式进行转码，下面用python对从path获取的网页根据utf-8的方式进行转码。  html=urllib.request.urlopen(path).read().decode("utf-8")  正常来说，一个网页只会有一种编码方式，通过上述的代码就能完成对其的转码。但是，如果在一个网页里存在两种编码方式的字符会怎么样？对，就是这个原因导致我在这个问题上卡了将近一天。对我这个刚接触爬虫的人，怎么也不会想到一个网页里会存在两种编码方式，所以我无论用哪种方式进行转码都会失败。当找到原因后，使用了lxml库自带的转码方法后可以顺利完成转码。  html=urllib.request.urlopen(path).read()  htmletre = etree.HTML(html)  result = etree.tostring(htmletre, encoding='utf-8', pretty\_print=True, method="html")  但是后来发现，这种方法依然有很大的问题。就是如果一个网页包含有两个编码的字符时，如果对其中一种编码方式进行转码，当第一次遇到另一个编码的字符时，转码就会结束，最后导致结果缺失，对于爬取信息来说是不可接受的，特别是当另一个编码的字符出现在前半部分时会导致后半部分信息全部缺失。放弃这种方法后，通过重新分析网页的代码，发现另一种编码方式只会出现在极个别地方，如果放弃这部分编码，只对大部分有效信息进行转码也是可以接受的。查阅相关文档后，利用以下代码就可以实现对大部分编码进行转码。  html=urllib.request.urlopen(path).read().decode('gbk',errors='ignore')  四.正则表达式  在获取html代码后，就可以利用正则表达式进行信息提取了，正则表达式相关的知识这学期我正好在一门教Unix的课上学了一些。这里想要说一下lxml库的使用，虽然大部分信息都可以通过正则表达式进行提取，但如果需要对代码本身的结构进行提取和分析就需要用到lxml库了。虽然正则表达式也能进行相关的操作，在不知道lxml库之前也利用正则表达式对标签层次关系进行过分析，但还是推荐使用lxml库，个人觉得比正则表达式简单方便。  五.Chrom浏览器  Chrome浏览器对于写爬虫非常有用，Chrome自带的开发者工具能帮助分析网页的源码、资源和数据包。特别是要写表单提交一类功能时，Chrome自带的抓包功能非常好用，通过分析包结构可以实现各类复杂表单的模拟提交。  六.爬虫与机器学习  在学习爬虫前，一直在学习机器学习。通过这段时间对爬虫的学习，我觉得机器学习跟爬虫能够很好的结合起来。一方面对某些模型的训练可以通过爬虫来获取足够多的数据样本，另一方面，因为各种网站的结构有很大的差异，利用机器学习可以对各种结构的网页进行准确分析和信息提取，最终达到相辅相成的目的。  七．常用第三方库  对于爬虫初学者，建议在了解爬虫原理以后，在不使用任何爬虫框架的情况下，使用这些常用的第三方库自己实现一个简单的爬虫，这样会加深对爬虫的理解。  urllib和requests都是python的HTTP库，包括urllib2模块以巨大的复杂性代价获取综合性的功能。相比于urllib2,Requests模块更能简约的支持完整的简单用例。关于urllib和requests的优缺点和区别，大家可以去网上查一下。  BeautifulSoup和lxml都是python页面解析的库。BeautifulSoup 是基于 DOM 的，会载入整个文档，解析整个DOM树，因此时间和内存开销都会大很多。而lxml只会进行局部遍历，使用xpath能够很快定位标签。bs4 是用 python 写的，lxml 是 c 语言实现的，也决定了lxml比bs4要快。  八．爬虫框架  python常用的爬虫框架就是scrapy和pyspider两个。 关于框架的使用方法及详细介绍，可参考官方文档。  九．动态页面渲染  1. url请求分析  （1）认真分析页面结构，查看js响应的动作; （2）借助浏览器分析js点击动作所发出的请求url; （3）将此异步请求的url作为scrapy的start\_url或者yield reques再次进行抓取。  2. selenium  Selenium是一个Web的自动化测试工具，最初是为网站自动化测试而开发的，类型像我们玩游戏用的按键精灵，可以按指定的命令自动化操作，不同是Selenium可以直接运行在浏览器上，它支持所有主流的浏览器(包括PhantomJS这些无界面的浏览器)。  Selenium可以根据我们的指令，让浏览器自动加载页面，获取需要的页面，甚至页面截屏，或者判断网站上某些动作是否发生。  Selenium自己不带浏览器，不支持浏览器的功能，它需要与第三方浏览器结合在一起才能使用。  3. phantomjs  使用selenium调用浏览器进行抓取页面时，由于要执行打开浏览器并渲染页面的操作，当进行大规模数据抓取时效率较低，无法满足需求。这时我们可以选择使用phantomjs。  PhantomJS是一个基于Webkit的"无界面"(headless)浏览器，它会把网站加载到内存并执行页面上的JavaScript，因为不会展示图形界面，所以运行起来比完整的浏览器更高效。  如果我们把Selenium和PhantomJS结合在一起，就可以运行一个非常强大的网络爬虫了，这个爬虫可以处理JavaScript、Cookie、headers，以及任何我们真实用户需要做的事情。  4. splash  Splash是一个Javascript渲染服务。它是一个实现了HTTP API的轻量级浏览器，Splash是用Python实现的，同时使用Twisted和QT。Twisted（QT）用来让服务具有异步处理能力，以发挥webkit的并发能力。  python连接splash的库叫做scrapy-splash，scrapy-splash使用的是Splash HTTP API， 所以需要一个splash instance，一般采用docker运行splash，所以需要安装docker。  5. spynner  spynner是一个QtWebKit的客户端，它可以模拟浏览器，完成加载页面、引发事件、填写表单等操作。  十．爬虫防屏蔽策略  1. 修改User-Agent  User-Agent是一种最常见的伪装浏览器的手段。  User-Agent是指包含浏览器信息、操作系统信息等的一个字符串，也称之为一种特殊的网络协议。服务器通过它判断当前访问对象是浏览器、邮件客户端还是网络爬虫。在request.headers里可以查看user-agent，关于怎么分析数据包、查看其User-Agent等信息，这个在前面的文章里提到过。  具体方法可以把User-Agent的值改为浏览器的方式，甚至可以设置一个User-Agent池（list，数组，字典都可以），存放多个“浏览器”，每次爬取的时候随机取一个来设置request的User-Agent，这样User-Agent会一直在变化，防止被墙。  2. 禁止cookies  cookie其实是储存在用户终端的一些被加密的数据，有些网站通过cookies来识别用户身份，如果某个访问总是高频率地发请求，很可能会被网站注意到，被嫌疑为爬虫，这时网站就可以通过cookie找到这个访问的用户而拒绝其访问。  通过禁止cookie，这是客户端主动阻止服务器写入。禁止cookie可以防止可能使用cookies识别爬虫的网站来ban掉我们。  在scrapy爬虫中可以设置COOKIES\_ENABLES= FALSE，即不启用cookies middleware，不向web server发送cookies。  3. 设置请求时间间隔  大规模集中访问对服务器的影响较大，爬虫可以短时间增大服务器负载。这里需要注意的是：设定下载等待时间的范围控制，等待时间过长，不能满足短时间大规模抓取的要求，等待时间过短则很有可能被拒绝访问。  设置合理的请求时间间隔，既保证爬虫的抓取效率，又不对对方服务器造成较大影响。  4. 代理IP池  其实微博识别的是IP，不是账号。也就是说，当需要连续抓取很多数据的时候，模拟登录没有意义。只要是同一个IP，不管怎么换账号也没有用，主要的是换IP。  web server应对爬虫的策略之一就是直接将IP或者整个IP段都封掉禁止访问，当IP被禁封后，转换到其他IP继续访问即可。方法：代理IP、本地IP数据库（使用IP池）。  5. 使用Selenium  使用Selenium来模拟人工点击访问网站，是种很有效的防止被ban的方式。但是Selenium效率较低，不适合大规模数据抓取。  6. 破解验证码  验证码是现在最常见的防止爬虫的手段。有能力的小伙伴可以自己写算法破解验证码，不过一般我们可以花点钱使用第三方打码平台的接口，轻松实现验证码的破解。  无论是爬虫还是数据分析我都是一个初学者，上述都是这段时间的一个心得和总结，一方面希望对其他同学有一点帮助，另一方面也希望有任何说错的和还能改进的地方大家能够指正。  虽然项目实训时间很短，要求也并不简单，但通过小组内每个人的熬夜狂肝，团队协作，最终还是呈现了一个不错的项目效果，很高兴大家都那么给力。平日的纪律、考勤也十分严格，体会到了公司中工作的辛劳，同时也对陈老师考勤统计的付出表示感谢。另外，曹老师的教学和答疑也为我们的设计项目框架、初期切入及后续推进提供了莫大帮助，十分感谢！ |
| 学生本人签名：万理　 2021年7月12日 |

二、指导老师评语：

1. 学生实践期间的思想品德表现，身心健康与安全生产方面的意识
2. 学生沟通能力、协作能力等综合能力的锻炼情况
3. 学生实践过程中的工作内容、执行情况、主要成果评价

指导教师签字：

（实习单位盖章）

三、实习评分（总分100分）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 评分项 | 项目阶段考评成绩（30分） | | | 项目成员考核（50分） | | 平时表现（10分） | | 创新创业（10分） | 总分 |
| 需求分析（20%） | 分析设计（30%) | 编码与测试(50%) | 各种技术的熟练应用程度与完成质量（60%） | 编码习惯和熟练程度（40%） | 职业素养（50%） | 创新和钻研精神（50%） |
| 得分 |  |  |  |  |  |  |  |  |  |

年 月 日