

****

信 息 学 院

《社交网络技术与应用》

期末大作业 项目报告

**题 目 国际舆情——疫情分析**

**组 员 学号 姓名**

**学号 姓名**

**学号 姓名**

**学号 姓名**

**学号 姓名**

## 实验步骤

### 数据爬取

##### 爬虫代码

代码：

注意：这里用到了URL爬取网页需要的两个第三方库requests和json，如果系统里头还未安装这两个库，请先完成安装再执行下面的步骤。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 3 | 4 | 4 | 3 | 3 |
| 3 | 4 | 5 | 5 | 4 | 4 |
| 4 | 5 | 3 | 3 | 5 | 5 |
| 5 | 6 | 4 | 4 | 6 | 6 |
| 6 | 7 | 5 | 5 | 7 | 7 |

# 导入所需要的模块

输出：

图1-9 输出的数据

##### 分析数据格式，定义自己要保存的数据项，并抽取数据项保存到本地CSV文件

**步骤1** 导入第三方库csv用于解析网页

# 导入所需的Python第三方库

代码：

**步骤2**定义自己要保存的数据项，初始化一个csv写入对象

代码：

import csv

打开Python执行路径，可以看到一个新创建的文件名为“data”的csv文件：

图1-10 新创建的csv文件

**步骤3** 将API返回的相关数据写入CSV文件

代码：

打开Python执行路径文件名为“data”的csv文件，可以看到写入的数据项：

图1-11 csv文件中的数据

##### 设置调用无界面浏览器Chrome所需的环境

下载chrome：https://www.google.cn/intl/zh-CN/chrome/，查看Chrome的版本，如下图为80.0.3987.132版本的Chrome。

图1-12 下载Chrome并查看版本

下载ChromeDriver：http://chromedriver.storage.googleapis.com/index.html

因为selenium要用到浏览器的驱动来控制浏览器模拟用户登录微博，下载与你本机Chrome版本相对应的驱动，chromedriver的版本一定要与Chrome的版本一致，不然就不起作用，如本机的版本为80.0.3987.132，下载地址中并无完全吻合的版本，最后下载了80.0.3987.16版本对应的亦可兼容本机，解压压缩包，找到chromedriver.exe复制到chrome的安装目录下Application，将chromedriver.exe文件的路径并加入到电脑的环境变量中去

图1-13 下载对应版本的ChromeDriver

##### 查看pc端微博登录页面

打开pc端微博登录页面：http://login.sina.com.cn/，右键点击“检查”查看网页元素，从该源码片段中发现用户名输入框的name为username，密码输入框的name为password，登录按钮为input元素且class为W\_btn\_a btn\_34px，如下图所示：

图1-14 查看pc端微博登录页面相关元素

##### 在Sublime Text中创建开发环境（可使用其他开发环境）

左上角点击“File”->“New file”，将文件名改为“weiboSpiderDayKWS.py”（文件名可以任取），在Python文件中编写Python爬虫代码。

##### 爬虫代码

**步骤1**调用无界面浏览器Chrome打开pc端微博登录页面

这里用到了模拟用户登录所需的第三方库，请先完成安装再执行下面的步骤。

# 导入所需要的模块

输出：

图1-15 运行弹出浏览器打开登录页面

**步骤2** 模拟用户名和密码进行登录

代码：

# 模拟用户登录微博

输出：注意中间输出如下页面，无需验证，等待片刻浏览器会自动验证，如果等待10秒之后浏览器没有自动验证，需要手动进行验证

图1-16 用户登录后的验证界面

图1-17 运行窗口输出信息

**步骤3** 打开pc端微博登录页面：https://s.weibo.com/，右键点击“检查”查看网页元素，从该源码片段中发现搜索框为input元素且type为text，如下图所示：

图1-18 查看搜索页面元素

**步骤3** 模拟打开关键词搜索页面，输入你感兴趣的关键词，模拟输入回车键进行搜索

代码：

# 打开搜索页面

输出：

图1-19 模拟键入关键词搜索后的页面

**步骤4** 解析搜索返回结果的页面，获取微博的具体内容

代码：

图1-20 获取微博具体内容的代码

输出：

### 数据预处理与存储

数据读取这里读取的数据是与项目文件同级目录下，或同一个文件夹中。

# 数据读取

输出：

日本：

美国：

印度：

澳大利亚：

成功输出可见读取已经成功

数据去重

通过如下代码，实现文本数据集的重复值处理。

#重复值处理

输出：

运行前数据量为29051条，运行后为25402条，可见经过代码运行后，有4000+条重复数据被去除

美国数据去重21条

印度无重复数据

澳大利亚

无重复数据

中文分词

分词就是将一段连续的文本按照一定的规范重新组合成词序列的过程。我们知道，在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂得多、困难得多。常见的中文分词工具有许多，本实验使用jieba库用于中文分词。

分词后如图

重复值处理

1.字符串匹配去重：drop\_duplicates()；

2.基于中文分词后的关键词字符串去重：先将文本数据转化为关键词字符串，再使用字符串匹配去重。

#重复值处理

重复词处理后数据量由上文的25402条下降至23091条，处理成功

美国数据量较前同样有所下降 处理成功

印度数据量较前同样有所下降 处理成功

澳大利亚数据量较前同样有所下降 处理成功

处理停用词

#去停用词  
for i in range(len(segdata)):

输出：

#保存预处理后的文件

### 词云图

该部分原先的处理流程是，将数据集中的“微博内容”特征列直接利用jiaba的jieba.analyse.extract\_tags函数进行词频统计排序，涉及的代码主要如下：

得到的四个国家的词频数据和词云图如下：

印度：

美国：

澳大利亚：

日本：

从图中红框圈中的数据可以看到，微博爬取的内容“全文”，“视频”，“网页”，“视频”四个无实际意义的词汇排得靠前，故将这四个词除去，此外，加上中文分词以及去停用词处理：

重新得到结果：

### 聚类

在该部分，实现了两种分类算法，其一是DBSCAN，另一是KMeans

处理流程：中文分词、去停用词、将文本转化成词频矩阵，利用PCA降成三维数据，相应的聚类算法实现。

转化成词频矩阵：

PCA降维：

降维后数据显示：

（一）DBSCAN算法:

将数据进行DBSCAN训练：

结果显示：

（二）KMeans：

利用inertia\_度量指标来确定K值：

结果显示：

算法选择依据：最终，选定的算法是xxx。

### 情感分析

在该部分，处理的一般流程为：

中文分词、去停用词：

词性标注：

利用SnowNLP库进行情感分析：

将情感分布以图展示：

### 知识抽取和可视化

## 任务分工

请描述清楚各自的分工和完成情况

请每个成员各自描述实现自己任务过程种遇到的难点和解决方法

请阐述项目过程中遇到的团队协作问题和解决方法