**Python实验**

1. **用到的库**

**Import requests（向网页发出请求，从而获取网页返回的信息）**

**Import re（正则表达式的库）**

**Import bs4（免疫当的库？，提取网页返回的特定信息）**

**Import pandas as pd（科学数据分析的库）**

**Import numpy as np（科学数据分析的库）**

**Import string（字符串的库）**

**From wordcloud import WordCloud（文本处理，数学处理，可视化的库）**

**Import jieba**

**From PIL import Image**

1. **函数**
2. **def get\_data(cid):**

**‘’’分析网页，并获取网页文件（为后面分析网页的函数html服务）’’’**

**Url = ‘http://comment.bilibili.com/cid编码.xml’**

**Headers（字典） = {（内容是服务器的用户头代理User-Agent；方法：随意找到一个js文件并打开，找到User-Agent，复制粘贴过来就可以了；写法：删除其中的空白字符，并改为字符串的形式）（注意：用户头在一个主机里是一致的）}**

**Response = requests.get(url（存储弹幕信息的位置）,headers = headers).content.decode(‘utf-8’)**

**#需要访问content，并对content进行解码，因为所获取的网页信息一般都会存在编码问题**

**#调用requests的get方法，并创建一个response对象（接收requests）**

**Return response**

1. **def parse\_html(response):**

**‘’’解读网页文件，并获取关键信息（提取弹幕数据信息）’’’**

**#法一：美丽汤**

**Lst = [soup.findAll(name = ‘d’)]#选取所有d标签及其内容**

**Danmuku = [i.text for i in lst]**

**#法二：正则Re**

**Pattern = re.compile(‘<d p = “.\*?”>(.\*?)</d>’)#自己编译一个模式，配对标签，用.\*?来匹配未知字符的未知次数**

**Danmuku = re.findall(pattern,response)#用正则根据所编译的格式去配对并提取在response网页文件里面的弹幕数据信息，并传递给弹幕回传列表**

**Return danmuku**

1. **def save\_data(danmuku):**

**‘’’存储数据位csv（将弹幕信息存储为csv文件）’’’**

**#用pandas的dataframe的数据框存储为csv文件**

**Dict = {‘danmuku’ : danmuku}**

**Pd\_data = pd.DataFrame(Dict)**

**#将字典所包含的内容转化为一个dataframe的数据框，并赋值**

**Pd\_data.to\_csv(‘弹幕文件.csv’, index = False, header = False, mode = ‘w’, encoding = ‘utf-8-sig’)**

**#pd\_data存储为一个csv弹幕文件，索引设置为false（否则设置时会自动添加，在读取文件时出现你没有见过的索引，影响后续数据的写入）**

1. **def data\_preprocess(danmuku):**

**‘’’数据处理（即文本处理）’’’**

**#数据去重**

**#大小写替换——去标点符号、空白——去停用词——词干化**

**#分词——词频统计——构建bog of words——分类聚类**

**#可视化（词云图、统计分析图表）**

**With open(‘弹幕文件.csv’, mode=’r’,encoding = ‘utf-8’) as f:**

**#加载用户自定义词典（在分词时，将词典内部词语进行整合，而不会让他们被机械的分开）**

**Jieba.load\_userdict(r‘u’)（绝对路径：Ctrl+shift+c）**

**#创建一个阅读器，来存储f.Read()获得的字符串内容**

**Reader = f.read().replace(‘\n’,’ ’)**

**#加载停用词表（去掉一些没有意义的词语）**

**Stopwords = [line.scrip() for line in open(‘停用词表’，encoding = ‘gbk’).readlines()]**

**#去标点、空白、数字**

**Pun\_num = string.punctuation + string.digits**

**#访问存储所有的标点符号，数字**

**Table = str.makestrans(‘ ’,’ ’,pun\_num)**

**#替换标点，数字**

**Reader = reader.Translate(table)**

**#全部去掉标点，数字**

**Seg\_list = jieba.cut(reader, cut\_all = False)**

**#不精确分词模式（查）**

**Sentence = ‘’**

**For word in seg\_list:**

**If word not in stopwords and word.isspace() == False:**

**Sentence += word**

**Sentence += ‘,’**

**#如果该词不在停用词表和空白字符的话，就把它加到sentence里**

**Sentence = sentence[:-1]（查）**

**Return sentence**

1. **def count\_words(txt):**

**‘’’词频统计’’’**

**Adict = {}#创建字典存储词语——词频键值对**

**Words = txt.split(‘,’)#将字符串分隔为一个个词语组成的列表，并赋值**

**For word in words:**

**Adict[word] = adict.get(word,0) + 1（查）**

**#针对words里的每个word单词，我们对前面创建的字典里相应的键去查找，如果存在就返回0，如果没有的话，就加一**

**Pd\_count = pd.DataFrame(adict, index=[‘times’]).T.sort\_values(‘times’, ascending = False)**

**Pd\_count,csv(‘count\_word.csv’)**

1. **def gen\_wcd(txt):**

**‘’’生成词云图’’’**

**mask = pd.array(image.open(r’’))**

**Wcd = WordCloud(**

**Colormap = ‘reds’,**

**Font\_path = r’’**

**Background\_color = ‘white’**

**Repeat = True**

**Mode=’RGBA’**

**Scale = 5**

**Collations = False**

**)**

**Image\_produce = wcd.generate(txt).to\_image()**

**Image\_produce.show()**

1. **控制台调用函数**

**if \_name\_ == ”\_\_main\_\_”**

**Cid = int(input(‘请输入你想查询的视频CID号：’))**

**Response = get\_data(cid)**

**Danmuku = parse\_html(response)**

**Save\_data(danmuku)**

**Sentence = data\_preprocess(danmuku)**

**Count\_words(sentence)**

**Gen\_wcd(sentence)**

1. **实现步骤：**
2. **获取网页的文本文件**

**找到存储相应视频弹幕信息的网页的url（统一资源定位符，即网页地址）**

**①挑选所要分析的视频对象**

**②按下F12，打开开发者选项工具栏，之后再点Network，并且用Ctrl+R刷新**

**③在所有数据中找到该视频的cid编码，按Ctrl+F，输入CID，此时界面显示的数值就是我们所要的cid编码**

**④打开http://comment.bilibili.com/cid编码.xml（所有的弹幕信息都储存在该xml文件中）**

**⑤编写代码，对该地址进行一个发送和请求并获取服务器返回的xml文件**

1. **解读网页文件，并提取关键信息**

**法一：美丽汤**

**①soup = bs4.BeautifulSoup(response)**

**把网页文本转化为美汤对象**

**弹幕信息夹在</d>标签内部**

**Lst = [soup.findAll(name = ‘d’)]#选取所有d标签及其内容**

**Danmuku = [i.text for i in lst]**

**法二：正则表达式Re**

**Pattern = re.compile(‘<d p = “.\*?”>(.\*?)</d>’)#自己编译一个模式，配对标签，用.\*?来匹配未知字符的未知次数**

**Danmuku = re.findall(pattern,response)#用正则根据所编译的格式去配对并提取在response网页文件里面的弹幕数据信息，并传递给弹幕回传列表**

1. **存储数据位csv（将弹幕信息存储为csv文件）**
2. **数据处理（即文本处理）**
3. **词频统计**