**五邑大学计算机学院**

**School of Computer Science**



**专业综合实训报告**

**题 目：** 基于Python的豆瓣电影短评爬取和分析

**学 院**： 计算机学院

**专 业**： 电子信息工程（信息安全）

**班 级**： 140807

**学 号**： 3214002400

**姓 名**： 林 恩

**指导教师**： 刘 耀 宗

**起止日期**： 2017年12月19日 至2018年1月19日

递交的材料清单：

1. 源代码一份
2. 设计报告一份

**指导教师审阅意见及成绩评定：**

指导老师签章：

日 期： 年 月 日

1. **实训内容**

用Python语言爬取豆瓣电影《寻梦环游记》的短评数据，并对其结果进行分析。

1. **实训目标**
2. 抓取豆瓣电影网页数据，得到《寻梦环游记》电影的ID
3. 抓取《寻梦环游记》电影的短评数据
4. 清理数据并保存数据
5. 对数据进行词频统计
6. 用词云进行展示并保存图片
7. 数据分析
8. **详细实现**

软件：python版本3.6.4

1. 抓取豆瓣电影网页数据，得到《寻梦环游记》电影的ID

代码如下：

**from urllib.request import urlopen**

**from bs4 import BeautifulSoup**

**def getNowPlayingMovie\_list(url):**

**position='https://movie.douban.com/cinema/nowplaying/'+url**

**html = urlopen(position)**

**# 输出页面HTML代码**

**# bsObj = BeautifulSoup(html.read())**

**# print(bsObj)**

**# 输出“正在上映”板块的信息**

**bsObj=BeautifulSoup(html, "html.parser")**

**nowplaying\_movie = bsObj.findAll('div',{'id':'nowplaying'})**

**# print(nowplaying\_movie)**

**# 输出每部电影的详细信息**

**nowplaying\_movie\_list = nowplaying\_movie[0].findAll('li',{'class':'list-item'})**

**# print(nowplaying\_movie\_list)**

**# print(nowplaying\_movie\_list[0])**

**# 找到电影的名字和id**

**nowplaying\_list=[]**

**for item in nowplaying\_movie\_list:**

**nowplaying\_dict={}**

**nowplaying\_dict['id']=item['id']**

**nowplaying\_dict['name']=item['data-title']**

**nowplaying\_list.append(nowplaying\_dict)**

**# print(nowplaying\_list)**

**return nowplaying\_list**

第一步，对网页进行访问。

urllib是Python的标准库，当安装Python就可以引用，不需要额外安装。它包含了从网络请求数据，处理cookie，甚至改变像请求头和用户代理这些元数据的函数。urlopen用来打开并读取一个从网络获取的远程对象，因为它是一个非常通用的库（它可以轻松读取HTML文件、图像文件，或其他任何文件流）。这里使用urlopen对网页进行访问。因此，通过代码" from urllib.request import urlopen"来查找Python的request模块（在urllib库里面），并从中导入urlopen函数。

观察代码：

**position='https://movie.douban.com/cinema/nowplaying/'+url**

**html = urlopen(position)**

**# 输出页面HTML代码**

**# bsObj = BeautifulSoup(html.read())**

**# print(bsObj)**

首先使用position变量存放拼接后的字符串，该字符串就是豆瓣时下热门电影在江门的购票网址，具体网址是：<https://movie.douban.com/cinema/nowplaying/jiangmen/>。

然后使用urlopen函数对地址进行访问， 并将结果存放在变量html中。其中，html是字符串类型的变量，里面存放了网页的html代码。可以输入print(html)查看，结果如图1所示：



图1

第二步，使用BeautifulSoup 库解析得到的html代码，从中提取我们需要的数据。

BeautifulSoup库不是Python标准库，因此需要单独安装。首先到：https://www.crummy.com/software/BeautifulSoup/上下载BeautifulSoup相关文件，解压到Python安装文件所在目录下，在cmd命令行中，进入到BeautifulSoup中的setup.py所在根目录，输入命令" python install setup.py "，完成BeautifulSoup库的安装。在Python的IDLE环境中，输入" from bs4 import BeautifulSoup "，如果结果如图2所示，没有报错信息，证明BeautifulSoup库安装成功。

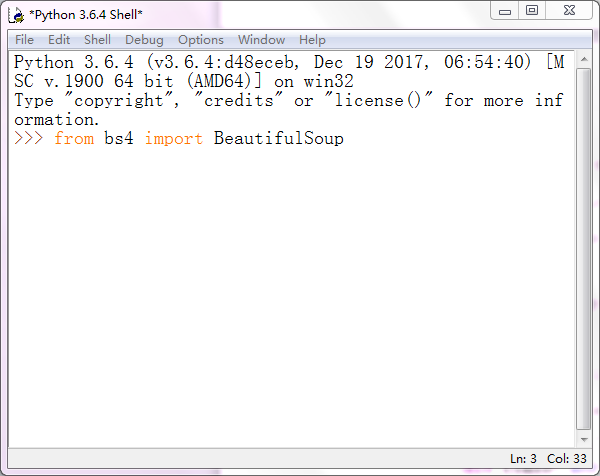


图2

观察网页源代码可以发现，所需要的电影名，电影的ID都在标签div id="nowplaying"中class="list-item"的li中，如图3所示：



图3

观察代码：

**# 输出“正在上映”板块的信息**

**bsObj=BeautifulSoup(html, "html.parser")**

**nowplaying\_movie = bsObj.findAll('div',{'id':'nowplaying'})**

**# print(nowplaying\_movie)**

前面已经导入urlopen，在这里调用html.read()获取网页的HTML内容，然后把HTML内容传到BeautifulSoup对象。通过findAll方法，将id为nowplaying的div标签的相关内容存放在变量nowplaying\_movie中。可以输入print(nowplaying\_movie)查看里面的内容，结果如图4所示：



图4

观察代码：

**# 输出每部电影的详细信息**

**nowplaying\_movie\_list = nowplaying\_movie[0].findAll('li',{'class':'list-item'})**

**# print(nowplaying\_movie\_list)**

**# print(nowplaying\_movie\_list[0])**

通过findAll方法，从nowplaying\_movi列表中找出class为**list-item**的li标签，并将相关内容存放在变量nowplaying\_movie\_list中。变量nowplaying\_movie\_list 是一个列表。存放所有li标签的内容。可以输入print(nowplaying\_movie\_list)查看里面的内容，结果如图5所示：



图5

观察nowplaying\_movie\_list中的内容可以发现，所需要的电影名字在data-title属性中，电影ID在id属性中（注：打开电影短评的网页时需要用到电影的id，所以需要对它进行解析）。

观察代码：

**# 找到电影的名字和id**

**nowplaying\_list=[]**

**for item in nowplaying\_movie\_list:**

**nowplaying\_dict={}**

**nowplaying\_dict['id']=item['id']**

**nowplaying\_dict['name']=item['data-title']**

**nowplaying\_list.append(nowplaying\_dict)**

**# print(nowplaying\_list)**

通过for循环将电影名字和id取出来放到数组nowplaying\_dict中。可以输入print(nowplaying\_list)进行查看，结果如图6所示：

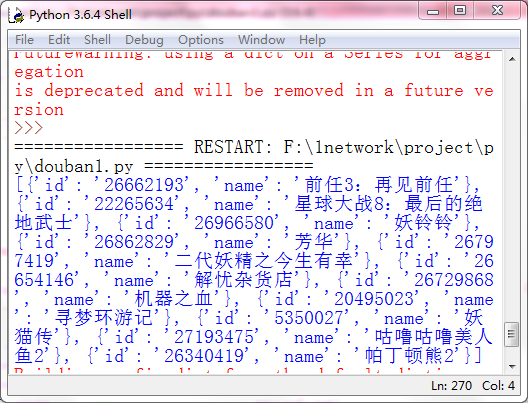


图6

1. 抓取《寻梦环游记》电影的短评数据

代码如下：

def getCommentsById(movieId, pageNum):

# 通过pageNum判断抓取前几页的短评数据

if pageNum>0:

start = (pageNum-1) \* 20

else:

return False

# 短评数据网址

requrl='https://movie.douban.com/subject/'+movieId +'/comments'+'?' +'start=' + str(start) + '&limit=20'

print(requrl)

# 获取短评数据

reap = urlopen(requrl)

bsreap=BeautifulSoup(reap, "html.parser")

comment\_div\_list = bsreap.findAll('div',{'class':'comment'})

eachCommentList=[]

for item in comment\_div\_list:

if item.findAll('p')[0].string is not None:

eachCommentList.append(item.findAll('p')[0].string)

return eachCommentList

def main(url):

commentList = []

NowPlayingMovie\_list = getNowPlayingMovie\_list(url)

# 获取前五页的短评数据

for i in range(5):

num = i + 1

commentList\_temp = getCommentsById(NowPlayingMovie\_list[7]['id'], num)

commentList.append(commentList\_temp)

# 主函数

position='jiangmen'

main(position)

第一步，对获取到的id进行处理，拼接成短评网址。

以《寻梦环游记》网址为例：https://movie.douban.com/subject/20495023/comments?start=0&limit=20。其中，20495023就是电影的id，start=0表示评论的第0条评论。

第二步，获取短评。

观察网页源代码可以发现，关于评论的数据是在class="comment"的div标签下的p标签中，如图7所示：



图7

观察代码：

reap = urlopen(requrl)

bsreap=BeautifulSoup(reap, "html.parser")

comment\_div\_list = bsreap.findAll('div',{'class':'comment'})

eachCommentList=[]

for item in comment\_div\_list:

if item.findAll('p')[0].string is not None:

eachCommentList.append(item.findAll('p')[0].string)

return eachCommentList

首先，对短评网址进行解析，变量bsreap里面存放了网页的html代码，变量comment\_div\_list存放了class="comment"的div标签中的html代码，eachCommentList存放了所有p标签中的短评数据。可以输入print(eachCommentList)进行查看。结果如图8所示：



图8

通过函数main中的for循环，获取电影前五页短评数据。

1. 清理数据并保存数据

代码如下：

import re

import os

f = open('comments.txt','a',encoding='utf-8')

def writeFile(s,parameter):

f.write(s)

f.write('\n')

f.write(parameter)

f.write('\n\n\n')

def main(url):

#将列表中的数据转换为字符串

comments=''

for k in range(len(commentList)):

comments=comments+(str(commentList[k])).strip()

writeFile('原始短评数据：',comments)

# 数据清洗，正则匹配，只留汉字

pattern = re.compile(r'[\u4e00-\u9af5]+')

filterdata=re.findall(pattern,comments)

cleaned\_comments = ''.join(filterdata)

writeFile('数据清洗后的短评数据：',cleaned\_comments)

第一步，将列表中的短评数据放在字符串数组comments中，可以输入print(comments)进行查看，结果如图9所示：



图9

第二步，通过正则表达式选出所有的汉字，达到数据清洗的目的。

python中正则表达式是通过re模块来实现的。所以需要通过代码"import re"导入该模块。

观察代码：

pattern = re.compile(r'[\u4e00-\u9af5]+')

filterdata=re.findall(pattern,comments)

cleaned\_comments = ''.join(filterdata)

writeFile('数据清洗后的短评数据：',cleaned\_comments)

首先定义需要的正则表达式，并将规则放置在变量pattern中。然后用re模块的findall方法，将字符串中符合定义正则规则的字符提取出来，存放在变量filterdata中。清洗后的短评数据存放在变量cleaned\_comments中，可以输入print(cleaned\_comments)进行查看，结果如图10所示：

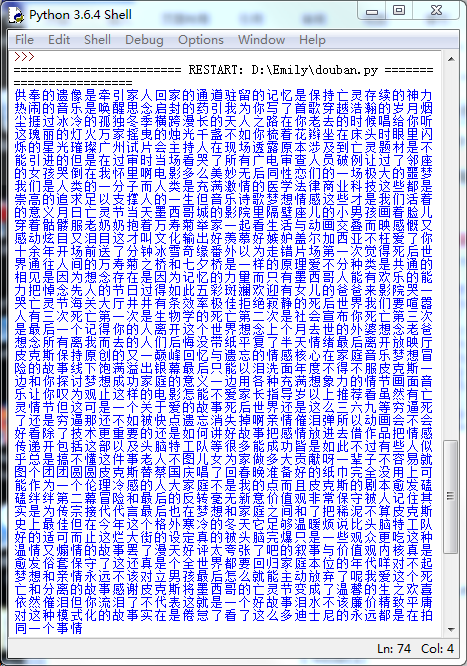


图10

观察代码：

import os

f = open('comments.txt','a',encoding='utf-8')

def writeFile(s,parameter):

f.write(s)

f.write('\n')

f.write(parameter)

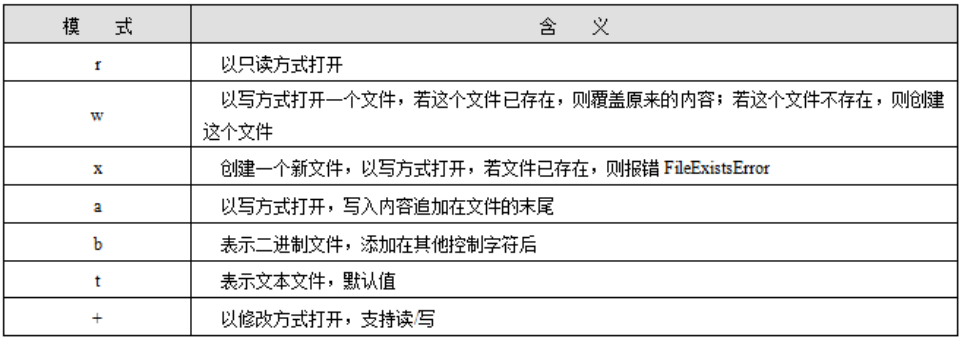
f.write('\n\n\n')

按照文件的编码方式，可将文件分为文本文件和二进制文件。

Python 3.x版本中，文件的默认编码格式是UTF-8，字符串使用的是Unicode编码。所有的文本类型，都使用Unicode编码，可以直接使用str.encode()进行编码，encode()后可看到字符的UTF-8编码，再使用bytes.decode()可解码为文本。

文件的打开方式可见表1：

表1



通过代码"import os"引入os模块，使用open方法，以写方式打开文件“comments.txt”，写入内容加在文件的末尾。通过writeFile方法，将所需数据内容写入文件中。

1. 对数据进行词频统计

代码如下：

def main(url):

# 进行中文分词操作

segment = jieba.lcut(cleaned\_comments)

print(segment)

f.write('进行中文分词操作：\n')

for item in segment:

f.write(item)

f.write('\t')

f.write('\n\n\n')

# 为中文分词加上序列号

words\_df=pd.DataFrame({'segment':segment})

print(words\_df)

words\_df\_trans = str(words\_df)

writeFile('为中文分词加上序列号：',words\_df\_trans)

# 去除停用词

stopwords=pd.read\_csv("stopwords.txt",index\_col=False,quoting=3,sep="\n",names=['stopword'], encoding='utf-8')

words\_df=words\_df[~words\_df.segment.isin(stopwords.stopword)]

words\_df\_trans = str(words\_df)

writeFile('去除停用词后的分词：',words\_df\_trans)

# 词频统计

words\_stat=words\_df.groupby(by=['segment'])['segment'].agg({"计数":numpy.size})

words\_stat=words\_stat.reset\_index().sort\_values(by=["计数"],ascending=False)

words\_stat\_trans=str(words\_stat)

writeFile('词频统计：',words\_stat\_trans)

第一步，进行中文分词操作。

这里使用结巴分词包。首先通过命令"pip install jieba"安装结巴分词包，然后通过代码"import jieba"导入结巴分词包。使用结巴分词的lcut方法，将清洗后的数据进行分词操作，结果返回的是一个分词列表，存放在变量segment中。可以输入print(segment)进行查看，结果如图11所示：

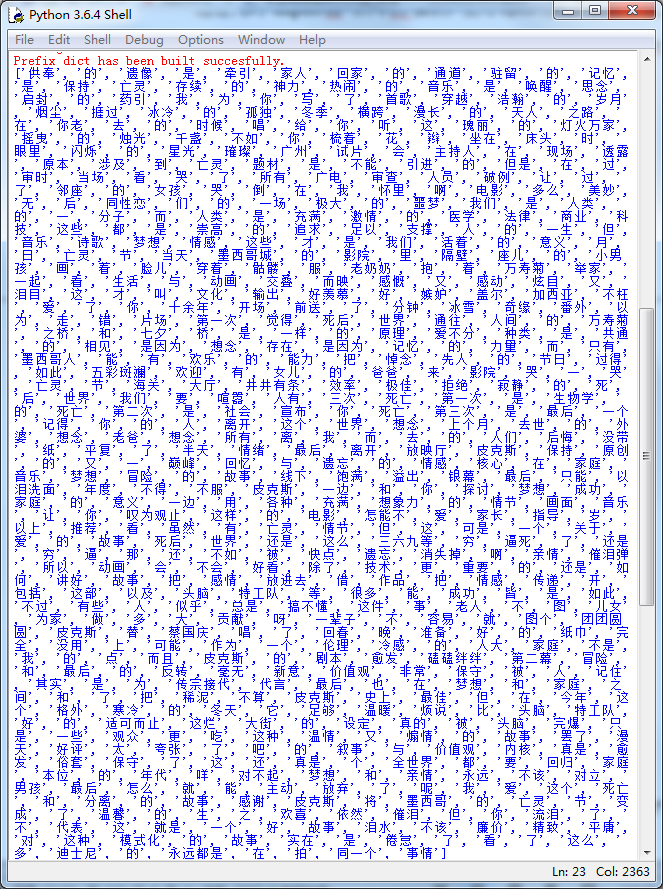


图11

最后，通过for循环将分词逐个写入文件中。

第二步，使用pandas对分词进行编号。

首先通过命令"pip install pandas"进行安装，然后通过代码"import pandas as pd"导入pandas包。返回带有序列号的分词的DataFrame对象words\_df。可以输入print(words\_df)进行查看，结果如图12所示：

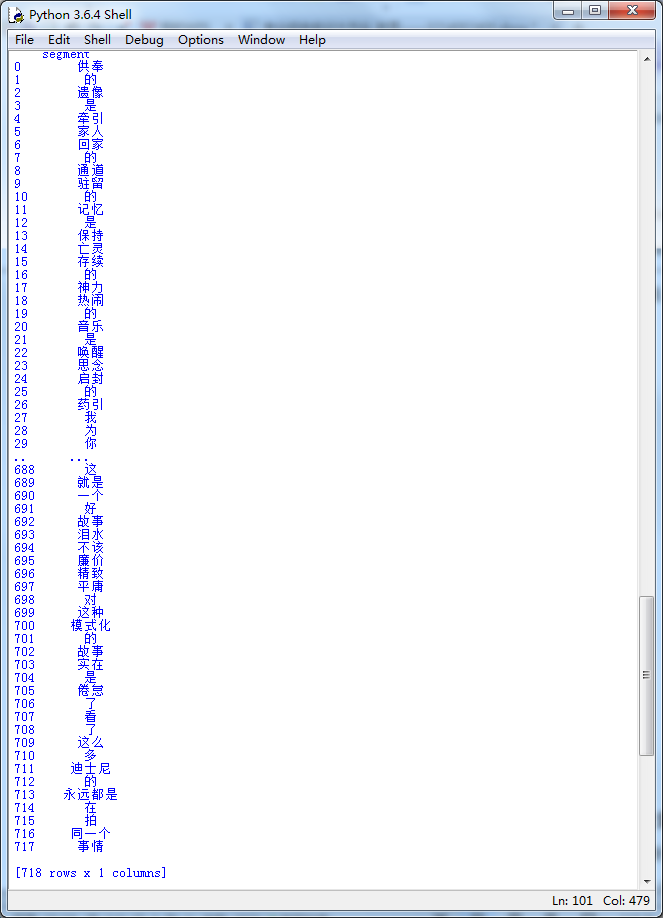


图12

最后，将words\_df转为str类型，然后通过writeFile方法将其写入文件中。

第三步，去掉分词中的停用词。

观察代码：

stopwords=pd.read\_csv("stopwords.txt",index\_col=False,quoting=3,sep="\n",names=['stopword'], encoding='utf-8')

words\_df=words\_df[~words\_df.segment.isin(stopwords.stopword)]

words\_df\_trans = str(words\_df)

writeFile('去除停用词后的分词：',words\_df\_trans)

先读取停用词文件并将其存在变量stopword中，接着判断分词是不是在停用词中，最后进行取反操作，将有用的分词保存在变量words\_df中。可以输入print(words\_df)进行查看，结果如图13所示：

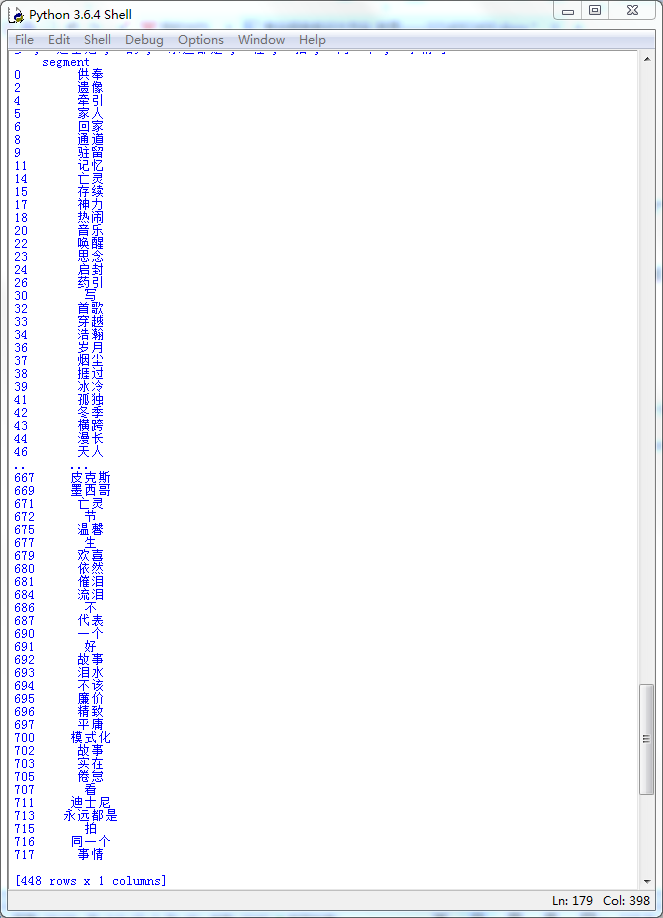


图13

最后，将words\_df转为str类型，然后通过writeFile方法将其写入文件中。

第四步，进行词频统计。

首先通过代码"import numpy"导入numpy计算包。使用groupby，将数据进行重构合并，使用numpy计算分词出现次数，并将最终结果保存在变量words\_stat中。可以输入print(words\_stat)进行查看，结果如图14所示：

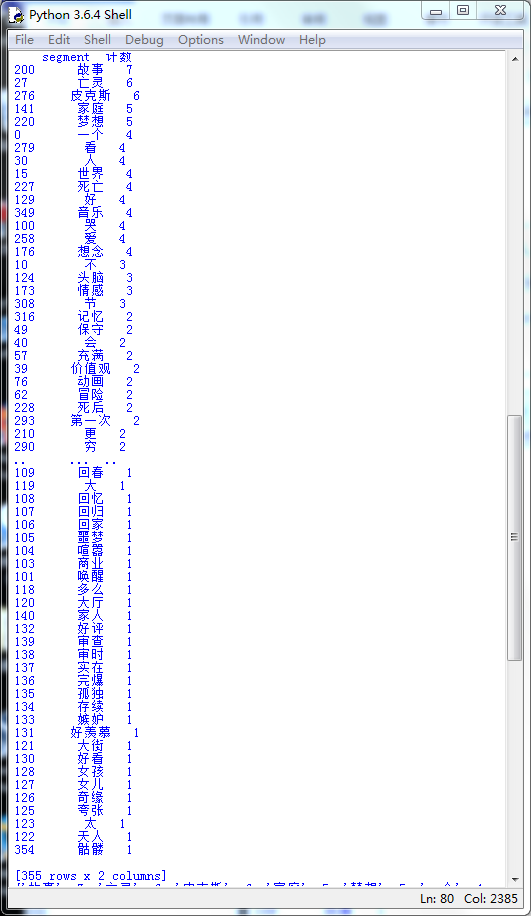


图14

1. 用词云进行展示并保存图片

def main(url):

# 用词云显示

wordcloud=WordCloud(font\_path="simhei.ttf",background\_color="white",max\_font\_size=80)

word\_frequence = {x[0]:x[1] for x in words\_stat.head(1000).values}

word\_frequence\_trans=str(word\_frequence)

writeFile('词频字典：',word\_frequence\_trans)

f.close()

wordcloud=wordcloud.fit\_words(word\_frequence)

plt.imshow(wordcloud)

plt.show()

# plt.imsave('F:/1network/project/py/jiangmen.jpg',wordcloud)

plt.imsave('jiangmen.jpg',wordcloud)

第一步，指定词云显示字体类型、字体大小和字体颜色。

第二步，根据词频统计的数据，选取前1000个分词，生成词频字典。可以输入print(word\_frequence)进行查看，结果如图15所示：

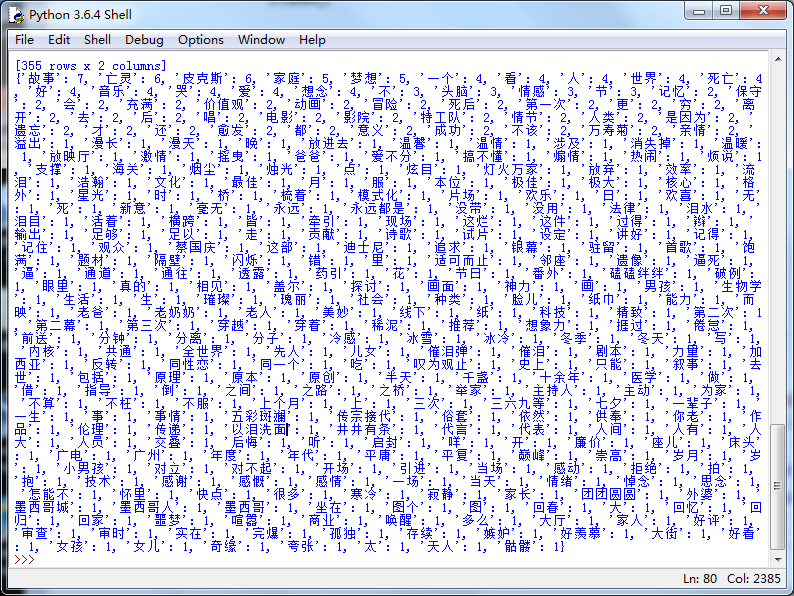


图15

第三步，将词频字典传给词云对象，并将词云对象显示出来，同时保存图片。如图16所示：



图16

1. 数据分析

从最后的词云显示图可以看出《寻梦环游记》是皮克斯动画工作室的动画片，是一部迪士尼电影，电影内容是与家庭有关，是一部饱含情感和梦想的影片。

1. **不足**

项目只是截取电影前五页的短评数据，并没有爬取所有短评数据，所以结果不精准。代码中没有对httpError以及AttributeError这类比较常见的异常进行处理，代码的可扩展性和可用性比较差。

1. **心得体会**

这次专业综合实训，我对网络爬虫有了一定的了解。

网络爬虫是一个自动提取网页的程序，它为[搜索引擎](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)从万维网上下载网页，是搜索引擎的重要组成。传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列,直到满足系统的一定停止条件。聚焦爬虫的工作流程较为复杂，需要根据一定的网页分析算法过滤与主题无关的链接，保留有用的链接并将其放入等待抓取的URL队列。然后，它将根据一定的搜索策略从队列中选择下一步要抓取的网页URL，并重复上述过程，直到达到系统的某一条件时停止。

在进行网络爬虫的时候，要特别注意以下几个问题：对抓取目标的描述或定义；对网页或数据的分析与过滤；对URL的[搜索策略](https://baike.baidu.com/item/%E6%90%9C%E7%B4%A2%E7%AD%96%E7%95%A5" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/_blank)。

通过本次实训，我了解了利用python进行数据抓取的过程。初步了解了html标签及其属性，明白数据清洗的目的及其方法，对正则表达式的使用有了比较深刻的印象。

**参考书籍**

Python网络数据采集/（美）米切尔（Mitchell.R）. –人民邮电出版社