# 词云实战——爬取豆瓣影评生成词云

## 功能介绍

“词云”就是对网络文本中出现频率较高的“关键词”予以视觉上的突出，形成“关键词云层”或“关键词渲染”，从而过滤掉大量的文本信息，使浏览网页者只要一眼扫过文本就可以领略文本的主旨。

豆瓣电影提供了最新的电影介绍及评论，包括上映影片的影讯查询及购票服务，观众可以记录想看、在看和看过的电影/电视剧，以及打分、写影评。豆瓣电影会根据观众的口味推荐好电影。本程序使用Python爬虫技术获取豆瓣电影(<https://movie.douban.com/>)中最新电影的影评，经过数据清理和词频统计后对电影的影评信息进行词云展示。

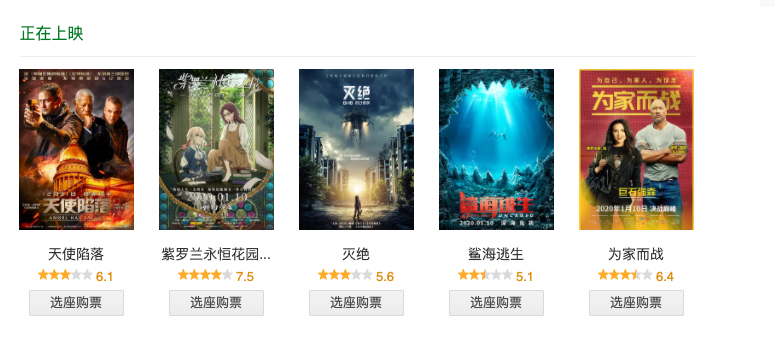
## 程序设计思路

本程序主要分为3个过程。

1. 抓取网页数据

使用Python爬虫技术获取豆瓣电影中最新上映电影的网页，其网址如下：

<https://movie.douban.com/cinema/nowplaying/qingdao/>



通过其HTML解析出每部电影的ID号和电影名，获取某ID号就可以得到该部电影的影评网址，形势如下：

<https://movie.douban.com/subject/26900949/comments>

[https://movie.douban.com/subject/26871938/comments](https://movie.douban.com/subject/26871938/?from=playing_poster)

其中，26900949、26871938就是电影《天使陷落》、《灭绝》的ID号，这样仅仅获取了20哥影评，可以指定开始号start来获取更多影评，例如：

<https://movie.douban.com/subject/26900949/comments?start=40&limit=20>

这意味着获取从第40条开始得20个影评。

1. 清理数据

通常将某部影评信息存入eachCommentList列表中。为便于数据清理和词频统计，把eachCommentList列表形成字符串comments，将comments字符串中的“也”“太”“ 的”等虚词（停用词）清理掉后进行词频统计。

1. 用词云进行展示

最后使用词云包对影评信息进行词云展示。

## 三、关键技术

1. 安装WordCloud

WordCloud使用最常规的pip install wordcloud命令安装。

或者在IDE中安装（各IDE安装方法不同）

1. 使用WordCloud

（1）WordCloud的基本用法

class wordcloud.WordCloud(font\_path=None, width=400, height=200, margin=2,

ranks\_only=None, perfer\_horizontal=0.9, mask=None, scale=1,

color\_func=None, max\_words=200, min\_font\_size=4, stopwords=None,

random\_state=None, backround\_color='black', max\_fone\_size=None,

font\_step-1, mode='RGB', relative\_scaling=0.5, regexp=None,

collocationgs=True, colormap=None, normalize\_plurals=True)

这是WordCloud的所有参数，下面具体介绍一下参数。

font\_path：需要展现什么字体就把该字体路径+扩展名写上，例如：font\_path=‘黑体.ttf’

width：输出的画布宽度，默认为400像素

height：处处的画布高度，默认为200像素

prefer\_horizontal：词语水平方向排版出现的频率，默认为0.9（所以词语垂直方向排版 出现的频率为0.1）

mask：如果该参数为空，则使用二维遮罩绘制词云；如果该参数非空，设置的宽/高值将 被忽略，遮罩形状将被mask取代。除了全白（#FFFFFF）部分不会悔之意外，其余部分 会用于绘制词云。例如bg\_pic=imread(读取一张图片.png)，背景图片的画布一定要设置 为白色，然后显示的形状为不是白色的其它颜色。可以用PS工具将自己要显示的形状复 制到一个纯白色的画布上，然后保存。scale：按照比例放大画布，例如设置为1.5，则长和宽都 是原来画布的1.5倍

min\_font\_size：显示的最小的字体大小

font\_step：字体步长，如果步长大于1，会加快运算，但是可能导致结果出现较大的误差

max\_word：要显示的词最大个数

stopwords：设置需要屏蔽的词，如果为空，则使用内置的STOPWORDS

background\_color：背景颜色，例如background\_color=‘white’，背景颜色为白色，默认为黑色

max\_font\_size：现实的最大的字体大小

mode：但参数为‘RGBA’并且backgrounde\_color不为空时背景透明

relative\_scaling：词频和字体大小的关联性

color\_func：生成颜色的函数，如果为空，则使用self.color\_func

regexp：使用正则表达式分隔输入的文本

collocations：是否包括两个词的搭配

colormap：给每个单词随机分配颜色，若制定color\_func，则忽略该方法

WordCloud提供方法如下：

fit\_words(firequencies)：根据词频生成词云

generate(text)：根据文本生成词云

generate\_from\_firequencies(firequencies[,...])：根据词频生成词云

generate\_text(text)：根据文本生成词云

process\_text(text)：将长文本分词并去除屏蔽词（此处指英语，中文分词还需要自己用其它库进行实现， 使用上面的fit\_words）

recolor([random\_state,color\_func,colormap])：对现有输出重新着色，重新着色会比重新生成整个词云快 很多

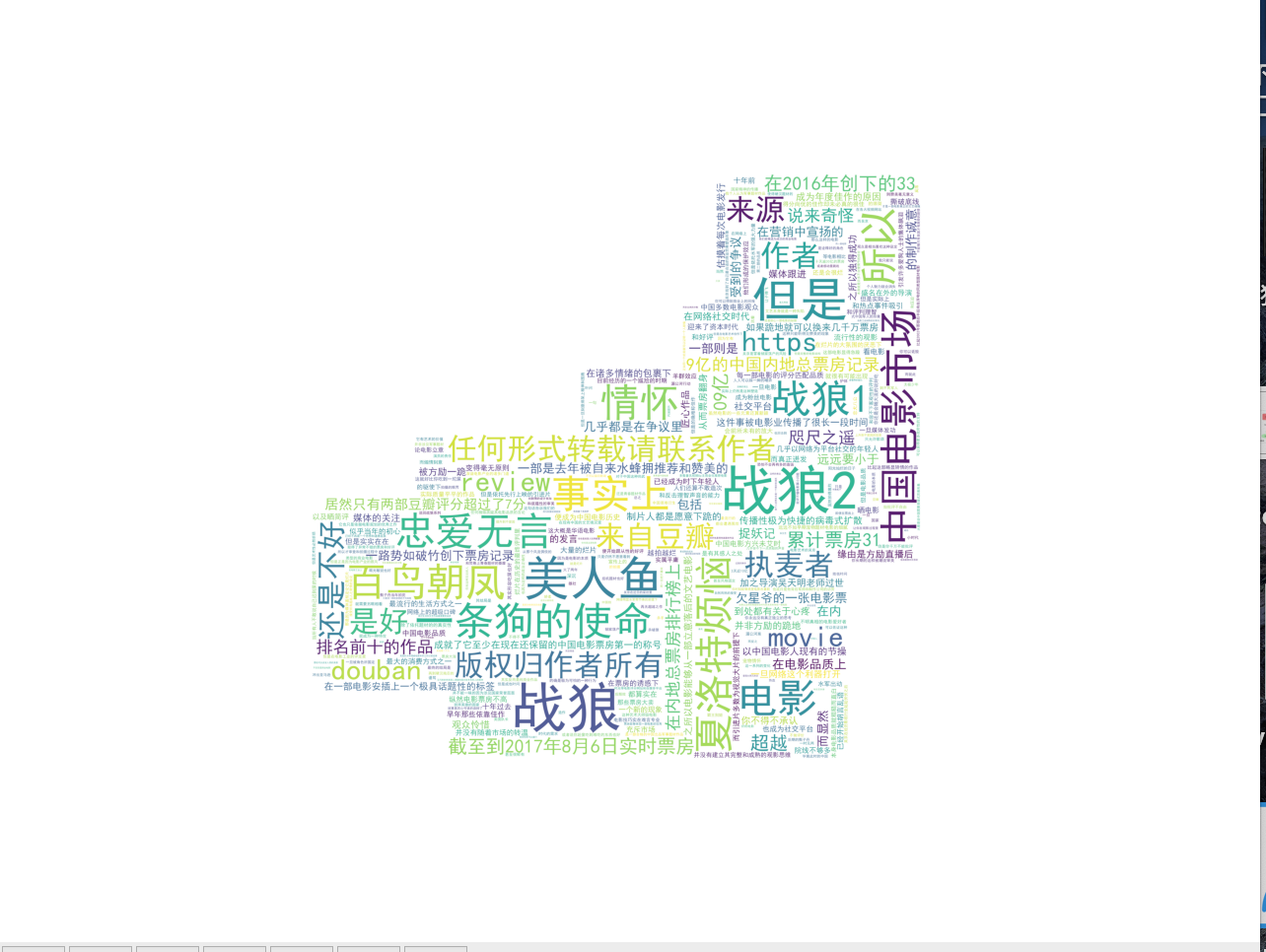
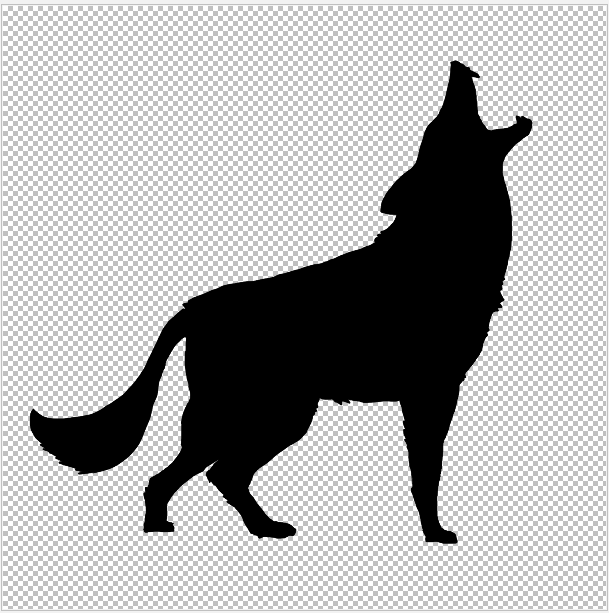
to\_array()：转化为numpy\_array

to\_file()：输出到文件

1. WordCloud的应用举例

*# 导入wordcloud模块和matplotlib模块*from wordcloud import WordCloud,ImageColorGenerator,STOPWORDS  
import matplotlib.pyplot as plt  
import numpy as np  
from PIL import Image  
  
*# 读取一个txt文件，注意修改文件的编码格式*text=open(**'text.txt'**,**'r'**,encoding=**'utf-8'**).read()  
*# 读取背景图片*bg\_pic=np.array(Image.open(**'wolf.png'**))  
**'''设置词云样式'''**wc=WordCloud(  
 *# background\_color参数用于设置背景颜色，默认颜色为黑色* background\_color=**'white'**,  
 mask=bg\_pic,  
 *# 有中文这句代码必须添加，否则绘制出现方框* font\_path=**'simhei.ttf'**,  
 max\_words=2000,  
 max\_font\_size=150,  
 random\_state=30,  
 scale=1.5  
)  
*# 根据文本生成词云*wc.generate\_from\_text(text)  
image\_colors=ImageColorGenerator(bg\_pic)  
*# 显示词云图片*plt.imshow(wc)  
plt.axis(**'off'**)  
plt.show()  
print(**'display success!'**)  
*# 保存图片*wc.to\_file(**'test\_img1.jpg'**)

只有在设置了mask的情况下才会得到一个拥有图片形状的词语你，此程序使用的模版图片是wolf.png，生成的词云形状如下：



1. 设置停用词

用户也可以设置停用词（“太”“的”等虚词），使得词云中不限时该虚词，例如

from os import path  
from PIL import Image  
import numpy as np  
import matplotlib.pyplot as plt  
from wordcloud import WordCloud,STOPWORDS,ImageColorGenerator  
*# 读取整个文章*text=open(**'text.txt'**,**'r'**).read()  
*# 读取遮罩/彩色图像*pic=np.array(Image.open(**'wolf.png'**))  
*# 设置停用词*stopwords=set(STOPWORDS)  
*# 人工添加停用词*stopwords.add(**"的"**)  
stopwords.add(**"了"**)  
*# 可以通过mask参数来设置词云形状*wc=WordCloud(  
 background\_color=**'white'**,  
 max\_words=2000,  
 mask=pic,  
 stopwords=stopwords,  
 max\_font\_size=40,  
 random\_state=42,  
 font\_path=**'simhei.ttf'**)  
*# 生成词云*wc.generate(text)  
image\_colors=ImageColorGenerator(pic)  
plt.imshow(wc,interpolation=**'bilinear'**)  
plt.axis(**"off"**)  
plt.show()

（4）WordCloud使用词频

import jieba.analyse  
from PIL import Image,ImageSequence  
import numpy as np  
import matplotlib.pyplot as plt  
from wordcloud import WordCloud,ImageColorGenerator  
lyric=**''**f=open(**'text.txt'**,**'r'**)  
for i in f:  
 lyric+=f.read()  
*# 用jieba对文章足分词，提取词频高的50哥词*result=jieba.analyse.textrank(lyric, topK=50, withWeight=True)  
keywords=dict()  
for i in result:  
 keywords[i[0]]=i[1]  
print(keywords)

image=Image.open(**'wolf.png'**)  
graph=np.array(image)  
wc=WordCloud(  
 font\_path=**'simhei.ttf'**,  
 background\_color=**'white'**,  
 max\_words=50,  
 mask=graph  
)  
*# 词频生成词云*wc.generate\_from\_frequencies(keywords)  
image\_color=ImageColorGenerator(graph)  
plt.imshow(wc)  
plt.imshow(wc.recolor(color\_func=image\_color))  
plt.axis(**'off'**)  
plt.show()  
wc.to\_file(**'test\_img2.jpb'**)

## 四、程序设计的步骤

1、抓取网页数据

首先要对网页进行访问，在Python中使用的是urllib库，代码如下

import requests\_html  
url = **'https://movie.douban.com/cinema/nowplaying/qingdao/'**session = requests\_html.HTMLSession()  
r = session.get(url)  
html\_data=r.html.html

其中，“https://movie.douban.com/cinema/nowplaying/qingdao/”是豆瓣电影最新上映的电影页面，用户可以在浏览中输入该网址进行查看；html\_data是字符串类型的变量，里面存放了网页的HTML代码。

然后对得到的HTML代码进行解析，提取出自己需要的数据。在Python中使用BeautifulSoup库进行HTML代码的解析。



BeautifulSoup使用的格式如下：

BeautifulSoup(html, **'html.parser'**)

第一个参数为需要提取数据的HTML，第2哥参数是指定解析器，然后使用find\_all()读取HTML中的内容。

但是HTML中有那么多的标签，该读取那些呢？其实，最简单的办法是打开爬取页面的HTML代码，然后查看需要的数据在那个HTML标签里面。

由上图可以看出，从<div id="nowplaying">标签开始是想要的数据，里面有电影的名称、评分、主演等信息，所以相应的代码编写如下：

from bs4 import BeautifulSoup  
soup= BeautifulSoup(html\_data, **'html.parser'**)  
nowplaying\_movie=soup.find\_all(**'div'**, id=**'nowplaying'**)  
nowplaying\_movie\_list=nowplaying\_movie[0].find\_all(**'li'**, attrs={**'class'**:**'list-item'**})

其中，nowplaying\_movie\_list是所有电影信息的一个列表，可以用print(nowplaying\_movie\_list[2])产看第3部电影《天使陷落》的内容。



从上图中可以看到在data-subject属性里面放了电影的ID号，而在img标签的alt属性里面放了电影的名字，因此通过这两个属性来得到电影的ID和名称（在打开电影短评的网页是需要用到电影的ID，所以需要对他进行解析），编写代码如下：

nowplaying\_list=[]  
for item in nowplaying\_movie\_list:  
 *# 以字典的形式存储每部电影的ID和名称* nowplaying\_dict={}  
 nowplaying\_dict[**'id'**]=item[**'data-subject'**]  
 for tag\_img\_item in item.find\_all(**'img'**):  
 nowplaying\_dict[**'name'**]=tag\_img\_item[**'alt'**]  
 nowplaying\_list.append(nowplaying\_dict)

在列表nowplaying\_list中存放了最新电影的ID和名称，可以使用print(nowplaying\_list)进行查看，结果如下：

[{'id': '26871938', 'name': '灭绝'}, {'id': '33424345', 'name': '紫罗兰永恒花园外传：永远与自动手记人偶'}, {'id': '26900949', 'name': '天使陷落'}, {'id': '27186353', 'name': '鲨海逃生'}, {'id': '26971054', 'name': '为家而战'}]

可以看到得是和豆瓣网址上面匹配的，这样就得到了罪行的电影信息。接下来对罪行电影短评进行分析。例如《天使陷落》的短评地址为”<https://movie.douban.com/subject/26900949/comments?start=0&limit=20>”，其中26900949就是《天使陷落》电影的ID，start=0表示第0条评论。

查看上面的短评页面的HTML代码，可以发现关于《天使陷落》评论的数据在标签<div class="comment">下：



因此对该标签进行解析，代码如下：

requrl=**'https://movie.douban.com/subject/'**+nowplaying\_list[0][**'id'**]+**'/comments?start=0&limit-20'**session=requests\_html.HTMLSession()  
response=session.get(requrl)  
html\_data=response.html.html  
soup=BeautifulSoup(html\_data, **'html.parser'**)  
comment\_div\_list=soup.find\_all(**'div'**,attrs={**'class'**:**'comment'**})

此时在comment\_div\_list列表中存放的就是class=’comment’的所有div标签里面的HTML代码了。在上面的图中还可以发现<div class=’comment’>标签里面的p标签下面<span>中存放了网友对电影的评论，因此对comment\_div\_list代码中的HTML代码继续进行解析，代码如下：

eachCommentList=[]  
for item in comment\_div\_list:  
 *# 获取p标签内部的span标签（即评论内容）* b=item.find(**'p'**).find(**'span'**)  
 if b.string is not None:  
 *# eachCommentList.append(item.find\_all('p')[0].string)* eachCommentList.append(b.string)

使用print(eachCommentList)查看eachCommentList列表中的内容，可以看到里面存放了大家想要的影评。

到此已经爬取了豆瓣电影最精播放电影的评论数据，接下来就要对数据进行清洗和词云展示了。

1. 数据清洗

数据清洗就是消去与数据无关的信息，这里为了方便进行数据清洗，将列表中的数据放在一个字符串中，代码如下：

comments=**''**for k in range(len(eachCommentList)):  
 comments+=(str(eachCommentList[k])).strip()

使用print(comments)进行查看，可以看到所有的评论已经变成一个字符串，但是评论中还是有很多标点符号等。这些符号对词频统计根本没用，因此要将它们清除，所用的方法是使用正则表达是，Python中的正则表达是通过re模块实现的。其代码如下：

import re  
pattern=re.compile(**r'[\u4e00-\u9fa5]'**)  
filterdata=re.findall(pattern,comments)  
cleaned\_comments=**''**.join(filterdata)  
print(cleaned\_comments)

继续使用print(cleaned\_comments)语句进行查看，可以看到此时评论数据中已经没有那些标点符号了，数据被洗“干净”了。

因为要进行词频统计，所以进行中文分词操作。在这里使用jieba分词（需要安装对应类库）。中文分词的代码如下：

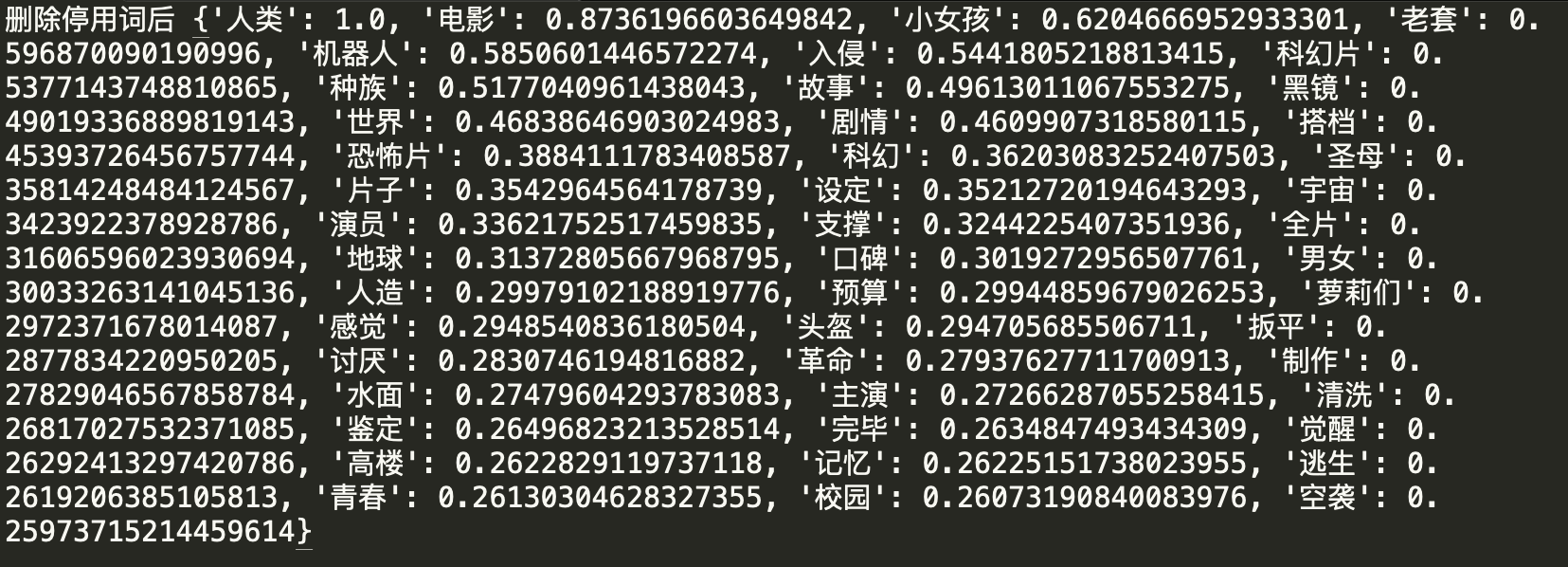
import jieba.analyse  
*# 使用jieba分词进行中文分词*result=jieba.analyse.textrank(cleaned\_comments,topK=50,withWeight=True)  
keywords=dict()  
for i in result:  
 keywords[i[0]]=i[1]  
print(**"删除停用词前"**,keywords)

结果如下：



从结果可以看到进行词频统计了，但数据中还有一些“没有”、“时候”等（停用词），这些词在任何场景中都是高频词，并且没有实际的含义，所以要把他们清除。

我们把停用词放在一个名为stopword.txt的文件中，将数据与停用词进行比对捷克。删除停用词的代码如下：



由于前面只是爬取了第1也的评论，所以数据有点少，在最后给出的完整代码中爬取了10也的评论，所得数据比较有参考价值。

1. 用词云进行显示

*# 用词云进行显示*import matplotlib.pyplot as plt  
import matplotlib  
*# 显示图像的最大范围*matplotlib.rcParams[**'figure.figsize'**]=(10.0,5.0)  
*# 词云包*from wordcloud import WordCloud  
*# 指定字体类型、字体大小和字体颜色*stop\_words=[]  
for line in open(**'stopword.txt'**,**'r'**,encoding=**'utf-8-sig'**):  
 stop\_words.append(line.rstrip(**'**\n**'**))  
wordcloud=WordCloud(  
 font\_path=**'simhei.ttf'**,  
 background\_color=**'white'**,  
 max\_font\_size=80,  
 stopwords=stop\_words  
)  
word\_frequence=keywords  
myword=wordcloud.fit\_words(word\_frequence)  
*# 展示云图*plt.imshow(myword)  
plt.axis(**'off'**)  
plt.show()

其中，simhei.ttf用来指定字体，用户可以在百度上搜索simhei.ttf进行下载，然后放入程序的中，常用的停用词已经为大家准备好。

首先在项目创建package为com.qst.douban\_wc，在此包下创建Python代码文件核心代码如下：

*# 分词包*import jieba  
import jieba.analyse  
*# numpy计算包*import numpy  
import re  
import matplotlib  
import matplotlib.pyplot as plt  
import requests\_html  
import requests  
from bs4 import BeautifulSoup as bs  
matplotlib.rcParams[**'figure.figsize'**]=(10.0,5.0)  
*# 词云包*from wordcloud import WordCloud  
import os  
  
*#分词网页函数*def getNowPlayingMovie\_list():  
 url = **'https://movie.douban.com/cinema/nowplaying/qingdao/'** session = requests\_html.HTMLSession()  
 r = session.get(url)  
 html\_data = r.html.html  
 soup= bs(html\_data, **'html.parser'**)  
 nowplaying\_movie = soup.find\_all(**'div'**, id=**'nowplaying'**)  
 nowplaying\_movie\_list = nowplaying\_movie[0].find\_all(**'li'**, attrs={**'class'**: **'list-item'**})  
 nowplaying\_list = []  
 for item in nowplaying\_movie\_list:  
 nowplaying\_dict = {}  
 nowplaying\_dict[**'id'**] = item[**'data-subject'**]  
 for tag\_img\_item in item.find\_all(**'img'**):  
 nowplaying\_dict[**'name'**] = tag\_img\_item[**'alt'**]  
 nowplaying\_list.append(nowplaying\_dict)  
 return nowplaying\_list  
  
*# 爬取评论函数  
# 参数为电影id号和要爬取评论的页码*def getCommentsById(movieId,pageNum):  
 eachCommentList = []  
 if pageNum > 0:  
 start=(pageNum-1)\*20  
 else:  
 return False  
 requrl = **'https://movie.douban.com/subject/'** + movieId + **'/comments?start='**+str(start)+**'&limit=20'** print(requrl)  
 session = requests\_html.HTMLSession()  
 response = session.get(requrl)  
 html\_data = response.html.html  
 soup = bs(html\_data, **'html.parser'**)  
 comment\_div\_list = soup.find\_all(**'div'**, attrs={**'class'**: **'comment'**})  
 for item in comment\_div\_list:  
 *# 获取p标签内部的span标签（即评论内容）* b = item.find(**'p'**).find(**'span'**)  
 if b.string is not None:  
 *# eachCommentList.append(item.find\_all('p')[0].string)* eachCommentList.append(b.string)  
 return eachCommentList  
  
def main():  
 *# 循环获取第2个电影的前10页评论* commentList=[]  
 NowPlayingMovie\_list=getNowPlayingMovie\_list()  
 *# 前10页* for i in range(10):  
 num=i+1  
 *# 指定那部电影，因为索引号从0开始，所以是第2个电影，num是爬取那一页评论* commentList\_temp=getCommentsById(NowPlayingMovie\_list[1][**'id'**],num)  
 commentList.append(commentList\_temp)  
 *# 将列表中的数据转换为字符串* comments=**''** for k in range(len(commentList)):  
 comments += (str(commentList[k])).strip()  
 *# 使用正则表达是去掉标点符号* pattern = re.compile(**r'[\u4e00-\u9fa5]'**)  
 filterdata = re.findall(pattern, comments)  
 cleaned\_comments = **''**.join(filterdata)  
 *# 使用jieba分词进行中文分词* result = jieba.analyse.textrank(  
 cleaned\_comments,  
 topK=50,  
 withWeight=True  
 )  
 keywords=dict()  
 for i in result:  
 keywords[i[0]]=i[1]  
 print(**"删除停用词前"**,keywords)  
 *# 停用词集合* stop\_words=[]  
 for line in open(**'../../../data/stopword.txt'**, **'r'**, encoding=**'utf-8'**):  
 stop\_words.append(line.rstrip(**'**\n**'**))  
 keywords={x:keywords[x] for x in keywords if x not in stop\_words}  
 print(**'删除停用词后'**,keywords)  
 *# 使用词云显示* wordcloud=WordCloud(  
 font\_path=**'../../../data/simhei.ttf'**,  
 background\_color=**'white'**,  
 max\_font\_size=80,  
 stopwords=stop\_words  
 )  
 word\_frequence=keywords  
 myword=wordcloud.fit\_words(word\_frequence)  
 *# 展示词云图* plt.imshow(myword)  
 plt.axis(**'off'**)  
 plt.show()  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 print(os.getcwd())  
 main()

如上为词云展示的核心文件，同学们需要在核心代码的基础上进行扩展