爬取《哪吒》豆瓣短评，我得到了什么？

原创

痴痴痴痴痴痴。 于 2019-08-14 15:09:14 发布 91 收藏

点击上方“蓝字”，感谢关注！

这段时间，《哪吒》爆火。

于是，就想看看，关于《哪吒》的评价。

为什么选择豆瓣？

质量和口碑还不错吧。

可是，折腾一波之后，发现了这个。

豆瓣从2017.10月开始全面禁止爬取数据，仅仅开放500条数据，白天1分钟最多可以爬取40次，晚上一分钟可爬取60次数，超过此次数则会封禁IP地址。

登录状态下，按网页按钮点击“后页”，参数“start”最多为480，也就是20*25=500条；非登录状态下，最多为200条。

行吧，500条就500条吧，Let's go。

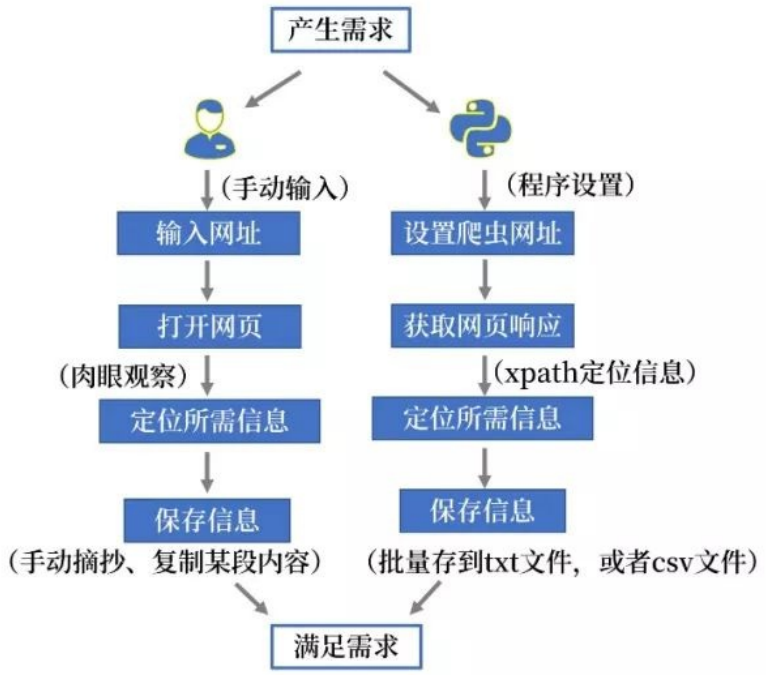
整个过程分为两部分：

1 获取豆瓣短评数据

2 词云可视化

1 获取短评数据

1）爬虫原理简单分析



2）需求分析

好了，爬虫的基本思路我们已经了解了。

现在，开始干活了。

首先，我们打开短评的url地址：

https://movie.douban.com/subject/26794435/comments?status=P

我们想要获取以下内容：

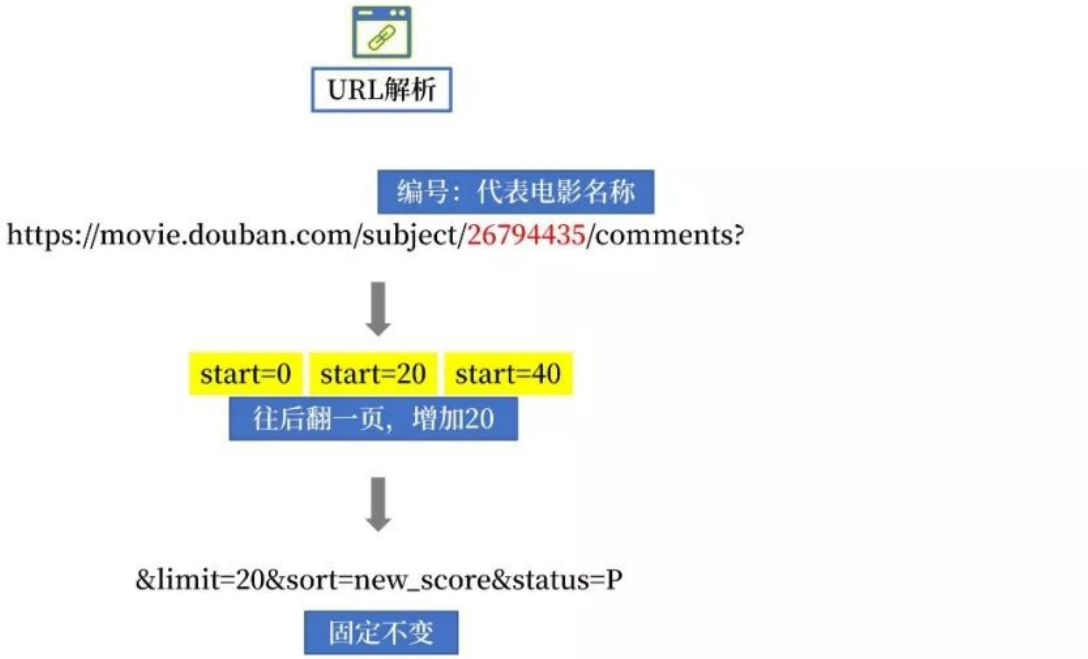
-用户名称

-评分

-短评内容

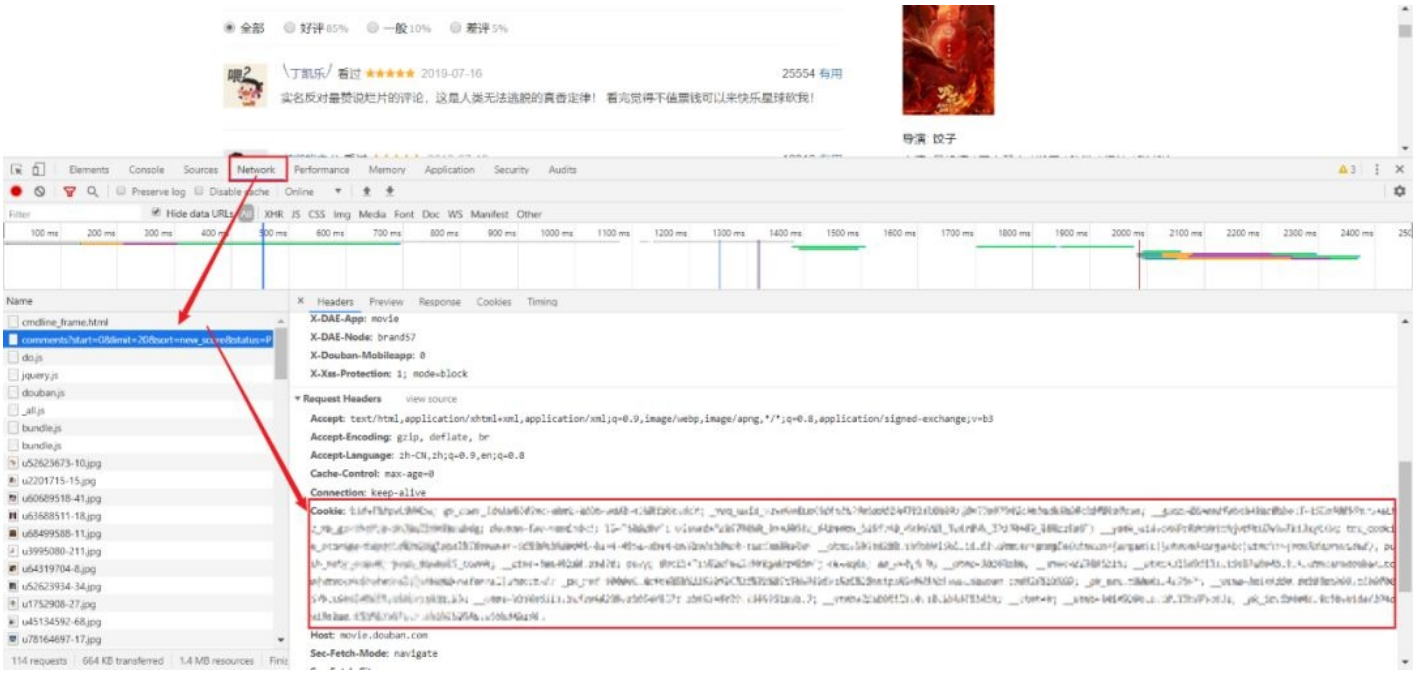
3）URL解析

要想获取数据，我们先来分析一下URL。



4）发送请求，获取响应

根据url，我们可以发送请求了，**注意携带cookie**。



先来爬一页，看看结果。

```
import requests

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
           'Cookie': '你的cookie'}

for i in range(0, 1):

    url = 'https://movie.douban.com/subject/26794435/comments?start={}&limit=20&sort=new_s' \
          'core&status=P'.format(i*10)

    reponse = requests.get(url, headers=headers)
    print(reponse.content.decode())
```

5）定位信息

从图中，我们可以看到对应的标签和属性。

利用**xpath**，我们可以很轻松地定位到我们想要的信息。**推荐《6节课学会python爬虫》，里边讲解得很好。**

先定位到，每一页的“20个短评”对应的xml位置。

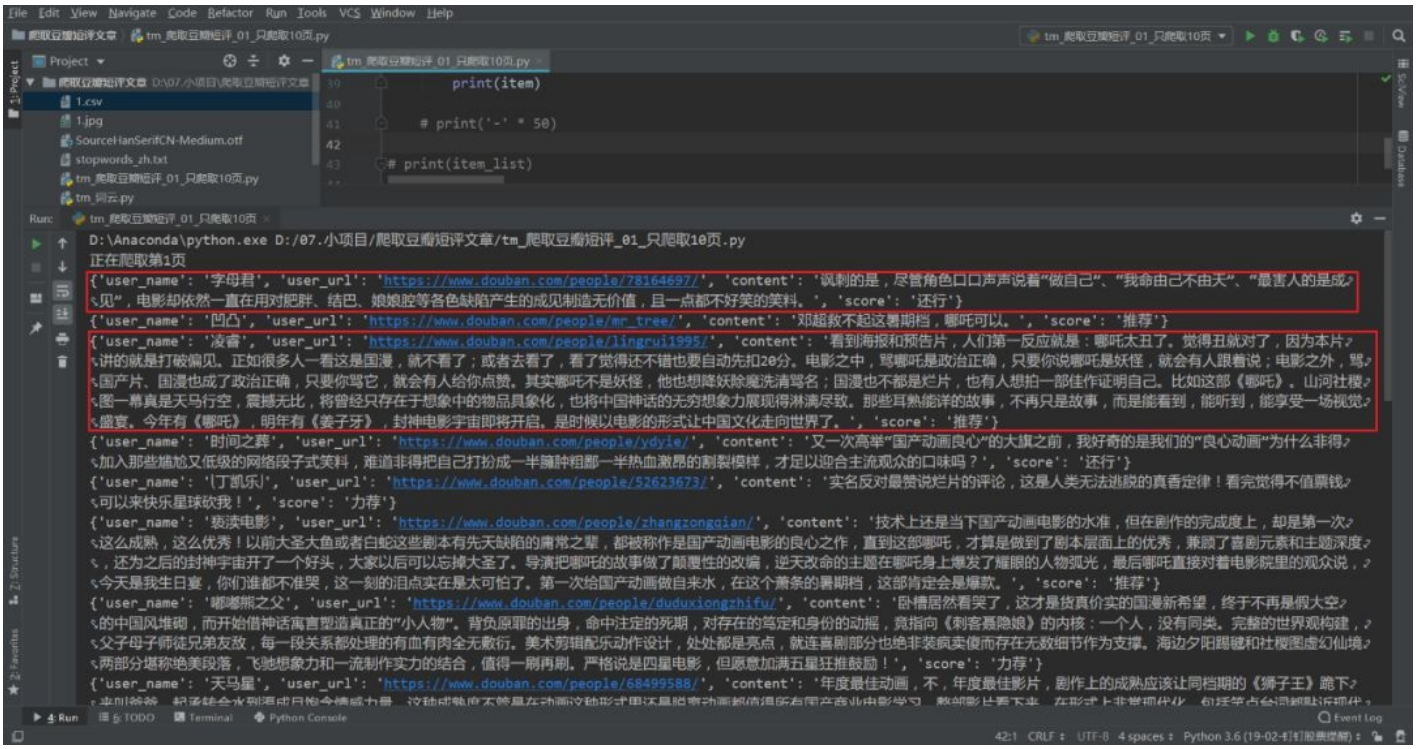
再遍历，每一个短评内容。

结合代码来看一下。

```
item_list = []

html = etree.HTML(reponse.content.decode())
div_list = html.xpath('//*[@id="comments"]//div[@class="comment"]')

# 定位大块
for div in div_list:

    # 遍历每一条信息
    item = {}

    # ./ 注意从当前节点，向下获取     # 用户姓名/用户主页的url/短评内容/评分     item['user_name'] = div.xpath('.
    item['user_url'] = div.xpath('.//span[@class="comment-info"]/a/@href')[0]
    item['content'] = div.xpath('.//span[@class="short"]/text()')[0].replace('\n', '')
    item['score'] = div.xpath('.//span[@class="comment-info"]/span/@title')[0]

    item_list.append(item)
    print(item)
```

5）保存结果

上边，已经把每一条数据，整理成一个字典。然后，把字典放在一个大的列表里。



这样，我们可以很轻松的把数据导出为csv文件。

```
# 把数据存成csv文件
```

```
import pandas as pd
df = pd.DataFrame(item_list)

# 保证不乱码
df.to_csv('哪吒短评数据.csv', encoding='utf_8_sig')
```

2 词云可视化

1）jieba分词

参考博客：

https://blog.csdn.net/dnxbjyj/article/details/72854460

结巴分词 是针对字符串进行处理的，分词后 会返回一个列表或者迭代器，你需要用 字符串的join方法，把词语列表 重新拼接成一个字符串，然后把内容给到 wordcloud 生成词云。

```
import pandas as pd
import jieba


# 读取数据
df = pd.read_csv('哪吒短评数据.csv', encoding='utf-8-sig')

text = ''
# 获得wordcloud 需要的 文本格式
for line in df['content']:
        text += ' '.join(jieba.cut(str(line), cut_all=False))  # 结巴分词
```

2）词云展示

创建一个词云对象，添加一些基本设置。比如，中文字体，背景图片，停用词等等。然后，根据上文中的 text，生成词云。

我们可以看一下，文本中最高频的50个词。并把词云保存为本地图片。

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# 停用词
words = pd.read_csv('stopwords_zh.txt', error_bad_lines=False, encoding='gbk', engine ='python', names=['st

stopwords = set('')
stopwords.update(words['stopword'])

backgroud_Image = plt.imread('豆瓣.jpg')  # 背景图

# 词云的一些参数设置
wc = WordCloud(
      background_color='white',
      mask=backgroud_Image,
      font_path='SourceHanSerifCN-Medium.otf',
      max_words=200,
      max_font_size=200,
      min_font_size=8,
      random_state=50,
      stopwords=stopwords
    )

# print(text)

# 生成词云
word_cloud = wc.generate_from_text(text)

# 看看词频高的有哪些
process_word = WordCloud.process_text(wc, text)
sort = sorted(process_word.items(), key=lambda e: e[1], reverse=True)
print(sort[:50])

plt.imshow(word_cloud)
plt.axis('off')

wc.to_file('结果.jpg')
print('生成词云成功!')
```

看一下高频词的结果。

```
[('哪吒', 24), ('电影', 9), ('这部', 8), ('故事', 8),
('动画', 6), ('国产 动画', 6), ('不由', 5), ('国漫', 5),
('想象力', 5), ('国产', 5), ('人物', 5), ('我命', 4), ('一部', 4),
('中国', 4), ('观众', 4), ('更是', 4), ('角色', 3), ('成见', 3),
 ('笑料', 3), ('暑期', 3), ('不错', 3), ('有人', 3), ('神话', 3),
('形式', 3), ('良心', 3), ('热血', 3), ('动画电影', 3), ('成熟', 3),
('优秀', 3), ('白蛇', 3), ('喜剧', 3), ('改编', 3), ('内核', 3),
('最佳', 3), ('饱满', 3), ('作品', 3), ('高潮', 3), ('场面', 3),
('大圣 归来', 3), ('缺陷', 2), ('本片', 2), ('打破', 2),
('偏见', 2), ('政治', 2), ('正确', 2), ('妖怪', 2),
('烂片', 2), ('社稷', 2), ('震撼', 2), ('封神', 2)]
```

看一下词云。

总结来了

我把文章中的一些重要的内容，总结在了下面的一张图里，方便大家保存、查阅。

《哪吒》豆瓣短评

- 获取豆瓣短评数据
  - 爬虫原理简单分析
  - 需求分析（获取用户名、短评内容等信息）
  - URL分析（页数发生变化）
  - 获取响应（request发送请求，获取相应，注意带cookie）
  - 定位信息（xpath解析工具）
  - 保存到本地（把每一个内容写成一个字典，再把所有的字典汇总成一个列表--item_list)
- 词云可视化
  - jieba分词（字符串）
  - 词云可视化（背景图片、中文字体、停用词）

几点注意：

- 豆瓣为了防爬虫，页面只能显示500条数据
- 不登录的情况下，最多能爬200条，所以，要在登录之后，复制cookie
- 爬多了会封IP的，建议先一页一页爬

@小痴印记

我把源文件及相关数据都打包好了，后台回复【**哪吒短评**】，一键提取。

这个小例子，挺基础的，适合入门的同学，但知识点也挺多的。

-END-

|推荐阅读|

python批量保存公众号内容为PDF

python实用资源汇总

高效搜索神器：listary vs 火柴

|今日互动|

《哪吒》怎么那么厉害？

小痴印记

知 识 来 自 于 记 忆 ， 智 慧 来 自 于 领 悟 。

喜欢

就转发一下呗