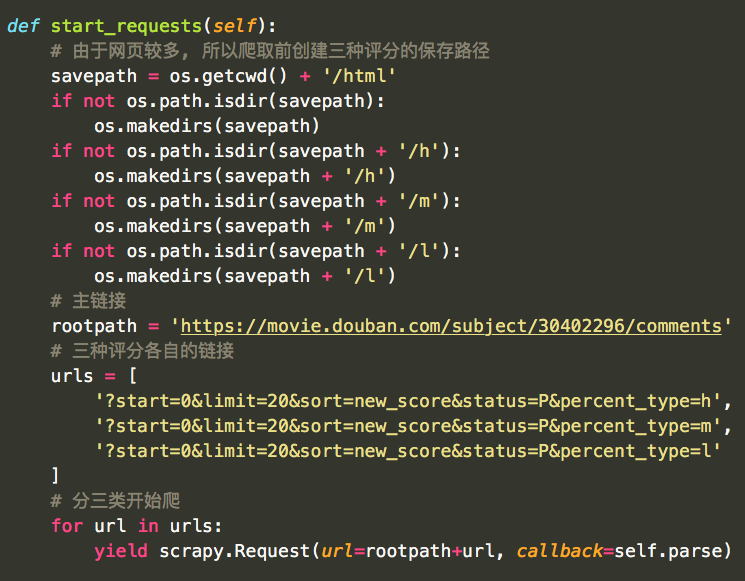
天气之子/Weathering With You 影评分析 代码说明

2018202147 崔冠宇

Step 1. 爬虫程序.

整个项目第一步是编写爬虫程序(WWYSpider.py). 其中两个主要函数如下所示:



start\_requests()函数的逻辑是: 首先建立数个用于保存文件的路径, 然后定义三种评论的起始链接, 产生三个初始请求.



parse()函数的逻辑是: 按一定命名规律保存爬下来的html文件，打开获得的文件的内容，如果还在范围内, 从中抽提“下页”按键的链接，并生成新的请求.

在项目目录下，运行 scrapy crawl WWYSpider, 得到html文件，然后进行下一步.

Step 2. HTML 评论提取.

得到了html文件, 第二步是编写网页解析程序(get\_comments.py). 其中三个主要函数如下所示:



get\_comments()函数的功能是: 打开网页, 用BeautifulSoup获得结构化的网页, 然后用相应的语句得到用户名及评价.

write\_txt() 和 write\_csv() 是按照相应格式, 将得到的用户名－评价字典写入文件.

最后是主函数:



主函数的功能是: 从三种评论的html文件中, 分类读取, 获得用户名和文件, 并且将每一类的评价合并写入一个csv文件, 用于后续处理.

在项目文件夹内，运行get\_comments.py，在csv文件夹内得到三种评价的csv文件, 接下来打开readme.ipynb进行后续处理.

Step 3. 在Jupyter Notebook 里进行分词并绘制词云.

这一部分主要工作是加载进csv文件, 用jieba 包分词（同时添加新词和过滤影响我们分析的一些词），并将分词后的结果按要求写入txt文件. 然后进行用wordcloud包绘制词云. 这一部分详见readme.ipynb文件，代码有注释和说明.

最后放一下我得到的词云和对应的分析:



**好评词云分析:**

首先观察几个最大的词, 分别是“世界”、“故事”、“少年”、“喜欢”、“你的名字”以及“画面”等等. 由此可见好评的用户应该是看过新海诚执导的电影的，属于电影的“铁粉”. 除此之外, 明显的词汇中看不到负面的用词, 整体评价积极.



**中评词云分析:**

和好评的词云有部分重叠, 比如“世界”、“故事”、“少年”和“你的名字”, 可见用户对电影有肯定之处; 同时“剧情”一词相比于好评词云大了许多, 有可能是这部分用户进行了客观的分析. 与此同时, 一些负面词汇开始出现, 比如“失望”、“不行”、“可惜”等, 总体而言是中偏上的评价.



**差评词云分析:**

同样的, 还是有不少重叠, 如“故事”、“剧情”、“世界”等, 但态度估计不一定是正向的. 除此之外, “中二”一词比较大, 除此之外, 也有一些“幼稚”、“矫情”、“莫名其妙”等负向情感较明显的词汇, 整体而言情绪是中偏下的.

最后是一些问题:

1.在写爬虫时, 遇到过没有USER AGENT而被拒绝访问(403)的情况. 在实践中都有哪些常见的反爬虫手段和反反爬虫手段？

2.在添加新词和停用词时, 我是手动一边观察效果, 一边添加词的. 在实践中, 有没有更好的方案? 比如说能不能从演员列表, 角色列表中获得一些新词? 对于不能表达情感的那些词, 要如何找出来以过滤呢?