这部电影值得看吗？ | 词云可视化分析豆瓣《大空头》影评

精于析，诚于心，锐于创，执于信。

很多人在看电影前都会去看看影评，了解电影内容和风评好坏，再决定是否要看。豆瓣电影常常是查看影评的首选网站，很多时候我们都能发现想看的电影有上万条影评，一条条看显然是不现实……

这里，我们首先使用爬虫批量下载豆瓣短评，再借助词云对影评进行可视化，获得直观的影评分析结果，快速发现电影特点。

**爬虫分析**

要用爬虫下载影评，首先要有影评地址，我们在豆瓣电影里搜索《大空头》，可以发现有三万多条短评，点进去查看全部短评，可以看到短评是分页的，每一页有20条短评。

我们注意到网址是这样的：<https://movie.douban.com/subject/26303622/comments?start=20&limit=20>

最后的start=20&limit=20表示从第20条评论开始，limit表示每页20条（不能改），所以我们只要让start依次取值0、20、30……就可以得到每个短评页的网址。

拿到网址之后，我们把他传给负责下载短评的get\_movie方法。get\_movie方法首先用requests类的get方法获取网页的内容，它有两个参数，其中url是网址，headers是一个携带浏览器信息的字典。

接着，用PyQuery的pyquery方法分析网页内容，分析结果存在doc变量中。对doc使用标签选择器，可以选中我们感兴趣的内容，而items()方法可以把一系列感兴趣的内容打包在一起，在后面的循环里，我们逐个查看了打包的内容，对于每条内容，需要再用标签选择器把短评部分摘取下来，这里评论结果保存在了all\_comment变量中。

get\_movie会把下载的短评返回到main函数，我们把短评保存到一个文本文件中，方便后续查看。

后面的处理就涉及到词云了。我们先用jieba库的lcut方法把短评从句子分解成独立的单词，用空格作为分隔符。

然后，用imageio库的imread方法读入了一张图片，用作结果的轮廓。接着，设置了停用词stopwords，让那些没有意义的虚词不显示在结果里。

最后，我们传入相关参数创建了WordCloud对象，用WordCloud类的generate函数产生了词云，并通过to\_file方法保存了结果。

**结果分析**

在词云里，我们看到最大的一个词是“金融”，上面还有“金融危机”字样，电影的主题是金融无疑了。

字号仅次于金融的是“剪辑”二字，看来这部电影的制作吸引了网友们的广泛关注啊，就是不知道是剪辑好呢，还是剪辑不好……

再看看别的词，可以找到“喜欢”“最佳”“好看”“有意思”这些赞美电影的词语，可能还真是一部好看的电影。但我们发现“枯燥”这个词也出现了，到底好不好看呢，可能要自己权衡一番了。

另外还能发现一类词语，“专业”“术语”“理解”……感觉这部电影会有点烧脑呀，不过也很正常，毕竟没有一些专业知识就看金融方面的电影是比较难理解的。