山东大学 软件 学院

信息检索 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301299 | 姓名： | | 班级： 2018级AI班 |
| 实验题目：第五、六周： | | | |
| 实验学时：5 | | 实验日期：2020年12月13日星期日 | |
| 硬件环境：  PC | | | |
| 软件环境：  Python+Flak+Mysql | | | |
| 实验内容：  一、实验综述  本实验基于对B站视频基本信息和弹幕内容的爬虫，进行数据分析。完成的实验内容共有  1）给定视频的BV号，爬取视频的基本信息，包括视频的名称、up主、播放量、弹幕数量、点赞、投币、收藏、分享的数量以及视频发布日期；  2）给定视频的BV号，爬取视频的弹幕内容，进行分词和过滤停用词，并生成特定样式的词云；  3）给定up主的名称，爬取其指定页面或全部页面的视频基本信息和弹幕内容，生成其基于所有视频的词云。  4）爬取哔哩哔哩排行榜，包括分类如下：全站、番剧、国产动画、国创相关、纪录片、动画、音乐、舞蹈、游戏、知识的各前100名视频，爬取他们的排名、基本信息和综合得分(b站给出)，并保存在数据库中。  5）将以上内容可视化，由于篇幅限制，只展示了“全部”排行榜的前10个视频。  灵感来源于在B站看视频，于是有了想分析b站弹幕，看看大家对up主和视频的想法。现在bilibili已经成为年轻人手机里必备的APP，bilibili网站（以下简称B站）除了动漫、鬼畜、生活等新颖有创意的内容以外，还有一大特色就是B站的弹幕。弹幕指的是悬浮于视频上方的实时评论，可以给观看视频的人“实时互动”的感觉，弹幕真正让B站从一个单向的视频播放平台，变成了双向的情感连接平台，进而形成了B站的社区文化，创造了许多好玩的梗。于是我想通过爬取B站的弹幕，来分析不同的分区不同的up主不同的视频中，有哪些“弹幕文化”，有哪些好玩的梗，同时分析大多数观众对该视频的感情色彩。这个主题较为新颖，同时也很贴近现实，可以通过大家弹幕中发表的言论，分析大家对该up主或该事件的态度，也可以通过大家发弹幕的时间位置判断视频的精彩瞬间，具有较大的现实价值。  二、实验步骤（仅给出关键代码）  1. 给定视频的BV号，爬取视频的基本信息  本实验尝试了用request库通过网络连接获取网页内容，然后尝试用lxml和beautifulsoup的方法解析。      2）给定视频的BV号，爬取视频的弹幕内容  查询资料，得到8个参数的意义  第一个参数代表弹幕出现的时间 以秒数为单位。这个时间就是视频播放的时间，也就是弹幕是在视频播放的第几秒发出的；  第二个参数代表弹幕的模式1..3 滚动弹幕 4底端弹幕 5顶端弹幕 6.逆向弹幕 7精准定位 8高级弹幕；  第三个参数代表字号12非常小,16特小,18小,25中,36大,45很大,64特别大；  第四个参数代表字体的颜色，以HTML颜色的十位数为准；  第五个参数代表Unix格式的时间戳。基准时间为 1970-1-1 08:00:00。也就是你发出弹幕的实时日期时间；  第六个参数代表弹幕池 0普通池 1字幕池 2特殊池 【目前特殊池为高级弹幕专用】；  第七个参数代表发送者的ID，用于“屏蔽此弹幕的发送者”功能；  第八个参数代表弹幕在弹幕数据库中rowID 用于“历史弹幕”功能。      3）生成特定样式的词云，模板使用和背景图对称的图片  采用jieba分词  )结巴分词支持三种分词模式：  a)精确模式，试图将句子最精确地切开，适合文本分析；  b)全模式，把句子中所有的可以成词的词语都扫描出来, 速度非常快，但是不能解决歧义；  c)搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词  2)算法原理  基于Trie树(前缀树/字典树)结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG)  采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合  对于未登录词，采用了基于汉字成词能力的HMM模型(隐马尔可夫模型)，使用了Viterbi算法        4）爬取哔哩哔哩排行榜（本周内容）  可以注意到排行榜网址的最后一个参数随着排行榜的类别进行变化，因此可以比较方便的获得各个类别的排行榜（100名）视频的基本信息。        每个视频的基本信息：    并存放在数据库中。    5）可视化部分  由于之前我没有学习过前端的有关知识，所以想要可视化展示出来成了我的一大难题。在网上找了一些模板，但是都不是我想呈现的效果，所以最后还是自己用基本元素写了。  另外还学习了Flask的应用，参考：  <https://www.w3cschool.cn/flask/flask_http_methods.html>  a. 连接数据库，获取排行榜信息    b. 提交搜索框后判断如果提交的是视频的BV号，则爬取视频的基本信息和弹幕，进行处理后显示成词云；若判断是up主的名字，则对up主的所有视频进行爬取，并对所有弹幕统共分析生成词云，更能表示该up主的特点。    c. 处理完成后，打开下一个界面进行展示    关于Flask，使用的过程中遇到了很多坑，对此做一下总结：  (1)需要将文件进行分层，对于模板类的文件，对于被render\_template引用的html，统一放在templates文件夹下。但是对应的css文件放在template文件夹下总是出现路径错误，特别是其中的图片会出现路径错误，参考：<https://www.jianshu.com/p/5722fef5fa96>的讲解，将css文件和image文件放在static文件夹下，可以将图片放在/static/images文件夹下。修改后正确的文件存放方式如下，其中/static/images/result存放构建的词云。    (2)修改静态文件夹中的文件后，重启程序仍然没有变化，参考博客后：    得到解决方案清除浏览器缓存。所以每次更改static中文件的配置后就需要清除浏览器的缓存，感觉比较麻烦，不过Flask好像没有更好的解决办法。  c. 界面采用html配合css | | | |
| 结论分析与体会：  1. 本次实验完整的做了一个爬虫项目。掌握了对网页的解析、分词、分析等信息检索最重要的部分，同时学习了前后端的连接，第一次尝试写前端，并用Flask连接。对网页解析有了更深的认识，自己也学会查看、查找网页源代码和动态的响应了，收获颇丰。  2. 改进目标：对up主名和视频号的区分，现在是直接判断前两个字符是不是“BV”，简单但包容性不强。  3. 参考资料汇总  中文NLP资料库：  <https://github.com/fighting41love/funNLP>  中文分词词性对照表：  <https://blog.csdn.net/kevin_darkelf/article/details/39520881>  <https://blog.csdn.net/accp_cn/article/details/71774109?utm_medium=distribute.pc_relevant.none-task-blog-searchFromBaidu-1.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-searchFromBaidu-1.control>  繁体简体转化：  <https://blog.csdn.net/wds2006sdo/article/details/53583367?utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~all~sobaiduend~default-1-53583367.nonecase&utm_term=langconv%20python%20%E5%AE%89%E8%A3%85&spm=1000.2123.3001.4430>  <https://github.com/skydark/nstools/blob/master/zhtools/langconv.py>  <https://github.com/skydark/nstools/blob/27c45fceebeef551368d3ca57dc995456dc2ff34/zhtools/zh_wiki.py>  如何解决Python中利用Wordcloud无法生成中文词云的问题：  <https://blog.csdn.net/liubinbin_521/article/details/79321378>  html:  <https://github.com/beyondray/Crawler>  flask:  <https://www.w3cschool.cn/flask/flask_http_methods.html>  flask图片：  <https://www.jianshu.com/p/5722fef5fa96>  排行榜：  <https://www.cnblogs.com/mengxiaoleng/p/11598832.html#_label0_5>  表格：  <https://blog.csdn.net/Golf_research/article/details/52997100?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-4.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-4.control> | | | |
| 最终成果展示：  1. 首页界面    上面的输入框可以输入视频的BV号或是输入up主名称，下面的部分展示了全部排行榜的前10个视频的基本信息，包括排行榜名次、视频名称、播放量、up主、综合得分（由b站给出）  输入视频的BV号后：（例如BV1yp4y1z7yH）    左边显示弹幕的词云（虽然看起来很像背景，但确实是生成的），可以看到李子柒的这条视频中很多都是“许愿”的弹幕，中间是视频详情，包括视频名称（以及链接）、播放量、up主、弹幕数量、点赞、投币、收藏、分享数目。词云看起来比较稀疏，经过分析后发现是弹幕中有很多的重复弹幕。  再例如BV1HT4y1w7HX：    生成的词云图保存在本地：    如果输入up主的名称，可以获取其所有视频的基本信息及弹幕：  共有148个视频，将基本信息保存在json文件中：    共爬取66002条弹幕    生成基于六万条弹幕的特征词云： | | | |