使用 Python 爬虫分析 Bilibili 每日排行进行数据可视化

胡云海 201700201008

摘要:在一个学期的 python 与数据分析课程后，学习到许多有用的 python 编程方法。掌握了 matplotlib、numpy、requests 等库后，尝试自己进行一次实践。选择对目前热门的自媒体网站 bilibili，抓取它的每日排行榜，对其每日排行的相关信息进行数据可视化，并对总榜 top5 的作者的视频评论绘制词云，分析其内容要点。

关键字:python 与数据分析；爬虫； matplotlib；jieba；wordcloud

1 项目概述

Python 是一个强大的语言，语法简单、易于学习，且有极大的第三方库，可以提供大量的功能给使用之使用。较为知名的 python 的三方库有，用于词频分析的 jieba 库，可以高效的分词，在自然语言处理中起到较大的作用；matplotlib 库，被誉为可以替代的 matlib 的强大存在，提供了极其丰富的数学绘图功能；Wordcloud 库是知名的词云生成工具，可以生成美观清晰地词云，给文本分析后提供极好的展现平台。

在学习了这些工具后，我也尝试使用自己的只是，去进行一些基础的数据获取、分析、可视化的工作流。于是我选择对现在了非常火热的自媒体平台 bilibili 作为目标，对其每日视频的排行榜进行爬取，得到其排行榜信息，并对每个不同的排行榜的 top20 的作者的点击量进行数据可视化比较不同领域内流量分布的情况最后又选择总榜 top5 使用词云展示其作品中的热点词汇，看评论中热点信息。

2 排行榜爬取

2.1 库的调用

第一步先写出排行榜爬取的程序所需要的库，这次使用到了 requests、re、json、os、urllib3、datetime，其中 requests 是用于请求和解析网页，re、json、os 是文件系统所需的库，而 urllib3 是关闭 https 的报错，datetime 库用于获取当日时间。

2.2 函数

2.2.1 获取 B 站排行榜 url

通过查阅资料之道 B 站排行榜的 url 是‘https://www.bilibili.com/ranking/all/榜单名/0/1’这样的组成，因此定义 state 为榜单名，即可快速写出对应的函数。

```
def get_url():
    state_url='https://www.bilibili.com/ranking/all/'
    true_url = state_url+state+'/0/1'  #组成相应的url
    print(true_url)
    return true_url
```

2.2.2 获取 B 站排行榜的 html

第二步得到 url 后便要去请求网页得到 HTML 文件，并使用 utf-8 编码。

```
def get_html(url):
    html = requests.get(url)
    html.encoding='utf-8'
    content=html.text;
    print(html)
    return content
```

2.2.3 用正则表达式获取所需内容

得到 html 文件需要解析，针对其数据的特点，和对其中有效数据进行分析，提取了排名、视频编号、作者、标题、播放量、观众、作者空间等信息。正则表达式就是提取重要结构将可以变化的地方用符号替代。

```
def pick_up_state(html):
    pattern=re.compile('<*?num.*?>(\d+).*?<a href=.*?alt=.*?</a>.*?<a href="https://www.bilibili.com/video/(.*?)".*?tit
                       +'.*?<.*?play.*?</i>(.*?)</span>.*?<.*?view.*?</i>(.*?)</span>.*?'+
                       '<.*?href="//space.bilibili.com/(.*?)".*?>.*?<.*?author.*?</i>(.*?)</span>.*?',re.S)
    items =re.findall(pattern, html)
    print(items)
    for item in items:
        yield{
            'ranking':item[0],
            'palyHerf':item[1],
            'title':item[2],
            'play':item[3],
            'view':item[4],
            'authorHome':item[5],
            'author':item[6],
     }
```

2.2.4 保存文件

将解析的数据保存在榜单和时间命名的文件中。

```
def writeTxt(state,stateAll):
    today=datetime.date.today()
    time =today.strftime("%Y-%m-%d")
    save=stateAll+time+'.txt'
    with open(save,'a',encoding='utf-8') as f:
        f.write(json.dumps(state, ensure_ascii=False) + '\n')
        f.close()
```

2.3 主程序

列出 B 站所有排行榜极其代码对应的网页 url 值，导入 url 获得函数送入 html 获取函数得到 html 文件，再进行解析，将解析的文件用保存函数保存。这样第一个主要程序完成。

```
if __name__ == '__main__':
    all=['0','1','168','3','129','4','36','188','160','119','155','5','181']    #查看B站相关页面可以知道
    stateAll={'0':'全站',
              '1':'动画',
              '168':'国创相关',
              '3':'音乐',
              '129':'舞蹈',
              '4':'游戏',
              '36':'知识',
              '188':'数码',
              '160':'生活',
              '119':'鬼畜',
              '155':'时尚',
              '5':'娱乐',
              '181':'影视'}
    for state in all:
        true_url= get_url()  #组成相应的url
        html=get_html(true_url)
        for stateDate in pick_up_state(html):
            writeTxt(stateDate,stateAll[state])
```

3 评论获取

下一步开始编写评论获取的程序，在这里使用到了和上面相似的库，不再赘述。

3.1up 主视频号获取

B 站的 up 主有独立的空间号，并通过相关链接可以得到其空间内的视频 aid/AV 号。格式如下："http://space.bilibili.com/ajax/member/getSubmitVideos?mid=" + str(mid) + "&pagesize=" + str(num) + "&page=" + str(n)

据上述信息可以写出函数：

```
def getAllaidList(mid, num, page):
    for n in range(1,page+1):
        space_url = "http://space.bilibili.com/ajax/member/getSubmitVideos?
        r = requests.get(space_url)
        text = r.text
        list_text = json.loads(text)        # 遍历JSON格式信息，获取视频aid
        for item in list_text["data"]["vlist"]:       #获取data，vlist文件信息
            #print(item)
            aid_list.append(item["aid"])
    print(aid_list)
```

3.2 获取 aid 号下的评论

B站提供了 api 可以快速得到视屏下的评论：http://api.bilibili.com/x/reply?type=1&oid="+str(item) + "&pn=1&nohot=1&sort=0。对其获取后，根据其格式可以推导出如下的程序：

```
def catchCommentList(item):
    comment_url = "http://api.bilibili.com/x/reply?type=1&oid=" + str(item)
    print(comment_url)
    r = requests.get(comment_url)
    numtext = r.text
    list_text = json.loads(numtext)
    commentsNum = list_text["data"]["page"]["count"]
    page = commentsNum // 20 + 1
    for n in range(1,page):
        url = "https://api.bilibili.com/x/v2/reply?jsonp=jsonp&pn="+str(n)+
        req = requests.get(url)
        text = req.text
        json_text_list = json.loads(text)
        for i in json_text_list["data"]["replies"]:
            info_list.append([i["member"]["uname"],i["content"]["message"]]
    print(info_list)
```

3.3 保存程序

同样进行数据保存，方便后续使用。

```
def saveTxt(fname,fcontent):
    filename = str(fname) + ".txt"
    for content in fcontent:
        with open(filename, "a", encoding='utf-8') as txt:
            txt.write(content[0] +' '+content[1].replace('\n','') + '\n\n'
        #print(content)
```

3.4 主程序

主程序调用 up 主的 AV 号的获取函数，设定某一 up 主的空间号，访问其空间内的作品，得到 aid 号，在用评论函数，得到其视频下的评论并保存。

```
if __name__ == "__main__":
    getAllaidList(284813366,5,1)          #输入Up主的空间号和拉取数目及页数
    for item in aid_list:                 #获取aid列表后输出到api逐个获取评论
        print(item)
        info_list.clear()                 #每次清空上一次的评论列表
        catchCommentList(item)
        saveTxt(item,info_list)
```

4 对各榜单可视化分析

下来，尝试先对各榜单榜单的 top20 的作者的作品的点击率进行可视化的程序编写。建立在上述基础上，增添了 authors 和 play 两个数组，分别记录排行榜的排名和播放量。这里 author 本是想要记录作者的由于名字过长，还是选择排名可视化效果更佳。使用 matplotlib 库绘制折现图，并以榜单名+top20up 主播放量折线图命名保存。

```python
if __name__ == '__main__':
    play=[]
    authors=[]
    authorName=[]
    authorHome=[]
    all=['0','1','36','160','119','155','5','181']     #查看B站相关页面可以知道，存在一些问题，由于网站
    stateAll={'0':'全站',
              '1':'动画',
              '36':'知识',
              '160':'生活',
              '119':'鬼畜',
              '155':'时尚',
              '5':'娱乐',
              '181':'影视'}
    key=0
    for state in all:
        stop=0
        true_url= get_url()  #组成相应的url
        html=get_html(true_url)
        print(key)
        #print(authors)
        authors.append([])
        authorName.append([])
        play.append([])
        authorName.append([])
        for stateDate in pick_up_state(html):
            print(stateDate)
            #print(stop)
            authors[key].append(stateDate['ranking'])
            authorName[key].append(stateDate['author'])
            play[key].append(eval(stateDate['play']))
            print(play)
            #print(authors)
            stop+=1
            if stop == 20:
                break
        plt.title(stateAll[state]+'TOP20up主播放量折线图')
        plt.xlabel('ranking')
        plt.ylabel('play times/万')
        print(authors[key],play[key])
        plt.plot(authors[key],play[key])
        plt.savefig(stateAll[state]+'TOP20up主播放量折线图'+'.png')
        plt.show()
        key+=1
```

5 获取总榜单 top5 作者的视频评论可视化

之后，将一号和二号程序进行整合，将 top5 的作者名和和其空间号分别保存在 authors 和 authorName 两个二维数组中，可以分别用与对视频 aid 的请求和后续文件名用。写出如下程序：

```
: if __name__ == '__main__':
    key=0
    state='0'
    stop=0
    true_url= get_url()  #组成相应的url
    html=get_html(true_url)
    #print(key)
    #print(authors)
    authors.append([])
    authorName.append([])
    for stateDate in pick_up_state(html):
        #print(stateDate)
        #print(stop)
        authors[key].append(eval(stateDate['authorHome']))
        authorName[key].append(stateDate['author'])
        stop+=1
        if stop == 5:
            break
    key+=1

    stater=0
    for state in authors:
        auth=0
        #print(stater,auth)
        for author in state:
            #print(authors)
            info_list.clear()
            getAllaidList(author,1,1)
            for item in aid_list:
                #print(item)
                catchCommentList(item)
            #print(info_list)
            saveTxt(stater,auth,info_list)
            auth+=1
        stater+=1
```

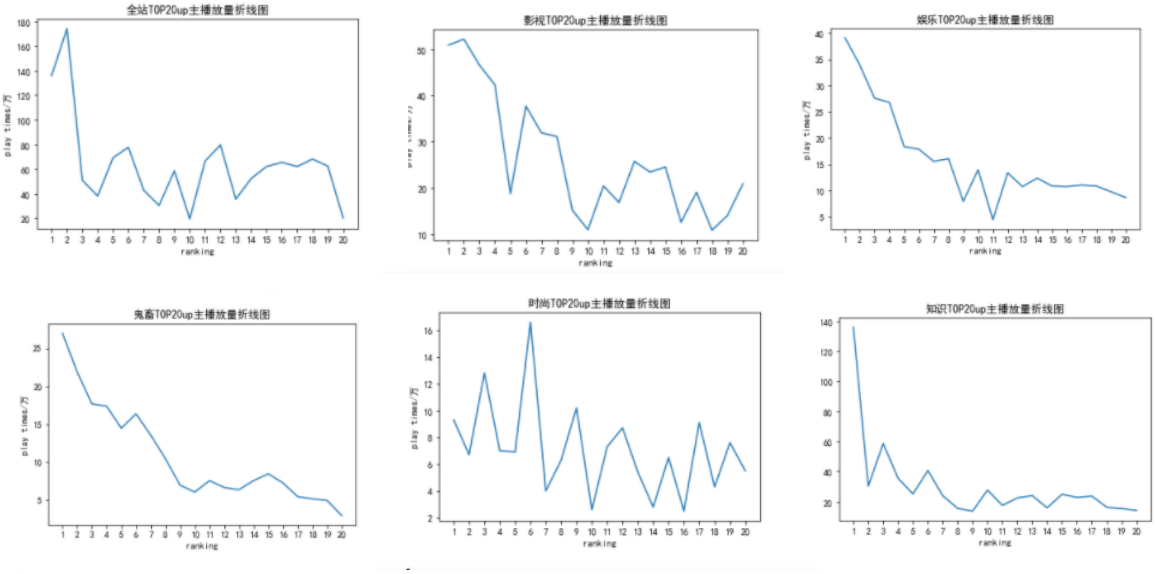
在进一步将保存下的文件重新导入 python，使用 jieba 分词，用 wordcloud 制作词云，top5 作者的视频作品的主要评论词云便跃然于眼前了。代码如下：

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import jieba
list=['互动大王',
      '我是郭杰瑞',
      '怪兽电台MonsterBlog',
      '小白测评',
      '庄不纯']
for i in list:
    text_from_file_with_apath = open(i+'.txt',encoding='utf-8').read()
    wordlist_after_jieba = jieba.cut(text_from_file_with_apath, cut_all = True)
    wl_space_split = " ".join(wordlist_after_jieba)
    my_wordcloud = WordCloud(background_color="white",width=1000, height=860, margin=2,font_pat

    plt.imshow(my_wordcloud)
    plt.savefig(i+'视频评论词云'+'.png')
    plt.axis("off")
    plt.show()
```

6 结论

完成了这一系列过程后，得到了一些简单地可视化成果。我们看到和总榜相比不同榜单都有着自己独特的地方。下面列出了获得的一部分图，以全站榜单作为参考，可以看到互联网的二八法则在这里展示的很突出，大部分的榜单中都是 top1 获得了压倒性的流量数，其中知识区的集中度最夸张，而时尚区也是少数的分布较均匀的榜单，这可能是偶然性也可能是由于内容受众的差异造成的。



看完了榜单可视化后，让我们看看全站榜的top5的视频评论吧（选取了4张展示）。可以看得出词云很好的展示了 up 主的视频的主要内容，和观众们的热点信息，比如在两张图中出现了 doge，彰显了这一个网络用语的流行性；在小白评测评论下最多的是手机一词，显示其内容的关联性；大大的朝阳、美国，反映了时政热点集中在现在的北京和美国，这可能与疫情相关。