基于 Python 对 B 站视频数据的抓取与分析

胡思源 张桂花

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘 要】在互联网普及程度逐渐全球化的二十一世纪，随着时间的推进，网络上的数据种类与量级以指数级的趋势增加。因此，如何准确、有效的获取到所需要的数据，进行相应的数据清洗，并进行可视化来直观的显现出数据所包含的隐藏信息，对此前数量大，但价值密度低且一直无法利用的数据来说，变为目前十分有意义的事。在这种背景下，本文从 Python 语言强大且多元的第三方库入手，利用 Selenium 库对哔哩哔哩视频网站进行视频信息数据和用户评论抓取，Pandas 库进行数据的清洗与预处理，Matplotlib，Wordcloud 进行数据可视化，由此获取到的数据隐藏信息，对目前哔哩哔哩网站视频创作者的创作方向与观众的喜爱趋势度有所帮助和指导。

【关键词】Python；爬虫；数据可视化；动态网页；反爬；

引言

在二十一世纪 4G 将末，5G 将至，数据量的在 4G 时代已经或 5G 时代将要爆发式增长的大环境下，对于数量巨大，但价值密度极其低下且无法被利用的数据的有效应用与挖掘分析，能够变废为宝的大数据技术成为各行各业所关注的焦点，通过对数据的挖掘和分析，往往能够得到有用的信息或规律，进而转化为有价值的信息和知识，来帮助部门决策者进行正确的决策。然而数据的获取却成为这一步最大的问题。本文将从 Python 语言的第三方库角度，对于如何获取所需数据进行详细讲解。

B 站作为近年来年轻人聚集占比最多的大型视频网站，汇聚了二十一世纪主流的年轻人的思想与见解，本文通过对 B 站视频信息数据与评论的获取，对获取数据进行数据分析与挖掘，不仅能对目前哔哩哔哩网站视频创作者的创作方向与观众的喜爱趋势度有所帮助和指导，而且能够了解相应种类视频下年轻人主流思想与见解。

1 网络爬虫及数据存储

1.1 爬虫简介

网络爬虫（又称为网页蜘蛛，网络机器人，在 FOAF 社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。另外一些不常使用的名字还有蚂蚁、自动索引、模拟程序或者蠕虫。

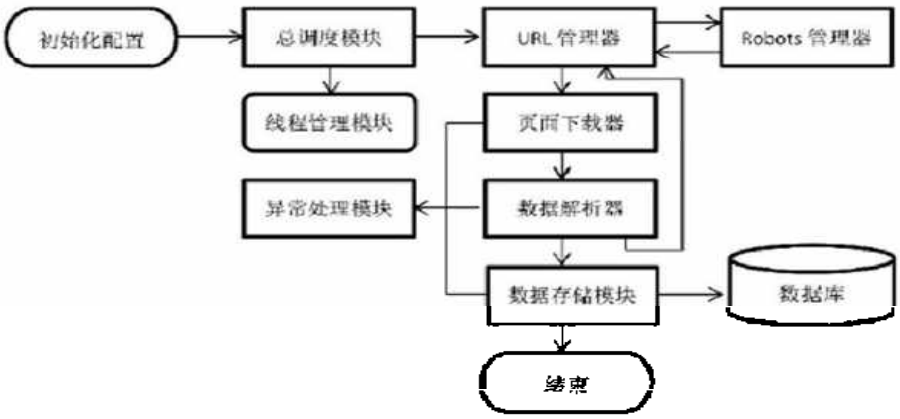

图 1[1] 网络爬虫基本架构图

Fig1 Basic architecture diagram of web crawler

URL 管理器：管理将要爬取的 URL，防止重复抓取和循环抓取。

网页下载器：这是下载网页的组件，用来将互联网上 URL 对应的网页下载到本地，是爬

虫的核心部分之一。

网页解析器：这是解析网页的组件，用来从网页中提取有价值的数据，是爬虫的另一个

核心部分。

流程：

由图 1 所示：

从爬虫调度端开始，对内部三个组件进行调度；

URL 管理器对 url 进行抓取，储存；

URl 管理器将 url 传入网页下载器进行下载；

网页下载器将访问得到的响应返回给网页解析器；

网页解析器进行网页解析，获取并存储所需数据，最终得到有用数据。

1.2 哔哩哔哩视频网站网页结构

视频搜索 url 结构

原始 url：

https://search.bilibili.com/all?keyword=%E7%BE%8E%E9%A3%9F&from_source=web_search&page=2

解析后可得：https://search.bilibili.com/all?keyword= 搜索内容 &page= 页数

通过以上所解析出的 url，我们可以由此对视频网页进行访问获取 url

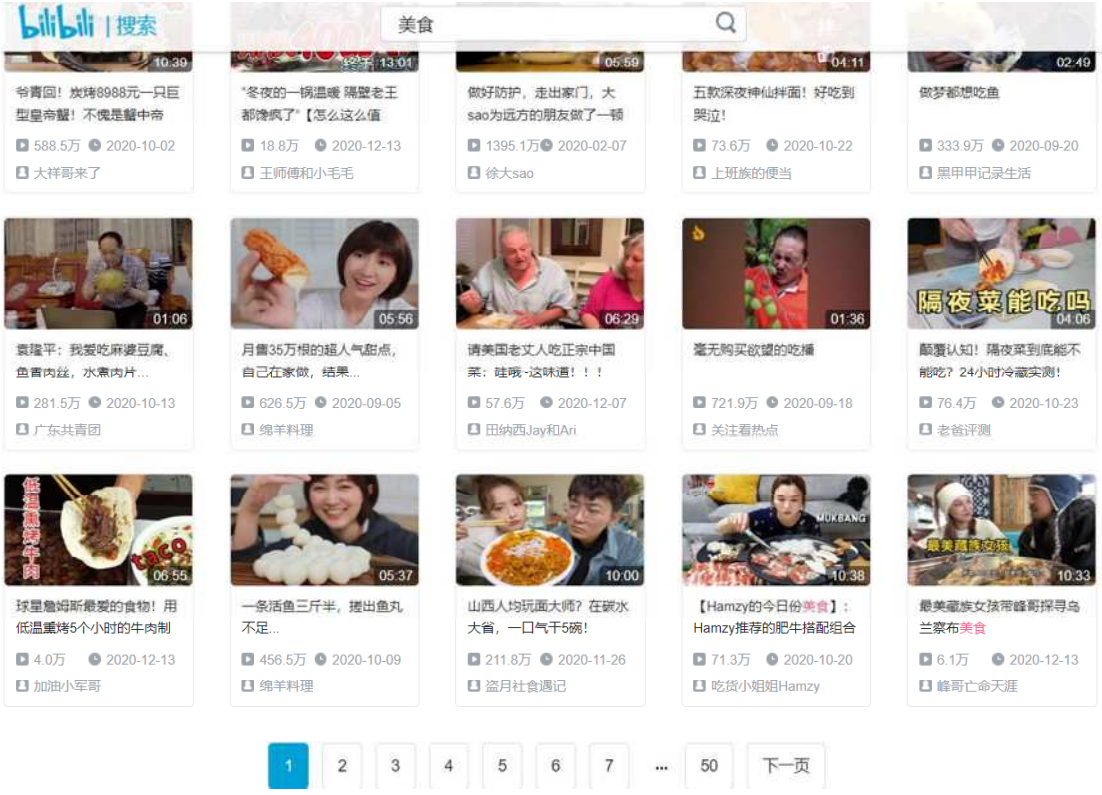

图 2 视频页

Fig2 The Video page

如图 2 所示，该网页每页拥有 20 个视频，那么每页具有 20 个 URl 视频链接。

直接键入检索关键字与期望爬取视频链接页数 [2]

1.3 获取视频网页链接

在本项目中，由于我们是分页使用 requests 进行获取视频的 url 链接，且所抓取网站为动态网页，所请求得到的网页内容会因时间，操作等因素发生改变，因此我们需要为 requests 的请求头添加 Cookie 来使服务器将该爬虫识别为特定用户，不会为每次访问 Cookie 的不同而返回相同的视频信息；其次，直接一次性将需要抓取的所有页数的视频 url 进行储存，减少遗漏或重复，获取视频链接代码如下。

```
# 保存视频主页的 url
        html_obj = etree.HTML(str_data)
        # 单页 20 个视频的 url 的一个列表 v
        url_list = html_obj.xpath( '//li[@class=" video-item matrix "]/a/@href' )

        for url_ in url_list:  # 将获取到视频的 url 添加上 https：
            url_ = "https:" + url_
            all_video_url_list_.append(url_)
        print(f" bilibili 搜索关键字视频：{data_}, 第 {page+1} 页 url 获取完成。 ")
```

1.4 获取视频链接反爬措施

```
# 用户代理
headers_ = {
    'User-Agent' :" Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    'Cookie' :" INTVER=1;
    'Referer' : f' https://search.bilibili.com/
```

```
all?keyword={data_}&page={page}'
    }
```

如上述代码所示，在构造爬虫时进行以下请求头的伪装：

添加 user-agent 来将自己的爬虫脚本伪装成浏览器的正常访问；

2 添加 Cookie 使服务器通过识别 Cookies 并鉴定出是合法用户；

3 添加 Referer 设定每次请求 response 的跳转页面为上一页视频页面；

4 添加 time 模块，利用 time.sleep() 来进行手动延迟，降低请求频率。

1.5 利用线程池来加快数据抓取速度

```
# 创建 5 个线程来进行网页访问和数据获取
pool = Pool(5)# 创建拥有 5 个进程数量的进程池
all_video_url_list_ = pool.map(crawl_video_
url,range(pages))
```

如上代码所示，引入 python 第三方库 multiprocessing 的 Pool 模块，利用 pool.map（函数名，可迭代对象）来加快抓取速度。

1.6 对抓取到的 url 列表进行去重

利用 set 函数的相同元素唯一的特性进行 url 去重：

```
list(set(new_all_video_url_list_))
```

1.7 使用 selenium 进行用户行为的模拟操作

利用 selenium 库的 webdriver 模块创建 Chrome 浏览器对象，结合 selenium 库的反爬虫机制 [3]

用户模拟行为操作：

如下方代码所示，执行 js 代码，拖动进度条到页面底端，获取此时网页数据。

```
for i in range(10): # 0-9
    time.sleep(0.5)
    j = (i+1)*1000
    js_ = f" document.documentElement.
scrollTop={j}"
    chrome_obj_p.execute_script(js_)
```

2 如下所示，通过调用浏览器对象的元素查找点击下一页。

```
chrome_obj_p.find_element_by_xpath( '//a[@
class=" next" ]' ).click()
```

1.8 使用 lxml 第三方库的 etree 模块将获取到的页面源代码转换为 html 对象

如下所示，利用 xpath 匹配方法获取所需数据。

```
# 视频标题
video_title = video_html.xpath( '//h1/@
title' )[0]
# 视频创作者
up_name = video_html.xpath( '//a[@report-
id=" name" ]/text()' )[0]
# 点赞数量
like_num = video_html.xpath( '//*[@id=" arc_
toolbar_report" ]/div[1]/span[1]/text()' )[0]
```

1.9 保存数据

1 如下所示，保存用户 ID 与评论为 CSV 文件。

```
with open(f" C:/Users/lenovo/Desktop/
B 站 _{data_}_ 评论 .csv", mode=' a+' ,
encoding=' utf-8' , newline=' ' ) as csv_file:
    f_csv = csv.writer(csv_
file,dialect=' excel' )
    for row in zip(user_name_list,reply_list):
        f_csv.writerow(row)
```

如下所示，保存视频信息为 Excel 文件。

```
writer = pd.ExcelWriter(file_path)
df = pd.DataFrame(video_info_list,columns=[" 视
频名称"," 创作者"," 点赞数"," 投币数"," 收藏
数"," 分享数"," 弹幕数"," 评论数" ])
# columns 参数用于指定生成的 excel 中列的顺序
df.to_excel(writer,index=False,
encoding=' utf-8' , sheet_name=' Sheet' )
writer.save()
```

2 数据处理

2.1 利用 Pandas 对视频信息数据进行预处理

直接操作 Excel 文件 [4,5] 读入 Excel 数据并转换为 Pandas 的 DataFrame 格式，并去除含有空值的行，代码如下代码所示：

```
filepath = ". / 数据 /B 站美食视频信息 .xlsx"
df = pd.read_excel(filepath,usecols=[" 视频播放
量"," 创作者"," 创作者关注数"," 点赞数"," 投
币数"," 收藏数"," 分享数"," 弹幕数"," 评论数" ])
full_df = df.dropna(axis=0,how=" any" )
```

2.2 进行类型转换

得到以上数据后，再用正则匹配数值，并将字符串类型的数据转换为浮点类型的数值，如视频的点赞数，收藏数等，如果含有万字，则将数值不变截取，如果不含万字，这将截取数据除以 10000。如下列代码所示：

```
# 数据处理
match_ = re.compile(" \d+(\.\d+)?" )
for i in range(full_df.shape[0]):
    for v in [0,2,3,4,5,6,7,8]:
        number = float(re.search(match_,
str(full_df.iloc[i,v])).group())
        if(" 万" in str(full_df.iloc[i,v])):
            full_df.iloc[i,v] = number
        else:
            full_df.iloc[i,v] = number/10000
            print(full_df.iloc[i,v])
```

2.3 得到清洗数据

如图 3 所示：

		视频播放量	创作者	作者关注	点赞数	投币数	收藏数	分享数	弹幕数	评论数
2	0	24.7	王小北酱	8.4	0.7791	0.1776	0.9363	0.1664	0.0226	0.0342
3	1	20.1	我是亦涵	5.9	0.8401	0.1457	0.5079	0.0506	0.1235	0.0738
4	2	182.6	美食作家	495.7	13.6	4.8	0.8727	0.2162	0.2	0.5047
5	3	166.9	二喵的饭	299.9	19	15.2	2.4	0.1378	0.2	0.2057
6	4	32	此次野花	2.1	1.3	0.3992	2.4	0.2094	0.0066	0.0186
7	5	13.7	胖龙的小	20.6	0.6706	0.1367	0.0252	0.2216	0.1903	0.0389
8	6	39.6	柴犬老丸	131.3	2.7	0.502	0.3406	0.1142	0.181	0.1071

图 3 清洗后数据（单位：万）

Fig3 Data after cleaning (unit: ten thousand)

评论数据已经不需要清洗，因此不做多余处理。

2.4 进行聚合操作

读入清洗后的数据，并按照创作者进行分组聚合求其均值再按照视频播放量进行降序排序，取其前 15 位，从而得到我们绘图需要的初始数据。如代码所示：

```
# 对数据进行分组聚合操作
df_inc=df.groupby('创作者').mean().sort_values(by="视频播放量",ascending=False)
```

3 数据可视化

3.1 作者视频平均信息柱状图

基于柱状图，我们可以更加直观的认识到抓取数据视频作者的视频质量与观众偏好度。

生成柱状图如图 4 所示：

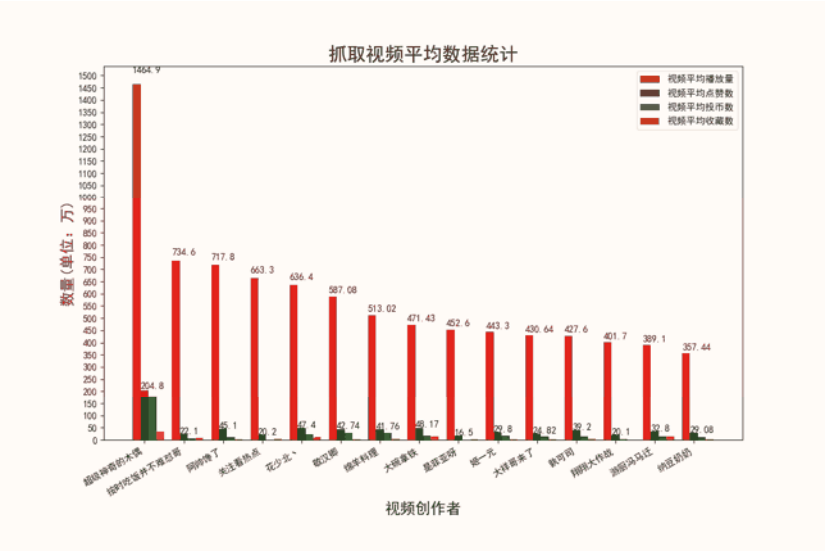

图 4 作者视频平均信息柱状图

Fig4 Author video average information histogram

从图 4，我们可以看出，视频的点赞，投币，收藏数与视频的播放量成正比，且对于质量较好的视频来说，其点赞数量一般在视频播放量的 5% 到 15% 之间浮动，而投币则比点赞的几率更低，基本在视频播放量的 2% 到 10% 之间浮动，收藏的几率也相对于投币数更低，基本在 1% 到 5% 之间浮动。其次，对于同样想进行美食类视频创作的新手创作者来说，完全可以借鉴排名前列的视频创作者来对自己的视频创作生涯进行相应的调整和规划。

3.2 抓取视频创作者占比饼图

基于饼图，我们可以直观的看出所抓取的 1000 个视频中，归属于那个作者最多。

生成饼图如图 5 所示：

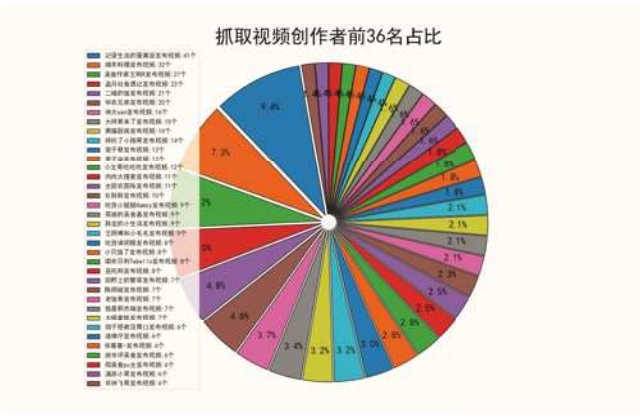

图 5 抓取视频创作者占比饼图

Fig5 Grab a pie chart of the percentage of video creators

从图 5，我们可以看出，抓取视频的发布数占比前三的创作者为记录生活的蛋黄派，绵羊料理，美食作家王刚 R，而同样的，作为分析结果，这三名创作者的关注数，以及视频质量，播放量等，也同样是名列前茅，很明显，天道酬勤，作为视频创作者，其粉丝数量和其播放量是在其发布视频数量的基础上逐渐递增的。

3.3 B 站美食视频评论词云图

词云图由于是由 Wordclound 进行绘制而成，因而也被称为 wordcloud 图，我们首先是用 jieba 库来进行分词统计，在将其生成的文本进行生成词云图，能够更加清晰的看出目前年轻群体时代主流词汇是哪些。

jieba 库进行中文分词排序核心代码如下所示：

```
for word in words:
if len(word) == 1:     # 单个词语不计算在内
    continue
else:
    counts[word] = counts.get(word, 0) + 1
# 遍历所有词语，每出现一次其对应的值加 1
```

生成词云图如图 6 所示：

从图 6，我们可以看出首先语气助词占比较多，我们暂不讨论，在美食视频的评论区里，首先当然是与美食相关的，如料理，绵羊，鸡蛋等食物频率较多；其次是 bili，当前网站为 B 站，出现 bili 等网站特色的词汇不足为奇；然后是中国，美国，作为二十一世纪的年轻人，所接触的最多的基本就是国家与国家之间的大事件作为谈资，首先我们可以知道，作为中国公民，爱国是现代年轻人基本的意识，所以中国这个词汇甚至在美食视频下都占比颇多，而美国作为目前世界第一的超级大国，与我国碰撞次数较多，因此占比也是不少；而经典 B 站特色词汇：好家伙，老公，朋友，关注，咸鱼，努力等词汇，都是 B 站年轻人群体的特色主流词汇。

结束语

本文研究的是使用 python 语言对国内最大，年轻人活跃众多的哔哩哔哩视频网站进行数据爬取，可视化与分析，通过对抓取视频信息进行柱状图、饼图排序后显示，展示大量视频信息数据间隐藏的各种关系与信息，可以对目前哔哩哔哩网站视频创作者的创作方向与观众的喜爱趋势度有所帮助和指导。

对评论内容进行词云图展示目前 21 世纪年轻人主流思想与热点词汇，可以对了解现代年轻群体主流态度及思维有所启发。

【参考文献】

[1] 李文华．基于 Python 的网络爬虫系统的设计与实现分析 [J]．内江科技，2021,42(02):58-59+26.

[2] 罗安然，林杉杉．基于 Python 的网页数据爬虫设计与数据整理 [J]．电子测试，2020(19):94-95+31.

[3] 许景贤，林锦程，程雨萌．Selenium 框架的反爬虫程序设计与实现 [J]．福建电脑，2021,37(01):26-29.

[4] 马宁，陈曦，张李铭．基于 Selenium 与 Openpyxl 的 Web 脚本自动化设计研究 [J]．电脑知识与技术，2020,16(01):51-53+70.

[5] 沈承放，莫达隆．beautifulsoup 库在网络爬虫中的使用技巧及应用 [J]．电脑知识与技术，2019,15(28):13-16.