摘要

最近的视频平台十分火热，成为人们生活中的一种潮流，同时也造就更多与此相关的职位，以 b 站为例，B 站作为一个创作、分享、讨论交流的平台，激励用户自制原创视频成为 UP 主，不管是全职 up 主，还是只是为了娱乐消遣，热爱剪辑，喜欢与人分享的业余 up 主。何之为是一个热门的 up 主，怎样才算是一个成功的视频，本文将会透过采集热门视频的标签弹幕等内容来进行了解分析

关键词：b 站 热门 数据采集

目录

1 商业背景

近年来，视频 +ugc 形式的火爆，造就了 b 站、抖音等新兴视频平台的崛起。然而，面对激烈的竞争，b 站作为一家成立 10 多年的企业，从最初的快速增长期已经进入增长放缓期。面对抖音、小红书等竞争对手的崛起， b 站的转型扩张成为必然。因此，b 站近年来推出“创作激励计划“，吸引越来越多的自媒体创作者加入。然而，b 站的“创作激励”仅仅在于金钱层面， 对于如何选择自己的发展领域、如何提升视频的制作质量以及引流技巧，b 站官方并没有给出直接的引导，所以，一般的新人 UP 主都需要花费巨大的时间成本来学习和失错，但成功的只有小部分，更多人则在投稿几次后收获惨淡流量后失去创作热情，埋灭在互联网中。其实很多时候不是创作者们没有潜力，而是 b 站作为一个年轻人居多的平台，存在一定的进入壁垒，在融入的过程相较于其他平台难度更高。因此，本研究通过 python 对 b 站的热门视频相关数据进行采集和挖掘，总结出热门视频的流量密码，利于新人UP 主快速融入平台，加速成长周期。

2 目标榜单

bilibili 的热门标签下的【综合热门】【入站必刷】【排行榜】，以获取所有视频中的部分标签、评论以及弹幕等所需要的数据，其中以上榜单每周都会进行更新，我们以最后一次爬取的时间为基准

图 1: 目标榜单

3 爬取用的库或工具

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from bs4 import BeautifulSoup
import requests
import time
import re
```

主要归纳为 selenium 库，beautifulSoup，time，request，以及虚拟驱动 Chromedriver，用法将在下文中提及

```
driver.get("https://www.bilibili.com/v/popular/all")
```

4 爬取目标

4.1 获取网址

使用了 Chromedriver 去模拟 Chrome 的操作，打开浏览器，并登入到 b 站网站"https://www.bilibili.com/v/popular/all"，同时使用了 selenium 中的 webdriverwait，去判断网站中所有的元素是否全部加载出来，再进行下一步操作

```
WebDriverWait(driver,100).until(EC.presence_of_all_elements_located)
```

4.2 进入每个视频中的网页

```
windows=driver.window_handles  # 获取当前窗口句柄集合（列表类型）
driver.switch_to.window(windows[0])  #进入第一个主要窗口title_name
= driver.find_element(by=By.XPATH,
    value='//*[@id="app"]/div/div[2]/div/ul/div[' + str(k) +
    ']/div[2]/p') # 标题名称
```

```
title_all.append(title_name.text) #把当前标题存入列表中
title_name.click()
driver.quit
```

循环使用 window_handles,switch_to_window(window[0])，quit 来进行不断地打开新视频网页，关闭，再打开下一个视频网页的动作，打开视频网站时在检查网站标题的 Xpath 路径，使用 beautifulSoup 来解析，并用 find_element 来找到视频的链接，再用 click 点进去

图 2: 检查元素

4.3 获取标题

新建好一个关于标题的空列表，把上文提及到的标题以.text 获取文本内容并存入标题的列表中

4.4 获取标签

```
label_all.append([]) # 生成子列表 用于存储每个视频的标签
tags = driver.find_elements(by=By.CLASS_NAME, value="tag") # 标签内容
for tag in tags:
    label_all[k-1].append(tag.text) #把当前标签存入列表中
driver.close()
```

新建好一个关于标签的空列表，并在每次视频的循环中新建一个空的子列表，进入到视频网页后，检查到大部分视频标签的 CLASS_NAME 均为“tag”，使用 find_element 去获取该部分标签，并存入对应视频的子列表里

图 3: 目标标签

4.5 获取弹幕

获取网页 cid 的函数

```
def get_cid(url): #拿取网页cid
  driver.get(url)
  WebDriverWait(driver,100).until(EC.presence_of_all_elements_located)
  html = driver.page_source
  cid = re.findall('https://upos-hz-mirrorakam.akamaized.net/upgcxcode/
                \d{2}/\d{2}/\d{9}',html) #透过re找到包含cid的html
  cid_num = cid.pop() #将所有找到的放在列表并取出最后一个
  cid_num = cid_num[-9:] #保留最后9位cid

  return cid_num
```

获取弹幕 cid 的函数

```
def get_danmu(cid):
  danmu = []
  url = "https://comment.bilibili.com/" + str(cid) + ".xml"
      #输入视频cid能取得所有弹幕的url

  request = requests.get(url)
  request.encoding='utf8'
  bs = BeautifulSoup(request.text, 'html.parser')
  results = bs.find_all('d')
  for result in results:
    danmu.append(result.text)
  return danmu
```

创建两个新列表来储存视频的 url 和弹幕的 url，使用 xpath 定位到每个视频的 url 名称，获取 href 并存入 url 的列表里，之后循环使用以上两个函数去透过输入 cid 来获取所有的弹幕

4.6 爬取评论

在已经进入了视频网页的基础上，下拉网页，待评论加载出来以后，通过 xpath 定位评论所在位置，并通过“.text”获取文本内容。

需要特别说明的是，在获取视频评论内容时，使用了两次“browser.execute_-script(“window. scrollBy(0,document.body.scrollHeight)”)”来实现下拉网页的效果。这是因为在实际爬取的过程中，我们发现网页在最初被打开并 加载的时候，只会显示视频主体、标签、播放列表等基本内容，而评论内容 需要下滑至网页底部才能加载出来；然而进行第一次的下拉操作以后，加上了”time.sleep”等待其加载完毕后依旧爬取失败，经过排错和多次尝试以 后，发现是因为视频开始播放以后，网页又会自动被拉回最顶部。再进行一 次下滑操作以后，评论内容爬取成功。

```
windows=browser.windor_handles
browser.switch_to.window(windows[-1])
browser.execute_script(“window.
    scrollBy(0,document.body.scrollHeight)”)
time.sleep(10)
browser.execute_script('window.scrollBy(0,document.body.scrollHeight)')
time.sleep(5)
for j in range(1,5):
    elem_text=browser.find_element_by_xpath(f'//*[@id="comment"]/div/div[2]
                                        /div/div[4]
                                        /div[{j}]/div[2]/p')
    print(elem_text.text)
    comments[i-1].append(elem_text.text) #把当前标签存入列表中
browser.close()
```

5 csv 写入

为之后的可视化处理，提前把爬取到的内容，包括标题，标签，弹幕，评论存入到 csv 里面

A	B	C	D	E	F	G	H
標題	弹幕1	弹幕2	弹幕3	弹幕4	弹幕5	弹幕6	弹幕7
约尔太太今	优雅！实在	优雅！实在	优雅！实在	优雅！实在	优雅！实在	优雅!实在太	优雅！实在
□□ 退 退	给老哥来颗	快递码应该	赚的其实都	这就像外卖	文件袋的快	村里代收一	好经典的小
【4K60FPS】	好甜好甜～	我都听哭了	太甜啦！！	太甜啦	好甜！！！	甜版凤凰传	我爱甜妹！
王心凌都没	火钳刘明	火钳刘明	《千手柱间	魔法师！！	火钳刘明	火钳刘明	哇，爆率真
丰收了 我真	他～会模发	他～～～会	请看作者，	滴滴，谁的	【国士无双	袁爷爷，我	三连了不敢
瘦了120斤后	期待跟我的	期待跟我最	期待跟我最	哈哈哈哈怎	姐姐惊到的	下午或者晚	姐姐那里，
厨师长教你	猫猫突然起	我不是她粉	对对对翻歌	还有《快乐	可怜的孩子	我这种四肢	天，看看当
【王晰X下洋	梦幻：又想	冷知识：作	每句歌词背	晰爹唱出了	实力唱功和	晰爹在无底	真的太厉害
【渊默行动	克里斯蒂安	百特曼！	真爱啊	卧槽我哭了	背后的纹身	百特曼！	真的好像！

图 4: 弹幕.csv 示例

6 可视化操作

通过可视化方式，将难以分辨的数据通过图形化的手段进行有效的表达，能高效简洁地展示我们的信息，帮助我们分析，挖掘爬出来的数的价值，把庞大的数据集引用起来。

6.1 标签处理

- 使用 tableau 工具对所获得所有的标签进行一个文字云可视，其中文字大小和文字颜色深浅均由 tableau 建立一个计算栏位进行统计的数量大小，可视化如下

图 5: 标签词云图

- 使用 piecharts 的 pie 构造一个玫瑰图，对于所爬取到的视频标签，我们进行了 fe 分区，其中包括’ 游戏’,’ 生活’,’ 知识’,’ 影视’,’ 音乐’,’ 其他’，他们之间的占比如下

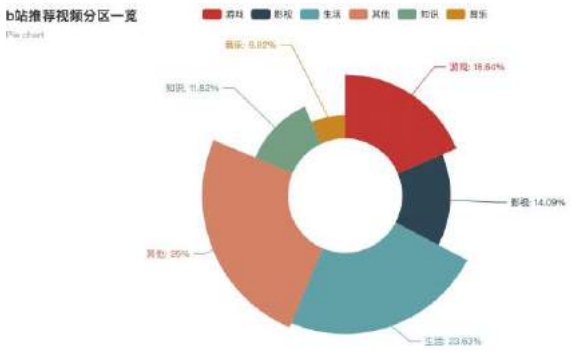

图 6: b 站推荐视频分区

6.2 弹幕处理统计

首先由于爬取到的视频数量过于庞大，其中累计的弹幕数量更是数不胜数，为了得到针对性的弹幕对于视频的作用，为此选取了几个案例进行分析。

```
import jieba
```

使用 jieba 的 python 库

读取到弹幕的 csv 文件，然后选取特定的视频案例的位置，在此选取的是“【4K60FPS】王心凌《爱你》经典现场！她太可爱了”的例子，把该视频标签下的所有弹幕进行筛选，存入到一个矩阵当中，建立一个 counts 的字典，使用循环把每一句的弹幕进行词汇的切割，并计算不是单个字的词的数量，把对应词汇和数字存入到字典当中，再次存入到一个新的 csv 里

```
counts={}
for i in range(1,360):
    words=jieba.lcut(data_x[i])
    for word in words:
        if len(word) == 1:
            continue
```

```
    else:
        counts[word] = counts.get(word, 0) + 1
```

6.3 弹幕词频可视化

依然是关于王心凌的案例，使用的是 tableau 工具对弹幕进行可视化，首先是筛选出排在前十二的词汇，（筛选条件为数量大于等于 10）作出一个条形图

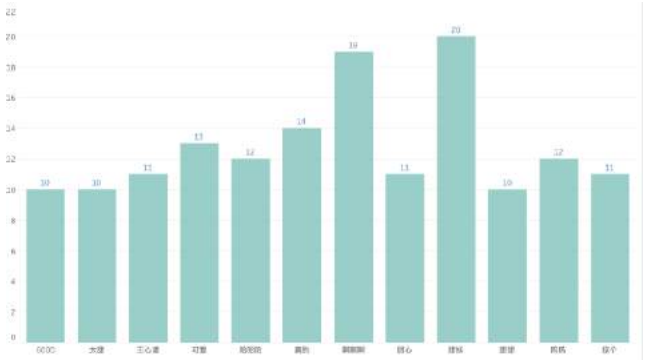

图 7: 案例下词汇条形直方图

再使用文字云进行所有弹幕词汇的可视，其中以文字的数量大小作为文字的大小以及文字颜色的深浅之分

图 8: 案例下弹幕词云图

6.4 播放量-标题云图

爬取了视频标题和视频播放量之后，将其合并称为字典格式，通过 word-cloud 制作“每周必看”视频的标题和对应的播放量的云图——其中颜色越深，代表播放量越高。结果如下图所示。

图 9: 2022 第 164 期“每周必看”——播放量与标题云图

6.5 热门视频 up 主粉丝数直方图

将“每周必看”入榜 up 主分为 5 个档次：第一档为小白 up 主，即粉丝数小于等于 1 万；第二档为成长期 up 主，即粉丝数大于一万且小于等于10万；第三档为小有名气的 up 主，即粉丝数大于 10 万且小于等于 100 万；第四档为知名 up 主，即粉丝数大于 100 万且小于等于 500 万；第五档为顶流 up 主，即粉丝数为 500 万以上。

考虑到可能有一些粉丝数较少的 up 主在视频列入“每周必看”榜单以后会出现显著的涨粉现象，于是对于过早的榜单不做研究，重点关注最近 4 期的每周必看的榜单信息。经过处理后，得到第 164 期-167 期（2022 年 5 月 6 日-6 月 2 日）“每周必看”up 主粉丝数分布直方图，如下图所示。

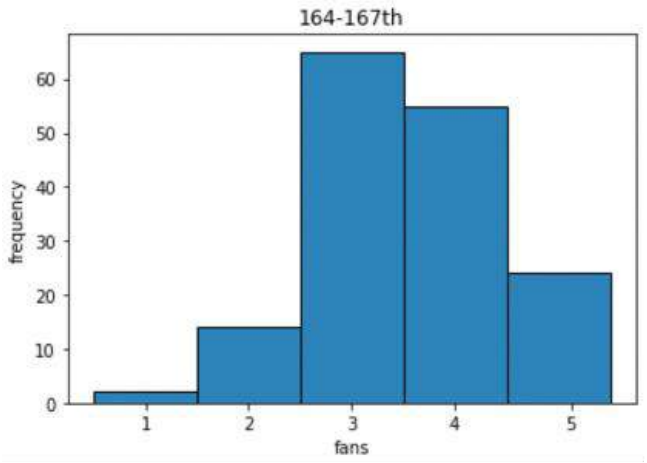

图 10: 第 164 期-167 期“每周必看”up 主粉丝数分布直方图

7 结论分析

- 比较轻松的生活区视频容易获得更多关注

  由采集数据可知 b 站上的视频主要分布在生活区（23.63%）、游戏区（18.64%）、影视区（14.09%）、知识区（11.82%）和音乐区（6.82%），且视频标签中“搞笑”一词出现频率最高，其次为“美食”、游戏相关和“科普”等。因此 b 站比较受欢迎的视频主要为偏向休闲娱乐，整体氛围较为轻松且带有一定内容的视频。

- 内容符合当下热点的视频的热度更高

  分析的三个案例视频《约尔太太今天约会 》《【4K60FPS】王心凌《爱你》经典现场！她太可爱了》和《丰收了我真的很想画这幅画》分别与当时热点《间谍过家家》、王心凌和袁隆平逝世一周年有关。因此，投制与当下热门讨论话题有关的视频更容易获得流量。此外，弹幕的

- 热门视频的标题更吸引眼球

  播放量最高的视频，在题材上，历史、美食、手工类以及热梗相关等大众接受度较高的话题的视频往往收获了最高的播放量，而游戏、赛车等话题的视频由于其领域门槛较高，播放量相较于榜单中其他视频较少；在标题命名上，播放量较高的视频在命名上普遍有以下几种特点和形式：

  第一种形式：以“投入 + 产出”格式命名（如“爆肝 30 天，我用 3000 块多米诺骨牌画了个龙珠”、“花五个月‘解锁‘《铠甲勇士激斗传》的最终结局！全部盔甲！”），这类标题通过比较夸张的数据，引起了观众的好奇心；

  第二种形式带有强烈语气（如“求求你别停产！这玩意我能吃一辈子！”；“小当家还是拍的太保守了！！”），通过强烈的情感吸引了观众的眼球；

  第三种形式：与热梗绑定（如：“《本草纲目》，但是废话文学“），紧跟时事。

- 热门视频的产出以小有名气的 up 为主

  1. 入选“每周必看”榜单的 up 主大多数以粉丝数位于 10 万到 500 万区间的 up 主。这类 up 主通常具有比较高的活跃度，投稿积极；

且拥有一定的创作能力，视频质量有保证，因此拥有相当数量的“死忠粉”，所以他们的视频播放量可以得到一定的保证。

2. 近 4 期的“每周必看”约有 10% 的入选视频来自于粉丝数小于10万的 up 主，他们的视频的流量数据甚至不输给知名 up 主。说明了粉丝少的 up 主的视频未必就是“被判死刑”，可以通过其视频的质量来提升其流量表现。

8 解决思路

- 定位视频内容的分区

  选择一个比较主流，占比较大的生活区、游戏区容易得到更多的关注

- 创作一个吸引眼球的标题

  参考上文分析出来的形式，可以透过新颖，夸张，紧跟时事或者强烈的情感去吸引观众的眼球，在众多同类视频中脱颖而出

- 视频内容方面可以选择大众接受度较高的话题

  若是一个新手 up 主可以先尝试一些诸如美食类的话题而之后再尝试一些专业性要求较高的话题

- 判断自己的成功

  从弹幕的角度来看，并非数量的多少才是决定内容的质量，有时弹幕里的内容和视频的内容关联度越高可以看出观众对于该内容认可

  从流量来看，有时粉丝数的多少并不完全决定流量的大小，仍是有部分较少粉丝的 up 主可以透过视频的质量在榜区中脱颖而出，不要以此灰心