班 级 **物联网1801**

学 号 21100180139

**河北化工医药职业技术学院**

**毕业论文**

**哔哩哔哩弹幕获取筛选器**

姓 名 张洪铭

专 业 物联网应用技术专业

指导教师 崔云

目录

[第一章：摘要 1](#_Toc54701898)

[1.1 说明 1](#_Toc54701899)

[1.2 关键词 1](#_Toc54701900)

[1.3 开发环境 1](#_Toc54701901)

[1.4 部署 1](#_Toc54701902)

[1.4 流程描述 2](#_Toc54701903)

[第二章：开发 2](#_Toc54701904)

[2.1 配置vscode 3](#_Toc54701905)

[2.2 配置git 3](#_Toc54701906)

[2.2.1 什么是git 3](#_Toc54701907)

[2.2.2 git 命令说明 3](#_Toc54701908)

[2.2.3 git 工作区，暂存区和版本库 4](#_Toc54701909)

[2.2.4 git 与github 4](#_Toc54701910)

[2.2.5 git 与 gitee 4](#_Toc54701911)

[2.3 开发过程 4](#_Toc54701912)

[2.3.1 资料获取 4](#_Toc54701913)

[2.3.2 cid获取 5](#_Toc54701914)

[2.3.3 编写爬虫 6](#_Toc54701915)

[2.3.4 分析弹幕文件结构 7](#_Toc54701916)

[2.3.5 解析xml 8](#_Toc54701917)

[2.3.6 使用jieba库提取关键字 8](#_Toc54701918)

[2.3.7 生成屏蔽文件 9](#_Toc54701919)

[2.3.8 导入 9](#_Toc54701920)

[2.4 测试 9](#_Toc54701921)

[2.4.1 测试一 10](#_Toc54701922)

[2.4.2 测试二 11](#_Toc54701923)

[2.4.3 vscode调试python 12](#_Toc54701924)

[第三章：jieba库 13](#_Toc54701925)

[3.1 使用 13](#_Toc54701926)

[3.2 模式 14](#_Toc54701927)

[第四章：总结 14](#_Toc54701928)

# 第一章：摘要

## 1.1 说明

Bilibili（下文简称B站）该网站于2009年6月26日创建，深受青年喜爱，但由于最近所谓“饭圈文化”、“快餐文化”等侵袭，导致该网站最大的特色“弹幕”也深受其害，充斥着各类不堪入目的词汇，尽管屏蔽词可以进行个别污言秽语的排查，但对于饭圈文化则无法规避，在此引入另一位B站用户的文章，本文不再过多赘述：

[1] <https://www.bilibili.com/read/cv4704353/>， 文手练习生赵命硬，B站

## 1.2 关键词

弹幕，爬虫，筛选

## 1.3 开发环境

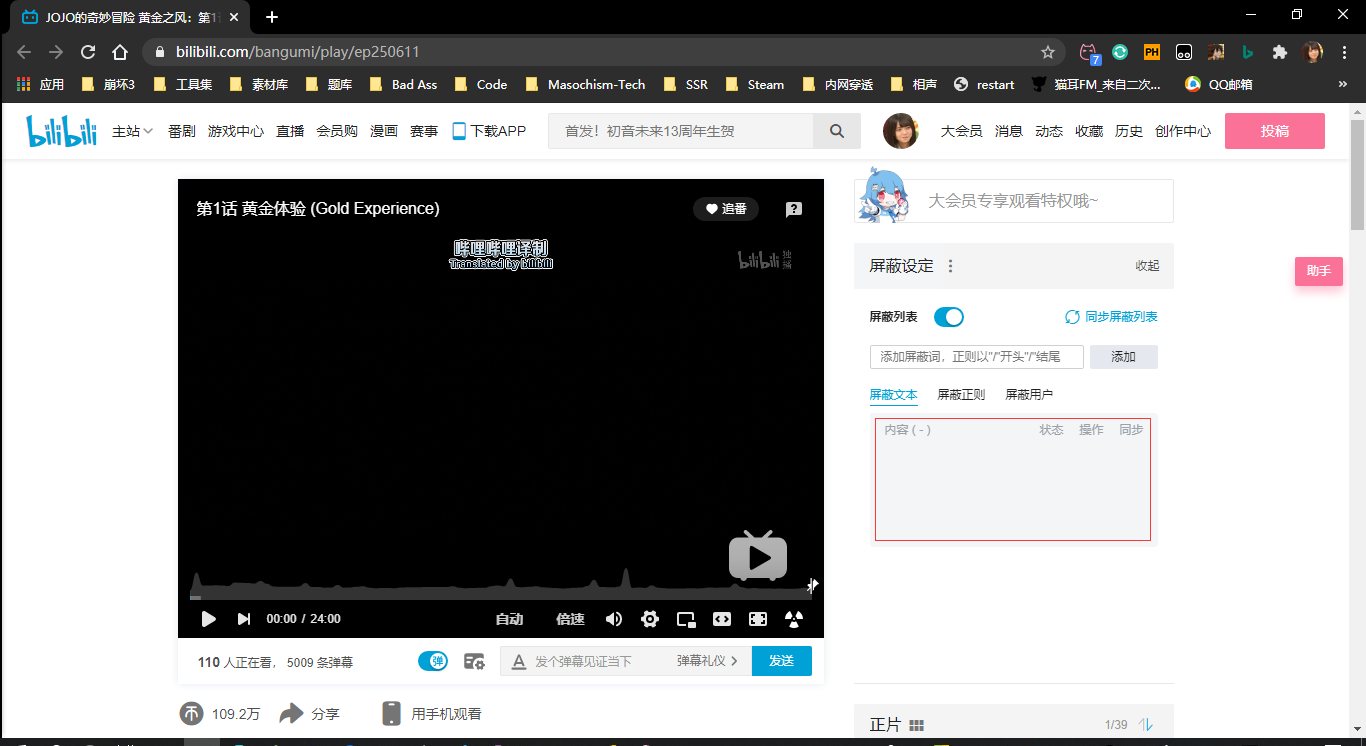
使用vscode和python3进行开发调试，使用python爬虫获取bilibili api下载弹幕数据，使用xml库进行数据提取与生成，使用jieba中文库进行关键词筛选。

## 1.4 部署

* + 前往<https://github.com/iPlanC/danmu_block/releases>下载压缩包；
  + 运行install.sh即可下载并安装python 3.8.5；
  + 请再次运行init.sh即可下载所需的代码库；
  + 运行block.py即可运行本程序；
  + 在控制台显示input BV (BID):时请输入视频(未测试)或番剧的BV号，**请勿使用Ctrl+C粘贴文本，会使程序结束(非bug)，必要时请使用右键粘贴；**

|  |
| --- |
| input BV (bid):BV1Ka4y1E75V |

* + 本程序将生成两个文件，bilibili.xml文件是获取到的弹幕库，barrages\_[bid].xml文件可以直接导入至哔哩哔哩屏蔽词；
    - bilibli.xml
    - barrages\_[bid].xml
  + **在图中红框处右键即可打开"导入\导出"菜单，即可选中相应的xml文件进行屏蔽。**

****

## 1.4 流程描述

首先使用CIDget(self, bvid)函数获取视频的cid，并使用此cid获取完整弹幕列表，将列表使用xpath提取后存入长字符串交由jieba.analyse进行筛选，筛选完成后存入列表，使用xml.dom相关组件保存屏蔽文件，由用户自行决定导入与否。

# 第二章：开发

## 2.1 配置vscode

安装python插件：从vscode拓展商店搜索python安装即可。

## 2.2 配置git

### 2.2.1 什么是git

Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。与常用的版本控制工具 CVS, Subversion 等不同，它采用了分布式版本库的方式，不必服务器端软件支持。

从<https://gitforwindows.org/>或国内镜像站<https://npm.taobao.org/mirrors/git-for-windows/>即可下载安装Git，使用Git可以轻松的控制开发流程及版本控制：

### 2.2.2 git 命令说明

* git init
  + git init命令可以在工作目录初始化git所需的文件
* git clone
  + git clone命令可以克隆远程git仓库
* git add
  + git add命令可以添加文件进行版本控制
* git commit
  + git commit命令可以暂存所有的更改并对其进行说明
* git push
  + git push命令可以将修改提交至远程服务器
* git pull
  + git pull命令可以将改动下载到本地

### 2.2.3 git 工作区，暂存区和版本库

* **工作区：**就是你在电脑里能看到的目录。
* **暂存区：**英文叫 stage 或 index。一般存放在 **.git** 目录下的 index 文件（.git/index）中，所以我们把暂存区有时也叫作索引（index）。
* **版本库：**工作区有一个隐藏目录 **.git**，这个不算工作区，而是 Git 的版本库。

### 2.2.4 git 与github

Git 并不像 SVN 那样有个中心服务器。

目前我们使用到的 Git 命令都是在本地执行，如果你想通过 Git 分享你的代码或者与其他开发人员合作。 你就需要将数据放到一台其他开发人员能够连接的服务器上。

### 2.2.5 git 与 gitee

大家都知道国内访问 Github 速度比较慢，很影响我们的使用。

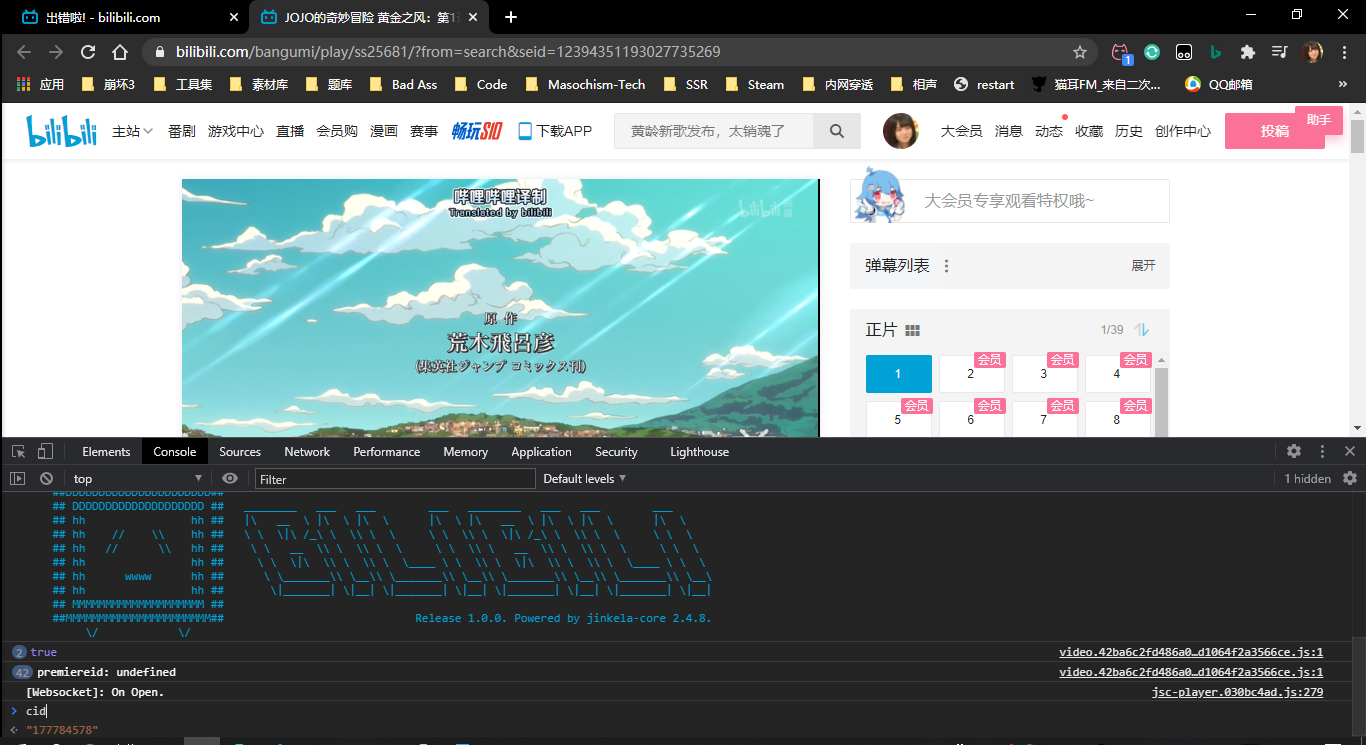
如果你希望体验到 Git 飞一般的速度，可以使用国内的 Git 托管服务——Gitee（gitee.com）。

Gitee 提供免费的 Git 仓库，还集成了代码质量检测、项目演示等功能。对于团队协作开发，Gitee 还提供了项目管理、代码托管、文档管理的服务，5 人以下小团队免费。

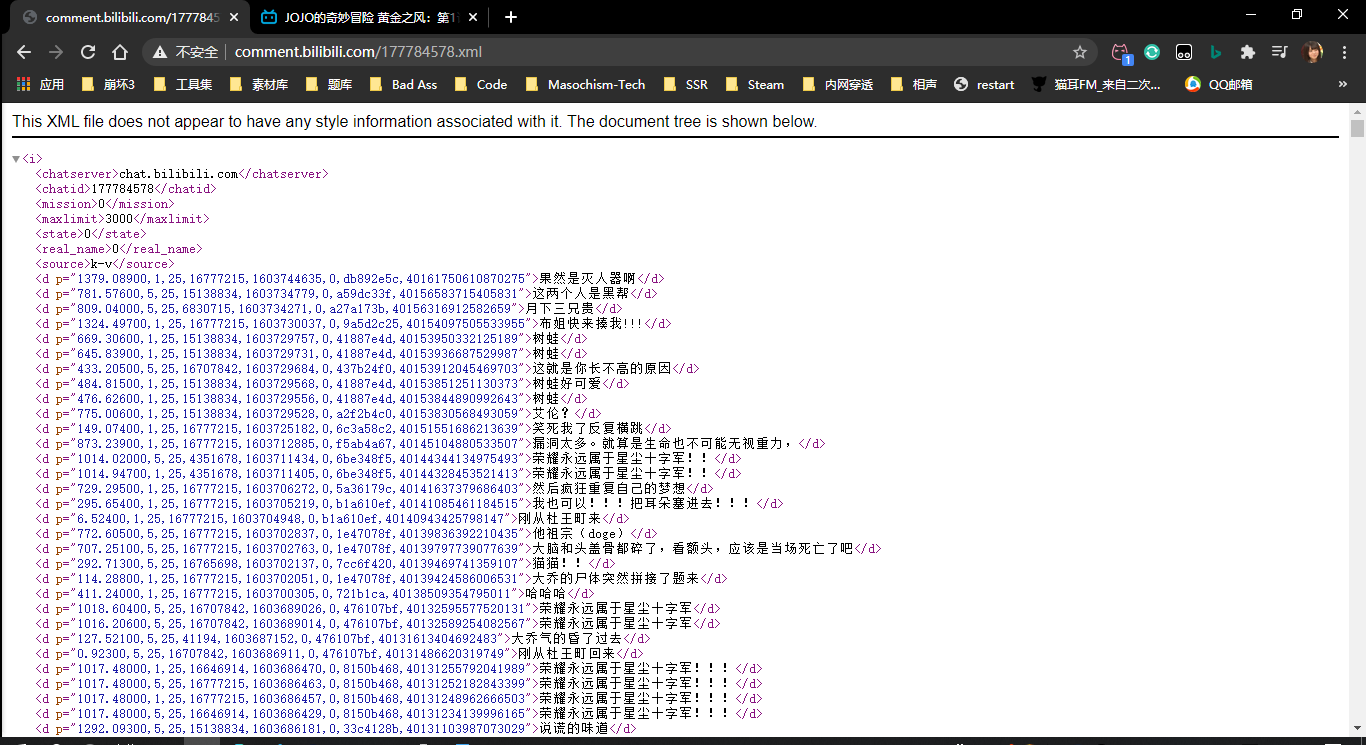
## 2.3 开发过程

### 2.3.1 资料获取

搜索资料获取B站弹幕库api，发现[http://comment.bilibili.com/[id].xml](http://comment.bilibili.com/%5bid%5d.xml)为相应视频的弹幕库，遂以BV号，也就是B站视频的唯一id号测试，发现报错404未找到，经查阅资料发现弹幕是以cid与视频绑定，通过网页控制台输入cid即可获得该视频的cid值。



使用此cid访问弹幕库，获得如下结果：



### 2.3.2 cid获取

在使用时每次需要通过控制台获取cid显然是不可以的，我们只能通过访问api通过BV号获取相应视频的cid

[https://api.bilibili.com/x/player/pagelist?bvid[bvid]&jsonp=jsonp](https://api.bilibili.com/x/player/pagelist?bvid%5bbvid%5d&jsonp=jsonp)

|  |
| --- |
| {"code":0,"message":"0","ttl":1,"data":[{"cid":177784578,"page":1,"from":"vupload","part":"01","duration":1441,"vid":"","weblink":"","dimension":{"width":1920,"height":1080,"rotate":0}}]} |

这段json文件其cid属性即为我们需要的cid号。

|  |
| --- |
| #获取视频cid      def CIDget(bvid):          url = "https://api.bilibili.com/x/player/pagelist?bvid=" + str(bvid) + "&jsonp=jsonp"          response = requests.get(url)          dirt = json.loads(response.text)          cid = dirt['data'][0]['cid']          return cid |

### 2.3.3 编写爬虫

|  |
| --- |
| # 新建一个类型名称为Bilibili  class Bilibili():  # 初始化爬虫  def \_\_init\_\_(self, cid):  self.headers = {           'Host': 'comment.bilibili.com',           'Connection': 'keep-alive',           'Cache-Control': 'max-age = 0',           'Upgrade-Insecure-Requests': '1',           'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.92 Safari/537.36',           'Accept': 'text/html, application/xhtml + xml, application/xml;q = 0.9, image/webp, image/apng, \*/\*;q = 0.8',           'Accept-Encoding': 'gzip, deflate, br',           'Accept-Language': 'zh-CN, zh;q = 0.9',           'Cookie': 'finger = edc6ecda; LIVE\_BUVID = AUTO1415378023816310; stardustvideo = 1; CURRENT\_FNVAL = 8; buvid3 = 0D8F3D74-987D-442D-99CF-42BC9A967709149017infoc; rpdid = olwimklsiidoskmqwipww; fts = 1537803390'          }          self.url = 'http://comment.bilibili.com/' + str(cid) + '.xml'          self.barrage\_reault = self.get\_page()  # 获取内容写入本地  def get\_page(self):          try:              time.sleep(0.5)              response = requests.get(self.url, headers = self.headers)          except Exception as e:              print('获取xml内容失败, %s' % e)              return False          else:              if response.status\_code == 200:                  with open('bilibili.xml', 'wb') as f:                      f.write(response.content)                  return True              else:                  return False |

该爬虫会将传入的cid解析并获取对应视频的xml弹幕库保存至本地，至此，数据获取完成。

### 2.3.4 分析弹幕文件结构

<d p="484.81500,1,25,15138834,1603729568,0,41887e4d,40153851251130373">树蛙好可爱</d>

以此弹幕为例，可以看到每一条弹幕内容都被<d>/<d>标签标记，而弹幕的属性也被保存在<d></d>的p属性中，下面我们来逐一分析一下

p=[1],[2],[3],[4],[5],[6],[7],[8]

[1]：由视频可知，此项为弹幕发送的时间；

[2]：弹幕类型；

[3]：字体大小

[4]：十六进制转十进制的颜色码

[5]：UNIX时间戳

[6]：弹幕池

[7]：发送者的UID经过CRC32校验的结果

[8]：顺序和历史弹幕

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Pool\type** | **1** | **4** | **5** | **6** | **7** | **9** |
| **0** | 滚动弹幕 | 底部弹幕 | 顶部弹幕 | 逆向弹幕 | 特殊弹幕 |  |
| **1** |  |  |  |  | 精准弹幕 |  |
| **2** |  |  |  |  |  | BAS弹幕 |

其他属性本工具暂无需求，本工具专注大量出现的重复词，故只筛选弹幕内容。

### 2.3.5 解析xml

|  |
| --- |
| # 解析xml      def param\_page(self):          time.sleep(1)          if  self.barrage\_reault:              # 文件路径，html解析器              html = etree.parse('bilibili.xml', etree.HTMLParser())              # xpath解析，获取当前所有的d标签下的所有文本内容              results = html.xpath('//d//text()')              return results |

使用xpath解析器获取内容并返回列表

### 2.3.6 使用jieba库提取关键字

|  |
| --- |
| # 词云制作      def make\_wordCloud(self, filename):          doc = xml.dom.minidom.Document()          root = doc.createElement('filters')          doc.appendChild(root)          barrages = self.param\_page()          text = ""          for barrage in barrages:              text = text + barrage          keywords\_textrank = jieba.analyse.textrank(text)          for word in keywords\_textrank:              nodeItem = doc.createElement('item')              nodeItem.setAttribute('enabled', 'false')              nodeItem.appendChild(doc.createTextNode('t=' + word))              root.appendChild(nodeItem)          fp = open('block\_' + filename + '.xml', 'w', encoding='utf-8')          doc.writexml(fp, indent = '\t', addindent = '\t', newl = '\n', encoding = "utf-8")          fp.close()          print(keywords\_textrank) |

将列表传递给jieba中文分词库，使用jieba.analyze分析根据重复量计算权重并存储进keywords\_textrank列表，根据列表生成xml文件以导入B站。

### 2.3.7 生成屏蔽文件

<https://harrynull.tech/bilibili/index.html>

通过此处提供的文件，了解到B站屏蔽文件的结构为：

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>      <filters>          <item enabled="false">t=树蛙</item>      </filters> |

每一项item结点都为一个屏蔽词，enabled属性决定导入后默认开启还是关闭，结点内容即为”t=[屏蔽词]”

在此，我们已经生成了一个可用的屏蔽文档，完整代码请见下方链接：

<https://github.com/iPlanC/danmu_block/blob/master/block.py>

### 2.3.8 导入

打开B站，点击右侧弹幕列表 -> 屏蔽设定，在方框内右键即可导入屏蔽表，默认情况关键词均为关闭。

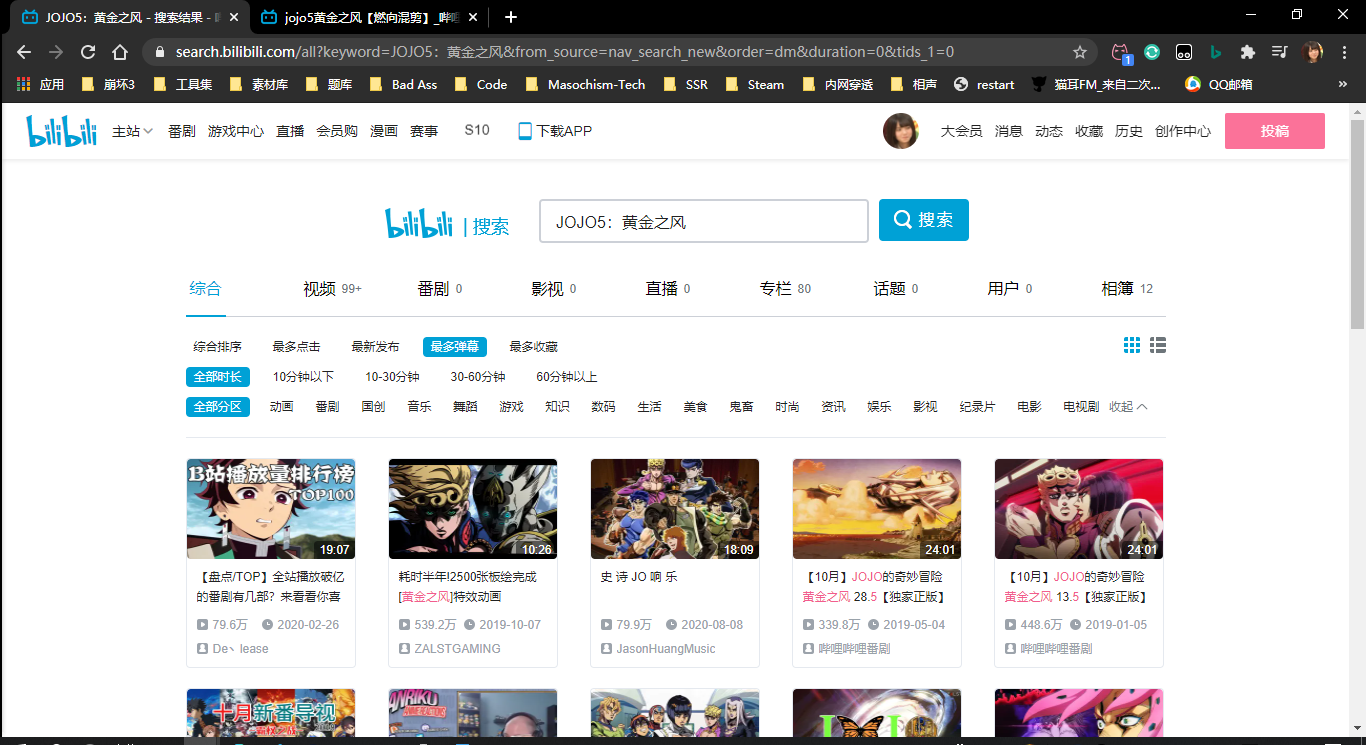
## 2.4 测试

### 2.4.1 测试一

以B站番剧《JOJO5：黄金之风》为例，其BV号为：BV1nW41127fv，一天两组一次，对这个视频进行十次提取观察其结果：

|  |  |  |
| --- | --- | --- |
| 1 | 2020-10-25 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 2 | 2020-10-25 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 3 | 2020-10-26 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 4 | 2020-10-26 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 5 | 2020-10-27 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 6 | 2020-10-27 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 7 | 2020-10-28 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 8 | 2020-10-28 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 9 | 2020-10-29 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |
| 10 | 2020-10-29 | ['无敌', '属于', '十字军', '星尘', '酸酸', '杜王町', '星辰', '替身', '飞机', '绅士', '乔家', '不见', '设定', '世界', '阶级', '回来', '高能', '电车', '意大利', '名画'] |

可以看到，短期内结果稳定，但是因为B站内部限流原因，其内部团队会自动清理冗余弹幕，所以不推荐使用番剧进行提取，以B站番剧《JOJO5：黄金之风》为例：



可以使用民间制作的视频来作为提取源进行筛选相对精确，实验将在下一节展示。

### 2.4.2 测试二

以视频《耗时半年!2500张板绘完成[黄金之风]特效动画 - ZALSTGAMING》为例，其BV号为：BV1SJ411F77h，一天两组一次，对这个视频进行十次提取观察其结果：

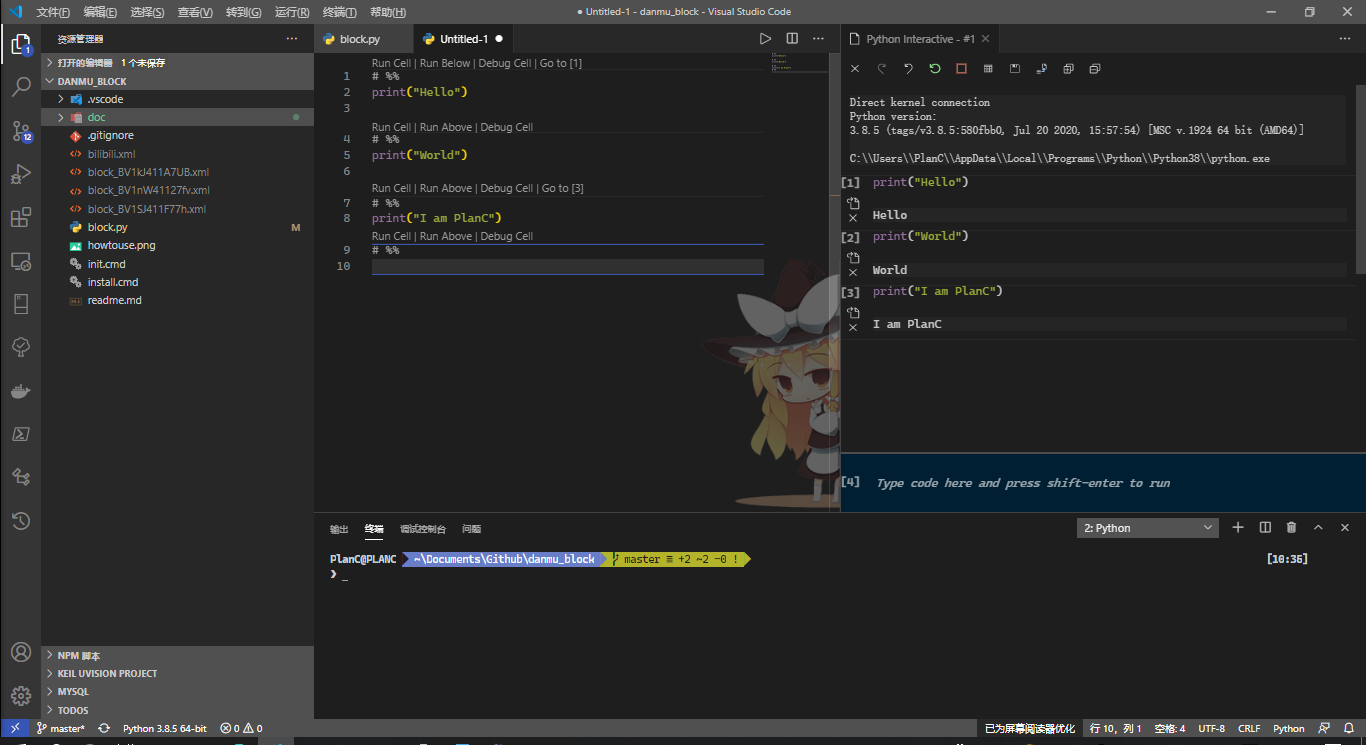
|  |  |  |
| --- | --- | --- |
| 1 | 2020-10-25 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 2 | 2020-10-25 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 3 | 2020-10-26 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 4 | 2020-10-26 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 5 | 2020-10-27 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 6 | 2020-10-27 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 7 | 2020-10-28 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 8 | 2020-10-28 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 9 | 2020-10-29 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |
| 10 | 2020-10-29 | ['属于', '十字军', '星尘', '镇魂', '替身', '硬币', '体验', '世界', '投币', '黄金', '没用', '收下', '时间', '比赞', '可能', '失败', '哀歌', '没有', '灵魂', '荒木'] |

测试结果如上，其筛选结果优秀，大量词汇均来自《jojo5：黄金之风》中的高频词。

### 2.4.3 vscode调试python

Vscode的python插件中自带块调试功能，需要安装ipykernel模块，在需要调试的块加”# %%”即可，例如：

|  |
| --- |
| # %%  print("Hello")  # %%  print("World")  # %%  print("I am PlanC") |



可以看到有类似python IDLE的窗口运行脚本

# 第三章：jieba库

## 3.1 使用

jieba分词的原理：

Jieba分词依靠中文词库

- 利用一个中文词库，确定汉字之间的关联概率

- 汉字间概率大的组成词组，形成分词结果

- 除了分词，用户还可以添加自定义的词组

|  |
| --- |
| import  jieba  txt = open(".\\统计源.txt", "r", encoding='utf-8').read()  words = jieba.lcut(txt)  counts = {}  for word in words:      if  len(word) == 1:          continue      else:          counts[word] = counts.get(word, 0) + 1    items = list(counts.items())  items.sort(key=lambda x: x[1], reverse=True)  for i in range(15):      word, count = items[i]      print("{0:<5}{1:>5}".format(word, count)) |

以上代码即可统计源文件中词组的出现频率并输出到控制台，但仍需要进一步处理，才能获得更加真实的结果。

## 3.2 模式

jieba支持四种分词模式：

* 精确模式，试图将句子最精确地切开，适合文本分析；
* 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；
* 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。
* paddle模式，利用PaddlePaddle深度学习框架，训练序列标注（双向GRU）网络模型实现分词。同时支持词性标注。paddle模式使用需安装paddlepaddle-tiny，pip install paddlepaddle-tiny==1.6.1。目前paddle模式支持jieba v0.40及以上版本。jieba v0.40以下版本，请升级jieba，pip install jieba --upgrade 。

本脚本使用基于TF-IDF算法进行提取的方式，对弹幕内容进行分割。

# 第四章：总结

通过该脚本的开发，充分学习了python爬虫与引入和使用代码库，使用git进行代码管理，用户手册的编写与代码风格优化，认识到了python的简洁之道，也理解了python没有成为主流语言的根本原因。通过实际项目的训练加强了对git的认识与理解，体会到了真正开发下各种开发工具的选择与使用。