B站视频弹幕爬虫+情感分析+词云图

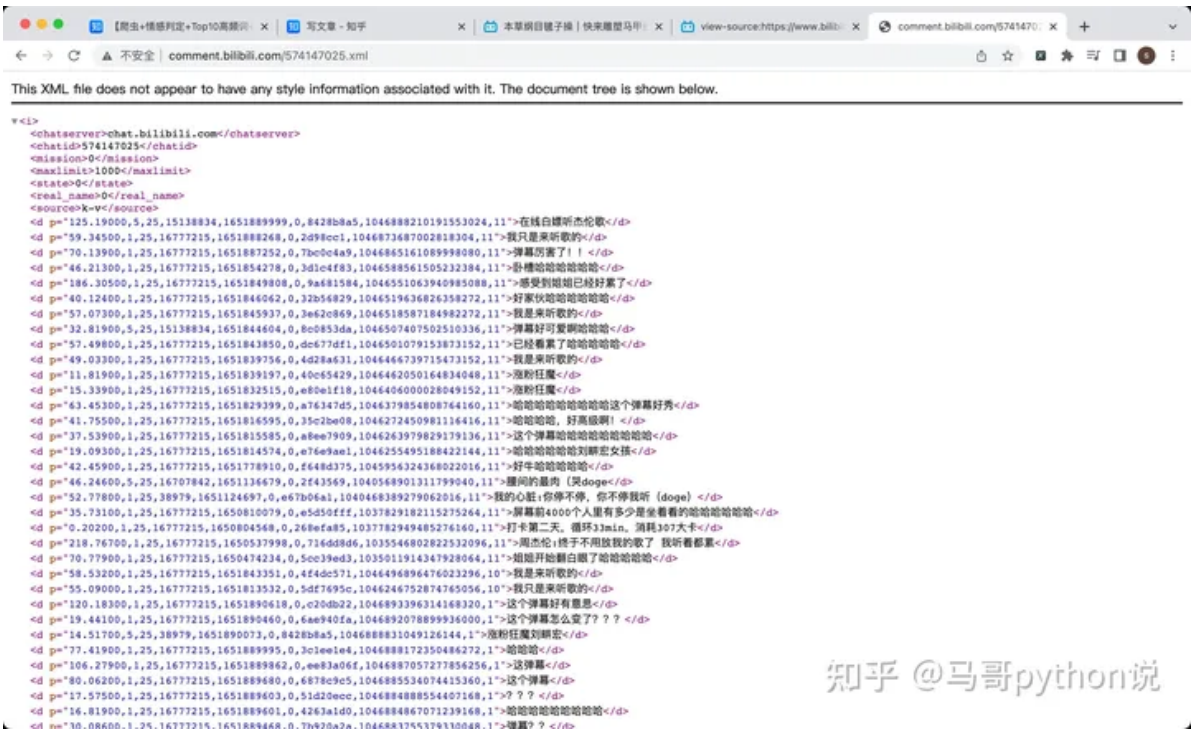
一、爬虫部分

1.1 获取视频评论的接口

首先分析B站弹幕接口。经过分析，得到的弹幕地址为：

```
http://comment.bilibili.com/{cid}.xml
```

以B站视频 https://www.bilibili.com/video/BV12A411N79s 为例，查看网页源代码，可以找到对应的 cid 为 315006821 ，所以该视频对应的弹幕接口地址是：http://comment.bilibili.com/315006821.xml



1.2 获取视频的 URL 地址

```
# 读取"新疆棉"弹幕数较多的视频BV号
with open('新疆棉播放量最多的几个视频的BV号.txt', 'r', ) as file:
    bv_list = [bv.strip() for bv in file.readlines()]

# 开始爬取
for bv in bv_list:
    get_bilibili_danmu(bv, result_file=csv_file)
```

我们首先从 bilibili 网站上找到弹幕数比较多的几个视频，并将对应视频的 BV 号存入一个 .TXT 文件中。然后，我们从文件中读取 BV 号，并调用爬虫函数。

1.3 爬取弹幕内容

首先，我们需要根据 BV 号获取对应视频的 HTML 文件，并从 HTML 文件中获取视频的 cid。

```
r1 = requests.get(url='https://api.bilibili.com/x/player/pagelist?bvid=' +
bv, headers=headers)
html1 = r1.json()
cid = html1['data'][0]['cid']                          # 获取视频的 cid
print(cid)
```

其次，得到 cid 之后，根据 b站的网址构成格式，得到弹幕的 URL 地址。从而利用 BeautifulSoup 库中的**解析器**对网页进行解析，得到弹幕内容的**列表**。

注意：弹幕地址对应的文件是 XML ，必须使用 **xml 解析器**进行解析。

```
danmu_url = 'http://comment.bilibili.com/{}.xml'.format(cid)    # 弹幕地址，
是 XML 文档
r2 = requests.get(danmu_url)
html2 = r2.text.encode('raw_unicode_escape')                    # 编码格式
soup = BeautifulSoup(html2, 'xml')                              # 必须使用
xml 解析器
danmu_list = soup.find_all('d')
```

接下来，我们就可以分解弹幕内容列表，从而获取我们所需要的元素。通过观察 XML 文件，我们可以发现：

- 弹幕内容是 `<d>` 标签的文本内容；
- 时间戳位于 `<d>` 标签的 `p` 属性中，是第五个数值；

```
time_list = []                                                  # 弹幕时间
text_list = []                                                  # 弹幕内容
for d in danmu_list:
    data_split = d['p'].split(',')                              # 按逗号分隔
    temp_time = time.localtime(int(data_split[4]))              # 转换时间格式，将整
数类型的时间戳转换为时间元组格式
    danmu_time = time.strftime("%Y-%m-%d %H:%M:%S", temp_time)
    time_list.append(danmu_time)
    text_list.append(d.text)
```

最后是数据的存储工作。我们定义一个 DataFrame 对象，用于实现数据的存储。

```
df = pd.DataFrame()                                    # 初始化一个
DataFrame对象
df['弹幕时间'] = time_list
df['弹幕内容'] = text_list
if os.path.exists(result_file):                        # 如果文件存在，不需
写入字段标题
    header = None
else:                                                  # 如果文件不存在，说
明是第一次新建文件，需写入字段标题
    header = ['弹幕时间', '弹幕内容']
df.to_csv(result_file, encoding='utf_8_sig', mode='a+', index=False,
header=header)
```

1.4 如何获取 cid

[教大家获取B站视频cid - 哔哩哔哩 (bilibili.com)]

二、情感分析部分

2.1 读取数据并转换数据格式

我们将数据从 CSV 文件中读出（此时是 DataFrame 对象），并将其转换为 Python **列表**。

```
# 获取评论内容列表:
# .values 将 DataFrame 列转换为 NumPy 数组
# .tolist() 方法将该数组转换为 Python 列表
df = pd.read_csv('新疆棉弹幕.csv')
barrage_list = df['弹幕内容'].values.tolist()
print('length of barrage_list is:{}'.format(len(barrage_list)))

barrage_list = [str(i) for i in barrage_list]                    # 将所有
元素转换成字符串
```

2.2 执行情感分析

首先，我们需要初始化一些参数。主要是便于存储数据和进行情感分析。

```
score_list = []                                        # 情感评分值
tag_list = []                                          # 打标分类结果
opt_count, neg_count, mid_count = 0, 0, 0
```

其次，我们**遍历弹幕列表**，并进行文本情感评分，根据评分结果进行分类。将得分和分类结果分别存入两个不同的列表中，以便于之后的处理。

```
for barrage in barrage_list:
    tag = ''
    # 创建一个 SnowNLP 类的实例，并用文本 barrage 计算情感分数，然后通
过.sentiments 属性返回文本的情感分数
```

```
    sentiments_score = SnowNLP(barrage).sentiments
    if sentiments_score < 0.3:
        tag = '消极'
        neg_count += 1
    elif sentiments_score >= 0.7:
        tag = '积极'
        opt_count += 1
    else:
        tag = '中性'
        mid_count += 1
    score_list.append(sentiments_score)                    # 得分值列表
    tag_list.append(tag)                                   # 判定结果列表
```

接下来，我们就可以利用 pandas 的强大能力对数据进行简单分析：统计正向、负向、中立弹幕的数量。

```
df['情感得分'] = score_list
df['分析结果'] = tag_list
grp = df['分析结果'].value_counts()                    # 计算“分析结果”中每个唯一值的频数
print('正负面评论统计：', grp)
```

最后，根据统计结果，我们绘制饼状图，并保存情感分析结果到 excel 文件中。

```
grp.plot.pie(y='分析结果', autopct='%.2f%%')          # 画饼图，数据精确到小数点后两位
plt.title('新疆棉弹幕_情感分布占比图', fontproperties='SimHei')
plt.savefig('新疆棉弹幕_情感分布占比图.png')             # 保存图片
df.to_excel('新疆棉弹幕_情感评分结果.xlsx', index=None)      # 把情感分析结果保存到excel文件
print('情感分析结果已生成')
```

三、词云部分

3.1 定义停用词和存储路径

```
stopwords = ['这个', '吗', '的', '啊', '她', '是', '了', '你', '我', '都', '也', '不', '在', '吧', '说',
             '就是', '这', '有', '就', '或', '哇', '哦', '这样', '真的']
outfile = 'C:/Users/DELL/Desktop/python爬虫基础/项目实战训练/b站视频弹幕爬虫+情感分析+词云/picture/新疆棉弹幕_词云图.jpg'
```

3.2 绘制词云

```
background_Image = np.array(Image.open(ciyun_background))    # 读取背景图片
wc = WordCloud(
    background_color="white",                                # 背景颜色
    width=1500,                                              # 图宽
    height=1200,                                             # 图高
    max_words=1000,                                          # 最多字数
    # font_path="C:\Windows\Fonts\simhei.ttf",               # 字体文件路径，
根据实际情况(Windows)替换
    stopwords=stopwords,                                     # 停用词
    mask=background_Image,                                   # 背景图片
)
# 对 v_str 进行分词，用空格连接分词结果
jieba_text = " ".join(jieba.lcut(v_str))
wc.generate_from_text(jieba_text)                            # 生成词云图
wc.to_file(v_outfile)                                        # 保存图片文件
print('词云文件保存成功: {}'.format(v_outfile))
```