**云计算大作业说明文档**

**研究问题：**

1. 一段时间内今日头条的新闻文章标题中热度最高的关键词
2. 一段时间内今日头条的新闻文章中热度最高的新闻类型

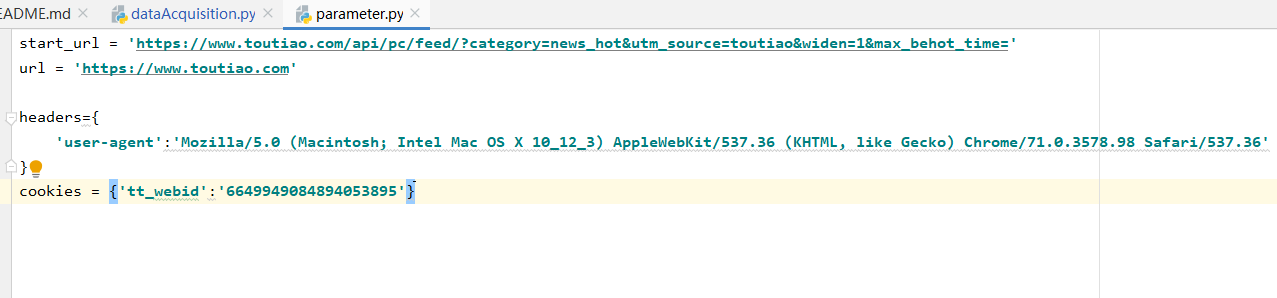
**研究问题一：**

1. 数据
2. 数据获取方式

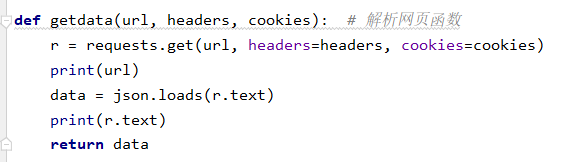
通过Python爬虫爬取今日头条的实时新闻文章数据（代码见dataAcquisition.py）

①获取as和cp的参数



②网页的url需要cp和as参数，通过**def** get\_as\_cp()获取，在parameter.py存储了需要的headers和cookies信息

③在main()函数里将获得的data存储到py文件的global数组或者字典里，使用getdata()获取了网页的源数据

1. 在getdata()里使用request.get()方法获得了数据源
2. 在main()里调用getdata()，然后按照其中一个global数组的方式存到全局数组或者字典里，然后打印相关的一部分数据，来实时获得进度



1. 数据格式

所得数据主要包含以下几方面内容：

①title，即标题

②url，即本条新闻的链接

③source，即本条新闻所属头条号

④media\_url，即所属头条号的连接

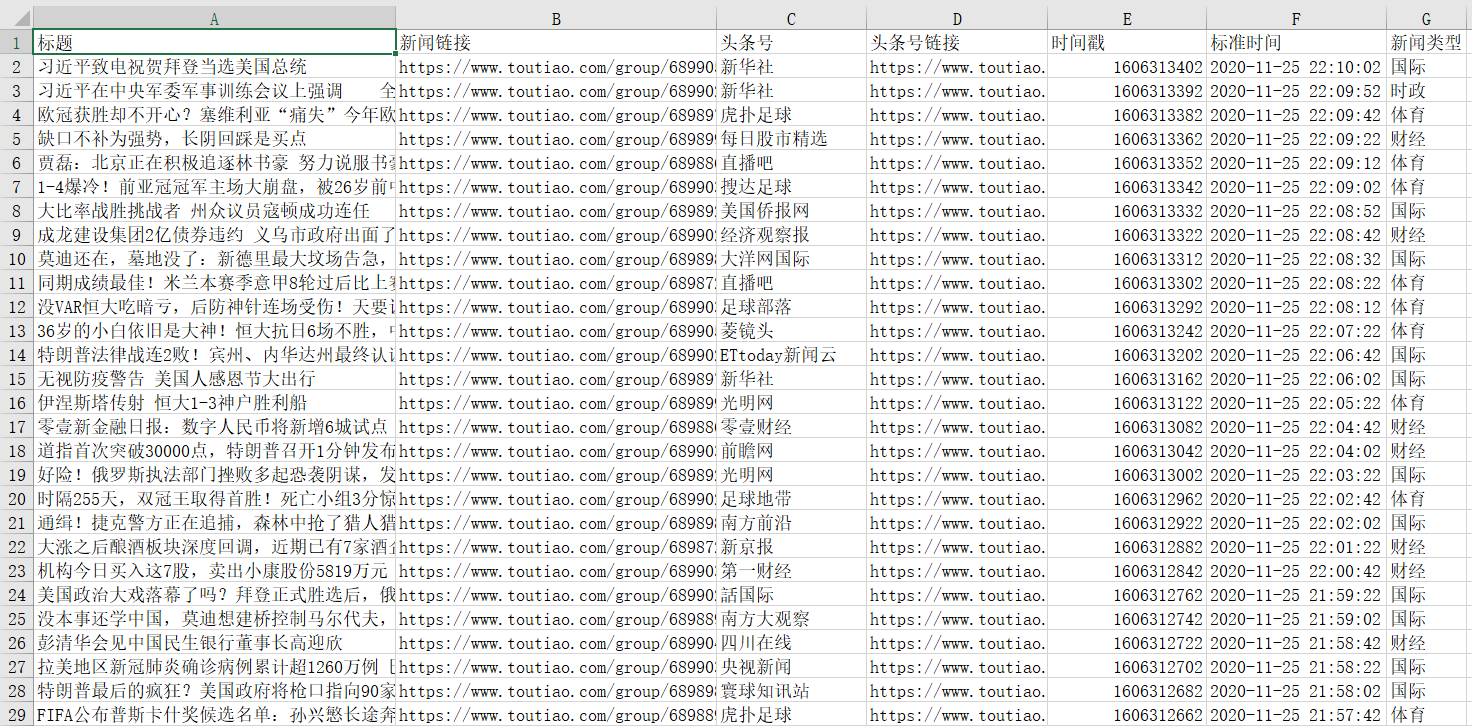
⑤time，即本条新闻发布时间

⑥tags，即本条新闻的标签

1. 存储方式

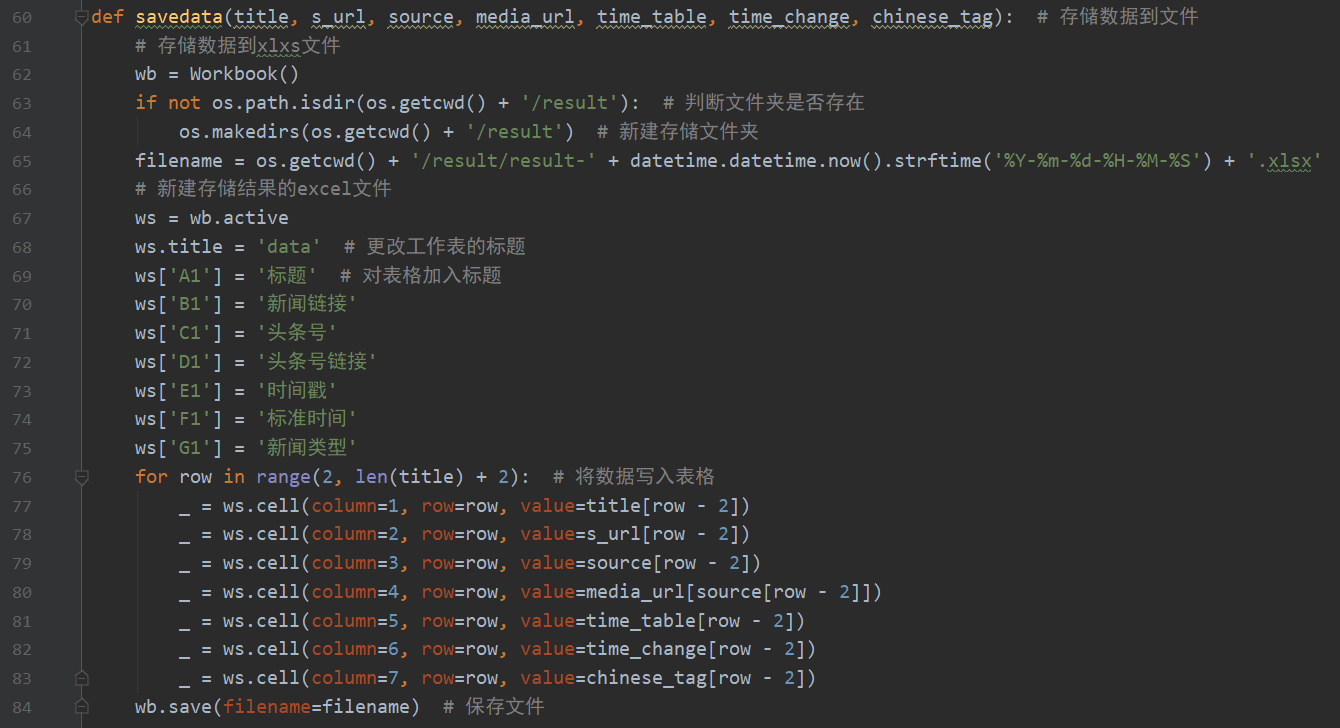
数据用以下三种方式进行了存储：

1. xlsx表格文件（输出到代码所在目录的/result子目录下）

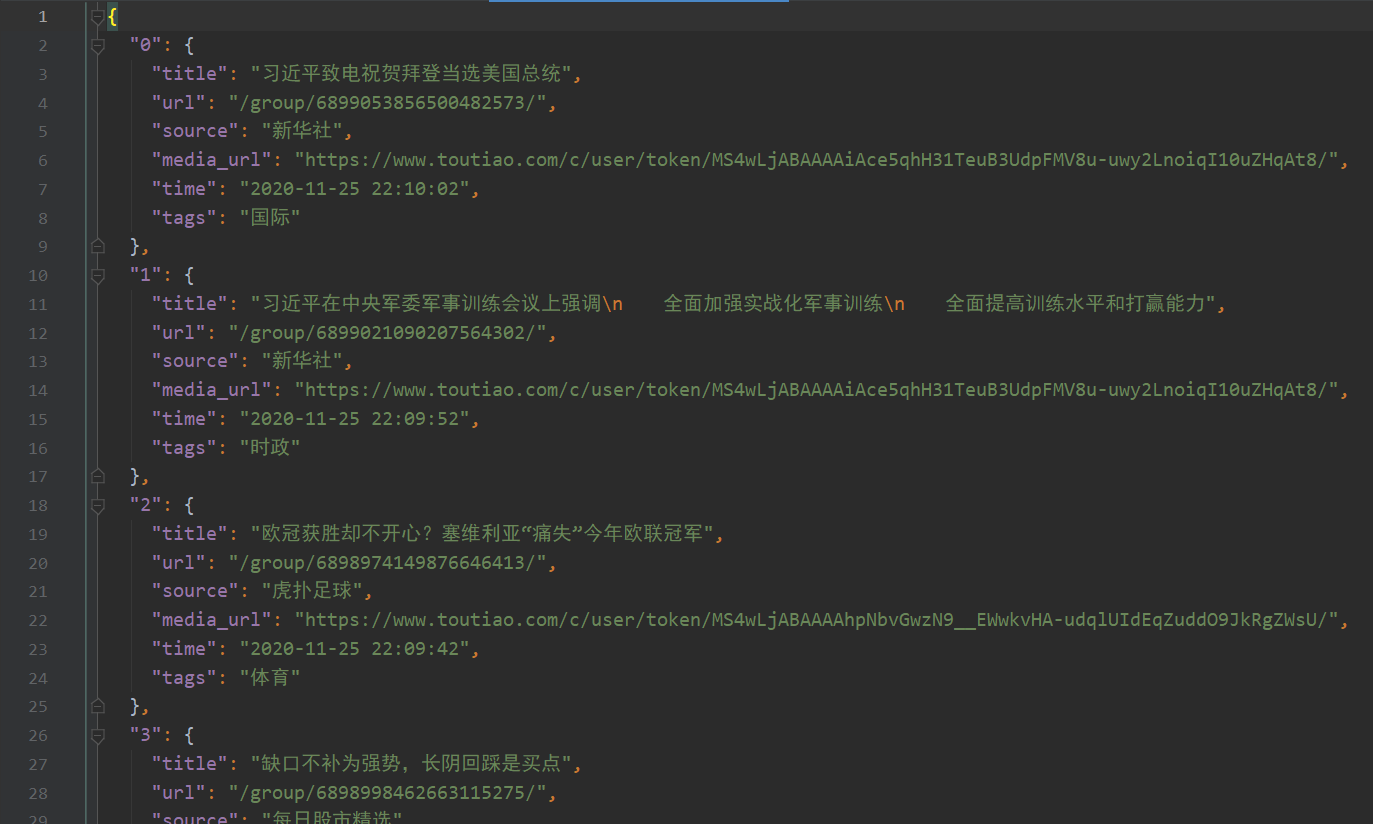


代码如下：

将存取的global数组传进savadata()函数，创建以时间为名字的xlsx文件，并按行存储



1. json文件（输出到代码所在目录的/json子目录下）

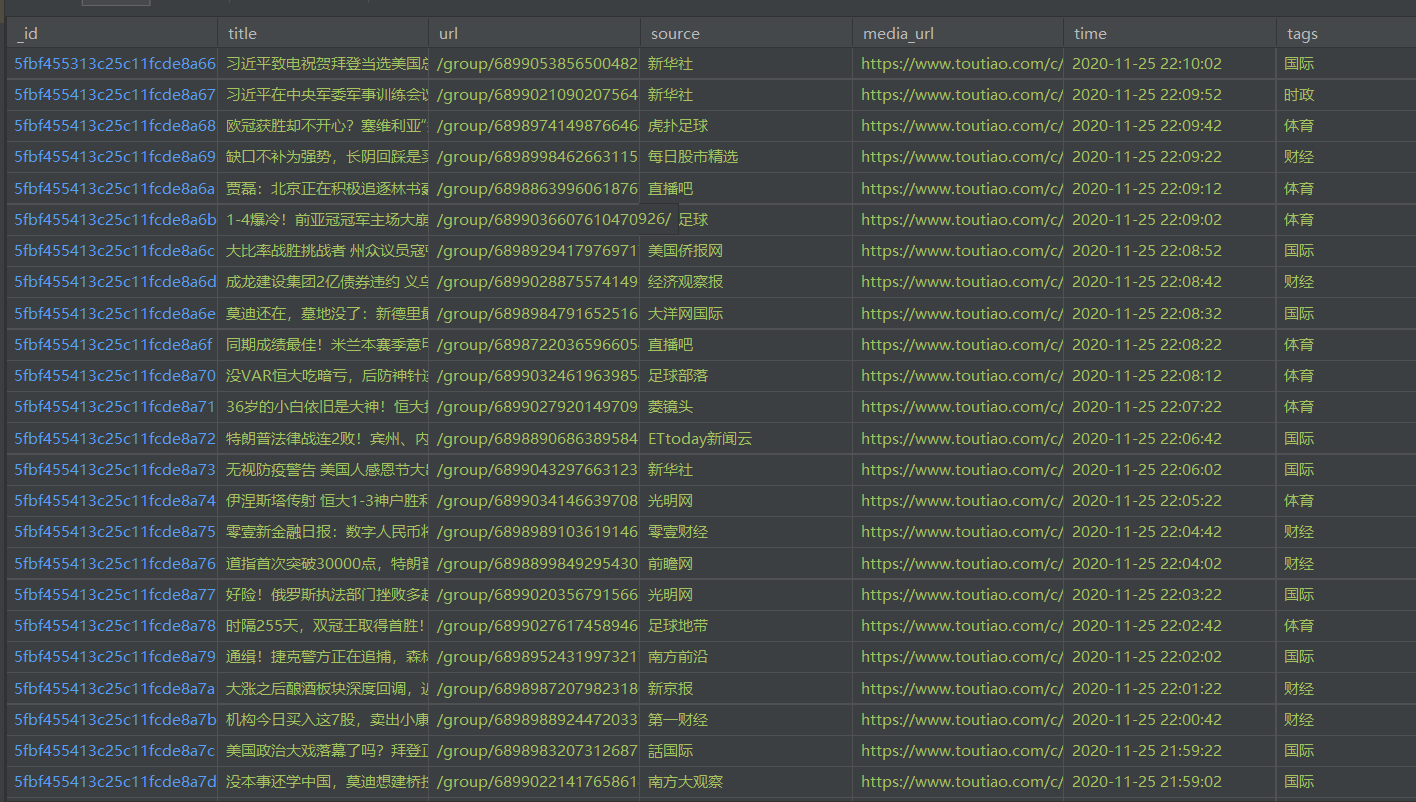


代码如下：

因为数据存取的需要，使用了convertJson()函数，同样创建了一个以时间为名字的json,按顺序存取为字典，方便读取



③写入MongoDB



代码如下：

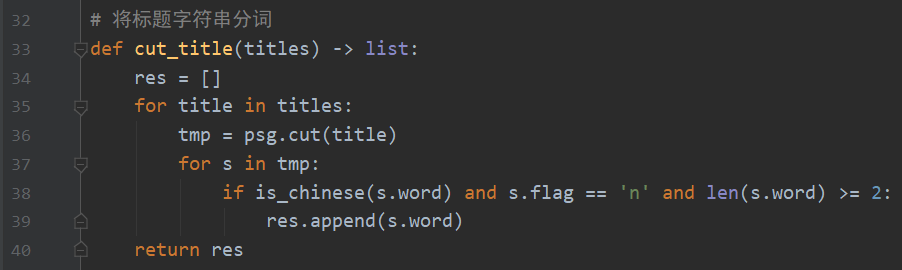


1. 预处理过程

先从MongoDB中读取出爬到的数据中的标题数据



然后调用Python中的jieba.posseg库，对每个标题进行中文分词，保留两个字及以上的中文名词



1. 流计算
2. 流准备和监听

Spark程序从SparkContext开始，SparkContext的初始化需要SparkConf对象，通过创建SparkConf对象来配置应用，基于SparkConf创建一个SparkContext对象，然后用SparkContext对象创建StreamingContext对象。

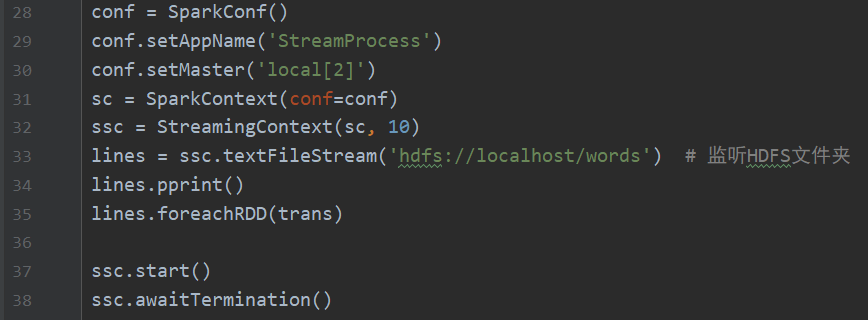
setAppName(“StreamProcess”)用来设置应用程序名称，即“StreamProcess”；setMaster(“local[2]”)括号里的参数“local[2]”表示运行在本地模式下，并且启动2个工作线程；

testFileStream(‘hdfs://localhost/words’)设置监听的文件夹，此处为HDFS上的/words文件夹；

streamingContext.start()来开始接收数据和处理流程；

streamingContext.awaitTermination()方法来等待处理结束

用PySpark每10s监听一次HDFS上的words文件夹，若words文件夹中有新增文件就会被检测到，并读取进行分析处理。



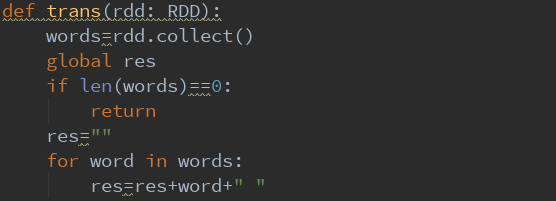
将jieba库分词得到的结果以txt文件形式上传至HDFS上被监听的words文件夹。PySpark监听并读取到words文件夹中的新增文本文件后，就会开始后续对其的流处理。



1. 流计算

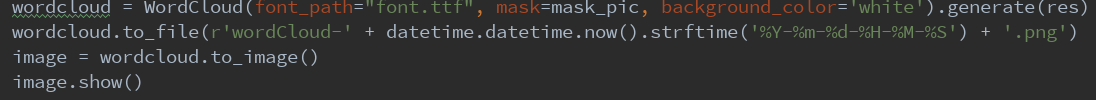
lines是一个DStream，对于其中的每一个RDD做自定义的trans()操作，trans()操作会将RDD转化为list，然后再将其中的每一个元素（字符串）添加进结果字符串res中。





1. 计算结果可视化

res为结果字符串，通过调用python中的wordcloud库统计出其中各个词出现的次数，然后直接将统计结果以词云的形式展示，并把生成的的词云图片保存到代码所在目录下。



生成的词云效果如下图：



1. 探究结果

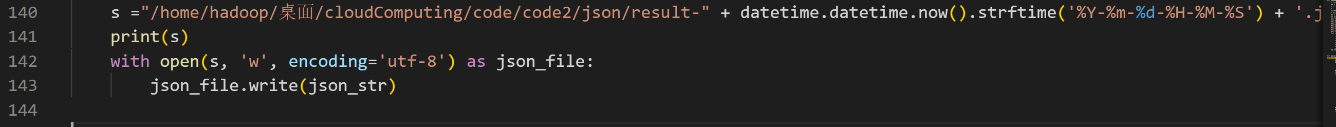
经过对今日头条新闻标题的分词及统计分析，我们发现“新冠肺炎”、“病例”依然占很大比例，说明最近一段时间，国内国外的疫情情况仍然是人们讨论的主要话题；除此而外，可以从生成的词云结果看出“体育”、“互联网”、“股市”也是今日头条新闻文章中关注较多的领域。

1. 困难及解决方案
2. 由于前期的数据处理过程以及最后的词云生成部分都需要调用Python的库，而起初计划是用scala或Java写Spark代码，那样会导致不便于调用Python的库，经过查资料，发现Spark也提供了Python开发使用的API，即PySpark，所以之后便转而使用PySpark完成了Spark部分的代码
3. 爬虫获取阶段，看到了获取json源数据格式的api，但是api是跟时间有关的，看到了相关加密的js，经过一番搜索才知道如何解析和获得正确的参数。后面的request获取，获取的数据存取需要基本学习一下。

**研究问题二：**

1. 数据

数据爬取使用dataAcquisition.py，爬虫函数格式同研究问题一的dataAcquisition.py。唯一改动点为将爬取下来的json文件保存到子项目的json目录下，代码如图



1. 流计算
2. 流准备和监听

Spark程序从SparkContext开始，SparkContext的初始化需要SparkConf对象，通过创建SparkConf对象来配置应用，基于SparkConf创建一个SparkContext对象，然后用SparkContext对象创建StreamingContext对象。

setAppName(“TestDStream”)用来设置应用程序名称，即“TestDStream”；

setMaster(“local[2]”)括号里的参数“local[2]”表示运行在本地模式下，并且启动2个工作线程；

checkpoint("file:///home/hadoop/桌面/cloudComputing/code/code2/checkpoint")表明将检查点设置到checkpoint目录下，为跨批次之间维护状态做准备

testFileStream(‘file:///home/hadoop/桌面/cloudComputing/code/code2/json’)设置监听的文件夹，此处为本地的/home/hadoop/桌面/cloudComputing/code/code2/json文件夹；

streamingContext.start()来开始接收数据和处理流程；

streamingContext.awaitTermination()方法来等待处理结束

用PySpark每20s监听一次本地的/home/hadoop/桌面/cloudComputing/code/code2/json文件夹，若该文件夹中有新增文件就会被检测到，并读取进行分析处理。

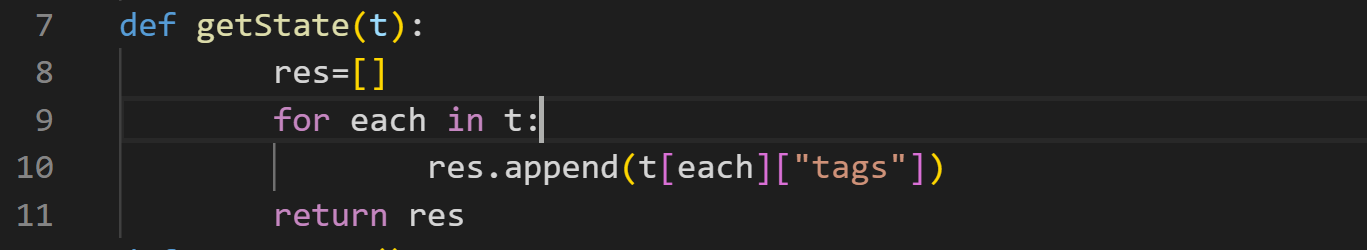


1. 流计算

首先使用json.loads()将字符串加载成json对象。

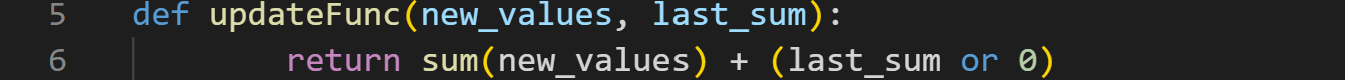


getState函数对json文件进行处理，提取出其中的新闻tags。



然后统计各tag的数量，其中使用updateStateByKey操作在跨批次之间维护状态。





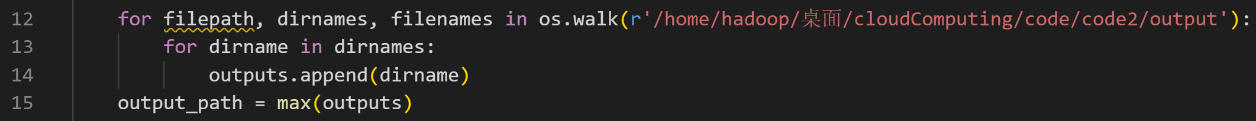
将所得结果存在output目录下。



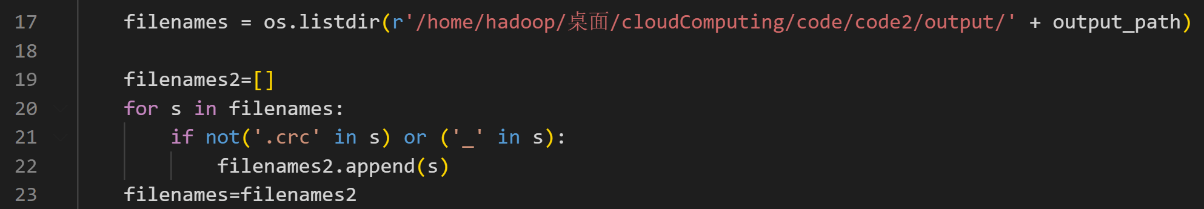
1. 计算结果可视化

之前的流计算结果已存于output目录下。

首先遍历output目录，找到最新生成的output.txt目录。



在这个最新的output.txt目录里获取所有part文件路径。



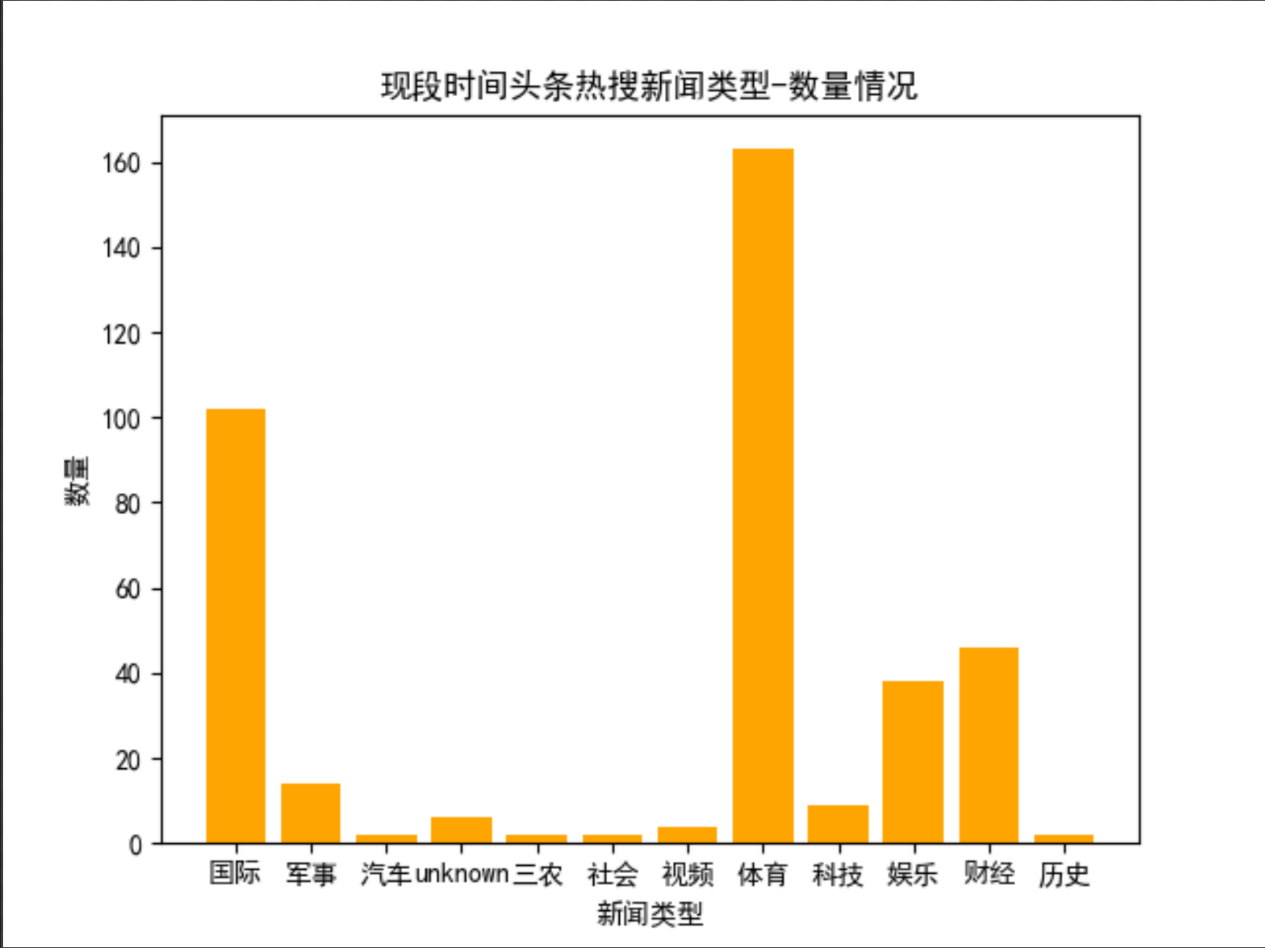
遍历所有part文件获取文件内容，分析得到新闻类型和对应的数量，存入两个数组中。



调用matplotlib.pyplot画直方图并保存到/home/hadoop/桌面/cloudComputing/code/code2/painting文件夹下。



生成的统计直方图效果如下图：



1. 探究结果

根据新闻类型-数量统计直方图可知，在头条热搜新闻中，类型为“体育”的新闻数量最多，说明体育新闻是今日头条中大家最关注的领域，可能和今日头条用户中体育迷占很大比重有关。数量第二多的是“国际”类型的新闻，说明国际新闻也是今日头条中大家十分关注的领域。类型为“体育”、“国际”的新闻占了最近一段时间头条新闻热搜的绝大多数，接下来是“财经”、“娱乐”、“军事”和“科技”，其余类型新闻则较少出现在头条热搜上。

1. 困难及解决方案
2. 在调试newsCpount.py时，一开始使用往被监听文件夹中移动现有json文件的形式来调试，发现没有监听到移动进去的文件，后经多次尝试以及查阅资料得知，被监听文件夹中的文件要想被监听到，其创建时间必须晚于streaming开启时间。
3. 爬虫函数存储结果时一开始采用os.getcwd()的方式得到当前工作路径，发现没有得到预期路径，之后决定使用绝对路径存储。
4. 进行结果可视化时发现中文字体无法正常显示，后经查阅资料，在matplotlib库里新加了SimHei字体，之后在paintNewsCount.py里加上plt.rcParams['font.sans-serif']=['SimHei'] ，最后在打开part文件时加上属性encoding="UTF-8"。
5. 之前为了一边爬虫一边运行streaming，采用多线程方式，发现无法正常运行。之后采取开两个窗口，先在一个窗口中运行streaming，再在第二个窗口中运行爬虫的方式，成功解决问题（我们是在VSCode里运行的，所以需要开两个窗口）。