

盐城师范学院

毕业论文

2018-2019 学年度

新闻热点爬取与可视化的研究与实现

学生姓名	蔡同波
学 院	数学与统计学院
专 业	信息与计算科学
班 级	数 15(5) 信计
学 号	15213526
指导教师	丁炫凯

2019 年 6 月 5 日

毕业论文承诺书

本人郑重承诺：

- 1、本论文是在指导教师的指导下，查阅相关文献，进行分析研究，独立撰写而成的.
- 2、本论文中，所有实验、数据和有关材料均是真实的.
- 3、本论文中除引文和致谢的内容外，不包含其他人或机构已经撰写发表过的研究成果.
- 4、本论文如有剽窃他人研究成果的情况，一切后果自负.

学生（签名）：_____

2019 年 6 月 5 日

新闻热点爬取与可视化的研究与实现

摘 要

随着互联网和智能硬件的普及，各种信息数量正不断增加，每天从不同的话题之中都会产生大量的新闻数据，如何从众多的新闻数据中快速地了解信息的要点，有效提高获取重要信息的效率和减少在阅读时间上的消耗正变得越来越重要。通过对新闻数据进行分析，可以得出人们对某个领域的关切程度和社会需要解决的问题，有利于了解当前的舆论焦点，有助于政府了解民意，便于国家对舆论进行正确引导。本文以中国新闻网网站为例，通过爬虫技术抓取社会新闻版块数据，再对采集到的新闻内容进行处理及词频词性分析来计算关键字，最后借助数据可视化技术提供直观的呈现，得出热点所涉及的相关人或事物，以此分析出社会领域关注问题和需要解决的问题情况。

【关键词】网络爬虫；新闻；热点；Python；可视化

Research and implementation of news hotspot crawling and visualization

Abstract

With the popularity of the Internet and intelligent hardware, the amount of information is increasing. Every day, a lot of news data will be generated from different topics. How to quickly understand the main points of information from numerous news data, effectively improve the efficiency of obtaining important information and reduce the consumption of reading time is becoming more and more important. Through the analysis of news data, we can get the degree of people's concern in a certain field and the problems that society needs to solve, which is conducive to understanding the current focus of public opinion, helping the government to understand public opinion, and facilitating the correct guidance of public opinion by the state. Taking CNN website as an example, this paper grabs the data of social news section by crawler technology, then processes the collected news content and calculates the keywords by word frequency and part of speech analysis. Finally, it provides an intuitive presentation with data visualization technology, and obtains the relevant people or things involved in hot spots, so as to analyze the problems concerned and need to be solved in the social field.

[Key words] web crawler; news; hotspot; Python; visualization

目 录

引 言.....	1
1 研究现状、总体任务和结构安排	1
1.1 新闻热点可视化研究现状	1
1.2 本文的主要工作.....	2
1.3 本文的组织结构.....	2
2 基于新闻 API 以及 BeautifulSoup 模块的数据抓取	3
2.1 基于新闻 API 的新闻链接抓取	3
2.2 基于 BeautifulSoup 模块的新闻内容的抓取	3
2.3 实验结果	5
2.4 小结.....	6
3 基于交叉信息熵算法的关键字提取.....	7
3.1 交叉信息熵算法.....	7
3.2 分词和词频词性统计.....	7
3.3 实验结果	8
3.4 小结.....	10
4 实验结果及数据可视化.....	11
4.1 实验结果	11
4.2 数据可视化	12
5 总结.....	19
参考文献.....	20
附 录.....	21
附录一：中国新闻网爬虫实现类	21
附录二：新闻文本数据分割类	24
附录三：新闻数据可视化实现类	27

引言

随着 5G、AI 智能时代的到来，数据以爆炸的方式增长。各大新闻媒体产生的新闻数量正变得越来越多，一方面是用户数量不断增加，另一方面是因为各种物联网设备的出现给人们生活方式带来了很大改变，产生了许多与社会生活相关数据信息。2019 年 2 月末，中国互联网络信息中心（CNNIC）发布了第 43 期《中国互联网发展统计报告》，报告显示截至到去年 12 月我国互联网普及率已经达到了 59.6%，用户数量达到 8.29 亿，中国的互联网发展正迎来前所未有的发展变革，5G 网络技术的普及将更彻底地改变人们的生活和生产方式，产业互联将触及经济和社会的各个方面。与传统的报纸、电视和广播相比，网络新闻在传播、互动和渗透力方面拥有更多的优势，使人们获取新闻信息更及时和便捷。

但是，随着信息量的不断增加，在同一新闻主题下拥有各种各样的新闻报道，大量的新闻出现在人们的生活中。由于在互联网上发布新闻对个人用户没有太多的限制，一些不法分子常常通过恶意“刷榜”、编造谎言恶意引导舆论方向，这可能会给社会带来不少的不良影响和危机。因此，很有必要检测分析当前新闻中的热门信息情况，为热点聚焦和防控提供一定的参考。本文通过可视化技术可以直观的计算统计出当前新闻的热点情况，为新闻决策和异常监控提供依据，以减小负面新闻对人们生活的影响，及时掌握事物的发展情况。

1 研究现状、总体任务和结构安排

1.1 新闻热点可视化研究现状

在大数据时代，面对复杂繁多的新闻信息，人们很希望能够快速从新闻报道中提取出关键的信息。数据可视化分析通过大数据的技术手段，针对数据的分布区域和在各行业的应用进行分析。长久以来，新闻数据可视化分析一直存在不少的问题，不仅需要使用先进的技术，还需要能够获得海量的数据，媒体组织应该共享彼此之间的信息，有效降低获取数据的成本，提高新闻数据的价值。另外，目前的新闻媒体发展仍受到关注度不高、新闻质量参差不齐的限制，应该努力提供创造性和个性化的新闻数据体验，采用多种模式来满足不同的用户，这也是未来数据可视化的发展目标和方向。

1.2 本文的主要工作

本文主要通过对新闻文本数据采集、分词和词频词性统计得到新闻热点词，并对热点数据提供可视化信息和分析，以帮助公众了解重要事件关注情况。

本文主要研究的内容为：

- 1、利用多线程技术和爬虫算法实现了对中国新闻网的新闻的并行爬取。
- 2、利用中科院的 ICTCLAS 分词、交叉信息熵算法分别对抓取的 2019 年 3 月 1 日到 31 日期间的社会新闻进行数据处理、词频词性统计以及 JSON 化处理。
- 3、利用 PyEcharts 技术对数据结果进行分类统计并绘制可视化视图。

1.3 本文的组织结构

本文的结构共分有五个部分，具体安排如下：

第一部分，引言。介绍了本课题的研究背景和意义，研究现状、总体任务和结构安排。

第二部分，基于新闻 API 以及 BeautifulSoup 模块的数据抓取。介绍了网络爬虫的概念、采取的爬虫算法和抓取新闻的方式。

第三部分，基于交叉信息熵算法的关键字提取。详细介绍介绍分词前的各种处理、分词算法以及词频词性统计过程，并对抓取的新闻数据进行 JSON 化处理。

第四部分，实验结果及数据可视化。设计并实现数据的可视化呈现数据背后的深层含义，得出研究的最终结果。

第五部分，总结。通过对数据的可视化结果分析、得出研究的结论。

2 基于新闻 API 以及 BeautifulSoup 模块的数据抓取

新闻 API 作为新闻数据的入口，是后台向前端传输数据的通道。而对应此规则的数据信息则是爬虫按照某一特定算法抓取的目标新闻内容，新闻爬虫依次通过发送 API 请求、获取和解析新闻页面数据、最后将格式内容存储进数据库中，直到页面数据遍历采集完毕才停止执行。

2.1 基于新闻 API 的新闻链接抓取

从中国新闻网网站上，获取的新闻 API 如图 2-1 所示：

```
daylist = get_day_list() #获取时间段
# 所有时间段对应的页面
urls = ['http://www.chinanews.com/scroll-news/sh/2019/{}/news.shtml'.format(day) for day in daylist]
# 从所有时间段页面找到每个新闻的页面
```

图 2-1 中国新闻网社会新闻 API

使用 python 的 requests 提供的 get() 方法我们可以非常简单的获取的指定网页的内容，使用 BeautifulSoup 的 select 方法得到该页面下所有标签中的所有 URL，并将 URL 存入到 url_list 的表中。如图 2-2 所示：

```
def get_single_links(url):
    """
    通过某一天的主页面URL得到该页面下的所有URL，并将URL存入到url_list的表中去
    :param url: 某天的主页面
    :return:
    """
    try:
        # 基本抓取流程
        wb_data = requests.get(url)
        #print(wb_data.status_code)
        wb_data.encoding = wb_data.apparent_encoding
        soup = BeautifulSoup(wb_data.text, 'lxml')
        links = soup.select('li > .dd_bt > a') #获取链接所在的标签获得链接
        for link in links:
            #print(link.get('href'))
            url_lists.insert_one({'links': link.get('href')}) #将链接插入到数据库中去
    except:
        print("error")
```

图 2-2 新闻 URL 抓取

2.2 基于 BeautifulSoup 模块的新闻内容的抓取

本文的爬虫系统主要由三个部分构成，分别是抓取，分析和存储。程序首先通过访问 API 接口请求解析新闻数据，然后根据定义的格式信息分析存储，新闻的整体数据抓取过程如图 2-3 所示：

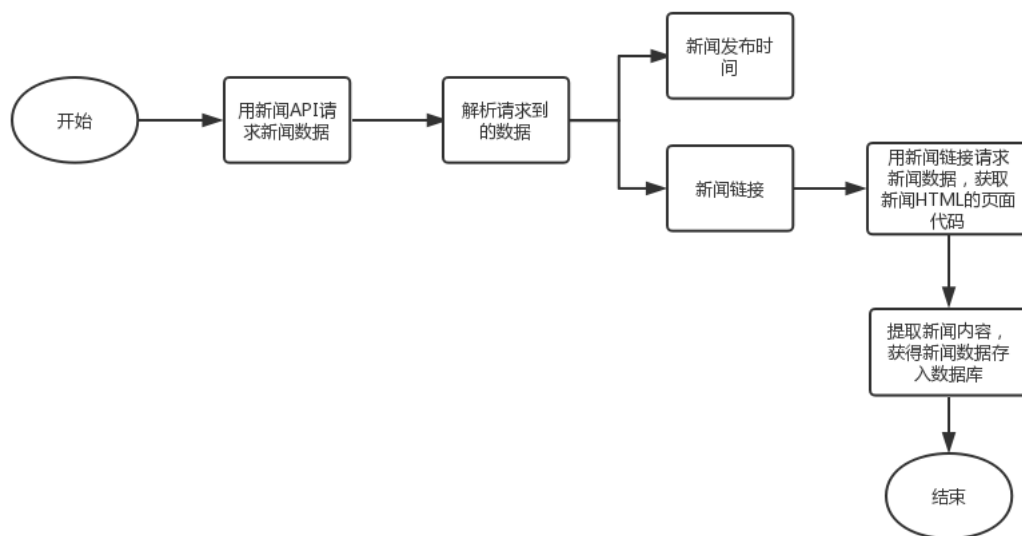


图 2-3 新闻整体数据抓取流程图

抓取到网页的内容后，要做的就是提取出想要的内容。首先导入 BeautifulSoup 库，使用 BeautifulSoup 可以非常简单的提取网页的特定内容。如图 2-4 所示：

```

def get_res(url):
    ...
    从单个新闻的链接中得到这个链接中的新闻内容
    :param url: 单个新闻的链接
    :return:
    ...
    try:
        contentTxt = ""
        wb_data = requests.get(url)
        wb_data.encoding = wb_data.apparent_encoding
        soup = BeautifulSoup(wb_data.text, 'lxml')
        contents = soup.select('.left_zw > p')
        for content in contents:
            contentTxt += content.get_text().strip()
        data = {
            "link": url,
            "content": contentTxt,
            "year": url.split('/')[2],
            "month": url.split('/')[2].split('-')[0],
            "day": url.split('/')[2].split('-')[1]
        }
        item_info.insert_one(data)
        print(data)
    except:
        print("error")
  
```

图 2-4 新闻内容抓取

数据库部分的存储集合包括两个部分，分别是新闻链接 url_lists、新闻内容 item_info 集合，两个集合的详细设计如下表：

表 2-1 新闻内容 item_info 集合

属性名	含义	类型	说明
_id	唯一标识	String	
Link	新闻链接	String	URL
Content	新闻内容	String	
Year	年份	String	
Month	月份	String	
Day	日期	String	

表 2-2 新闻链接 url_lists 集合

属性名	含义	类型	说明
_id	唯一标识	String	
Links	新闻链接	String	URL

2.3 实验结果

本文样本数据抓取的时间是 2019 年 3 月份的所有新闻数据。由图 2-5 可见，url_lists 中共有 433.6KB 的链接数据，item_info 中共有 17.6MB 的新闻内容数据。

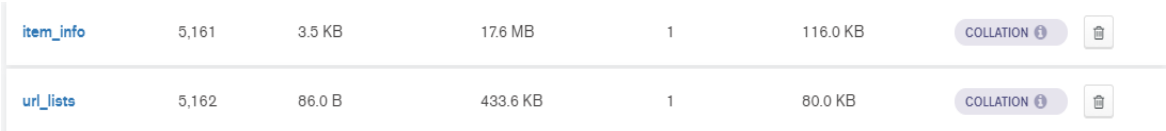


图 2-5 新闻数据大小

按照时间顺序抓取的新闻链接集合如图 2-6 所示：

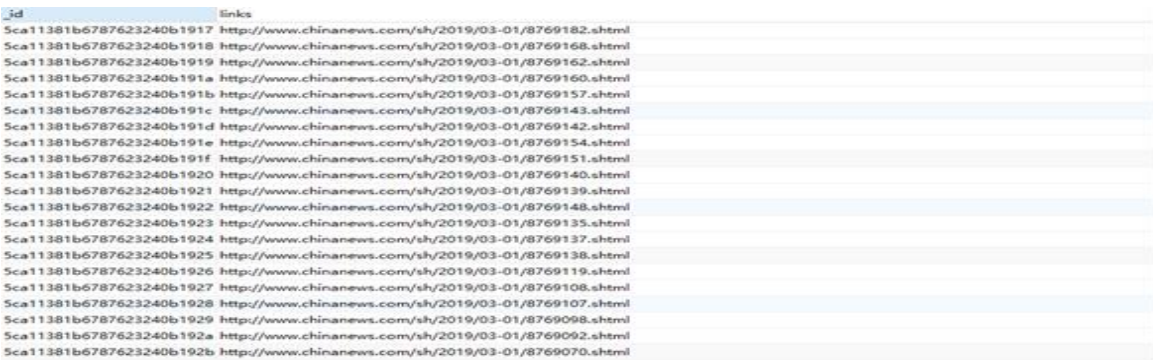


图 2-6 新闻链接 url_lists 集合

获得新闻链接后再根据链接进行深层次的内容解析获得新闻内容，如图 2-7 所示：

_id	link	content	year	month	day
5ca113d2b678760dc49c8	http://www.chinanews.com/sh/2019/03-01/8769182.shtml	3月1日, 福州市人民检察院对晋安区人民检察院	2019	03	01
5ca113d2b6787626a8348	http://www.chinanews.com/sh/2019/03-06/8773269.shtml	中新网湖州3月6日电(见习记者 施紫楠 通讯员	2019	03	06
5ca113d2b678760a1c031	http://www.chinanews.com/sh/2019/03-03/8770027.shtml	时值三月, 春暖花开, 姹紫嫣红, 央视新闻推	2019	03	03
5ca113d2b678761730a78	http://www.chinanews.com/sh/2019/03-08/8774700.shtml	半月谈微评 根治欠薪, 让农民工不再“忧薪”	2019	03	08
5ca113d4b678760dc49c8	http://www.chinanews.com/sh/2019/03-01/8769168.shtml	中新社广州3月1日电 (蔡敏建)记者3月1日从广	2019	03	01
5ca113d5b6787626a8348	http://www.chinanews.com/sh/2019/03-06/8773264.shtml	中新网福州3月6日电 (刘雅萍)福建省连血干细	2019	03	06
5ca113d5b678761730a78	http://www.chinanews.com/sh/2019/03-08/8774694.shtml	中新网3月8日电(记者 张尼) 三八节当天, 北	2019	03	08
5ca113d6b678760a1c031	http://www.chinanews.com/sh/2019/03-03/8770003.shtml	榜样是人格化的价值观, 看得见的正能量, 具	2019	03	03
5ca113d7b6787626a8348	http://www.chinanews.com/sh/2019/03-06/8773262.shtml	今天上午, 在十三届全国人大二次会议首场“代	2019	03	06
5ca113d7b678760dc49c8	http://www.chinanews.com/sh/2019/03-01/8769162.shtml	新华社北京3月1日电(记者陈菲、丁小溪)3月1	2019	03	01
5ca113d7b678761730a78	http://www.chinanews.com/sh/2019/03-08/8774692.shtml	【新时代 她时代】优秀而优雅的女科学家—记	2019	03	08
5ca113d8b678760a1c031	http://www.chinanews.com/sh/2019/03-03/8770001.shtml	新华社北京3月3日电 题: 两会连着你我的“小	2019	03	03
5ca113d9b6787626a8348	http://www.chinanews.com/sh/2019/03-06/8773238.shtml	中新社广州3月6日电 (王华 麦金萍)世界罕见	2019	03	06
5ca113d9b678760dc49c8	http://www.chinanews.com/sh/2019/03-01/8769160.shtml	3月1日, 检察机关对福建省福州市晋安区人民	2019	03	01
5ca113d9b678761730a78	http://www.chinanews.com/sh/2019/03-08/8774685.shtml	“毕生节俭, 只为一次奢侈”。在“感动中国201	2019	03	08
5ca113dab678760a1c031	http://www.chinanews.com/sh/2019/03-03/8769957.shtml	中新网3月3日电 据河北省公安厅官方微博消息	2019	03	03
5ca113dbb6787626a8348	http://www.chinanews.com/sh/2019/03-06/8773233.shtml	中新网郑州3月6日电(记者 韩章云)寻常见到的	2019	03	06
5ca113dbb678760dc49c8	http://www.chinanews.com/sh/2019/03-01/8769157.shtml	记者从最高检获悉, 3月1日, 检察机关对福建	2019	03	01
5ca113ddb678761730a78	http://www.chinanews.com/sh/2019/03-08/8774657.shtml	中新网昆明3月8日电 (记者 刘再阳)中国地震	2019	03	08
5ca113dcb678760a1c031	http://www.chinanews.com/sh/2019/03-03/8769960.shtml	中新网3月3日电 针对有网友在微信群称“有乞	2019	03	03
5ca113ddb678760dc49c8	http://www.chinanews.com/sh/2019/03-01/8769143.shtml	中新网邢台3月1日电 (张墨翔 李铁捷)河北省	2019	03	01
5ca113ddb6787626a8348	http://www.chinanews.com/sh/2019/03-06/8773232.shtml	中新网广州3月6日电 (记者 郭军)“鸣一”3月6	2019	03	06

图 2-7 新闻内容 item_info 集合

由图 2-6、2-7 可见, url_lists 文件中含有 2 列数据, 分别是 id (标识符)、links (链接), item_info 中含有 6 列数据, 分别是 id (标识符)、link (链接)、content (新闻内容)、year (年)、month (月)、day (日), 新闻内容所在列的数据较多。

2.4 小结

本章利用爬虫算法多线程抓取中国新闻网网站的社会新闻。主要实现页面链接和新闻的爬取。通过某一天的主页面 URL 得到该页面下的所有 URL, 并将 URL 存入到 url_list 的集合中去, 使用 BeautifulSoup 模块从单个新闻的链接中得到这个链接中的新闻内容存入到 item_info 集合中。

3 基于交叉信息熵算法的关键字提取

在上面第三章中，已经通过爬虫程序获取到了 2019 年整个 3 月份大约 5161 条新闻数据，通过新闻文本数据的分词和词频词性统计分析，便能很容易获得全部数据中的热点情况。

3.1 交叉信息熵算法

在信息理论中，熵是不确定事件的度量，是信息中拥有的平均信息量。一个词叫做关键词，原因在于这个词能搭配很多很丰富的其他词语表达更重要的含义，于是便产生了一个词的信息熵定义，数学表达如下：

$$H(w) = \sum p \log(p)$$

W 代表单词，p 代表单词周围出现的不同单词的数量。例如，在一篇文章中，现在有两个 XWY，一个 ZWS，那么 W 的左信息熵是：

$$-\frac{2}{3} \log\left(\frac{2}{3}\right) - \frac{1}{3} \log\left(\frac{1}{3}\right)$$

其中， $\frac{2}{3}$ 表示关键句 X 在 3 个句子中出现两次，而 Y 只出现一次，因此 Y 的信息熵为 $\frac{2}{3}$ ，对于一个 ZWS 来说，他们两个的信息熵则是相同的。如果是以 XWS，YWS 例，那么 W 的右边信息熵的值是 0，因为是 $-\log(1)$ 。

对文档中所有的词语计算上下文信息熵的情况，如果一个词的上下文信息熵都比较大，那么这个词就很容易被判断为关键词。

3.2 分词和词频词性统计

新闻数据在实现分词和词频词性统计时，首先需要读取数据并写入文本文件中，如图 3-1 所示：

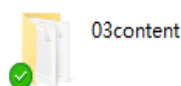
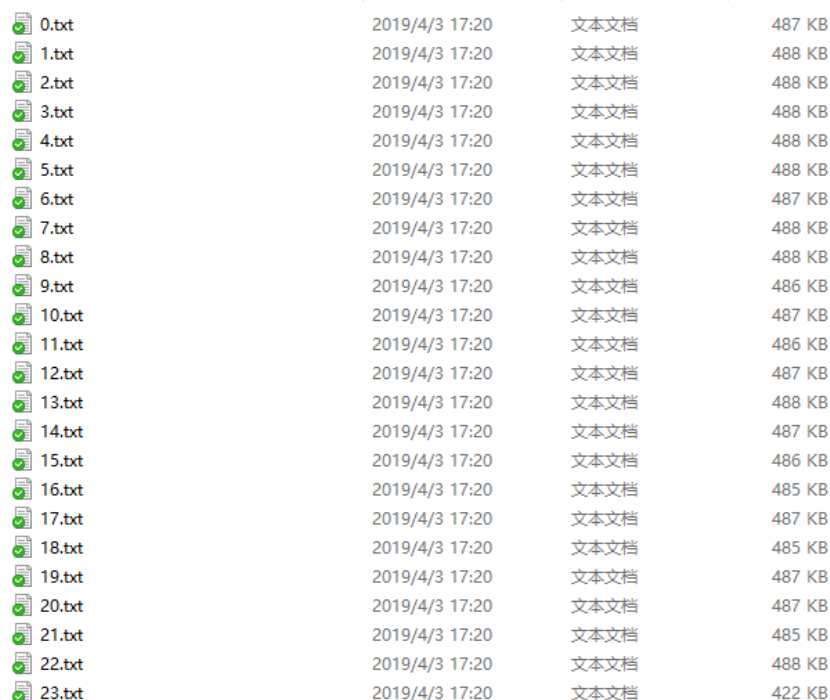


图 3-1 新闻文本数据

由于中科院计算所的 NLPIR 分词系统只能分割 500K 以内的数据(实际只有 490K), 所以不得不对大容量的 txt 文件进行分割成 500K 以下的文件。输出文件 500K 大小的文件, 在 GBK 编码中一个汉字占 2 个字节, 故定义文本大小 $lensize=256000$, 表示读取 256000 个字符, 这些字符在最后存在 GBK 文件里的时候文件大小为: $256000*2/1024 = 500K$, 实际情况下可能里面有一些英文字符所以实际大小为 490k 左右, 切分后的文本文件如下图 3-2 所示:



0.txt	2019/4/3 17:20	文本文档	487 KB
1.txt	2019/4/3 17:20	文本文档	488 KB
2.txt	2019/4/3 17:20	文本文档	488 KB
3.txt	2019/4/3 17:20	文本文档	488 KB
4.txt	2019/4/3 17:20	文本文档	488 KB
5.txt	2019/4/3 17:20	文本文档	488 KB
6.txt	2019/4/3 17:20	文本文档	487 KB
7.txt	2019/4/3 17:20	文本文档	488 KB
8.txt	2019/4/3 17:20	文本文档	488 KB
9.txt	2019/4/3 17:20	文本文档	486 KB
10.txt	2019/4/3 17:20	文本文档	487 KB
11.txt	2019/4/3 17:20	文本文档	486 KB
12.txt	2019/4/3 17:20	文本文档	487 KB
13.txt	2019/4/3 17:20	文本文档	488 KB
14.txt	2019/4/3 17:20	文本文档	487 KB
15.txt	2019/4/3 17:20	文本文档	486 KB
16.txt	2019/4/3 17:20	文本文档	485 KB
17.txt	2019/4/3 17:20	文本文档	487 KB
18.txt	2019/4/3 17:20	文本文档	485 KB
19.txt	2019/4/3 17:20	文本文档	487 KB
20.txt	2019/4/3 17:20	文本文档	487 KB
21.txt	2019/4/3 17:20	文本文档	485 KB
22.txt	2019/4/3 17:20	文本文档	488 KB
23.txt	2019/4/3 17:20	文本文档	422 KB

图 3-2 文本分割

另外, 由于 NLPIR 分词系统只支持 GBK 的编码, 还需要将 UTF-8 格式文本转换成 GBK 格式。相关代码如图 3-3 所示:

```
with codecs.open(output_file_name, "w", 'gbk') as file:
    s = str(url_list)
    res = s.encode('utf-8', 'ignore').decode('utf-8', 'ignore').encode('gbk', 'ignore').decode('gbk', 'ignore')
    file.write(res)
```

图 3-3 文本编码转换

3.3 实验结果

使用中国科学院的 ICTCLAS 分词系统对新闻文本数据分词计算操作的结果如图 3-4 所示:

```

工作/vn/128.72/419
进行/vx/122.80/194
中国/ns/109.76/262
2018年/t/107.76/146
服务/v/96.33/233
发展/vn/96.27/233
王磊/nr/92.16/149
孩子/n/82.47/224
一个/mq/81.58/239
2019年/t/81.32/96
通过/v/78.19/160
企业/n/76.64/135
女性/n/67.94/154
可以/v/67.32/142
开展/v/66.88/106
北京/ns/64.68/233
发现/v/64.57/143
生活/vn/64.45/153
全国/n/63.05/160
时间/n/62.79/146
社会/n/62.39/159
提供/v/61.72/103

```

图 3-4 分词原始数据

得到原始数据后我们并不能立即进行可视化的显示，还需要进一步借助 Excel 进行处理，这里主要是将分割后的关键字内容进行合并，如图 3-5 所示：

	A	B	C	D	E	F	G	H	I	J	K
1	words	cates	weights	nums							
2	工作	vn	8477	2486.81							
3	进行	vx	5504	3199.81							
4	中国	ns	5647	2112.92							
5	2018年	t	3227	2628.25							
6	服务	v	4436	1773.33							
7	发展	vn	5073	1991.03							
8	王磊	nr	360	323.06							
9	孩子	n	4471	1672.9							
10	一个	mq	5719	1806.06							
11	2019年	t	1694	1863.95							
12	通过	v	626	284.93							
13	企业	n	3868	1833.61							
14	女性	n	711	445.91							
15	可以	v	3458	1617.55							
16	开展	v	2598	1541.4							
17	北京	ns	2909	854.58							
18	发现	v	3553	1482.15							
19	生活	vn	2903	1240.73							
20	全国	n	2809	1156.07							
21	时间	n	3423	1420.31							
22	社会	n	3628	1545.9							
23	提供	v	2312	1299.73							
24	建设	v	3305	1498.7							

图 3-5 Excel 分解后的数据

但是经 Excel 处理后的新闻数据依然不能直接使用，需要转成 JSON 文本文件才行，通过 BeJSON 在线转换，可以将 Excel 表格数据转至 JSON 文本文件，结果如图 3-6 所示：

```

{"words":"工作","cates":"vn","weights":"8477","nums":"2486.81"},
{"words":"进行","cates":"vx","weights":"5504","nums":"3199.81"},
{"words":"2018年","cates":"t","weights":"3227","nums":"2628.25"},
{"words":"服务","cates":"v","weights":"4436","nums":"1773.33"},
{"words":"发展","cates":"vn","weights":"5073","nums":"1991.03"},
{"words":"王磊","cates":"nr","weights":"360","nums":"323.06"},
{"words":"孩子","cates":"n","weights":"4471","nums":"1672.9"},
{"words":"一个","cates":"mq","weights":"5719","nums":"1806.06"},
{"words":"2019年","cates":"t","weights":"1694","nums":"1863.95"},
{"words":"通过","cates":"v","weights":"626","nums":"284.93"},
{"words":"企业","cates":"n","weights":"3868","nums":"1833.61"},
{"words":"女性","cates":"n","weights":"711","nums":"445.91"},
{"words":"可以","cates":"v","weights":"3458","nums":"1617.55"},
{"words":"开展","cates":"v","weights":"2598","nums":"1541.4"},
{"words":"北京","cates":"ns","weights":"2909","nums":"854.58"},
{"words":"发现","cates":"v","weights":"3553","nums":"1482.15"},
{"words":"生活","cates":"vn","weights":"2903","nums":"1240.73"},
{"words":"全国","cates":"n","weights":"2809","nums":"1156.07"},
{"words":"时间","cates":"n","weights":"3423","nums":"1420.31"},
{"words":"社会","cates":"n","weights":"3628","nums":"1545.9"},

```

图 3-6 JSON 化的分词数据

3.4 小结

新闻关键字提取是一个比较大的过程，需要对新闻文本数据进行文本分割，本章实现了对新闻内容的抽取，在分词的时候使用了中国科学院的 ICTCLAS 分词系统，并做了新闻内容的词频和词性结果统计，最后将分词文本转成 JSON 格式的数据文件，为下一步的可视化实现提供数据支撑。

4 实验结果及数据可视化

本文主要是利用文本词频统计技术进行新闻热点问题可视化分析，希望通过可视化技术准确直观的显示当前媒体报道中的各类热点情况，以此为个人、社会和国家提供新闻热点的关注情况，为社会舆论更好发展提供一定的指导。

4.1 实验结果

结果显示 2019 年整个 3 月份，产生了大约 17 M 的新闻数据（如图 4-1 所示），共抓取了约 5161 条新闻信息，总计大约 608 万字（如图 4-2 所示），平均每天产生 166 多条新闻信息。



图 4-1 新闻文本大小



图 4-2 新闻数据总量

然后，通过中国科学院的 ICTCLAS 分词系统对这 608 万字的新闻数据文档处理，得到了 28 万 3 千多的关键词数据，如图 4-3 所示：

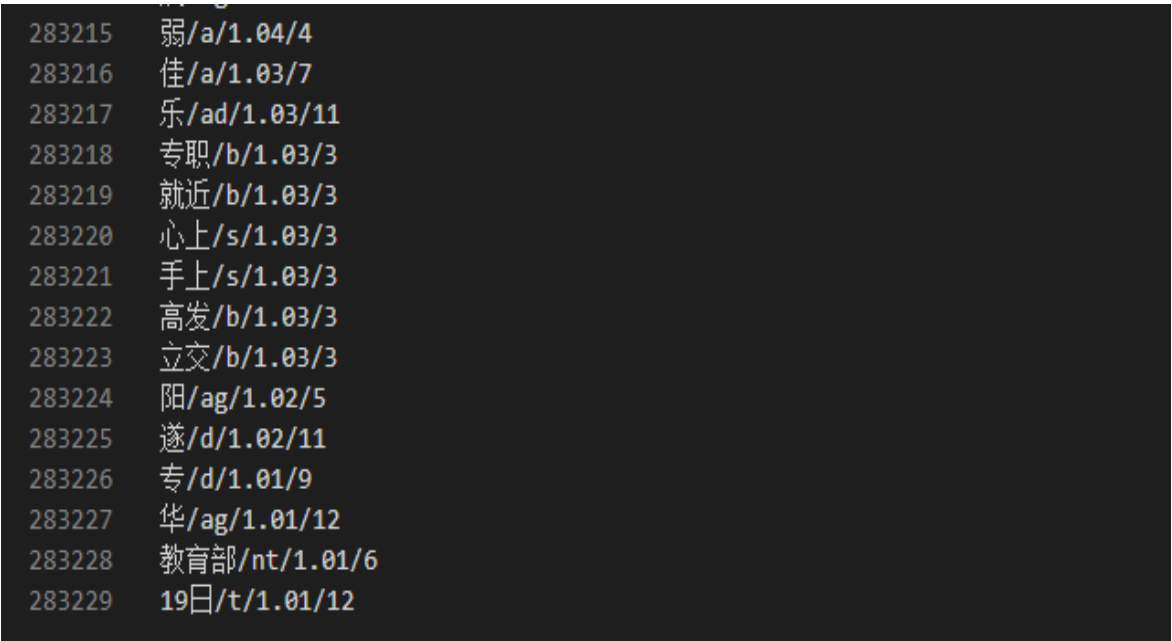


图 4-3 分词数据总量

4.2 数据可视化

为了可视化的显示提取后的数据，需要借助 PyEcharts 技术动态的将 JSON 化后的新闻数据显示在网页上，如图 4-4 所示，这是 3 月份在社会新闻版块重要的热点名词统计情况。

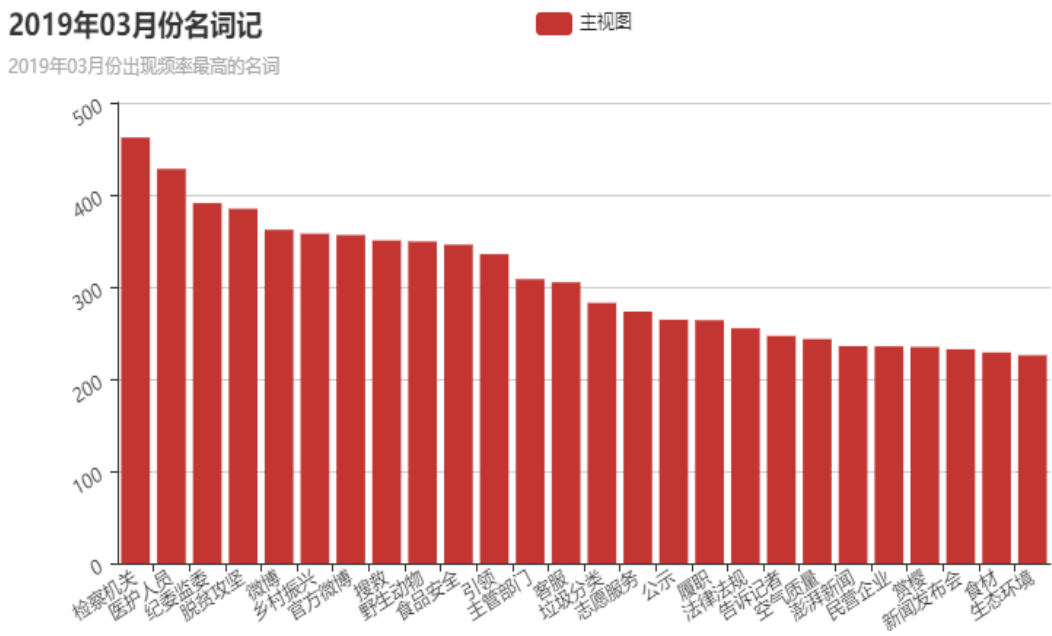


图 4-4 三月份热点名词

对应的热点名词数据视图表如下：

表 4-1 三月份热点名词

{ ' words' : ' 检察机关' , ' nums' : 462.19 }
{ ' words' : ' 医护人员' , ' nums' : 428.08 }
{ ' words' : ' 纪委监委' , ' nums' : 391.02 }
{ ' words' : ' 脱贫攻坚' , ' nums' : 384.82 }
{ ' words' : ' 微博' , ' nums' : 362.07 }
{ ' words' : ' 乡村振兴' , ' nums' : 357.79 }
{ ' words' : ' 官方微博' , ' nums' : 356.35 }
{ ' words' : ' 搜救' , ' nums' : 350.42 }
{ ' words' : ' 野生动物' , ' nums' : 349.31 }
{ ' words' : ' 食品安全' , ' nums' : 345.85 }
{ ' words' : ' 引领' , ' nums' : 335.61 }
{ ' words' : ' 主管部门' , ' nums' : 308.41 }
{ ' words' : ' 客服' , ' nums' : 305.03 }
{ ' words' : ' 垃圾分类' , ' nums' : 282.68 }
{ ' words' : ' 志愿服务' , ' nums' : 273.25 }
{ ' words' : ' 公示' , ' nums' : 264.36 }
{ ' words' : ' 履职' , ' nums' : 263.86 }
{ ' words' : ' 法律法规' , ' nums' : 255.11 }
{ ' words' : ' 告诉记者' , ' nums' : 246.86 }
{ ' words' : ' 空气质量' , ' nums' : 243.44 }
{ ' words' : ' 澎湃新闻' , ' nums' : 235.78 }
{ ' words' : ' 民营企业' , ' nums' : 235.5 }
{ ' words' : ' 赏樱' , ' nums' : 234.84 }
{ ' words' : ' 新闻发布会' , ' nums' : 232.34 }
{ ' words' : ' 食材' , ' nums' : 228.88 }
{ ' words' : ' 生态环境' , ' nums' : 225.82 }

根据以上信息我们可以清晰地看到占据了新闻名词榜的榜首的是“检察机关”，其中“医护人员”和“纪委监委”分别占据第2、3名。在过去的3月份，正是我们国家举行第十三届全国人民代表大会会议的时候，作为国家的权力机关与重要政府部门，出现频率之高反应其在国家政治生活中的重要作用，与此同时“纪委监委”作为重要的监察部门同样受到广泛的关注。对于关键词医护人员的出现也不奇怪，2019年国家卫健委对医护人员提出了更多更具体的要求，医护人员的薪资待遇方面也即将迎来改革，在各项突发事件中，医护人员首当其冲救死扶伤，在社会生活中发挥了巨大的作用。

那么，如何发现某一时间新闻是发生在哪些人身上，或者说某一事件的参与者受到的关注量是怎样的，可以很清晰的从图 4-5 获得想要的结果。



图 4-5 出现最多的人名

对应的人名数据视图表如下：

表 4-2 出现最多的人名

{ ' words' : ' 曹园' , ' nums' : 439.66 }
{ ' words' : ' 王磊' , ' nums' : 323.06 }
{ ' words' : ' 王某' , ' nums' : 213.76 }
{ ' words' : ' 陈某' , ' nums' : 177.56 }
{ ' words' : ' 孙强' , ' nums' : 149.13 }
{ ' words' : ' 刘某' , ' nums' : 136.96 }
{ ' words' : ' 王乃生' , ' nums' : 104.22 }
{ ' words' : ' 张英' , ' nums' : 94.12 }
{ ' words' : ' 沈巍' , ' nums' : 93.98 }
{ ' words' : ' 刘瑞强' , ' nums' : 87.5 }
{ ' words' : ' 王华州' , ' nums' : 84.76 }
{ ' words' : ' 张立' , ' nums' : 84.47 }
{ ' words' : ' 胡瑞娟' , ' nums' : 75.49 }
{ ' words' : ' 胡尹萍' , ' nums' : 74.34 }
{ ' words' : ' 刘瑞芹' , ' nums' : 73.16 }
{ ' words' : ' 胡某' , ' nums' : 73.02 }
{ ' words' : ' 李建明' , ' nums' : 71.1 }
{ ' words' : ' 陈伟起' , ' nums' : 70.01 }
{ ' words' : ' 罗应玖' , ' nums' : 68.94 }
{ ' words' : ' 陈培新' , ' nums' : 68.33 }
{ ' words' : ' 刘海蛟' , ' nums' : 65.61 }
{ ' words' : ' 张玉环' , ' nums' : 65.26 }

{ ' words' : ' 陈宗祥' , ' nums' : 63.79}
{ ' words' : ' 黄志发' , ' nums' : 62.93}
{ ' words' : ' 方彦格' , ' nums' : 62.28}
{ ' words' : ' 唐超' , ' nums' : 61.11}
{ ' words' : ' 李海涛' , ' nums' : 60.55}
{ ' words' : ' 李云峰' , ' nums' : 60.31}
{ ' words' : ' 潘志平' , ' nums' : 59.0}
{ ' words' : ' 刁继龙' , ' nums' : 55.96}
{ ' words' : ' 陈静瑜' , ' nums' : 54.18}
{ ' words' : ' 李某' , ' nums' : 52.47}
{ ' words' : ' 郑检凤' , ' nums' : 51.39}
{ ' words' : ' 黄淑芬' , ' nums' : 50.71}
{ ' words' : ' 袁长生' , ' nums' : 49.44}
{ ' words' : ' 蔡徐坤' , ' nums' : 47.49}
{ ' words' : ' 张静芬' , ' nums' : 45.79}
{ ' words' : ' 尹月娥' , ' nums' : 44.93}
{ ' words' : ' 费敏秀' , ' nums' : 44.91}
{ ' words' : ' 李雷' , ' nums' : 38.77}

通过上面的词云图我们发现王某、刘某、胡某在本月份出现的频率最高，同时这也是新闻媒体在新闻报道中最常用的指代性词语，继续探究其本后的情况，其实也间接表明中国的姓氏分布状况，因为王姓、刘姓的在全国姓氏人口中占据更大的一部分，所以相关联的姓氏人名新闻出现的概率便会更大一些。透过图中具体的人名信息，可以很清楚的了解这些人大多是3月份重大热门事件的新闻人物，媒体报道的数量自然也就更大一些。

如图4-6是新闻在全国各省市的统计分布图：

2019年03月份各省市新闻出现频率

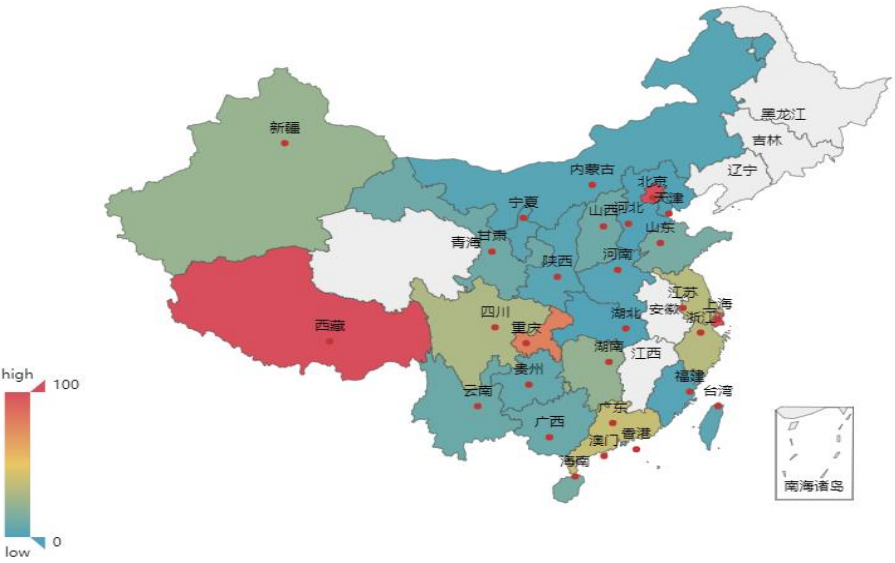


图 4-6 各省市新闻出现情况

通过不同的颜色，我们可以看到在不同的地方发生的热门事件。红色表示该地区的新闻频率非常高。北京，上海，重庆和西藏位居前列，而安徽，黑龙江和吉林的频率远远不够，继续看看每个城市的情况，如图 4-7 所示：

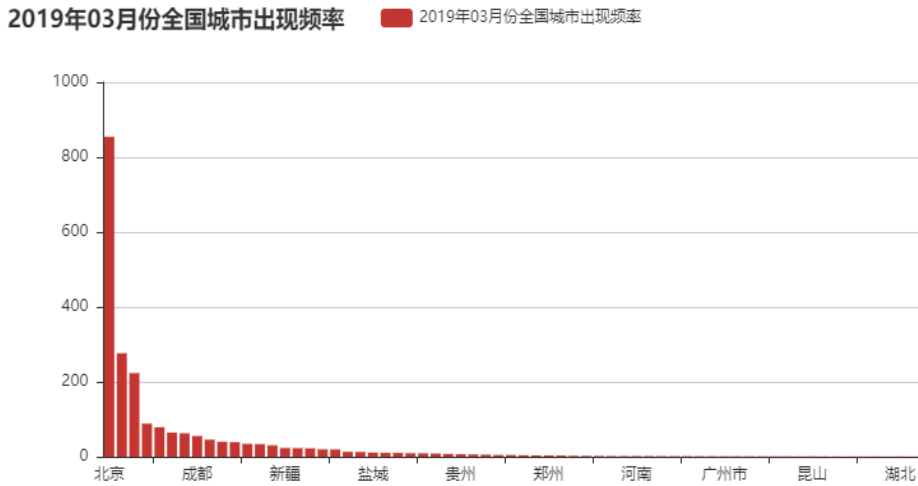


图 4-7 全国各城市出现频率

对应的全国省市数据视图表如下：

表 4-3 全国各城市出现频率

{ ' words' : ' 北京' , ' nums' : 854.58 }
{ ' words' : ' 上海' , ' nums' : 276.16 }
{ ' words' : ' 西藏' , ' nums' : 223.39 }
{ ' words' : ' 广州' , ' nums' : 88.38 }
{ ' words' : ' 重庆' , ' nums' : 78.85 }
{ ' words' : ' 海林' , ' nums' : 64.55 }
{ ' words' : ' 北京市' , ' nums' : 62.56 }
{ ' words' : ' 成都' , ' nums' : 55.4 }
{ ' words' : ' 杭州' , ' nums' : 45.52 }
{ ' words' : ' 武汉' , ' nums' : 40.05 }
{ ' words' : ' 广东' , ' nums' : 39.0 }
{ ' words' : ' 江苏' , ' nums' : 34.26 }
{ ' words' : ' 浙江' , ' nums' : 33.65 }
{ ' words' : ' 四川' , ' nums' : 30.42 }
{ ' words' : ' 新疆' , ' nums' : 23.36 }
{ ' words' : ' 深圳' , ' nums' : 22.98 }
{ ' words' : ' 湖南' , ' nums' : 22.43 }
{ ' words' : ' 南京' , ' nums' : 19.71 }
{ ' words' : ' 西安' , ' nums' : 19.51 }
{ ' words' : ' 海南' , ' nums' : 13.37 }
{ ' words' : ' 山东' , ' nums' : 12.83 }
{ ' words' : ' 盐城' , ' nums' : 11.14 }

```

{' words' : '甘肃', ' nums' : 10.36}
{' words' : '山西', ' nums' : 10.25}
{' words' : '香港', ' nums' : 9.38}
{' words' : '云南', ' nums' : 9.14}
{' words' : '广西', ' nums' : 8.66}
{' words' : '福州', ' nums' : 7.57}
{' words' : '贵州', ' nums' : 6.83}
{' words' : '武都区', ' nums' : 6.18}
{' words' : '拉萨', ' nums' : 5.57}
{' words' : '兰州', ' nums' : 4.78}
{' words' : '响水县', ' nums' : 4.68}
{' words' : '台湾', ' nums' : 4.1}
{' words' : '陕西', ' nums' : 3.64}
{' words' : '郑州', ' nums' : 3.6}
{' words' : '宁夏', ' nums' : 3.42}
{' words' : '英德', ' nums' : 2.83}
{' words' : '内蒙古', ' nums' : 2.68}
{' words' : '顺德', ' nums' : 2.6}
{' words' : '柳州', ' nums' : 2.27}
{' words' : '福建', ' nums' : 2.27}
{' words' : '河南', ' nums' : 2.24}
{' words' : '南海', ' nums' : 2.22}
{' words' : '重庆市', ' nums' : 2.14}
{' words' : '个旧', ' nums' : 2.03}
{' words' : '牡丹江市', ' nums' : 1.98}
{' words' : '易县', ' nums' : 1.83}
{' words' : '中山', ' nums' : 1.66}
{' words' : '广州市', ' nums' : 1.65}
{' words' : '黄山', ' nums' : 1.63}
{' words' : '云浮', ' nums' : 1.55}
{' words' : '天津', ' nums' : 1.48}
{' words' : '南宁', ' nums' : 1.46}
{' words' : '大连', ' nums' : 1.43}
{' words' : '石家庄', ' nums' : 1.31}
{' words' : '昆山', ' nums' : 1.29}
{' words' : '澳门', ' nums' : 1.25}
{' words' : '吉林市', ' nums' : 1.11}
{' words' : '沈阳', ' nums' : 1.09}
{' words' : '河北', ' nums' : 1.08}
{' words' : '盐城市', ' nums' : 1.04}
{' words' : '石家庄市', ' nums' : 1.03}
{' words' : '湖北', ' nums' : 1.03}
{' words' : '常州', ' nums' : 1.02}

```

过去三月哪些机构更频繁地出现在新闻中？可以通过此圆环饼图查看这些机构的情况，如图4-8所示：

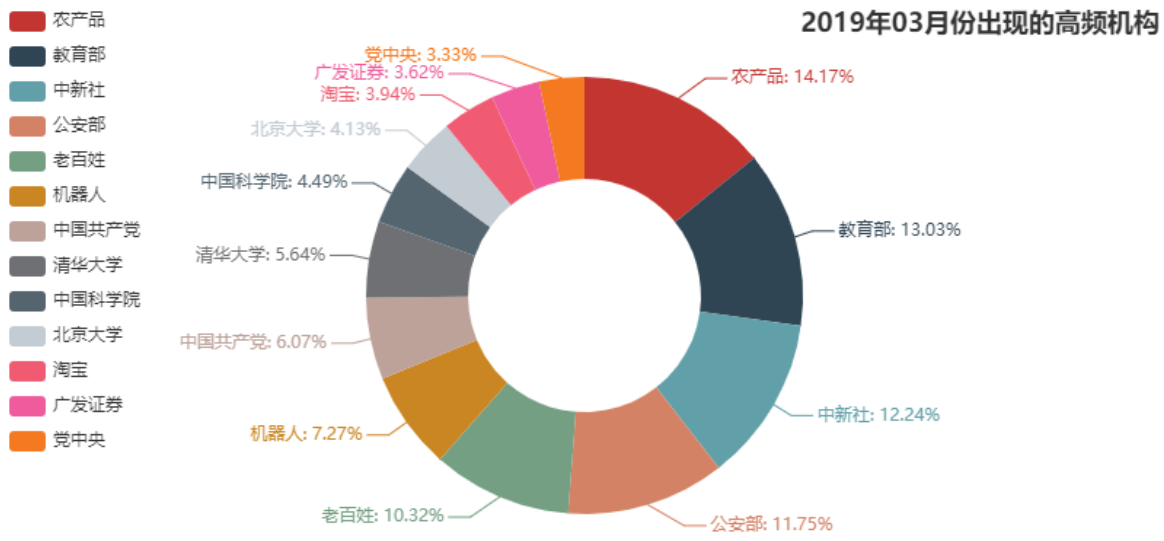


图 4-8 高频机构

对应的高频机构数据视图表如下：

表 4-4 高频机构

{ ' words' : ' 农产品' , ' nums' : 72. 25}
{ ' words' : ' 教育部' , ' nums' : 66. 46}
{ ' words' : ' 中新社' , ' nums' : 62. 45}
{ ' words' : ' 公安部' , ' nums' : 59. 92}
{ ' words' : ' 老百姓' , ' nums' : 52. 63}
{ ' words' : ' 机器人' , ' nums' : 37. 1}
{ ' words' : ' 中国共产党' , ' nums' : 30. 94}
{ ' words' : ' 清华大学' , ' nums' : 28. 77}
{ ' words' : ' 中国科学院' , ' nums' : 22. 89}
{ ' words' : ' 北京大学' , ' nums' : 21. 04}
{ ' words' : ' 淘宝' , ' nums' : 20. 08}
{ ' words' : ' 广发证券' , ' nums' : 18. 47}
{ ' words' : ' 党中央' , ' nums' : 17. 01}

通过上面的圆饼图可以清楚地看出，在过去 3 月里，关键词农产品所占比例最高，出现频率最高。我国作为农业大国，粮食生产一直是政府工作关注的重点，随着乡村振兴战略全面实施，农业信息化市场化的水平将得到显著的提升。教育部紧随其后的可能原因是研究生院校的复审阶段，同时这也是今年发布教育工作点的时候。除清华大学，中国科学院和北京大学外，其他大多数部门都是政府部门，大多数政府部门都能反映政府在我国的工作以及人们生活的重视和重要性。同时，召开相关会议也是这些热点词出现的重要原因。

5 总结

本文主要进行新闻热点问题分析,实现了新闻热点的发现和可视化分析。主要实现了如下任务:第一、利用多线程技术和爬虫算法实现了对中国新闻网的新闻的并行爬取。第二、利用中科院的 ICTCLAS 分词、信息熵算法分别对抓取的 2019 年 3 月 1 日到 31 日期间的社会新闻进行数据处理、词频词性统计以及 JSON 化处理。第三、利用 PyEcharts 技术对数据结果进行分类统计并绘制可视化视图。分析结果说明,2019 年 3 月份社会民生是民众的关注热点,在社会民生问题上,人们对中国在社会保障、教育、食品、舆论、机器人、互联网等方面的关切程度更加迫切,说明老百姓对国家更快更好发展的热切期盼,政府可以在这方面给予更多正确引导和支持。通过上述功能的实现,本文完成了毕业设计的预期任务,实现了预定目标。

新闻热点可视化分析是一种基于大量网络新闻数据的图表分析方式,本文以中国新闻网的新闻数据为来源,计算统计了某一领域的热点信息情况,借助 PyEcharts 强大的数据呈现技术,让普通的数据信息焕发出崭新的魅力。在研究方法上,发现还有如下能够改善的地方:可以采取多方向爬取不同源网站的各种新闻数据进行统计,通过聚合分析,可以获得更准确的统计数据;本文的分词和词频词性分析借助的是中国科学院的 ICTCLAS 分词,但是由于系统限制,需要对分词文本做切分为单个大小小于 500K 的文件,过程中难免会出现部分的文件中存在相同但有不同统计量的关键字,需要借助其他工具比如 Excel 进行相同词合并,操作实现上还比较繁琐;在数据可视化方面实现了对当前处理后静态数据的呈现,如何直接与程序的 API 接口连接,动态的根据传输的数据及时变化有更深层次的研究意义。

参考文献

- [1] .第 43 次 CNNIC 中国互联网报告发布[J].中国广播,2019(04):48.
- [2] 荣晗.基于分布式的网络爬虫系统的研究与实现[D].电子科技大学,2017.
- [3] 尚楚涵.互联网舆情信息挖掘技术研究与实现[D].华南理工大学,2013.
- [4] 孙健康.大数据背景下数据新闻的可视化研究[D].南昌大学,2017.
- [5] 王宝龙. 面向新闻领域的文本数据获取系统的设计与实现[D].北京邮电大学,2010.
- [6] 巫函.数据挖掘与可视化技术对新闻阅读体验的改善——以腾讯网在巴西世界杯期间的报道为例[J].西部学刊(新闻与传播),2016(07):53-55.
- [7] 高云.大数据时代数据新闻的发展现状探析[J].视听,2019(04):184-185.
- [8] 张进浩.数理统计在数据分析中的应用[J].金融经济,2017(12):130-131.
- [9] 杨凯利,山美娟.基于 Python 的数据可视化[J].现代信息科技,2019,3(05):30-31+34.
- [10] Thelwall, M. A web crawler design for data mining[J]. Journal of Information Science, 2001, 27(5):319-325.
- [11] Fatemeh Ahmadi-Abkenari, Ali Selamat. An architecture for a focused trend parallel Web crawler with the application of clickstream analysis[J]. Information Sciences, 2011, 184(1).
- [12] Lishan Deng. Innovative Application of Python in Data Crawling —Chinese Version of Movie Recommendation Platform[J]. Journal of Physics: Conference Series, 2019, 1168(3).
- [13] 甘凌博, 员婕.数据新闻的交互可视化策略探析——以 2018 年世界杯报道为例[J].新闻传播, 2019(03):37-38+41.
- [14] 邓若蕾.大数据时代数据新闻的可视化传播效果探究[J].传播力研究, 2019, 3(04):32.
- [15] 娄立原, 周选州.可视化数据新闻实践探析[J].青年记者, 2019(02):33-34.
- [16] 林文涛, 陈伟强, 刘杭燕, 叶楠.面向热点新闻的爬虫系统设计与实现[J].数字通信世界, 2019(01):261-262+132.
- [17] Ali Mesbah, Arie van Deursen, Stefan Lenselink. Crawling Ajax-Based Web Applications through Dynamic Analysis of User Interface State Changes[J]. ACM Transactions on the Web (TWEB), 2012, 6(1).
- [18] Shaojun Zhong, Zhijuan Deng. A Web Crawler System Design Based on Distributed Technology[J]. Journal of Networks, 2011, 6(12).
- [19] 鲁义轩.整合与分析:发挥大数据价值的两大关键[J].通信世界, 2013(20):48.

附 录

附录一：中国新闻网爬虫实现类

```
# -*- coding: utf-8 -*-
# Time: 2019/03/01 01:18
# Author: caiongbo
# File: news_spider.py
# 功能：
'''
    主要实现页面链接和新闻的爬取，是爬虫的主要内容
'''

import requests
from bs4 import BeautifulSoup
import time
import pymongo
import datetime

# 连接 mongoDB 数据库,并建立信息数据库，这里建立了 3 月份的数据库
client = pymongo.MongoClient('localhost', 27017)
news = client['chinanews_201903']
# 页面详细信息表
item_info = news['item_info']
# 网页连接 URL
url_lists = news['url_lists']

def get_day_list():
    '''
        得到设置时间段内的所有时间日期
        :return:以列表形式返回所有时间日期
    '''
    daylist = []    # daylist 存放这个时间段所有的日期
    # 设置了 03 月的起止日起
    start = '2019-02-28'
    end = '2019-03-31'

    # 将日期格式化
    datestart = datetime.datetime.strptime(start, '%Y-%m-%d')
```

```

dateend = datetime.datetime.strptime(end, '%Y-%m-%d')
while datestart < dateend:
    datestart += datetime.timedelta(days=1)
    daylist.append(datestart.strftime('%m%d'))
return daylist

def get_single_links(url):
    """
    通过某一天的主页面 URL 得到该页面下的所有 URL，并将 URL 存入到 url_list 的表中去
    :param url: 某天的主页面
    :return:
    """
    try:
        # 基本抓取流程
        wb_data = requests.get(url)
        #print(wb_data.status_code)
        wb_data.encoding = wb_data.apparent_encoding
        soup = BeautifulSoup(wb_data.text, 'lxml')
        links = soup.select('li > .dd_bt > a') #获取链接所在的标签获得链接
        for link in links:
            #print(link.get('href'))
            url_lists.insert_one({'links': link.get('href')}) #将链接插入到数据库中去
    except:
        print("error")

def get_all_links():
    """
    通过之前的时间段数据得到该时间段对应当天页面，然后通过 get_single_links 得到该页面
    下的所有链接，
    以此来循环得到这段时间所有新闻的所有的链接
    :return:
    """
    daylist = get_day_list() #获取时间段
    # 所有时间段对应的页面
    urls = ['http://www.chinanews.com/scroll-news/sh/2019/{}/news.shtml'.format(day) for day in
daylist]
    # 从所有时间段页面找到每个新闻的页面
    for url in urls:

```

```
print(url)
time.sleep(0.5)
get_single_links(url)
print("current database count:" + str(url_lists.count()))

def get_res(url):
    """
    从单个新闻的链接中得到这个链接中的新闻内容
    :param url: 单个新闻的链接
    :return:
    """
    try:
        contentTxt = ""
        wb_data = requests.get(url)
        wb_data.encoding = wb_data.apparent_encoding
        soup = BeautifulSoup(wb_data.text, 'lxml')
        contents = soup.select('.left_zw > p')
        for content in contents:
            contentTxt += content.get_text().strip()
        data = {
            "link": url,
            "content": contentTxt,
            "year": url.split('/')[ -3],
            "month": url.split('/')[ -2].split('-')[0],
            "day": url.split('/')[ -2].split('-')[1]
        }
        item_info.insert_one(data)
        print(data)
    except:
        print("error")
```

附录二：新闻文本数据分割类

```
# -*- coding: utf-8 -*-
# Time: 2019/03/05 09:48
# Author: caiongbo
# File: splitfile.py
# 功能:
'''
主要是由于中科院计算所的 NLPIR 分词系统只能分割 5000k 以内的数据(实际只有 490k),
所以不得不对大容量的 txt 文件进行分割成 500k 以下的文件
'''

import os
import time
import codecs

def split_file(file_name, filepath):
    '''
    输出文件 500k 大小的文件 read(len) len 表示一个字符, GBK 编码下一个汉字为 2 个字节
    故定义 lensize = 256000 表示读取 256000 个字符, 这些字符在最后存在 GBK 文件里的时候
    文件大小为:  $256000 * 2 / 1024 = 500k$ , 实际情况下可能里面有一些英文字符所以实际大小
    为
    490k 左右, 再测试发现再少 1024 个字符结果在分词处理时效果更好
    :param file_name: 要分割文件的名字
    :param filepath: 分割文件存放路径
    :return:
    '''
    url_list = []
    print(file_name)
    with codecs.open(file_name, 'r', 'utf-8') as f:
        lensize = 256000 - 1024
        count = 0
        words = f.read().strip()
        print('file size:', len(words), ' type is ', type(words))
        i = 0
        while i < len(words) - lensize:
            url_list.append(words[i : i + lensize])
            i = i + lensize
```

```

        output_file_name = filepath + str(count) + '.txt'
        with codecs.open(output_file_name, "w", 'gbk') as file:
            s = str(url_list)
            # NLPIR 分词系统只支持 GBK 的编码，将 utf-8 格式文本转换成 gbk,坑
            res = s.encode('utf-8', 'ignore').decode('utf-8', 'ignore').encode('gbk',
'ignore').decode('gbk', 'ignore')
            file.write(res)
            url_list = []
            count += 1
    url_list.append(words[i: -1])
    output_file_name = filepath + str(count) + '.txt'
    # print(i)
    # print(url_list)
    # print(output_file_name)
    with codecs.open(output_file_name, "w", 'gbk') as file:
        s = str(url_list)
        res = s.encode('utf-8', 'ignore').decode('utf-8', 'ignore').encode('gbk', 'ignore').decode('gbk',
'ignore')
        file.write(res)

def all_split_file():
    # 自动获得当前文件所在目录的父目录下的 contents 文件夹
    path = os.path.dirname(os.path.dirname(os.path.abspath(__file__))) + "\\contents\\"

    filename = "
    filepath = "
    # 设置要转换的文件，如果是 range(1, 12)表示装换 01content.txt-11content.txt
    # for i in range(1,12):
    for i in range(3, 4):
        file = "
        if i < 10:
            file = '0' + str(i)
        else:
            file = str(i)
        file += 'content'
        filename = path + file + '.txt'
        filepath = path + file + '\\'
        if not os.path.isdir(filepath):

```

```
        os.makedirs(filepath)
    split_file(filename, filepath)

if __name__ == '__main__':
    begin = time.time()
    all_split_file()
    end = time.time()
    # 统计要花费的时间
    print("spent time is " + str(end - begin))
```

附录三：新闻数据可视化实现类

```
# -*- coding: utf-8 -*-
# Time: 2019/03/10 17:05
# Author: caiongbo
# File: 03_paint.py
# 功能:
'''
主要实现数据可视化的呈现
'''

# -*- coding:utf-8 -*-
import pyecharts
from pyecharts import Bar
from pyecharts import online
# online()
import json
import codecs

path = '03_result.json'
with open(path, 'r', encoding='utf-8') as f:
    temp = json.load(f)
    list = []
    list = temp
temp = []
for i in list:
    temp.append(i)

print(len(temp))
for i in temp:
    try:
        i['nums'] = float(i['nums'])
    except:
        print('error')

dic_n_new = {} #新词
dic_nr = {} # 人名
dic_ns = {} # 地名
```



```

dic_nt = {} # 机构团体名
dic_nz = {} # 其它专名
dic_wh = {}

res_n_new = [] #新词
res_nr = [] # 人名
res_ns = [] # 地名
res_nt = [] # 机构团体名
res_nz = [] # 其它专名
res_wh = {}
for i in temp:
    try:
        if i['cates'] == 'n_new':
            if i['words'] in dic_n_new:
                dic_n_new[i['words']] = dic_n_new[i['words']] + i['nums']
            else:
                dic_n_new[i['words']] = i['nums']
        if i['cates'] == 'nr':
            if i['words'] in dic_nr:
                dic_nr[i['words']] = dic_nr[i['words']] + i['nums']
            else:
                dic_nr[i['words']] = i['nums']
        if i['cates'] == 'ns':
            if i['words'] in dic_ns:
                dic_ns[i['words']] = dic_ns[i['words']] + i['nums']
            else:
                dic_ns[i['words']] = i['nums']
        if i['cates'] == 'nt':
            if i['words'] in dic_nt:
                dic_nt[i['words']] = dic_nt[i['words']] + i['nums']
            else:
                dic_nt[i['words']] = i['nums']
        if i['cates'] == 'nz':
            if i['words'] in dic_nz:
                dic_nz[i['words']] = dic_nz[i['words']] + i['nums']
            else:
                dic_nz[i['words']] = i['nums']
        if i['cates'] == 'nz':

```

```

        if i['words'] in dic_wh:
            dic_wh[i['words']] = dic_wh[i['words']] + i['nums']
        else:
            dic_wh[i['words']] = i['nums']
    except:
        #print("eroor")
        pass

#处理各个词性的数据，最后逆序排列
def deal(res, dic):
    res.clear()
    c = {}
    for key in dic:
        c['words'] = key
        c['nums'] = dic[key]
        res.append(c)
        c = {}
    res.sort(key = lambda x:x['nums'])
    res.reverse()

# 输出 list 数据
def output(res, num = 200):
    for i in res[:num]:
        print(i)

deal(res_n_new, dic_n_new) #进行数据储存
output(res_n_new, 30) #数据输出
attr_n_new = []
value_n_new = []
for n_new in res_n_new[4:30]:
    attr_n_new.append(n_new['words'])
    value_n_new.append(n_new['nums'])
bar = Bar("2019 年 03 月份名词记", "2019 年 03 月份出现频率最高的名词")
bar.add("主视图", attr_n_new, value_n_new, xaxis_interval=0, xaxis_rotate=30, yaxis_rotate=30)
bar.render("2019 年 03 月份出现频率最高的名词.html")

from pyecharts import WordCloud

```

```

deal(res_nr, dic_nr)
output(res_nr)

attr_nr = []
value_nr = []
for nr in res_nr:
    attr_nr.append(nr['words'])
    value_nr.append(nr['nums'])

wordcloud = WordCloud("2019 年 03 月份人名记", "2019 年 03 月出现频率最高的人名",
width=900, height=620, title_pos="center")
wordcloud.add("人名", attr_nr[:40], value_nr[:40], word_size_range=[20, 120], shape='tree')
wordcloud.render("2019 年 03 月出现频率最高的人名.html")
# wordcloud

from pyecharts import Map
deal(res_ns, dic_ns)
output(res_ns)
attr_ns = []
value_ns = []
for ns in res_ns:
    attr_ns.append(ns['words'])
    value_ns.append(ns['nums'])
map = Map("2019 年 03 月份各省市新闻出现频率", width=800, height=600)
try:
    map.add("", attr_ns, value_ns, maptype='china', is_visualmap=True, visual_text_color='#000',
is_label_show=True)
except Exception as e:
    pass
map.render("2019 年 03 月份各省市新闻出现频率.html")
# map

from pyecharts import Geo
deal(res_ns, dic_ns)
# output(res_ns)
data = []
data.clear()

```

```
for ns in res_ns:
    tupl = (ns["words"], ns["nums"])
    data.append(tupl)

geo = Geo("2019 年 03 月份全国城市出现频率", "data from chinanews", title_color="#fff",
title_pos="center",
width=800, height=600, background_color='#434a59')
attr, value = geo.cast(data)

geo.add("", attr, value, visual_range=[0, 300], visual_text_color="#fff",symbol_size=15,
is_visualmap=True)
geo.render("2019 年 03 月份全国城市出现频率.html")
# geo

from pyecharts import Pie
deal(res_nt, dic_nt)
output(res_nt)
attr_nt = []
value_nt = []
for nt in res_nt:
    attr_nt.append(nt['words'])
    value_nt.append(nt['nums'])
pie = Pie("2019 年 03 月份出现的高频机构", title_pos='right')
pie.add("", attr_nt[2:15], value_nt[2:15], radius=[40, 75], label_text_color=None,
is_label_show=True,
legend_orient='vertical', legend_pos='left')
pie.render("2019 年 03 月份出现的高频机构.html")
# pie
```