新闻热点话题检测

摘要：

本文主要是针对新闻热点话题检测技术进行研究。首先是基于爬虫技术的新闻爬取，通过爬虫技术爬取了新浪、搜狐、新华网等新闻网页的大量实时新闻；其次是基于聚类算法的新闻聚合，利用 jieba 分词、TF-IDF 技术、聚类算法对清洗过的新闻文本进行特征提取与聚类分析，得到了新闻的聚合簇；最后是话题热度排行，使用 TextRank 算法对每一处聚合簇中的新闻贡献度进行排名，结合聚类结果与词向量分析，得到了对新闻热度的排名与分析。

1 课题研究背景与意义

随着计算机与互联网技术的蓬勃发展，互联网信息呈现出爆炸性增长，越来越多的人将互联网视为获取信息的最佳平台。如今，我们所处的不再是信息贫乏的时代，而是一个充斥着海量信息的新时代，随着“大数据”时代的到来，信息量增多，信息种类繁多，涉及范围广泛，新闻五花八门，同一主题下的新闻，多得数不胜数，人们可以根据自己的职业和爱好关注专业新闻网站最近的不同热点要闻。

虽然说关键词检索是目前从海量信息中获取有用信息的主要途径，但通过关键词检索得到的信息，其冗余度往往较高，同时有用信息也常常丢失。因此，人们迫切希望有一种方法可以自动处理大量信息并挖掘相关的话题，对话题相关信息进行有效的组织。话题检测技术通过对海量的文本信息进行处理，可以帮助网民筛选感兴趣的新闻信息。

2 基于爬虫技术的新闻爬取

2.1 互联网爬虫技术

如果已知网站的外部 API，就可以快速获取网站数据，获取的数据通常是 json 格式，只需通过 json 模块进行转化就可以获得网站信息的相关字典，通过字典键值对的形式就可以通过关键字获取到其对应的值。

不过网站很多地方往往没有设置数据获取的 API，所以还是需要通过各种爬虫手段来抓取网站页面的代码，然后对网页代码进行解析来获取到想要的数据。爬取网页数据的方法如下所示，

（1）正则表达式匹配。

（2）BeautifulSoup 模块。

（3）Lxml 模块。

比如一个网页的代码 html 有“<td>Topic Detection and Tracking </td>”，需要获取<td>元素中的内容，以上三种方法对应的表达式为，

（1）正则表达式匹配：re.findall('<td>(.*?)</td>', html)

（2）BeautifulSoup 模块：BeautifulSoup(html).find('td').text

（3）Lxml 模块：lxml.html.fromstring(html).cssselect('td')[0]

2.2 新闻采集

本设计爬虫爬取的新闻网站为新浪、搜狐、新华网。爬取的流程为，首先利用网站的新闻 API 爬取新闻标题、时间、url，然后利用 lxml 模块获取新闻的详细信息，最后，设置 3 个线程爬取新闻网站的新闻。

在爬取过程中，首先找到这三个网站解析新闻网页内容的 API，如图 2.1 所示，然后用 API 请求数据，最后就可以得到最近新闻的相关信息。

表 2.1 新华网、搜狐网、新浪网三个网站解析网页内容的 API

新华网	https://feed.mix.sina.com.cn/api/roll/get?pageid=153&lid=2516&k=&num={}&page={}&r={}
搜狐网	http://v2.sohu.com/public-api/feed?scene=CHANNEL&sceneId=15&page={}&size={}
新浪网	http://qc.wa.news.cn/nodeart/list?nid=11147664&pgnum={}&cnt={}&tp=1&orderby=1

在爬虫爬取过程中，选择 python 自带的 urllib 模块爬取信息，最后得到这三个网站的新闻字典 dict，该字典中包含新闻的标题、发布时间以及 url。除上述信息外，还要获取新闻的详细内容，本设计选择 lxml 模块的 html 解析器对 url 数据进行解析，得到该新闻页面的所有 html 信息，最后再通过 xpath 提取新闻内容，爬取新浪、搜狐、新华网的流程图如图 2.1 所示，

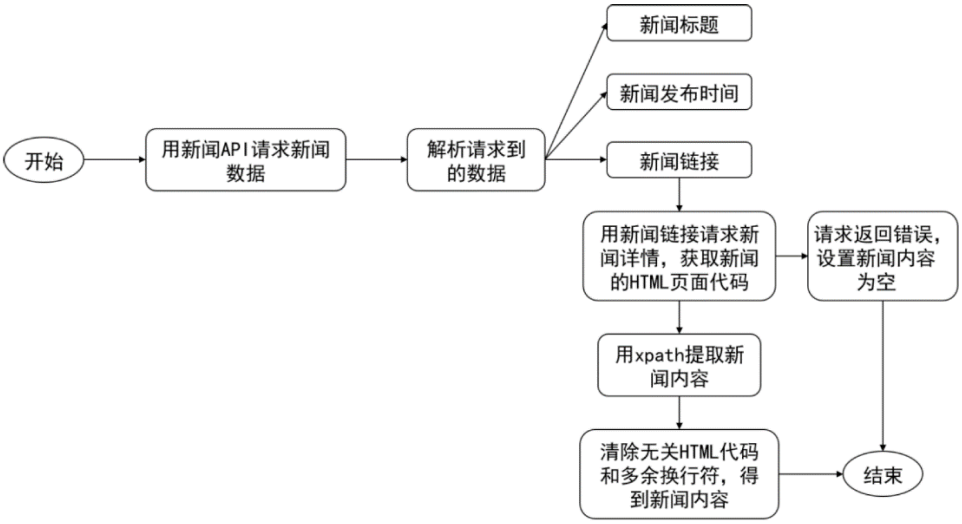

图 2.1 新闻网信息爬取流程图

为了提高三个网站新闻信息的爬取速度，本设计采用多线程信息爬取策略，新浪、搜狐、新华网每个网站分配一个线程，最后将三个网站的信息合并在一起，这样就可以大大增加新闻信息爬取的效率。

2.3 信息清洗

对爬取新闻信息的清洗包括两个流程：整体清洗和局部清洗，整体清洗适用于所有的信息记录，如：去除乱码记录，去除严重信息缺失记录，不合法的字段信息统一清除为 null，去掉前后的空格、去掉多个<p>因为缺失新闻内容连在一起的情况等；局部清洗适用于特定的类别，如发布时间字段的统一，将新闻中多出“图片来源”、“资料来源”等无关的信息替换为空字符等。

对于新闻的聚类，多数标点符号也会影响聚类效果，所以需要将新闻内容的标点符号全部删除以便剩下的中文、英文和数字能很好的反映新闻内容。另外，为了方便 jieba 分词的进行，在清洗时，需要将句子中特殊字符分离。

2.4 实验结果

通过定义好的爬取规则，对新闻网站的信息进行爬取，得到如图 2.2 所示的

大量新闻。

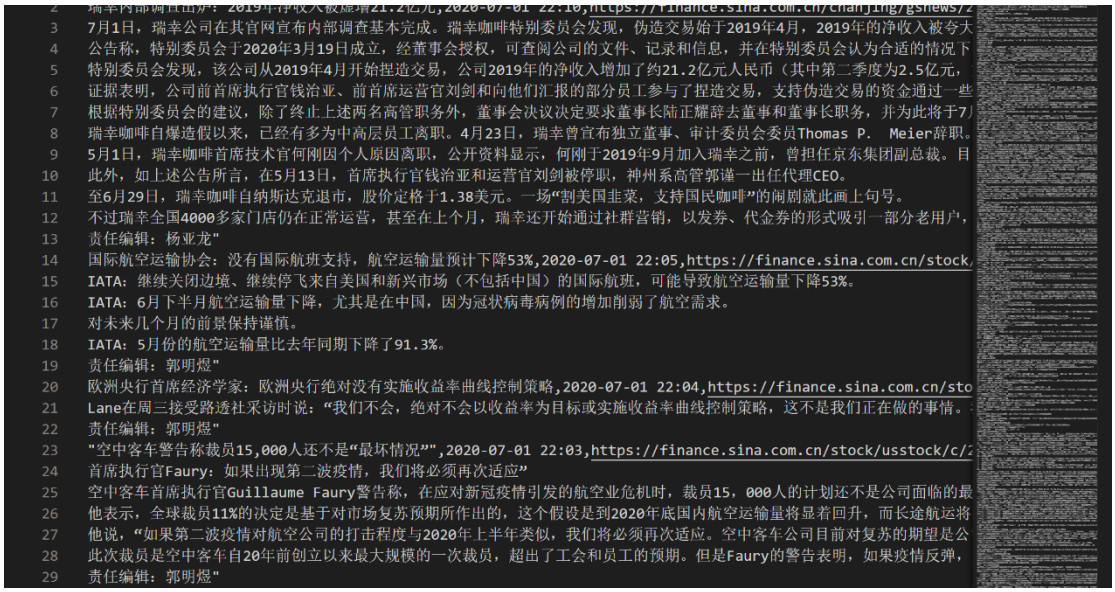

图 2.2 爬虫爬取结果

按照顺序分别从新华网、新浪、搜狐等新闻网页进行信息爬取。针对新闻的日期等因素进行限制，可以初步过滤新闻，将重复的新闻与时效性差的新闻进行滤除。根据新闻的标题、日期、原始 url 来源、新闻内容进行切分，最后将内容保存到本地的 csv 文件中，将新闻文件进行组合之后，得到大约 10M 的纯文本信息。

3 基于聚类算法的新闻话题聚合

3.1 jieba 分词

jieba 分词是基于前缀词典实现的词图扫描方法，该分词算法的步骤分为两个部分，一部分是针对登录词分词，登录词即 jieba 工程上的 dict.txt；另一部分是针对未登录词进行分词。

不管对于哪种词，首先要加载登录词典，建立 Trie 树分词模型，建立分词 DAG 词图，然后计算全局概率得到基于前缀词典的词频最大切分组合；针对未登录词，利用 Token 使中文部分和英文部分分开识别，加载隐马尔科夫链模型图，采用 Viterbi 算法动态规划取得分词和标注；针对登录词，按照字典标注识别，识别英文、数字以及时间形式的组合，最后给出结果。

另外，为了更好的聚类效果，在分词中需要设置停用词词典，即在分词之后，把在停用词词典中不需要的词去除；同时，本设计还使用消歧词典来将一个多个同一意思的词转化为同一个词；本设计也创建了一个保留单字词典，只保留在词典中的单字，把其他所有的单字都去除，例如“是”、“会”等。

3.2 文本特征计算

为了有效地获取文本数据的特征，需要对文本数据进行特征提取。

3.2.1 TF

词频表示词条在文档中出现的频率，对于特定文件里的某一词语来说，它的重要性可表示为：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3-1}$$

式中，$n_{i,j}$ 为该词在文件 d_j 中的出现次数，$\sum_k n_{k,j}$ 在文件中所有字词的出现次数之和。

在 python 中，通过 sklearn.Countvectorizer()函数来统计词频，然后转换为对应的向量。在统计过程中，把所有词库中的词当成列，并初始化向量为[0,0,…,0,0]，长度由词库的词的个数决定，如果文本中出现某一词，那么某一词的数就+1，最后得到词频的向量。

3.2.2 IDF

IDF 是一个词语普遍重要性的度量，

将一类文档包含某个词条的文档数据统计出来，若该数字较小，即 IDF 较大，就说明词条由非常好非常好的区分能力；若该数较大，即 IDF 的值就会较小，就说明该词条有比较差的区分能力。

某个词语的 IDF，计算方法为总文件数目除以包含该词语之文件的数目，再将该商取对数，

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{3-2}$$

式中，$|\mathrm{D}|$ 指语料库中的文件总数；$|\{j : t_i \in d_j\}|$ 指包含词语 t_i 的文件数目（即 $n_{i,j} \neq 0$ 的文件数目），若该词语不在语料库中，就会导致分母为零，因此一般情况下使用作为分母。

$$1 + |\{\mathrm{d} \in D : t \in d\}| \tag{3-3}$$

3.2.3 TF-IDF

TF-IDF 用来评估一字或者一个词对于一个文件集的重要程度。TF-IDF 是一种用于信息检索与数据挖掘的常用加权技术。

计算公式如下：

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{3-4}$$

该模型的训练过程分为两个步骤:jieba 分词和 TF-IDF 关键词提取。

通过 TF-IDF 特征提取之后，在所有的新闻中，每篇新闻都有一个向量标识。向量上的每一个值都是一个词的 TF-IDF 值。其向量获得方式为首先统计出所有的词，把每个词当成向量的每一个维度，如果该文档中有某词，就在某词的维度上计算它的 TF-IDF 值；如果不存在某词，那么某词的维度上的值就为 0。采用 TF-IDF 方法对所有的新闻进行特征提取，提取的结果是一个稀疏矩阵。

3.3 新闻聚类算法

在话题检测技术中，聚类方法有着非常非常重要的作用，即将 TF-IDF 提取的关键词进行聚类操作，把它们分到各自的话题中，从而达到在相同簇内相似度高不同簇内相似度低的效果。

密度聚类算法为了解决距离聚类算法对凹形数据集不是很好处理的问题，该算法区域内对象的密度大于阈值，那么就可以将这个区域聚类到与之相邻的类簇

中去。该算法优点突出，可以发现任何形状的簇，最常用的是 DBSCAN 密度聚类算法。

DBSCAN 聚类算法是一种基于高密度连通区域的聚类算法，能够将高密度的区域划分为簇，并在有噪声的数据中发现任意形状的簇。

其原理为：首先检查数据集中每点的邻域，如果其邻域包含的点大于阈值，则创建一个以该点为核心的簇；然后迭代地聚集从这些核心对象直接密度可达的对象；最后，当没有新点添加到簇时结束。

伪代码如表 3.1，

表 3.1 DBSCAN 算法

输入：数据集 D，给定点在邻域内成为核心对象的最小邻域点数：MinPts, 邻域半径：Eps

输出：簇集合

(1)将数据集 D 中的所有对象标记为未处理状态

(2) for（数据集 D 中每个对象 p） do

(3)　　if （p 已经归入某个簇或标记为噪声） then

(4)　　continue;

(5)　　else

(6)　　检查对象 p 的 Eps 邻域 NEps(p) ；

(7)　　if (NEps(p)包含的对象数小于 MinPts) then

(8)　　标记对象 p 为边界点或噪声点；

(9)　　else

(10)　　标记对象 p 为核心点，并建立新簇 C, 并将 p 邻域内所有点加入 C

(11)　　for (NEps(p)中所有尚未被处理的对象 q)　do

(12)　　检查其 Eps 邻域 NEps(q)，若 NEps(q)包含至少 MinPts 个对象，则将 NEps(q)中未归入任何一个簇的对象加入 C；

(13)　　end for

(14)　　end if

(15)　　end if

(16) end for

在本设计中，距离度量方法采用余弦相似度进行度量，本设计在设置 Eps 参

数时，一般都设置为 0.3-0.5 之间，因为超过 0.5 的半径值会使不属于同类新闻聚在一起，小于 0.3 则无法识别相同事件的新闻。设置 MinPts 参数时，可以人为的假定一个阈值，假设有 10 条新闻报道同一事件就认为它就是一个热点，那么可以设置 MinPts 值为 10。

3.4 实验结果

在信息爬取的过程中，我们针对爬取规则、重复新闻等特点，对新闻进行了第一次清洗。在对新闻进行处理的过程中，我们发现爬取到的新闻内容中，存在部分新闻的内容存在大量的颜文字、空白等非常规语义信息。同时，存在部分新闻内容由于编码方式不一致，导致在处理的过程中出现断词错误现象。因此，我们针对这些新闻内容进行了二次处理，将无法编码的内容与缺失的内容进行清洗。最后，将新闻内容中无效的信息（责任编辑、资料来源、标点符号）进行剔除，得到了可以进行分析的文本信息。

本实验采用了 jieba 分词的方法，将新闻文本内容进行分词。在实验的过程中，我们发现使用 jieba 进行自动分词，会出现部分专用名词也被切断的现象，对我们后续的结果分析带来了不利的影响。因此，在充分利用 jieba 分词的基础上，我们自己定义了一个专有名词词典，将新闻内容中被错分概率高的词汇进行有针对性地处理，这会大大降低程序处理错误的现象。

同时，在分词的过程中，我们得到了词汇的词性信息，因此根据词性信息，我们将对话题主题无关的连接词、副词与助词等词汇进行过滤，只保留对话题贡献度大的名词、动词等词语，这样大大降低了词向量的维度。最后，我们通过自己定义的同义词词典，将具有相同意义的词语进行合并，得到了最终的分词结果，如图 3.1 所示，

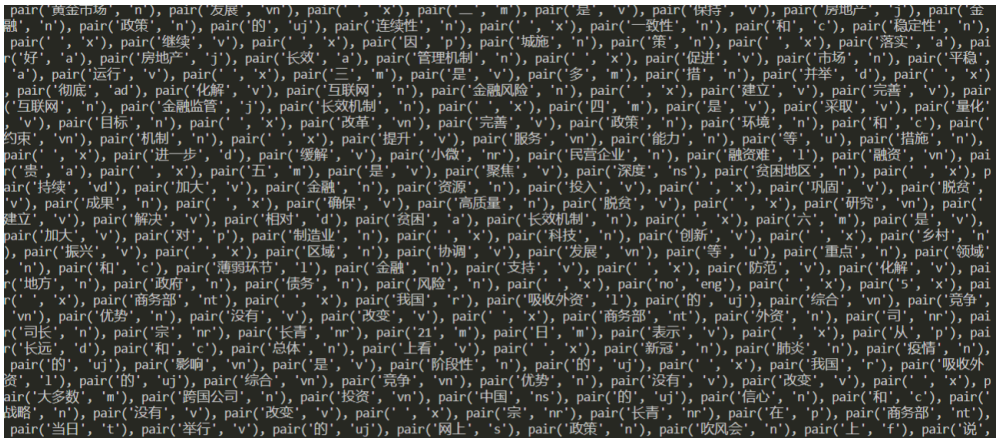

图 3.1 分词结果示意图

在得到了分词结果之后，对这些词语进行特征提取，通过提取到的特征来实现对新闻更好的表达。本实验选择了 TF-IDF 来对新闻进行特征提取，减小了新闻中无意义词的影响，可以更好的体现新闻内容的主题。首先统计出所有新闻中出现的所有词，将每个词当成向词向量的一个维度，若新闻中存在该词语，则在该维度上计算该词语的 TD-IDF 特征值；若不存在该词语，则将其置为 0。得到了可以表达所有新闻内容的稀疏矩阵。

得到所有新闻内容的稀疏矩阵后，我们通过基于密度的 DBSCAN 算法对新闻进行聚类分析，得到相应的话题。通过基于密度的聚类算法可以将任意形状的新闻点簇进行聚合，同时降低了离散点对整体聚类效果的影响，其结果示意图如图 3.2 所示，

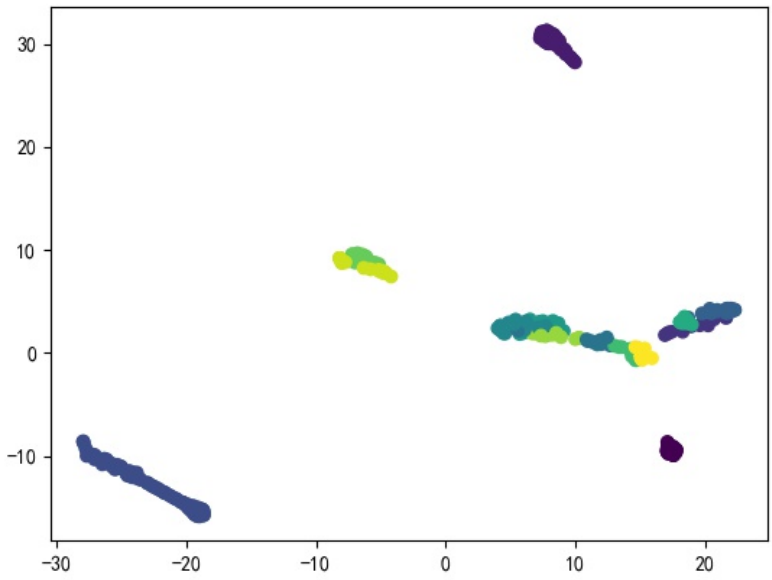

图 3.2 聚类结果示意图

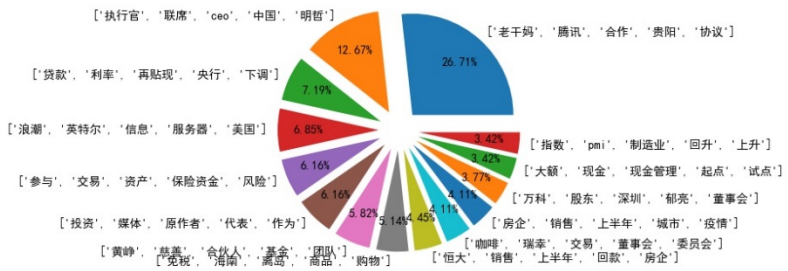

图 3.3 聚类簇示意图

将扫描半径设置为 0.43，设定类别最小值为 5，我们得到了上图所示的聚类结果。由新闻内同进行聚类共产生了 14 个簇，每个簇的结果如下：

1) 全面解读：腾讯帮老干妈做了什么广告？三人如何行骗的？
2) 马明哲辞任 CEO 平安联席 CEO 机制成熟
3) 巴西央行或将再次下调基准利率
4) 英特尔“断供”浪潮 国产服务器厂商隐忧加剧
5) 保险资金参与股指期货交易不得用于投机 应当以对冲或规避风险为目的
6) 新传媒：与旷视科技等合作参与投资人工智能产业基金
7) 黄峥与创始团队捐赠 2.37%股份成立繁星慈善基金
8) 图解|海南“离岛免税”涵盖这 45 类商品
9) 中国恒大上半年合约销售 3488 亿 销售、回款均创新高
10) 瑞幸咖啡宣布调查结果：去年虚增收入 21 亿元，要求陆正耀辞去董事长职务
11) 地产股 7 月“开门红”：百余只上涨，十余只涨停，房企上半年业绩超预期是主因
12) 万科董事会大换血，深铁 3 席位全换重申“不干预”
13) 对公账户起点为 50 万元 7 月 1 日起大额现金管理在河北试点
14) PMI 分析：制造业加速扩张 就业压力加剧

因此，我们将新闻聚合的结果作为相应一类新闻的话题，例如第一个话题即为“腾讯与老干妈事件”。综合近期的新闻热点分析，疫情对生产的影响确实是我们讨论的主题。

4 话题热度排行

4.1 TextRank 文本排行算法

TextRank 是基于图的文本的排序算法，类似于网页的排名，对于词语或者句子均可以得到排名，所以它可以进行关键词提取和自动文摘。其用于自动文摘时的思想是：将每个句子看成 PageRank 图的节点，若两个句子的相似度大于阈值，则认为它们有相似联系，它们间就有一条无向有权边，权值是相似度，再用 PageRank 算法得到句子的得分，把得分较高的句子作为文章的摘要。

4.2 基于 TextRank 自动文摘算法的热点话题排行

基于 TextRank 算法进行热点话题排行的过程如图 4.1 所示：

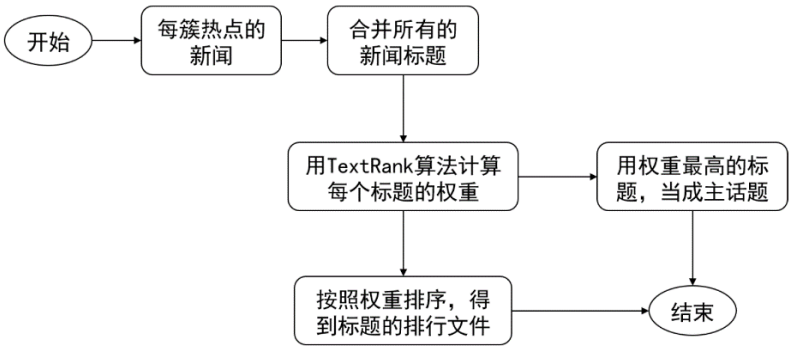

图 4.1　TextRank 算法进行热点话题排行流程图

在该过程中，首先合并所有的新闻标题，然后调用 TextRank 算法进行计算，得到达到字数要求的句子的排序，对于权重最高的句子，可以认为就是该处热点的主话题。

4.3 实验结果

本实验中，我们使用了 TextRank 算法对聚类出的新闻话题进行热度排序。首先，对新闻内容进行预处理，得到新闻的相应关键词集。通过如下公式然后计算新闻词句之间的相似度，

$$\text{相似度}=\frac{\text{重复词数量}}{\log(\text{内容1中词向量长度})+\log(\text{内容2中词向量长度})} \qquad (4.1)$$

设定相似度判断的基本阈值，将大于阈值的内容进行连接，其连接权重即为相似度。再计算词向量（节点）的权重，

$$\text{权重}=(1\text{-阻尼系数})+\sum\frac{\text{相似度}\times\text{节点权重}}{\text{所有连接节点边的权重和}} \qquad (4.2)$$

多次迭代式（4.2），即可得到稳定的节点权重。将节点按权重排序，将权重最高的节点作为重要向量。

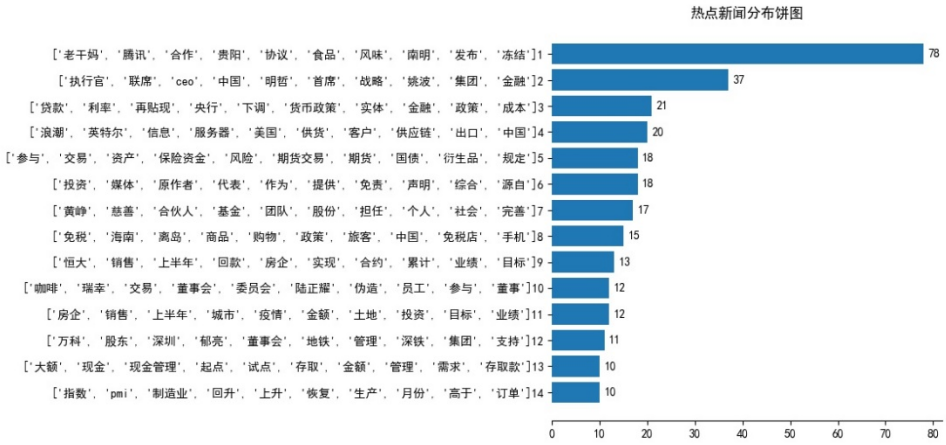

图 4.2 热点新闻分布图

将聚类得到的稀疏矩阵输入 TextRank 算法，对聚类出的话题进行内部贡献度排序。可以看到贡献最高的为腾讯与老干妈事件，贡献了 78 点热度。完整的贡献度分析如下：

热点：调查：腾讯与老干妈事件

相关词汇：['全面', '腾讯', '老干妈', '广告', '行骗']

相关词汇：['老干妈', '遭遇', '危机', '继承者', '乘风破浪']

相关词汇：['老干妈', '回应', '腾讯', '催收', '法院', '裁定', '不知情']

相关词汇：['腾讯', '伪造', '老干妈', '印章', '签合同', '刑拘', '腾讯', '辣椒酱']

相关词汇：['腾讯', '回应', '自筹', '老干妈', '征集', '线索', '支付宝', '反应']

相关词汇：['腾讯', '自掏腰包', '准备', '老干妈', '骗子', '线索']

相关词汇：['老干妈', '欠款', '腾讯', '百度', '搜索', '回应', '与我无关']

相关词汇：['腾讯', '面子', '被告', '老干妈', '里子', '一干二净']

相关词汇：['腾讯', '老干妈', 'QQ', '飞车', '追逐', '头部', '游戏']

相关词汇：['腾讯', '悬赏', '老干妈', '骗子', '线索', '律师', '分析', '案件', '疑点']

相关词汇：['腾讯', '老干妈', '纠纷案', '反转', '拖欠', '广告费']

相关词汇：['媒体', '环京', '松绑', '河北', '怀来县', '废止', '住房', '限购', '政策', '腾讯', '回应', '一事', '一言难尽', '老干妈', '骗子', '线索']

相关词汇：['腾讯', '回应', '老干妈', '事件', '一言难尽', '欢迎', '网友', '提供线索']

相关词汇：['腾讯', '老干妈', '公章', '金融', '萝卜', '事件']

相关词汇：['腾讯', '老干妈', '公章', '套取', '游戏', '礼包', '刑拘', '金融', '萝卜', '事件']

相关词汇：['腾讯', '员工', '自筹', '老干妈', '线索', '支付宝']

相关词汇：['律师', '老干妈', '腾讯', '事件', '嫌犯', '诈骗罪']

这些新闻都围绕着疫情期间的经济生产主题，在对词向量进行分析后，我们的得到了相应新闻的词向量图：

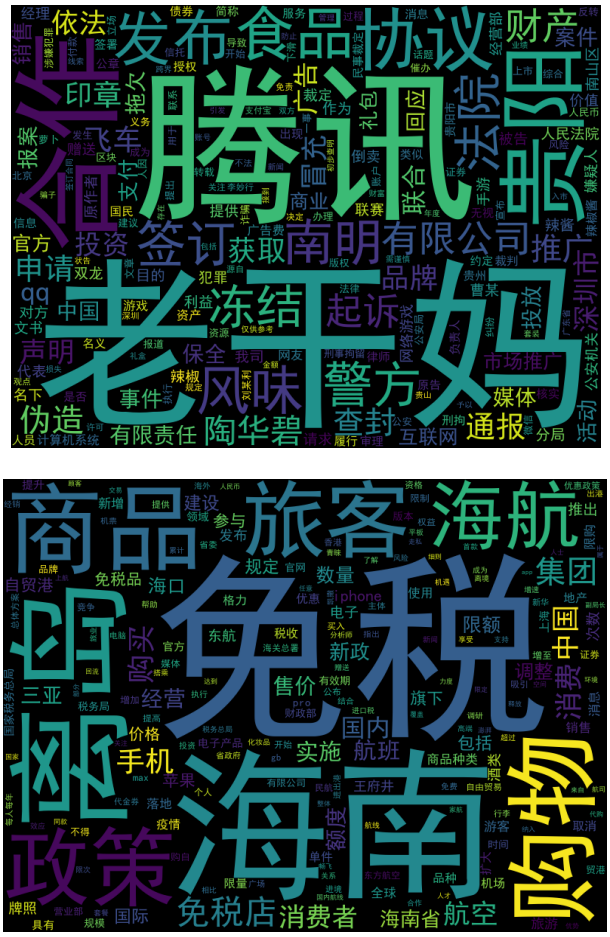

图 4.3 词云结果图

5 结论与展望

本文主要进行了针对新闻的热点话题分析，实现了对新闻内容的话题聚类与热度排行。具体实现了如下任务：

首先，通过爬虫技术爬取了新浪、搜狐、新华网等新闻网页的大量实时新闻。由于爬取到的内容存在大量的 dirty 数据，我们通过文本处理技术，对爬取到的内容进行清洗。

其次，利用 jieba 分词、TF-IDF 技术、聚类算法对清洗过的新闻文本进行特征提取与聚类分析。得到了新闻的聚合簇。

最后，使用 TextRank 算法对每一处聚合簇中的新闻贡献度进行排名，结合聚类结果与词向量分析，得到了对新闻热度的排名与分析。从结果上看，武汉新型冠状病毒的热度综合来看是最高的，这也与当前的形式环境密切相关，同时，这也证明了我们算法的有效性。

通过上述功能的实现，本文实现了基本的任务要求。同时在数据处理的过程中，我们也发现了技术上的一些不足。由于需要用户自主定义的特殊词，这在一定程度上降低了算法的普适性。在后期的学习过程中，可以通过引入深度学习的文本处理方法，得到更好的结果展示。