数据储存部分说明

数据存储设计，首先我们设计一个dowload函数用于储存

```
def download(self):
    df = pd.DataFrame(self.item_list)
    df.to_csv(f"../static/{self.name}.csv",mode="a+",encoding="utf-8-sig",index=False)
```

然后根据这些获取到信息，我们用字典的形式保存起来

```
        item = {}
        name = div.xpath('.//div[@class="content"]/div[@class="info"]//a[@class="name"]/text()')
        name = name[0] if len(name) != 0 else None
        time = div.xpath('.//div[@class="content"]/p[@class="from"]//a[1]/text()')
        time = str(time[0]).strip() if len(time) != 0 else None
        f = div.xpath('.//div[@class="content"]/p[@class="from"]//a[2]/text()')
        f = str(f[0]).strip() if len(f) != 0 else None
        content = div.xpath('.//div[@class="content"]/p[@class="txt"]//text()')
        content = str(content[0]).strip() if len(content) != 0 else None
        transmit = div.xpath('.//div[@class="card-act"]/ul/li[1]/a/text()')
        transmit = str(transmit[0]).strip() if len(transmit) != 0 else None
        comment = div.xpath('.//div[@class="card-act"]/ul/li[2]/a/text()')
        comment = str(comment[0]).strip() if len(comment) != 0 else None
        praise = div.xpath('.//div[@class="card-act"]/ul/li[3]/a/button/span[2]/text()')
        praise = str(praise[0]).strip() if len(praise) != 0 else None
```

```
item['name'] = name
item['time'] = time
item['f'] = f
item['content'] = content
item['transmit'] = transmit
item['comment'] = comment
item['praise'] = praise
item['spiderTime'] = datetime.datetime.now().strftime('%Y-%m-%d')
print(item)
self.item_list.append(item)
```

并且写好的字典内容，用列表的形式保存起来

传到一个总的列表里面

```
    print(item)
    self.item_list.append(item)
```

然后进行一个循环防错处理

当爬到尾页的时候我们就停下来，如果没有那么多页我们就把这些数据保存起来，用CSV的格式，如果出现错误，直接把数据保存起来，防止数据丢失的问题

```
if "51" in str(next_url):
    print("已经到达尾页，程序结束!")
    self.close()
    self.download()
    sys.exit()
else:
    self.url = next_url
    try:
        self.getData()
    except Exception as e:
        print(e)
        self.download()
```

最后每爬取一页，就做一个内容的保存，这样一来不会让数据丢失，二来也缓解了内存的压力

，方法用的是,mode="a+"的形式，也就说不断的追加形式，确保新的内容不会把旧的内容刷新，做到不断往表格里面添加新的数据，并且旧的数据也能被保留下来

以下分析步骤主要分为三个部分：

第一步部分，数据处理部分：

首先这是原始数据的部分：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	name	time	f	content	transmit	comment	praise	spiderTime										
2	澎湃新闻	04月08日	微博视频号	【英国首相	77	110	245	2022/4/9										
3	广东新闻频	04月08日	微博视频号	【广州：重	746	1969	33881	2022/4/9										
4	央视网	今天09:01	微博视频号	【	40	252	597	2022/4/9										
5	沪语说唱M	2秒前	iPhone 11	上海一学生	转发	评论	赞	2022/4/9										
6	Harry超越	2秒前	iPad客户端	预测一下，	转发	评论	赞	2022/4/9										
7	美杜莎麻麻	4秒前	nova8 我是	橙色是这边	转发	评论	赞	2022/4/9										
8	上海黄浦	6秒前	上海融媒体	【从机关至	转发	评论	赞	2022/4/9										
9	中新社吉林	8秒前	微博视频号	【	转发	评论	赞	2022/4/9										
10								2022/4/9										
11	睡在云朵上	9秒前	vivo手机	昨天下午当	转发	评论	赞	2022/4/9										
12	小只糯米驰	10秒前	微博视频号	#黄渤梅婷	转发	评论	赞	2022/4/9										
13	那么等风来	10秒前	iPhone 11	上海你这么	转发	评论	赞	2022/4/9										
14	环保三亚	11秒前	微博 weibo	发布了头条	转发	评论	赞	2022/4/9										
15	太极拳学	15秒前	Redmi K20	#广州 疫情	转发	评论	赞	2022/4/9										
16	哈尔滨经济	16秒前	iPhone客户	哈尔滨4月	转发	评论	1	2022/4/9										
17	夏天橙	16秒前	iPhone客户	从2/15日开	转发	评论	赞	2022/4/9										
18	南方周末	20秒前 转	微博视频号	【制造业人	13	24	58	2022/4/9										
19	中国新闻网	17秒前	微博 weibo	【	转发	评论	1	2022/4/9										
20	大餅餅是餅	17秒前	iPhone客户	这届疫情防	转发	评论	赞	2022/4/9										
21	连如今	17秒前		尸位素餐的	转发	评论	赞	2022/4/9										
22	陕西文艺	18秒前	HarmonyC	【	转发	评论	赞	2022/4/9										
23	艺能人_金	20秒前 转	微博视频号	《真不贵》	39	15	82	2022/4/9										
24	圈圈的娱乐	27秒前		疫情之后有	转发	评论	赞	2022/4/9										

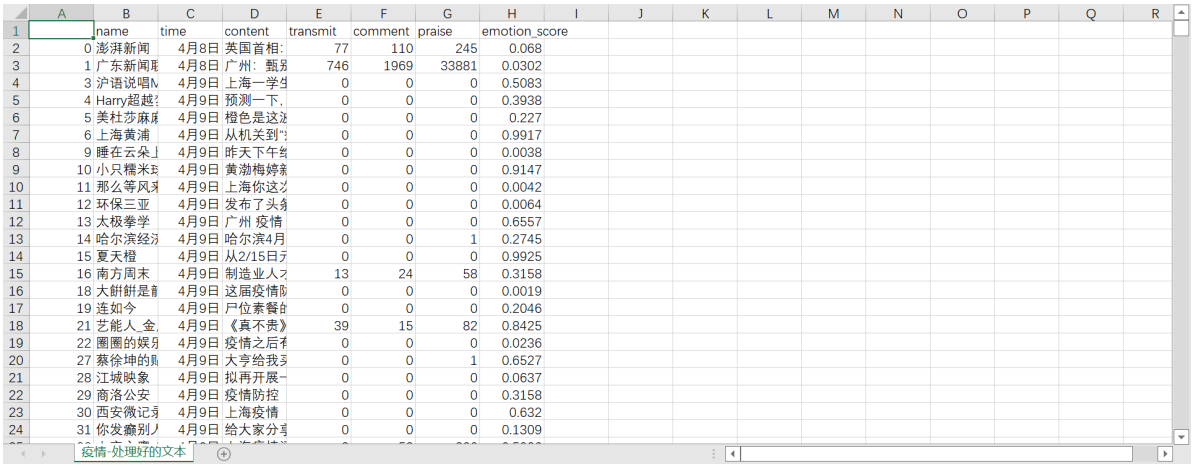
疫情关键词实时舆论

根据这些数据，我们要做一个大量的清洗步骤，这样才能确认我们后期数据分析的部分能够顺利做好

1. 我们要删除一个无效数据列，这些列并没有什么分析的价值，像第三列和最后一列其实并没有任何意义，这些就是无效列，我们要全部删除
2. 第二我们因为要分析时间趋势这些时间其实并不能准确的告诉我们时间的发生变化，所以我们还是要对其进行处理，把时间格式全部统一，并且替换成日期的形式

3. 在评论内容这边，有很多的无效字符，比如是【 这些无效字符，这些字符也会影响我们后期分析内容的结果，所以也是要删除的
4. 根据转发，评论，点赞，我们把这些中文字符转化为0
5. 再把空行全部删除
6. 最后我们进行一个情感分析的判断，去判断每个评论对应的情感分析，这里采用的是百度开源的senta_bilstm模型，这个是成熟NLP模型便于我们进行更为精准的计算，得出较为正确的分数，越接近1说明情感倾向偏正，越接近0说明情感倾向偏负

处理好的文本如下：

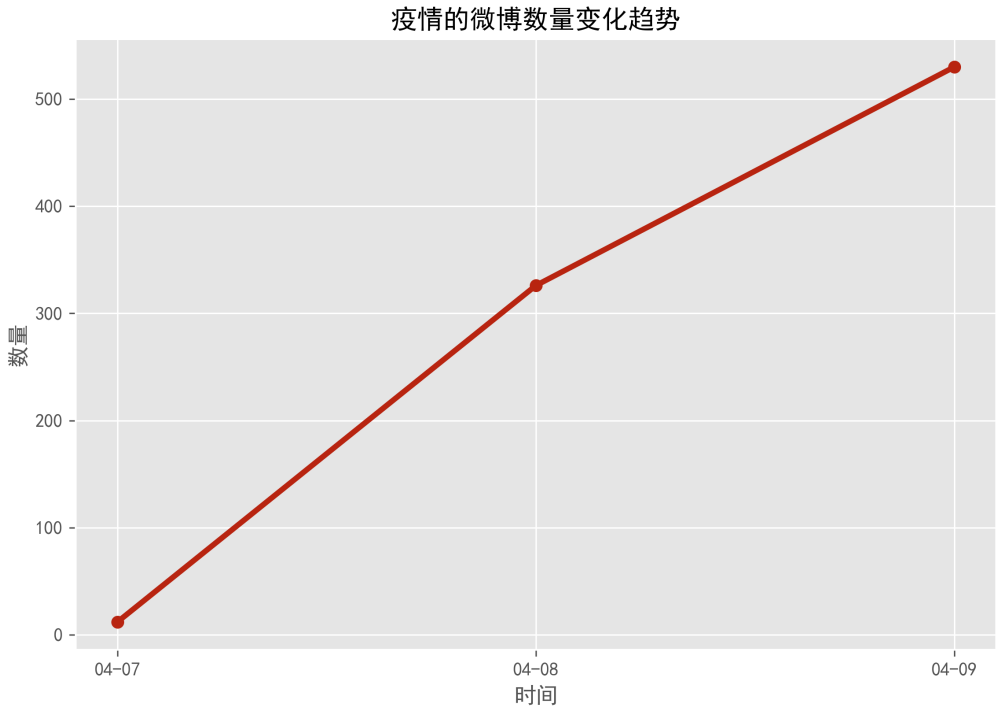


	A	B	C	D	E	F	G	H
1		name	time	content	transmit	comment	praise	emotion_score
2	0	澎湃新闻	4月8日	英国首相:	77	110	245	0.068
3	1	广东新闻联	4月8日	广州：甄别	746	1969	33881	0.0302
4	3	沪语说唱M	4月9日	上海一学生	0	0	0	0.5083
5	4	Harry超越	4月9日	预测一下，	0	0	0	0.3938
6	5	美杜莎麻麻	4月9日	橙色是这波	0	0	0	0.227
7	6	上海黄浦	4月9日	从机关到"	0	0	0	0.9917
8	9	睡在云朵上	4月9日	昨天下午给	0	0	0	0.0038
9	10	小只糯米球	4月9日	黄渤梅婷新	0	0	0	0.9147
10	11	那么等风来	4月9日	上海你这次	0	0	0	0.0042
11	12	环保三亚	4月9日	发布了头条	0	0	0	0.0064
12	13	太极拳学	4月9日	广州 疫情	0	0	0	0.6557
13	14	哈尔滨经济	4月9日	哈尔滨4月	0	0	1	0.2745
14	15	夏天橙	4月9日	从2/15日开	0	0	0	0.9925
15	16	南方周末	4月9日	制造业人才	13	24	58	0.3158
16	18	大餅餅是首	4月9日	这届疫情防	0	0	0	0.0019
17	19	连如今	4月9日	尸位素餐的	0	0	0	0.2046
18	21	艺能人_金_	4月9日	《真不贵》	39	15	82	0.8425
19	22	圈圈的娱乐	4月9日	疫情之后有	0	0	0	0.0236
20	27	蔡徐坤的鼠	4月9日	大亨给我买	0	0	1	0.6527
21	28	江城映象	4月9日	拟再开展一	0	0	0	0.0637
22	29	商洛公安	4月9日	疫情防控	0	0	0	0.3158
23	30	西安微记录	4月9日	上海疫情	0	0	0	0.632
24	31	你发癫别人	4月9日	给大家分享	0	0	0	0.1309

疫情-处理好的文本

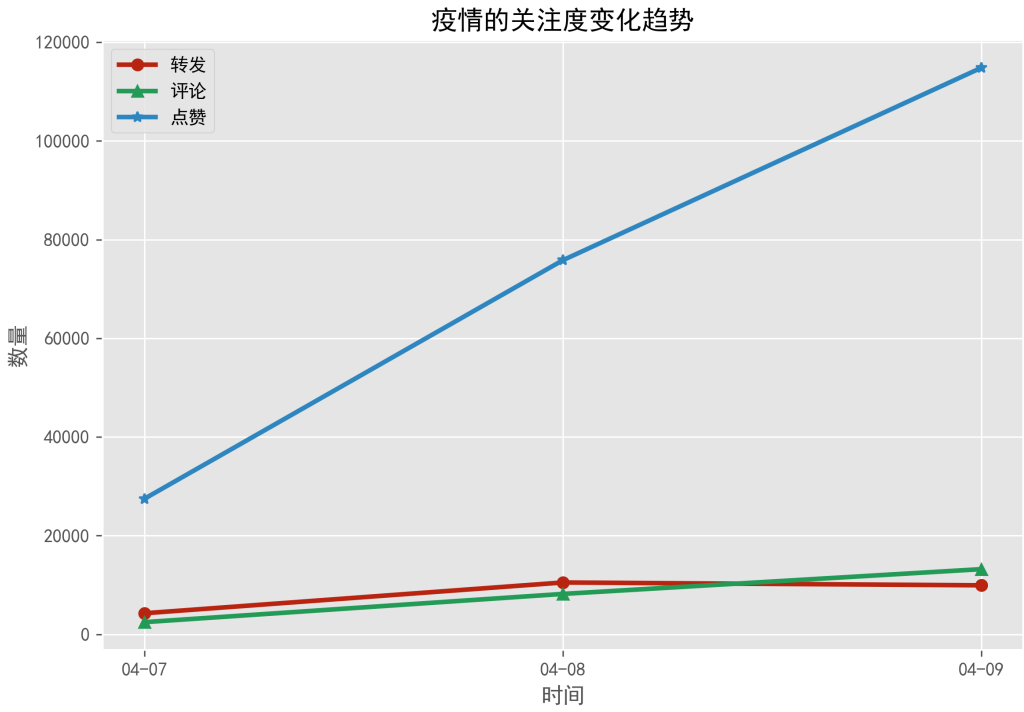
第二步部分，预警分析

首先根据这个微博数量的变化趋势，我们可以判断舆情目前的一个大致走向

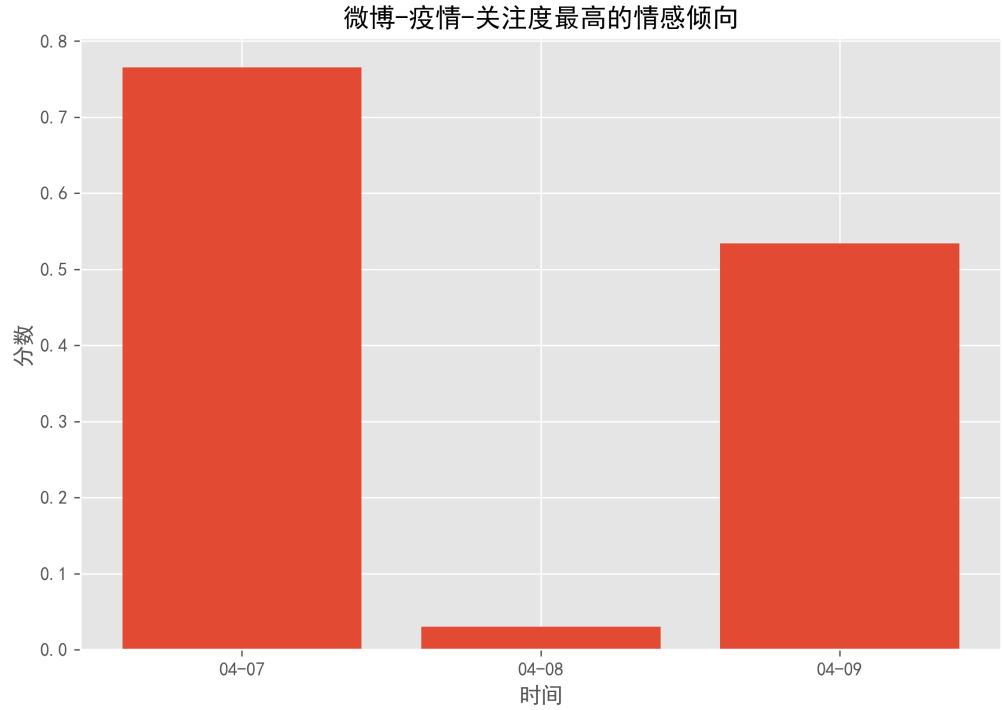


从这边我们不难看出，微博数量的变化，呈往上走的趋势，尤其是在第二天突然从几十条评论变成几百条评论，这时就已经到了我们的达到了我们的预警范围了，我们就要开始逐步对其进行一个详细的分析判断流程，去查看为什么在这一天突然变化如此之大，导致这个变化的原因是什么，是什么其他开始爆发了疫情，引起了人们激烈的讨论。

这时我们先头开始分析，首先先查看微博，人们的具体反应情况如下，单单看发帖数量不足以说明什么，还得查看人们的互动情况如何，从这些互动的情况，更能反应出，人们对疫情目前关心的程度如何



从该图看出，人们点赞率远远高于评论和转发，说明人们心里对于这些发帖人的发帖内容还是很认可他们的内容的，从而也反应出这次人们的关注程度，点赞的数量从3万多到12万左右，这些也间接说明，这个讨论的趋势越是越来越大，也达到了预警的范围，接着我们再去对后面，进行进一步探讨



这个是每天微博，关注度最高的那条微博的内容的一个情感倾向的判断，这里关注度最高是首先筛选，每天的微博，对其进行归类，然后再根据那天微博的评论内容，来判断这个是人们最关心的话题，因为单纯转发和点赞，其实并不能有一个强有力的说明，只有当人们都参与谈论了，这样才能证明这个人们最关心的话题，也是这样更容易来对其进行一个分析和判断

如图所示，这几天的情感倾向由原来正面突然转变为负面，很接近0那种，根据上面，转折点也是从第二天开始，第二天发帖数量突然暴涨，人们点赞数量也开始暴涨，说明这一天疫情的发展趋势，严重影响了人们的生活质量，导致大家都去到网上来宣泄自己的情感，这时全是负面的影响，符合了我们的预警，我们这时就要开始去探讨，人们不满的地方在哪里，去逐一分析判断，来缓解舆情所带来的负面影响

我们去查看我们的lda主题模型

这里我们先是做好了去重的处理和删除一些无效词

```
def clear_characters(text):
    return re.sub('\W', '', text)
def is_all_chinese(strs):
    for _char in strs:
        if not '\u4e00' <= _char <= '\u9fa5':
            return False
    return True

content = df['content']
content = content.drop_duplicates(keep='first')
content = content.astype(str)
content = content.apply(clear_characters)
content = content.dropna(how='any')
print(content)
```

处理好的文本也从868的数量到了706的数量，删除了很多无效的文本，让我们后续的分析变为准确

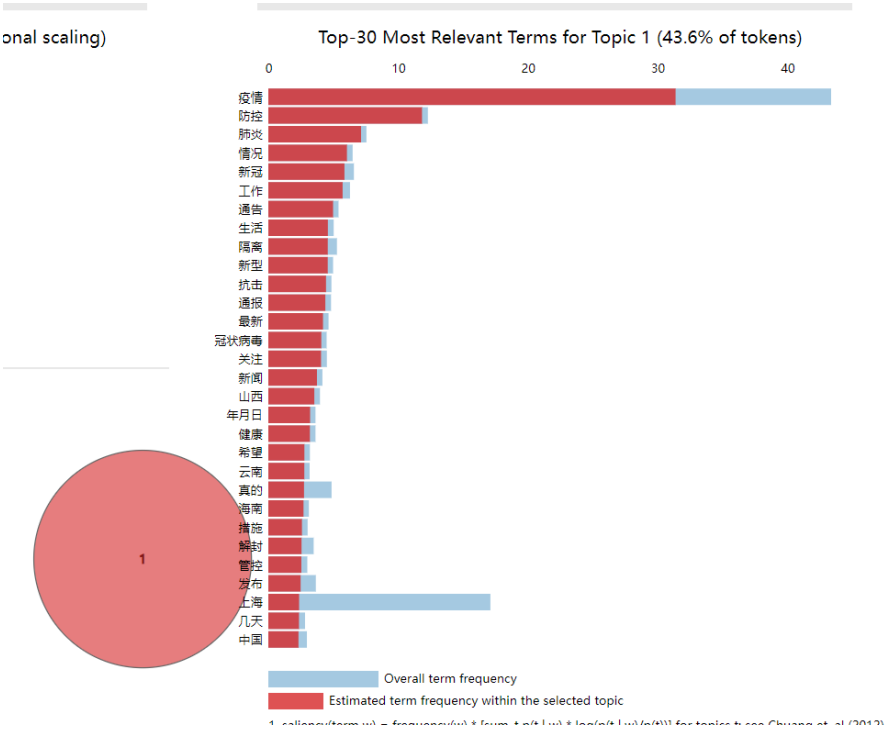
接着我们进行分词处理，并且采用tf-idf算法进行文本处理

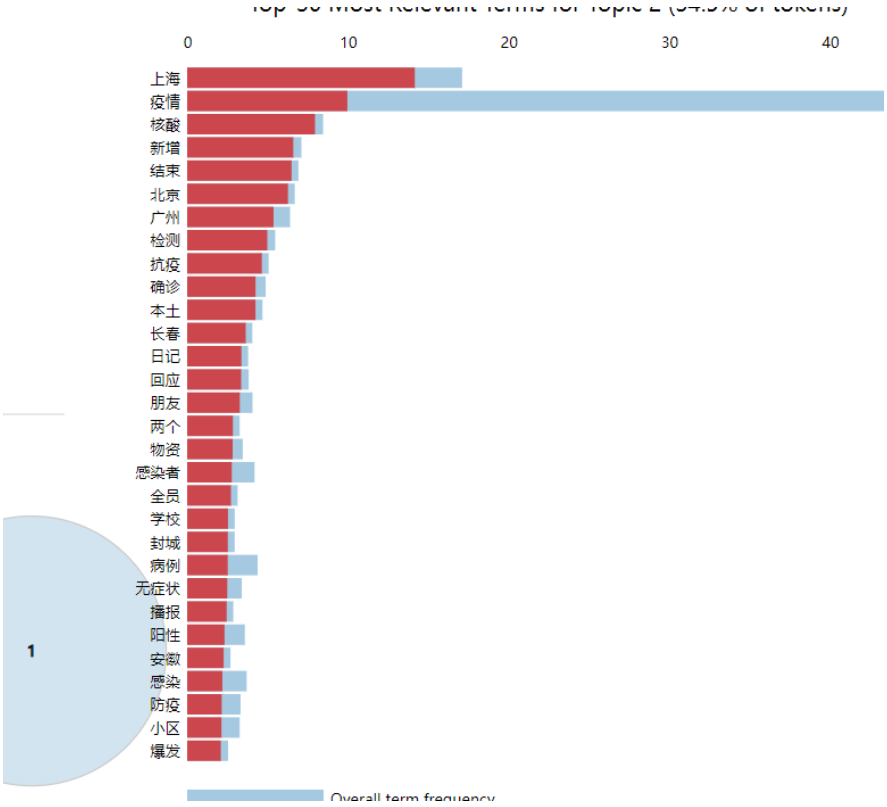
```
# 设置特征数
n_features = 2000

tf_vectorizer = TfidfVectorizer(strip_accents='unicode',
                                max_features=n_features,
                                stop_words=stop_words,
                                max_df=0.99,
                                min_df=0.002)  # 去除文档内出现几率过大或过小的词汇

tf = tf_vectorizer.fit_transform(corpus)
```

处理好的词之后我们再进行用LDA主题分析去查阅，每个主题下，哪些词影响最大

```python
# 主题-关键词分布
def print_top_words(model, tf_feature_names, n_top_words):
    for topic_idx,topic in enumerate(model.components_):  # lda.component相当于model.topic_word_
        print('Topic #%d:' % topic_idx)
        print(' '.join([tf_feature_names[i] for i in topic.argsort()[:-n_top_words-1:-1]]))
        print("")

# 定义好函数之后 暂定每个主题输出前20个关键词
n_top_words = 20
tf_feature_names = tf_vectorizer.get_feature_names()
# 调用函数
print_top_words(lda, tf_feature_names, n_top_words)

data = pyLDAvis.sklearn.prepare(lda,tf,tf_vectorizer)
print(data)
```

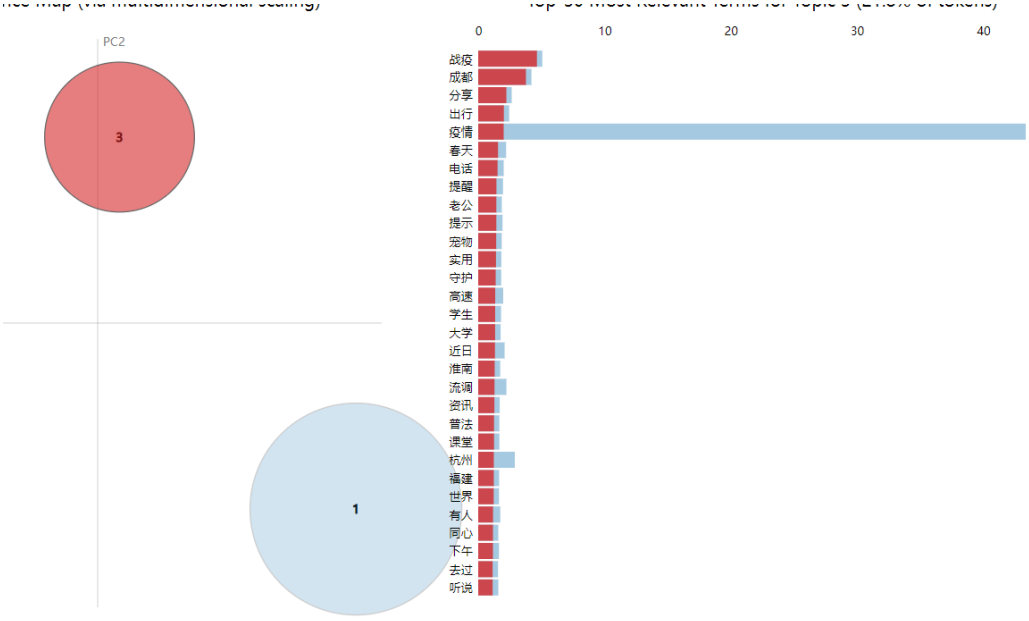
在我们的主题一中，主要是围绕整个疫情，防控，肺炎这些来进行一个探讨，这是一个大的主题





在主题二中，则是围绕着上海，北京，广州，这些个个地区的防疫情况如何，

而主题三中，则是人们的一些日常生活，比如出行，电话，春天这种，这个也说明疫情对人们造成的生活影响情况如何



第三步部分，行为分析

在这里我们主要是去查看这些分类后的情况，把这些分类结果，去判断人的一个具体行为

这里我们采用的是k-means模型，去进行一个聚类分析

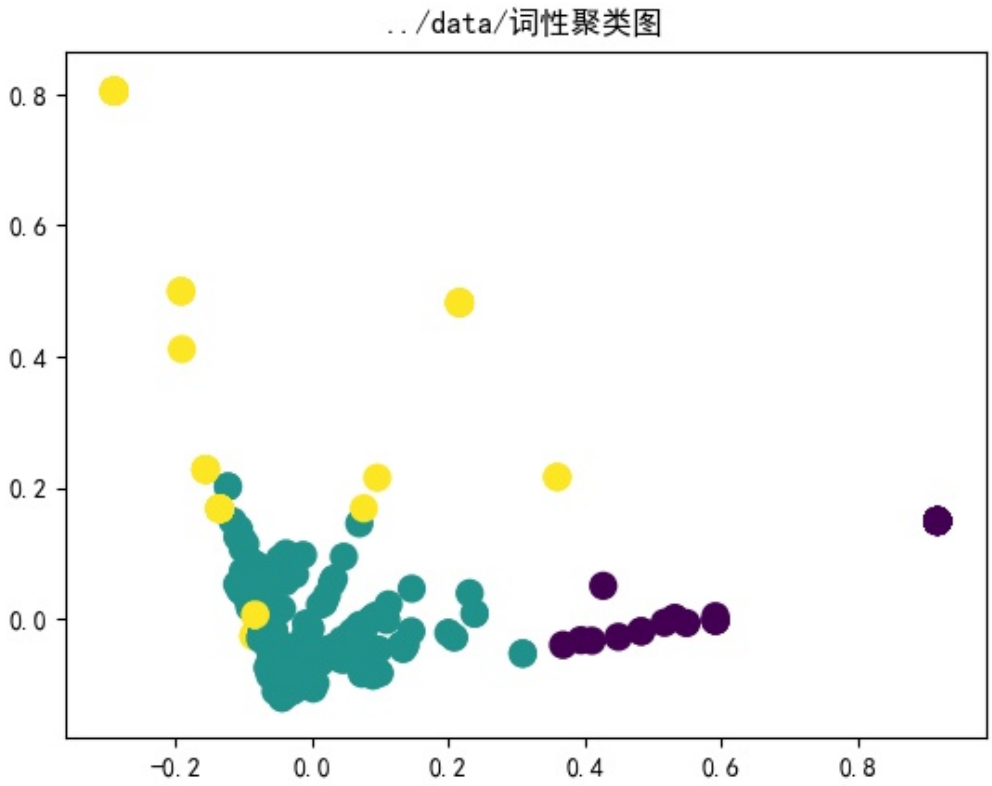
首先还是使用我们的tf-idf算法去计算每个词的权重

```
for line in open('../data/fenci.txt', 'r',encoding='utf-8').readlines():
    corpus.append(line.strip())
# 将文本中的词语转换为词频矩阵 矩阵元素a[i][j] 表示j词在i类文本下的词频
vectorizer = CountVectorizer()

# 该类会统计每个词语的tf-idf权值
transformer = TfidfTransformer()

# 第一个fit_transform是计算tf-idf 第二个fit_transform是将文本转为词频矩阵
tfidf = transformer.fit_transform(vectorizer.fit_transform(corpus))
# 获取词袋模型中的所有词语
word = vectorizer.get_feature_names()

# 将tf-idf矩阵抽取出来 元素w[i][j]表示j词在i类文本中的tf-idf权重
weight = tfidf.toarray()

# 打印特征向量文本内容
```

接着再根据这些权重处理好的值

```
print('Start Kmeans:')
from sklearn.cluster import KMeans

clf = KMeans(n_clusters=3)
print(clf)
pre = clf.fit_predict(weight)
df = pd.read_csv('../data/疫情-处理好的文本.csv')
result = pd.concat((df, pd.DataFrame(pre)), axis=1)
result.rename({0: '聚类结果'}, axis=1, inplace=True)
result.to_csv('../data/疫情-分类后的文本.csv',encoding="utf-8-sig")
print(pre)
```

进行一个聚类效果，这边根据可视化，聚3类的效果不错



这里可以明显的看出，除了少量的黄球和绿球混在一起了，其他的都完全分开了，如果是聚成其他类，那效果反而更差，因此这些，聚三类即可

聚好的文本如下：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		Unnamed:	name	time	content	transmit	comment	praise	emotion_s	聚类结果								
2	0	0	澎湃新闻	4月8日	英国首相:	77	110	245	0.068	1								
3	1	1	广东新闻耳	4月8日	广州：甄别	746	1969	33881	0.0302	1								
4	2	3	沪语说唱M	4月9日	上海一学生	0	0	0	0.5083	1								
5	3	4	Harry超越	4月9日	预测一下，	0	0	0	0.3938	1								
6	4	5	美杜莎麻屋	4月9日	橙色是这波	0	0	0	0.227	1								
7	5	6	上海黄浦	4月9日	从机关到"	0	0	0	0.9917	1								
8	6	9	睡在云朵上	4月9日	昨天下午参	0	0	0	0.0038	1								
9	7	10	小只糯米团	4月9日	黄渤梅婷煮	0	0	0	0.9147	1								
10	8	11	那么等风来	4月9日	上海你这么	0	0	0	0.0042	1								
11	9	12	环保三亚	4月9日	发布了头条	0	0	0	0.0064	1								
12	10	13	太极拳学	4月9日	广州 疫情	0	0	0	0.6557	2								
13	11	14	哈尔滨经济	4月9日	哈尔滨4月	0	0	1	0.2745	1								
14	12	15	夏天橙	4月9日	从2/15日开	0	0	0	0.9925	1								
15	13	16	南方周末	4月9日	制造业人才	13	24	58	0.3158	1								
16	14	18	大餅餅是餅	4月9日	这届疫情防	0	0	0	0.0019	1								
17	15	19	连如今	4月9日	尸位素餐的	0	0	0	0.2046	1								
18	16	21	艺能人_金_	4月9日	《真不贵》	39	15	82	0.8425	1								
19	17	22	圈圈的娱乐	4月9日	疫情之后看	0	0	0	0.0236	1								
20	18	27	蔡徐坤的贴	4月9日	大亨给我买	0	0	1	0.6527	1								
21	19	28	江城映象	4月9日	拟再开展一	0	0	0	0.0637	1								
22	20	29	商洛公安	4月9日	疫情防控	0	0	0	0.3158	2								
23	21	30	西安微记录	4月9日	上海疫情	0	0	0	0.632	2								
24	22	31	你发癫别人	4月9日	给大家分享	0	0	0	0.1309	1								

疫情-分类后的文本

然后我们去探讨每一类所代表的含义是什么，这样我们也方便后续的归类问题

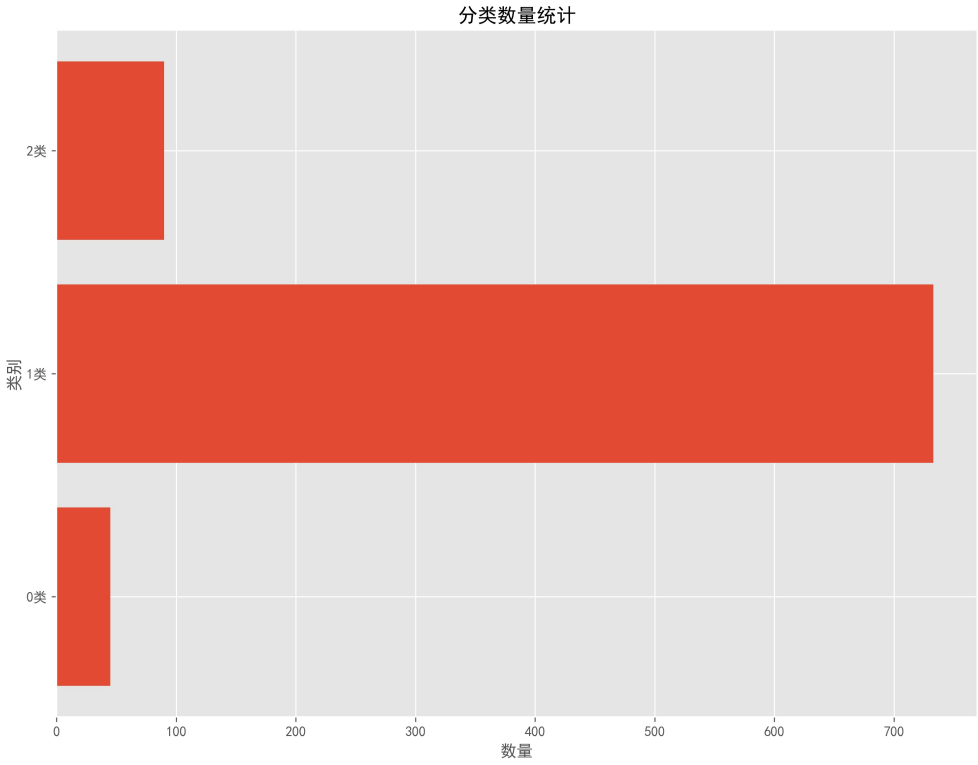
因为聚类是无监督学习的，也就是每次聚类的结果都不一样，所以我们不能把这些聚类说死，最好的归为便是聚类1，聚类2，聚类3,具体的背后含义，则是去查看它们对应的词云图去判断它们是什么

然后再去下判断

词云图一，主要则是针对上海的一个确诊情况和新增病例情况

词云图二则是上海疫情防控的情况如何

词云图三则是广州，上海，无锡，这些个个地名疫情防控的情况



这边便是它们聚类后的数量统计，其中聚类1数量是最多的，也就说目前微博讨论的主要内容则是上海疫情防控的情况如何，也符合我们的目前的形式判断。