操作步骤流程

1、首先根据获取到文档进行数据清洗

先对文章进行去重，用pandas中的drop_duplicates模块，对重复的链接进行去重，因为链接都是唯一标识，所以可以判断该文章是否存在重复，然后使用自然语言常用的模块，如re,nltk,spacy等模块，把一些标点符号，数字，无效字符全部去掉然后根据停用词表，去掉一些无意义的单词，例如is ,a,the as,这些无效的单词，以及一些与文章无关的中文单词

把上面的文本处理好之后，根据nltk库中的SentimentIntensityAnalyzer模块，做情感判断'

根据上面处理好的文章，而后进行情感判断，计算文章的复杂度，读取它的负面数值，正面数值，和中立数值，然后根据它上面的数值，进行情感归类，判断它是属于正面还是负面还是中立

并且处理好之后，重新生成新的表格

情感处理的代码

```
#开始情感判断
data['scores'] = data['new_内容'].apply(lambda commentText:
sid.polarity_scores(commentText))
#读取复杂度
data['compound'] = data['scores'].apply(lambda score_dict:
score_dict['compound'])
#读取负面
data['Negtive'] = data['scores'].apply(lambda score_dict: score_dict['neg'])
#读取正面
data['Postive'] = data['scores'].apply(lambda score_dict: score_dict['pos'])
#读取中立
data['Neutral'] = data['scores'].apply(lambda score_dict: score_dict['neu'])
#读取复杂度
data['comp_score'] = data['scores'].apply(emotional_judgment)
#对序列重新排序
new_df = data.dropna(subset=['new_内容'])
#保存最新文档
new_df.to_excel('./data/纽约日报.xlsx',encoding="utf-8-sig",index=None)
```

处理好之后，接着进行主题判断

首先采用nltk中的sent_tokenize，word_tokenize进行分词和分句

然后把分好词的用记事本的方式储存，便于后面的分析，这样可以在下次继续主题判断的之后，直接采取记事本的方式来读取，不用再去重新分词，节省运行时间，提高运行效率

分词的代码

```
def tokenize_only(text):  # 分词器，仅分词
    # 首先分句，接着分词，而标点也会作为词例存在
    tokens = [word.lower() for sent in nltk.sent_tokenize(text) for word in
nltk.word_tokenize(sent)]
    filtered_tokens = []
    # 过滤所有不含字母的词例（例如：数字、纯标点）
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    return ' '.join(filtered_tokens)
```

接着处理好相关的文本数据之后，接着开始主题模型的构建

主题模型在自然语言处理等领域是用来在一系列文档中发现抽象主题的一种统计模型。传统判断两个文档相似性的方法是通过查看两个文档共同出现的单词的多少，如TF（词频）、TF-IDF（词频-逆向文档频率）等，这种方法没有考虑到文字背后的语义关联，例如在两个文档共同出现的单词很少甚至没有，但两个文档是相似的，因此在判断文档相似性时，需要使用主题模型进行语义分析并判断文档相似性。

如果一篇文档有多个主题，则一些特定的可代表不同主题的词语会反复的出现，此时，运用主题模型，能够发现文本中使用词语的规律，并且把规律相似的文本联系到一起，以寻求非结构化的文本集中的有用信息。例如热水器的商品评论文本数据，代表热水器特征的词语如“安装”“出水量”“服务”等会频繁地出现在评论中，运用主题模型，把热水器代表性特征相关的情感描述性词语与应的特征词语联系起来，从而深入了解用户对热水器的关注点及用户对于某一特征的情感倾向

LDA主题模型

潜在狄利克雷分配，即LDA模型（Latent Dirichlet Allocation，LDA）是由Blei等人在2003年提出的生成式主题模型[⑱ Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3:2003.]⑱。生成模型，即认为每一篇文档的每一个词都是通过“一定的概率选择了某个主题，并从这个主题中以一定的概率选择了某个词语”。LDA模型也被称为三层贝叶斯概率模型，包含文档（d）、主题（z）、词（w）三层结构，能够有效对文本进行建模，和传统的空间向量模型（VSM）相比，增加了概率的信息。通过LDA主题模型，能够挖掘数据集中的潜在主题，进而分析数据集的集中关注点及其相关特征词。

LDA模型采用词袋模型（Bag Of Words，BOW）将每一篇文档视为一个词频向量，从而将文本信息转化为易于建模的数字信息。

定义词表大小为L，一个L维向量(1,0,0,...,0,0)表示一个词。由N个词构成的评论记为。假设某一商品的评论集D由M篇评论构成，记为。M篇评论分布着K个主题，记为。记a和b为狄利克雷函数的先验参数，q为主题在文档中的多项分布的参数，其服从超参数为a的Dirichlet先验分布，f为词在主题中的多项分布的参数，其服从超参数b的Dirichlet先验分布。LDA模型图如图所示。

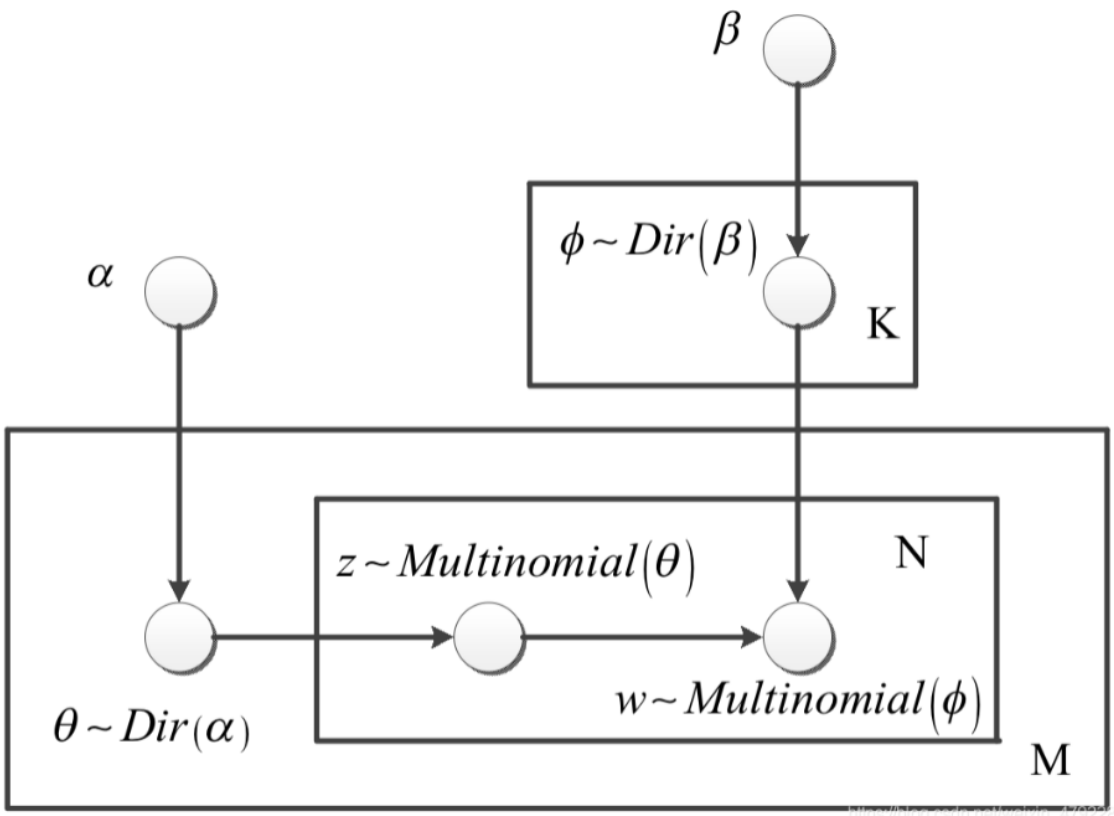


图6 LDA模型结构示意图

LDA模型假定每篇评论由各个主题按一定比例随机混合而成，混合比例服从多项分布，记为式（1）。

$$z \mid \theta = \mathrm{Multinomial}(\theta) \quad (1)$$

而每个主题由词汇表中的各个词语按一定比例混合而成，混合比例也服从多项分布，记为式（2）。

$$w \mid z, \phi = \mathrm{Multinomial}(\phi) \quad (2)$$

在评论dj条件下生成词wi的概率表示为式（3）。

$$P(w_j \mid d_j) = \sum_{s=1}^{K} P(w_i \mid z = s) \times P(z = s \mid d_j) \quad (3)$$

其中，$P(w_i \mid z = s)$表示词w_i属于第s个主题的概率，表示第s个主题在评论d_j中的概率。

其中，P（wi | z=s）表示词w i w*{i}wi属于第s个主题的概率，表示第s个主题在评论d j d*{j}d j中的概率。

LDA主题模型是一种无监督的模式，只需要提供训练文档，它就可以自动训练出各种概率，无需任何人工标注过程，节省大量人力及时间。它在文本聚类、主题分析、相似度计算等方面都有广泛的应用，相对于其他主题模型，其引入了狄利克雷先验知识。因此，模型的泛化能力较强，不易出现过拟合现象。

LDA主题模型可以解决多种指代问题，例如：在热水器的评论中，根据分词的一般规则，经过分词的语句会将“费用”一词单独分割出来，而“费用”是指安装费用，还是热水器费用等其他情况，如果简单的进行词频统计及情感分析，是无法识别的，这种指代不明的问题不能购准确的反应用户情况，运用LDA主题模型，可以求得词汇在主题中的概率分布，进而判断“费用”一词属于哪个主题，并求得属于这一主题的概率和同一主题下的其他特征词，从而解决多种指代问题。

寻找最优主题数

（2）寻找最优主题数

基于相似度的自适应最优LDA模型选择方法，确定主题数并进行主题分析。实验证明该方法可以在不需要人工调试主题数目的情况下，用相对少的迭代，找到最优的主题结构。具体步骤如下。

① 取初始主题数k值，得到初始模型，计算各主题之间的相似度（平均余弦距离）。

② 增加或减少k值，重新训练模型，再次计算各主题之间的相似度。

③ 重复步骤②直到得到最优k值。

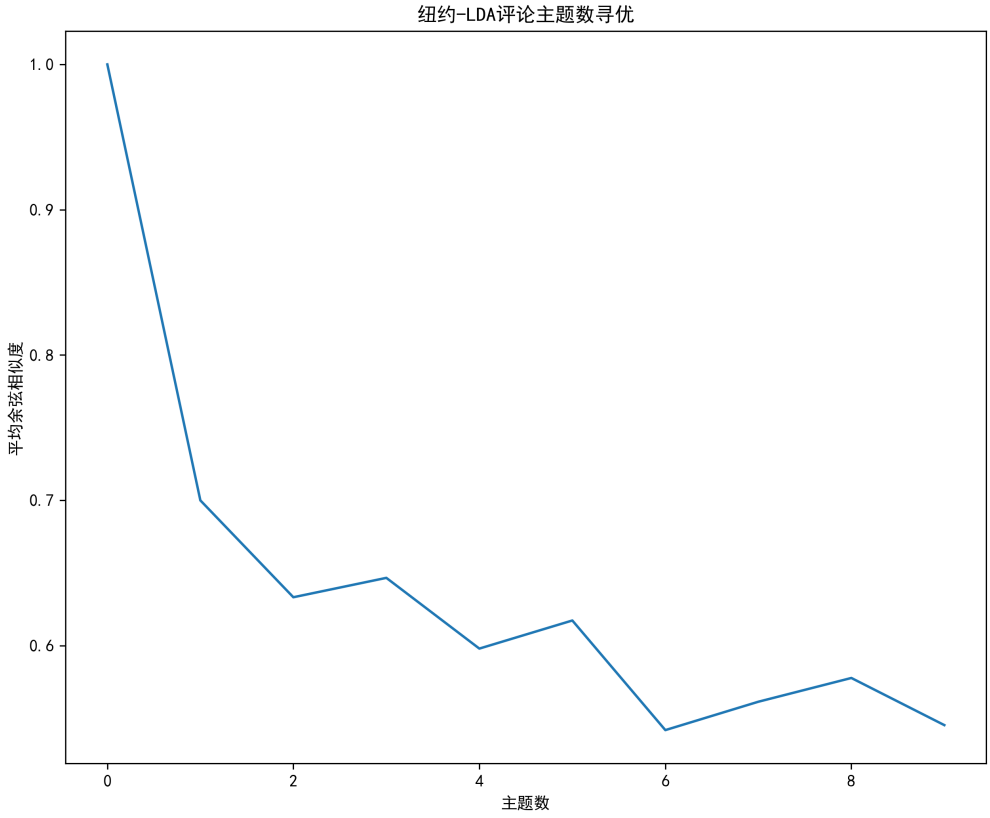
利用各主题间的余弦相似度来度量主题间的相似程度。从词频入手，计算它们的相似度，用词越相似，则内容越相近。
假定A和B是两个n维向量，A是，B是，则A与B的夹角θ的余弦值通过式（4）计算。

$$P（w_j \mid d_j）= \sum_{s=1}^{K} P（w_i|z=s）\times P(z=s|d_j)（3）$$

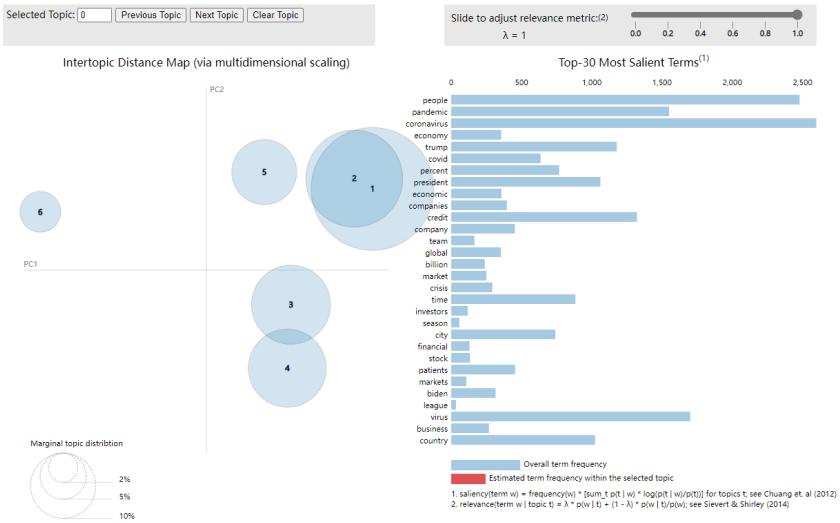
$$\cos\theta = \frac{\sum_{i=1}^{n} A_i B_i}{\sum_{i=1}^{n}(A_i)^2 \sum_{i=1}^{n}(B_i)^2} = \frac{AB}{|AB|}（4）$$

使用LDA主题模型，找出不同主题数下的主题词；每个模型各取出若干个主题词（比如前100个），合并成一个集合；生成任何两个主题间的词频向量；计算两个向量的余弦相似度，值越大就表示越相似；计算个主题数的平均余弦相似度，寻找最优主题数，如以下代码清单所示。

举例：

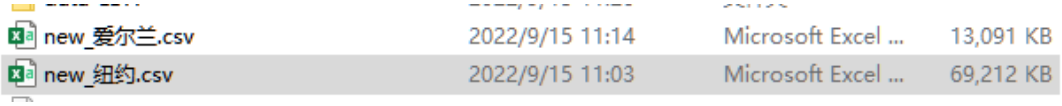


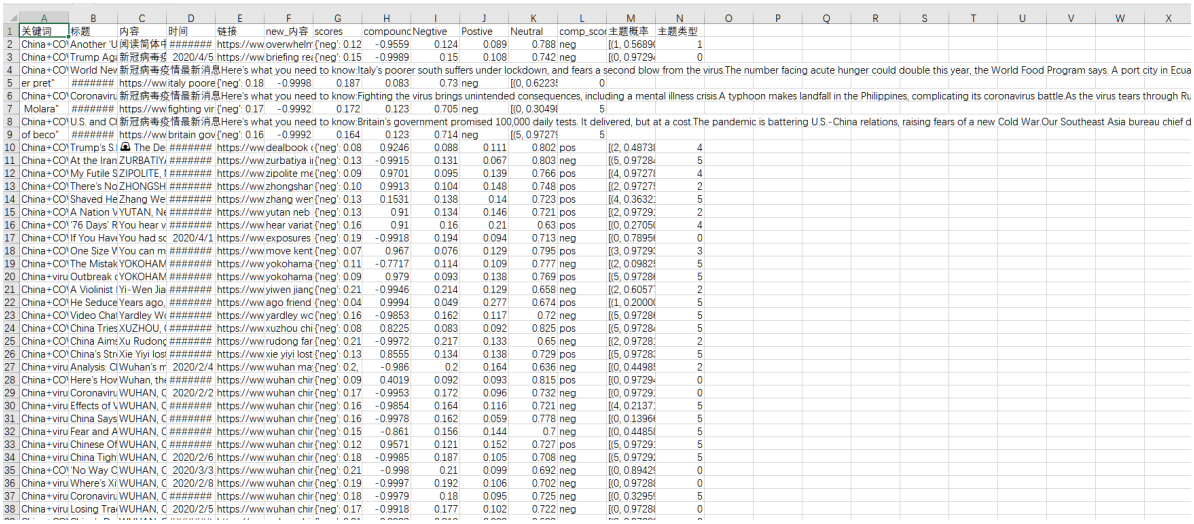
像纽约这个，它的最优主题数就是低谷的主题数，可以是4，可以是6，可以是2，实践中要勇于尝试，去尝试多个主题数，然后根据圆圈所展示的，选择最优的主题数，数量不一定是死的，要根据实际场景，灵活应用



在做好主题建模之后，就可以开始下一步的对比分析了

数据集在是这两个文件

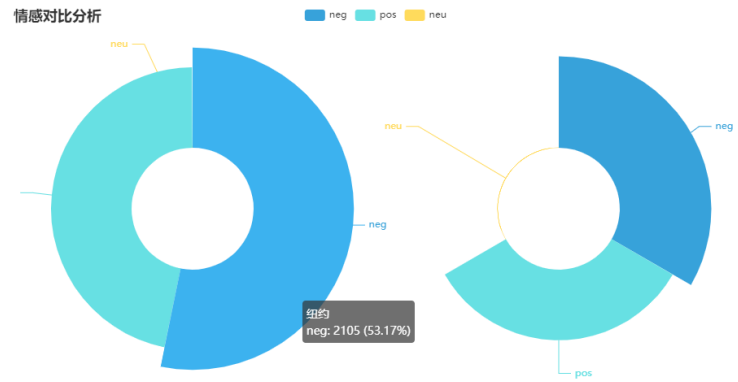




	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	关键词	标题	内容	时间	链接	new_内容	scores	compounc	Negtive	Postive	Neutral	comp_sco	主题概率	主题类型
2	China+CO	Another 'U	阅读简体中	#######	https://ww	overwhelm	{'neg': 0.12	-0.9559	0.124	0.089	0.788	neg	[(1, 0.5689(	1
3	China+CO	Trump Agi	新冠病毒疫	2020/4/5	https://ww	briefing re	{'neg': 0.15	-0.9989	0.15	0.108	0.742	neg	[(0, 0.9729(	0
4	China+CO	World Nev	新冠病毒疫情最新消息	Here's what you need to know	Italy's poorer south suffers under lockdown, and fears a second blow from the virus.The number facing acute hunger could double this year, the World Food Program says. A port city in Ecua									
5	er pret"	#######	https://ww	italy poore	{'neg': 0.18	-0.9998	0.187	0.083	0.73	neg	[(0, 0.6223(	0		
6	China+CO	Coronaviru	新冠病毒疫情最新消息	Here's what you need to know	Fighting the virus brings unintended consequences, including a mental illness crisis.A typhoon makes landfall in the Philippines, complicating its coronavirus battle.As the virus tears through Ru									
7	Molara"	#######	https://ww	fighting vir	{'neg': 0.17	-0.9992	0.172	0.123	0.705	neg	[(0, 0.3049(	5		
8	China+CO	U.S. and Cl	新冠病毒疫情最新消息	Here's what you need to know	Britain's government promised 100,000 daily tests. It delivered, but at a cost.The pandemic is battering U.S.-China relations, raising fears of a new Cold War.Our Southeast Asia bureau chief d									
9	of beco"	#######	https://ww	britain gov	{'neg': 0.16	-0.9992	0.164	0.123	0.714	neg	[(5, 0.9727(	5		
10	China+CO	Trump's S	The De	#######	https://ww	dealbook (	{'neg': 0.08	0.9246	0.088	0.111	0.802	pos	[(2, 0.4873(	4
11	China+CO	At the Iran	ZURBATIY	#######	https://ww	zurbatiya i	{'neg': 0.13	-0.9915	0.131	0.067	0.803	neg	[(5, 0.9728(	5
12	China+CO	My Futile S	ZIPOLITE, I	#######	https://ww	zipolite me	{'neg': 0.09	0.9701	0.095	0.139	0.766	pos	[(4, 0.9727(	4
13	China+CO	There's No	ZHONGSH	#######	https://ww	zhongshar	{'neg': 0.10	0.9913	0.104	0.148	0.748	pos	[(2, 0.9727(	2
14	China+CO	Shaved He	Zhang We	#######	https://ww	zhang wen	{'neg': 0.13	0.1531	0.138	0.14	0.723	pos	[(4, 0.3632(	5
15	China+CO	A Nation V	YUTAN, Ne	#######	https://ww	yutan neb	{'neg': 0.13	0.91	0.134	0.146	0.721	pos	[(2, 0.9729(	2
16	China+CO	'76 Days' R	You hear v	#######	https://ww	hear variat	{'neg': 0.16	0.91	0.16	0.21	0.63	pos	[(0, 0.2705(	4
17	China+CO	If You Have	You had sc	2020/4/1	https://ww	exposures	{'neg': 0.19	-0.9918	0.194	0.094	0.713	neg	[(0, 0.7895(	0
18	China+CO	One Size Y	You can m	#######	https://ww	move kent	{'neg': 0.07	0.967	0.076	0.129	0.795	pos	[(3, 0.9729(	3
19	China+CO	The Mistak	YOKOHAM	#######	https://ww	yokohama	{'neg': 0.11	-0.7717	0.114	0.109	0.777	neg	[(2, 0.0982(	5
20	China+viru	Outbreak (	YOKOHAM	#######	https://ww	yokohama	{'neg': 0.09	0.979	0.093	0.138	0.769	pos	[(5, 0.9728(	5
21	China+CO	A Violinist	Yi-Wen Jia	#######	https://ww	yiwen jiang	{'neg': 0.21	-0.9946	0.214	0.129	0.658	neg	[(2, 0.6057(	2
22	China+CO	He Seduce	Years ago,	#######	https://ww	ago friend	{'neg': 0.04	0.9994	0.049	0.277	0.674	pos	[(1, 0.2000(	5
23	China+CO	Video Cha	Yardley We	#######	https://ww	yardley wc	{'neg': 0.16	-0.9853	0.162	0.117	0.72	neg	[(5, 0.9728(	5
24	China+CO	China Tries	XUZHOU, t	#######	https://ww	xuzhou chi	{'neg': 0.08	0.8225	0.083	0.092	0.825	pos	[(5, 0.9728(	5
25	China+CO	China Aims	Xu Rudong	#######	https://ww	rudong far	{'neg': 0.21	-0.9972	0.217	0.133	0.65	neg	[(2, 0.9728(	2
26	China+CO	China's Str	Xie Yiyi lost	#######	https://ww	xie yiyi lost	{'neg': 0.13	0.8555	0.134	0.138	0.729	pos	[(5, 0.9728(	5
27	China+viru	Analysis: C	Wuhan's m	2020/2/4	https://ww	wuhan ma	{'neg': 0.2,	-0.986	0.2	0.164	0.636	neg	[(0, 0.4498(	2
28	China+CO	Here's How	Wuhan, the	#######	https://ww	wuhan chir	{'neg': 0.09	0.4019	0.092	0.093	0.815	pos	[(0, 0.9729(	0
29	China+viru	Coronaviru	WUHAN, C	2020/2/2	https://ww	wuhan chir	{'neg': 0.17	-0.9953	0.172	0.096	0.732	neg	[(0, 0.9729(	0
30	China+viru	Effects of \	WUHAN, C	#######	https://ww	wuhan chir	{'neg': 0.16	-0.9854	0.164	0.116	0.721	neg	[(4, 0.2137(	5
31	China+viru	China Says	WUHAN, C	#######	https://ww	wuhan chir	{'neg': 0.16	-0.9978	0.162	0.059	0.778	neg	[(0, 0.1396(	5
32	China+viru	Fear and A	WUHAN, C	#######	https://ww	wuhan chir	{'neg': 0.15	-0.861	0.156	0.144	0.7	neg	[(0, 0.4485(	5
33	China+viru	Chinese Of	WUHAN, C	#######	https://ww	wuhan chir	{'neg': 0.12	0.9571	0.121	0.152	0.727	pos	[(5, 0.9729(	5
34	China+viru	China Tigh	WUHAN, C	2020/2/6	https://ww	wuhan chir	{'neg': 0.18	-0.9985	0.187	0.105	0.708	neg	[(5, 0.9729(	5
35	China+CO	'No Way C	WUHAN, C	2020/3/3	https://ww	wuhan chir	{'neg': 0.21	-0.998	0.21	0.099	0.692	neg	[(0, 0.8942(	0
36	China+viru	Where's Xi	WUHAN, C	2020/2/8	https://ww	wuhan chir	{'neg': 0.19	-0.9997	0.192	0.106	0.702	neg	[(0, 0.9728(	0
37	China+viru	Coronaviru	WUHAN, C	#######	https://ww	wuhan chir	{'neg': 0.18	-0.9979	0.18	0.095	0.725	neg	[(0, 0.3295(	5
38	China+viru	Losing Trai	WUHAN, C	2020/2/5	https://ww	wuhan chir	{'neg': 0.17	-0.9918	0.177	0.102	0.722	neg	[(0, 0.9728(	0

有情感分类，主题类型然后我们根据月份去做判断

首先，先去做总的对比图

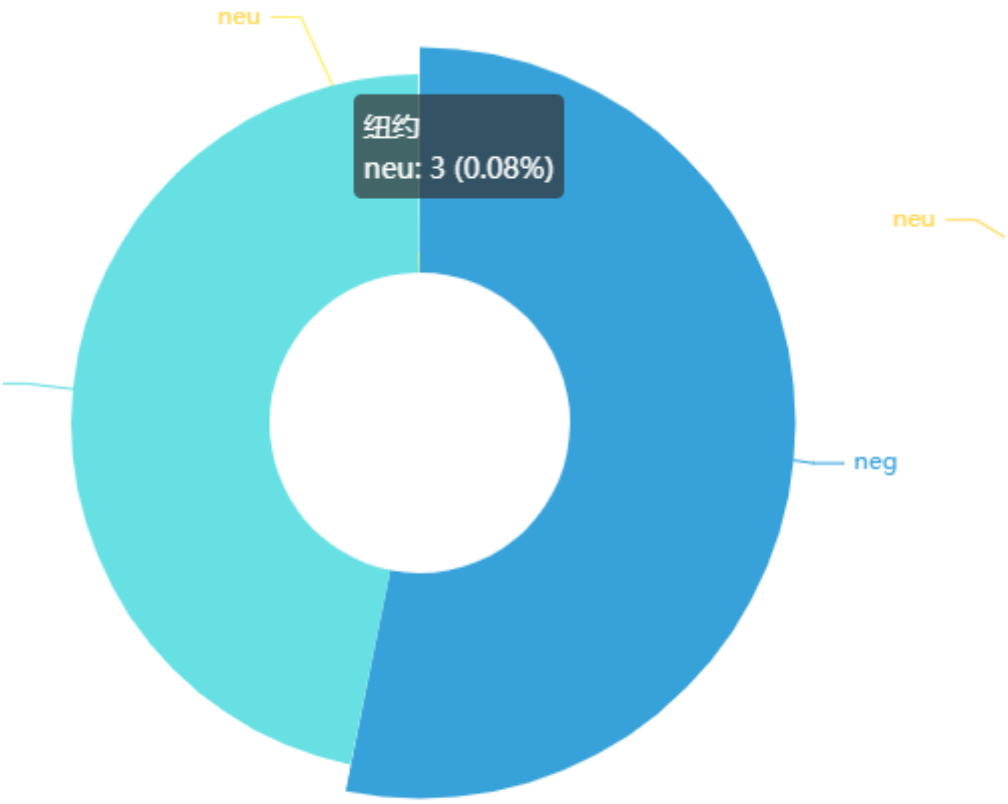


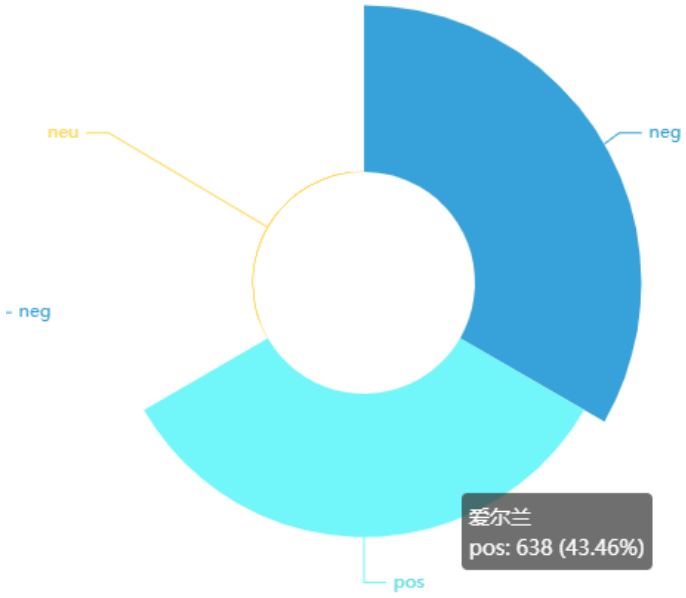
这里是情感分析的对比图，它是HTML文件，可以通过鼠标点击，展示所要查看的内容

第一个圆圈是纽约，第二个圆圈是爱尔兰

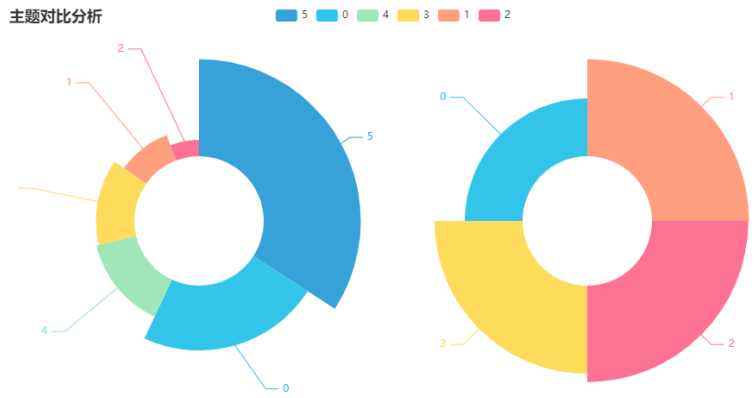
通过图像可以看出，纽约的负面占比大于正面占比，然后无中立

或者是中立的文章较少，可以忽略不计





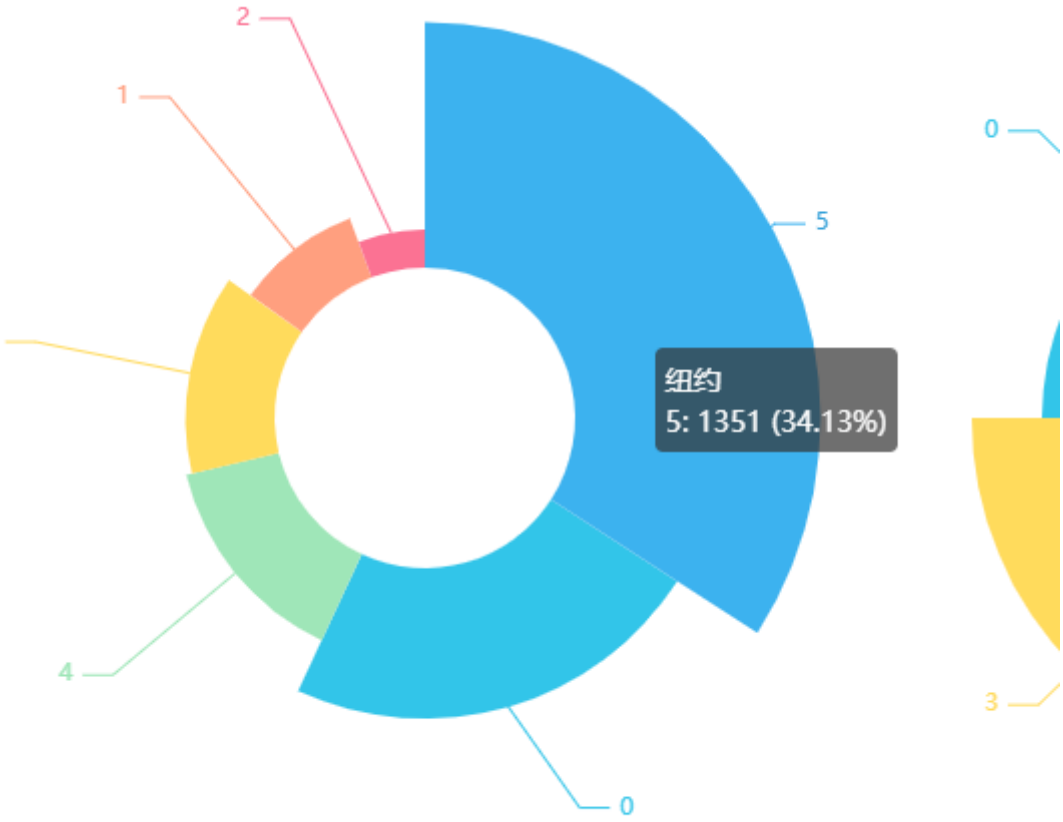
爱尔兰的负面占比同样是高于正面占比

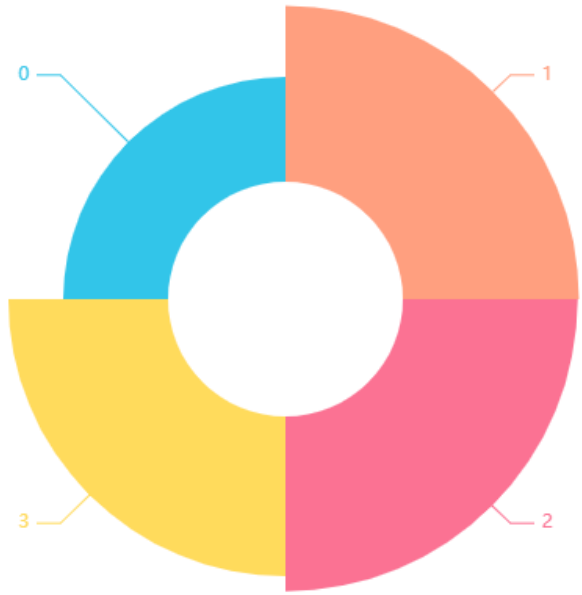


这个图，同上面的做法一样

都是通过这个圆圈来看看，主题数量的占比，

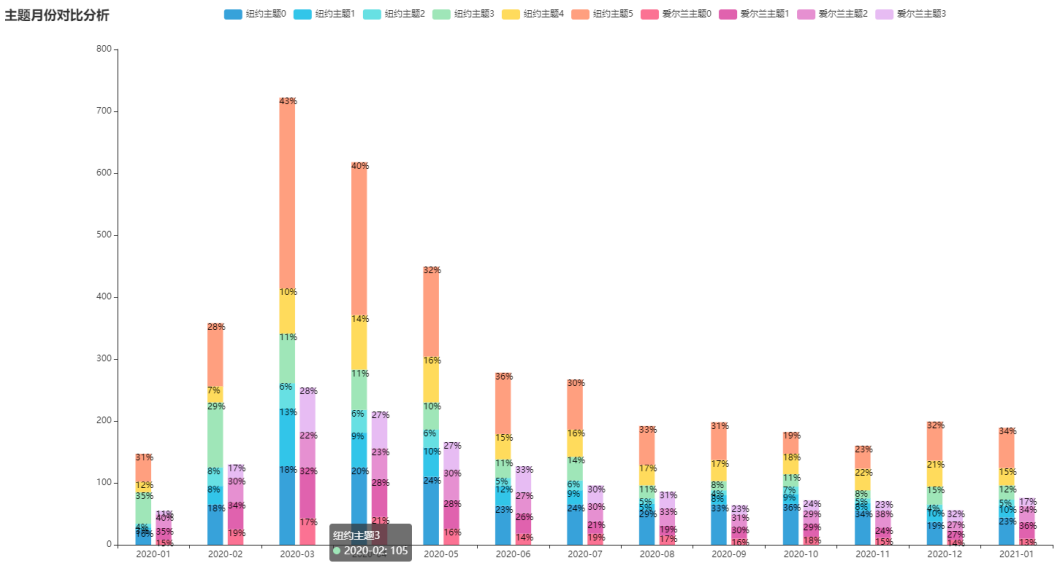
纽约的话，主题前3分别是5,0,4



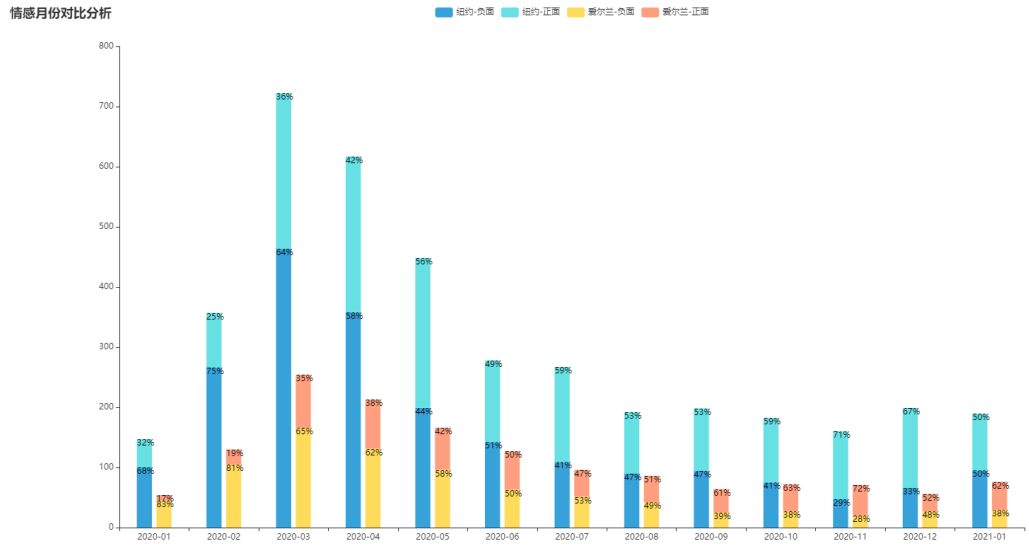


爱尔兰的话，则是 1,2，3的主题数量接近差距不是很大

月份对比的话，



主要是则是通过这种堆叠图来进行对比，查看每个月份，主题的占比数量如何，以及比例



情感也是一样，不过因为中立太少了，影响计算，所以我就把中立全部给删掉了

对整体影响并不大

上面的可视化是采用pyecharts库来做的

月份对比的主题代码如下：

```
def demo():
    def main2(x):
        df3 = x
        data = df3['主题类型'].value_counts()
        data.sort_index(inplace=True)
        x_data = list(data.index)
        y_data = list(data.values)
        d = []
        for x,y in zip(x_data,y_data):
            d1 = {
                "value": int(y), "percent": float(y / sum(y_data))
            }
            d.append(d1)
        return d
    new_df = df1.groupby('时间').apply(main2)
    x_data = list(new_df.index)
    y_data1 = []
    y_data2 = []
    y_data3 = []
    y_data4 = []
    y_data5 = []
    y_data6 = []
    for x in list(new_df.values):
        y_data1.append(x[0])
        y_data2.append(x[1])
        y_data3.append(x[2])
        y_data4.append(x[3])
        y_data5.append(x[4])
        y_data6.append(x[5])

    new_df1 = df2.groupby('时间').apply(main2)
```

```
    y_data11 = []
    y_data21 = []
    y_data31 = []
    y_data41 = []

    for x in list(new_df1.values):
        y_data11.append(x[0])
        y_data21.append(x[1])
        y_data31.append(x[2])
        y_data41.append(x[3])

    c = (
        Bar(init_opts=opts.InitOpts(width="1600px",
height="800px",theme=ThemeType.LIGHT))
            .add_xaxis(x_data)
            .add_yaxis("纽约主题0", y_data1, stack="stack1", category_gap="30%")
            .add_yaxis("纽约主题1", y_data2, stack="stack1", category_gap="30%")
            .add_yaxis("纽约主题2", y_data3, stack="stack1", category_gap="30%")
            .add_yaxis("纽约主题3", y_data4, stack="stack1", category_gap="30%")
            .add_yaxis("纽约主题4", y_data5, stack="stack1", category_gap="30%")
            .add_yaxis("纽约主题5", y_data6, stack="stack1", category_gap="30%")
            .add_yaxis("爱尔兰主题0", y_data11, stack="stack3",
category_gap="50%")
            .add_yaxis("爱尔兰主题1", y_data21, stack="stack3",
category_gap="50%")
            .add_yaxis("爱尔兰主题2", y_data31, stack="stack3",
category_gap="50%")
            .add_yaxis("爱尔兰主题3", y_data41, stack="stack3",
category_gap="50%")
            .set_global_opts(title_opts=opts.TitleOpts(title="主题月份对比分析"))
            .set_series_opts(
            label_opts=opts.LabelOpts(
                position="center",
                color='black',
                formatter=JsCode(
                    "function(x){return Number(x.data.percent * 100).toFixed() +
'%';}"
                ),
            )
        )
            .render("主题月份对比.html")
    )
```

圆形可视化的代码如下：

```
def main1():
    new_df = df1['主题类型'].value_counts()
    x_data = list(new_df.index)
    y_data = list(new_df.values)

    new_df1 = df2['主题类型'].value_counts()
    x_data1 = list(new_df1.index)
```

```
    y_data1 = list(new_df1.values)

    c = (
        Pie(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
            .add(
            "纽约",
            [(str(x),int(y)) for x,y in zip(x_data, y_data)],
            radius=["30%", "75%"],
            center=["25%", "50%"],
            rosetype="radius",

            # label_opts=opts.LabelOpts(is_show=False),
        )
            .add(
            "爱尔兰",
            [(str(x),int(y)) for x,y in zip(x_data1, y_data1)],
            radius=["30%", "75%"],
            center=["75%", "50%"],
            rosetype="area",
        )
            .set_global_opts(title_opts=opts.TitleOpts(title="主题对比分析"))
            .set_series_opts(
            tooltip_opts=opts.TooltipOpts(
                trigger="item", formatter="{a} <br/>{b}: {c} ({d}%)"
            ),
        )
            .render("主题对比分析.html")
    )
```

上述便是全部内容讲解说明了