整体思路分析报告

整个项目的思路一共分为以下几个步骤：

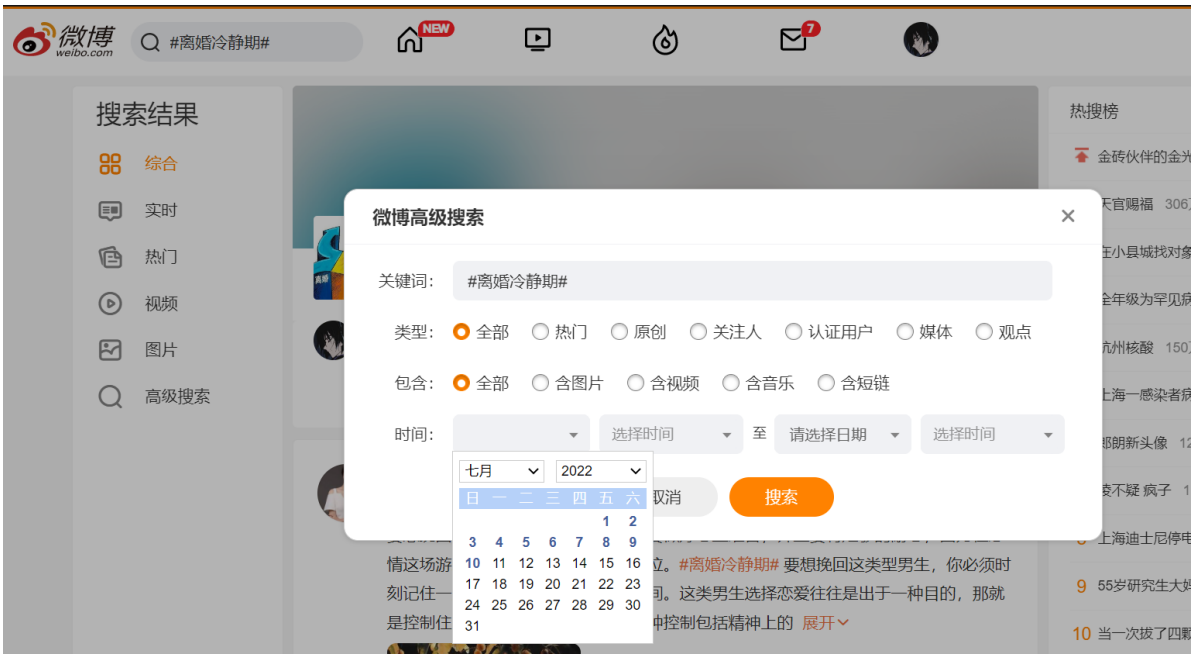
1. 先去微博获取数据
2. 对数据进行清洗，删除重复项内容
3. 对数据进行人工打标签
4. 对数据进行情感分析判断，该步骤采用百度开源NLP，飞浆NLP情感分析判断
5. 对数据进行可视化分析，分为无官号干预可视化分析，和官号干预可视化分析

获取数据方法

首先找到网页版微博，然后输入对应的标签，离婚冷静期

然后进行高级搜索

https://s.weibo.com/weibo?q=%23%E7%A6%BB%E5%A9%9A%E5%86%B7%E9%9D%99%E6%9C%9F%23&typeall=1&suball=1&



选择对应的时间，这里有一点是需要注意的，因为显示的缘故，一次最多只能显示50页

```
rng = pd.date_range(start='11/1/2020', end='2/28/2021')
list_time = []
for r in rng:
    r = str(r).split(" ")
    list_time.append(r[0])
for l in tqdm(range(len(list_time)-1)):
    for i in range(1,51,1):
        url = 'https://s.weibo.com/weibo?q=%23%E7%A6%BB%E5%A9%9A%E5%86%B7%E9%9D%99%E6%9C%9F%23&typeall=1&suball=1&timescope=custom:{}:{}&Refer=g&page={}'.format(list_time[l],list_time[l+1],i)
        main_ur(url)
```

所以程序这边设置了，去获取每一天的内容，每一天都爬取50页，确保信息收集齐全

然后根据这个时间段去爬取

这里采取的是xpath定位法

首先先定位该页面的一整个文本内容，获取每一页每一条正文发布的内容，不同页正文数量不同，所以这样自动化去定位到每一页具体的正文内容比较便利

然后再根据方框去定位，正文，时间，博主名字，博主认证信息，以及点赞，转发和评论内容

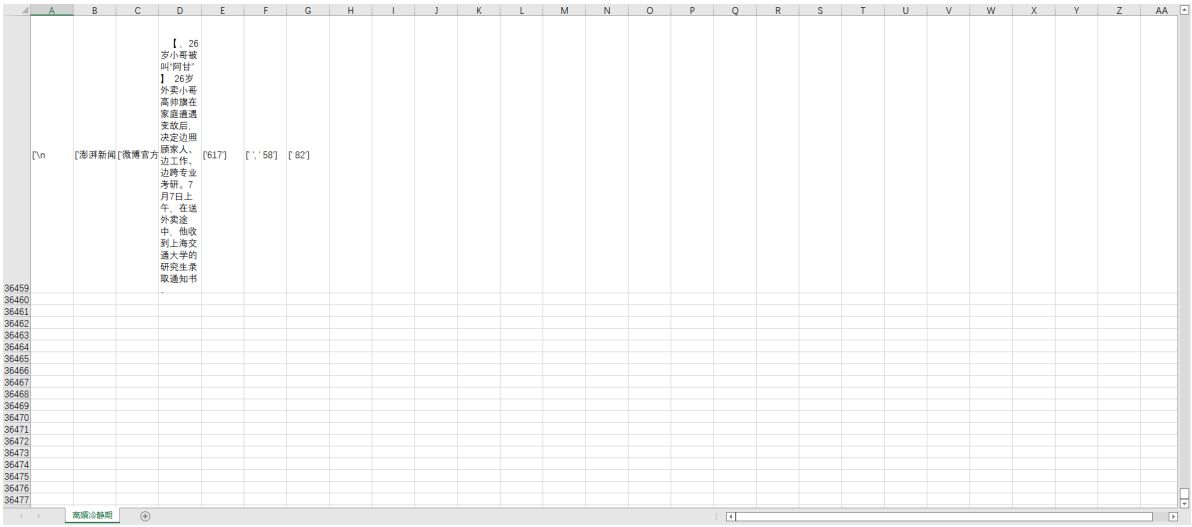
这一整块内容就是内容定位思路

```
soup = etree.HTML(content)
    node = soup.xpath('//div[@class="card-feed"]')
    act = soup.xpath('//div[@class="card-act"]/ul')
    for n,a in zip(node,act):
        name = n.xpath('./div[@node-type="like"]/div/div[2]/a/@nick-name')
        try:
            title = n.xpath('./div[@class="avator"]/a/span/@title')
        except:
            title = '无'
        timedate = n.xpath('./div[@node-
type="like"]/p[@class="from"]/a[1]/text()')
        comtent = n.xpath('./div[@node-type="like"]/p[@node-
type="feed_list_content"]/text()')
        comtent1 = ' '.join(comtent)

        dianzan = a.xpath('./li[3]/a/button/span[2]/text()')
        zhuanfa = a.xpath('./li[1]/a/text()')
        pinglun = a.xpath('./li[2]/a/text()')
```

获取好全部内容之后，保存为一个CSV文件，CSV文件名为离婚冷静期.csv

往下拉一共有3万多条文本内容，其实在获取的过程中，会发现很多内容是重复的，只是由不同账号发文，可以理解为疯狂刷热度获取水军，小号，这些并不是我们所需要的，所以下一步便是数据清洗的工作



数据清洗

数据清洗一共分8步骤

首先清洗时间列的内容

```
def main1(x):
    x1 = str(x)
    x1 = x1.replace("'",'').replace("[",'').replace("]",'').replace(" ",'').replace("\n",'')
    x1 = str(x1)
    x2 = x1.split('\\n')
    return x2[1]
```

把一些无意义的符号和回车全部删除

第二步便是清洗博主列

```
def main2(x):
    x1 = str(x)
    x1 = x1.replace("'",'').replace("[",'').replace("]",'').replace(" ",'').replace("\n",'')
    x1 = str(x1)
    x2 = x1.split('\\n')
    return x2[0]
```

同样是把无意义的符号和回车全部删除

然后其他的列同样如上

```
df['时间'] = df['时间'].apply(main1)
df['博主'] = df['博主'].apply(main2)
df['认证'] = df['认证'].apply(main3)
df['内容'] = df['内容'].apply(main4)
df['点赞'] = df['点赞'].apply(main5)
df['转发'] = df['转发'].apply(main6)
df['评论'] = df['评论'].apply(main7)
```

把一些无效的内容全部删除，保留有效数据

做好上面的工作之后，然后对内容这一列进行去重工作，把重复内容项全部删除

```
df = df.drop_duplicates(subset=['内容'],keep='first')
```

接着再对时间进行筛选

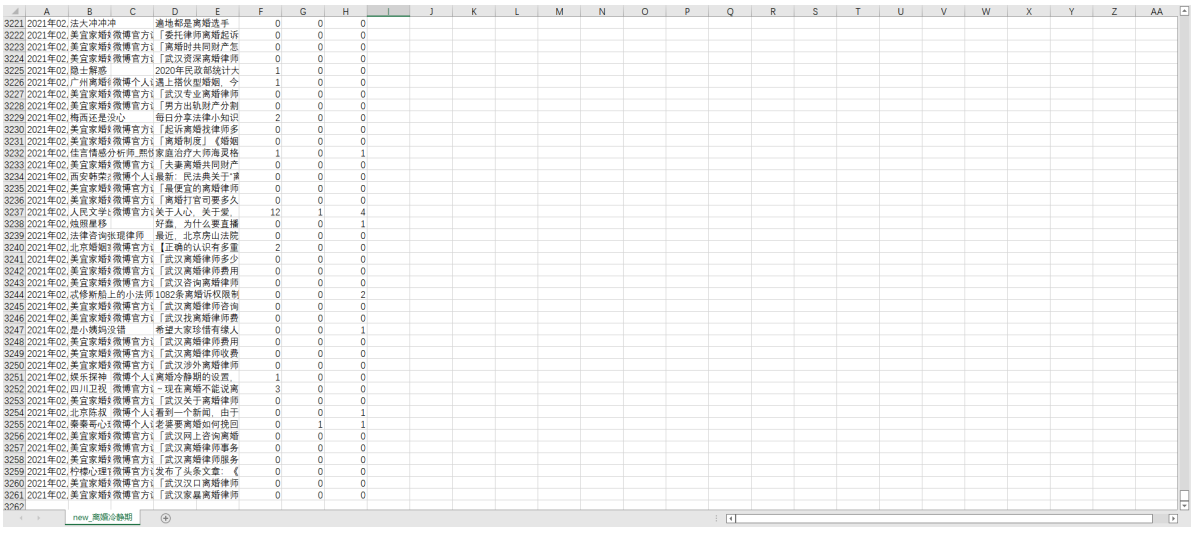
保留符合时间段的内容，时间段为2020年11月到2021年2月的全部内容，其他时间全部删除

```
def main8(x):
    x1 = str(x)
    if '2020年10月' in x1:
        return x1
    elif '2020年11月' in x1:
        return x1
    elif '2020年12月' in x1:
        return x1
    elif '2021年01月' in x1:
        return x1
    elif '2021年02月' in x1:
        return x1
    else:
        return np.NaN
```

把数据全部清洗好过后，剩下就是删除不符合逻辑的数据，和时间以及重复项内容，保存为一个新的文件

然后我们再去看新的文件大小如何

```
df['时间'] = df['时间'].apply(main8)
df1 = df.dropna(how='any',axis=0)
# df1['时间'] = df1['时间'].apply(main9)
df1.to_csv('new_离婚冷静期.csv',encoding='utf-8-sig',index=None)
```



新的文件如上，数据干净了不少，然后去完重之后，只剩3261条数据，比之前3万多条数据，缩小了10倍

说明重复内容还是很多的

数据分类

接下来就是数据打标签，这个是纯人力打标签的，也就是通过读取正文内容，来判断是属于其他，家暴，女性不平等，性别对立的哪一类，然后进行评判，由于是个人主观打标签的，不同的人有不同的理解，所以存在一定的误差

情感分析

情感分析的话，这边是采用百度成熟的开源NLP模型《senta_bilstm》去对评论内容进行情感分析打分

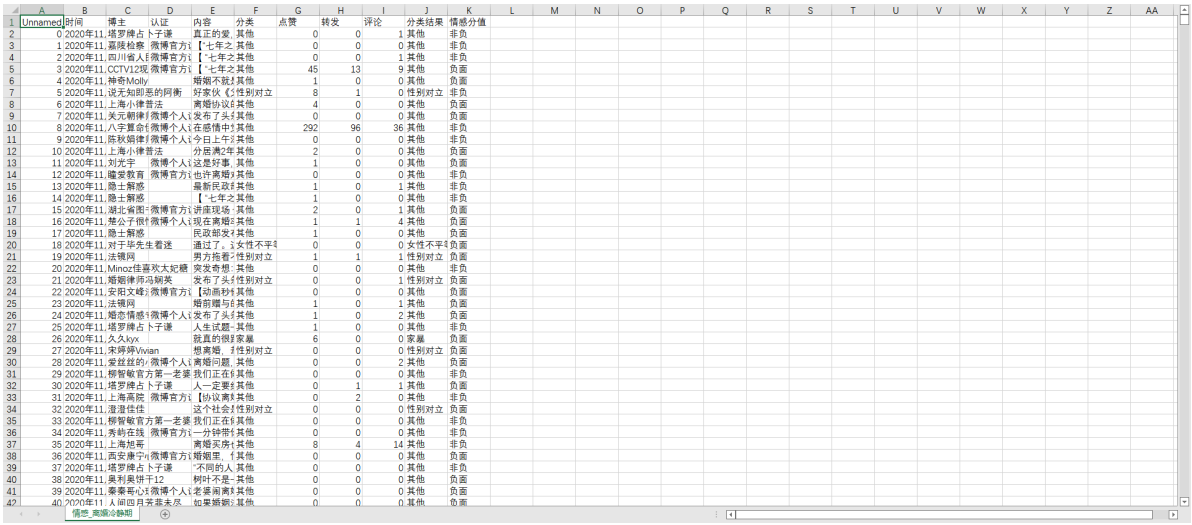
```
def tihuan(x):
    x = float(x)
    if x <= 0.3:
        return '负面'
    else:
        return '非负'

df['情感分值'] = df['情感分值'].apply(tihuan)
df.to_csv('情感_离婚冷静期.csv',index=None,encoding='utf-8-sig')
```

然后这边的话，根据情感分析，小于0.3分的则判断为负面，大于0.3的则判断为正面，

这边的分值范围是0-1范围之间，越接近1情感越接近正面，负面则反之

最后文件呈现的形式如下：



Unnamed	时间	博主	认证	内容	分类	点赞	转发	评论	分类结果	情感分值
0	2020年11	塔罗牌占卜子谦		真正的爱_	其他	0	0	1	其他	非负
1	2020年11	慕隆检察	微博官方认	【七年之	其他	0	0	0	其他	非负
2	2020年11	四川省人民	微博官方认	【七年之	其他	0	0	1	其他	非负
3	2020年11	CCTV12观	微博官方认	【七年之	其他	45	13	9	其他	负面
4	2020年11	神奇Molly		婚姻不就	其他	1	0	0	其他	负面
5	2020年11	说无知即恶的阿衡		好家伙《	性别对立	8	1	0	性别对立	非负
6	2020年11	上海小律普法		离婚协议	其他	4	0	0	其他	负面
7	2020年11	关元朝律师	微博个人认	发布了头	其他	0	0	0	其他	负面
8	2020年11	八字算命师	微博个人认	在感情中	其他	292	96	36	其他	非负
9	2020年11	陈秋娟律师	微博个人认	今日上午	其他	0	0	0	其他	非负
10	2020年11	上海小律普法		分居满2年	其他	2	0	0	其他	负面
11	2020年11	刘尤宇	微博个人认	这是好事_	其他	1	0	0	其他	负面
12	2020年11	瞻爱教育	微博官方认	也许离婚对	其他	0	0	0	其他	非负
13	2020年11	隐士解惑		最新民政	其他	1	0	1	其他	非负
14	2020年11	隐士解惑		【七年之	其他	1	0	0	其他	非负
15	2020年11	湖北省图书	微博官方认	讲座现场	其他	2	0	1	其他	负面
16	2020年11	楚公子很忙	微博个人认	现在离婚	其他	1	1	4	其他	负面
17	2020年11	隐士解惑		民政部发	其他	1	0	0	其他	负面
18	2020年11	对于毕先生看迷		通过了。	女性不平等	0	0	0	女性不平等	负面
19	2020年11	法镜网		男方拖着	性别对立	1	1	1	性别对立	负面
20	2020年11	Mmzz挂喜欢大妃糖		突发奇想	其他	0	0	0	其他	非负
21	2020年11	婚姻律师冯炯英		发布了头	性别对立	0	0	1	性别对立	负面
22	2020年11	安阳文峰区	微博官方认	【动画秒	其他	0	0	0	其他	负面
23	2020年11	法镜网		婚前赠与	其他	1	0	1	其他	负面
24	2020年11	婚恋情感	微博个人认	发布了头	其他	1	0	2	其他	负面
25	2020年11	塔罗牌占卜子谦		人生试题-	其他	1	0	0	其他	非负
26	2020年11	久久xyx		挺真的很	家暴	6	0	0	家暴	负面
27	2020年11	宋婷婷Vivian		想离婚_	性别对立	0	0	0	性别对立	负面
28	2020年11	爱丝丝的小	微博个人认	离婚问题_	其他	0	0	2	其他	负面
29	2020年11	柳智敏官方第一老婆		我们正在	其他	0	0	0	其他	非负
30	2020年11	塔罗牌占卜子谦		人一定要	其他	0	1	1	其他	负面
31	2020年11	上海高院	微博官方认	【协议离	其他	0	2	0	其他	非负
32	2020年11	星落佳佳		这个社会	性别对立	0	0	0	性别对立	负面
33	2020年11	柳智敏官方第一老婆		我们正在	其他	0	0	0	其他	非负
34	2020年11	秀屿在线	微博官方认	一分钟带	其他	0	0	0	其他	非负
35	2020年11	上海旭哥		离婚买房	其他	8	4	14	其他	非负
36	2020年11	西安康宁医	微博官方认	婚姻里_	其他	0	0	0	其他	负面
37	2020年11	塔罗牌占卜子谦		不同的人	其他	0	0	0	其他	非负
38	2020年11	曼利曼饼干12		树叶不是	其他	0	0	0	其他	负面
41	2020年11	秦霁蜀心	微博个人认	老婆闹离	其他	0	0	0	其他	负面
42	2020年11	人间四月芳菲未尽		如果婚姻	其他	0	0	0	其他	负面

情感_离婚冷静期

最后便是数据分析

这边一共分两步走，一个是无官方的推文，然后去分析，一个是全部推文然后去分析，这样形成对比分析，来查看，在没有官方干预和有官方干预的情况下，整体趋势是如何走势的

首先先去处理数据

```python
df = pd.read_csv('情感_离婚冷静期.csv')
df['数量'] = 1

def time1(x):
    x1 = str(x).split('日')
    x1 = x1[0] + '日'
    return x1

df['时间'] = df['时间'].apply(time1)
df['时间'] = pd.to_datetime(df['时间'],format='%Y年%m月%d日')
df.index = df['时间']

df = df[df['认证'] != '微博官方认证']
```

把时间序列处理好，这样好方便我们直接用时间来进行分析，这边因为时间集数量比较大，所以采用星期为周期进行统计，来查看11月到2月这段时间的一个时间走势如何

首先查看的是整体发文数量的一个时间走势，这样做的目的则是为了查看整体热度的一个情况走势，通过发文的数量表现，可以得知人们对这件事的热度是怎么样的

```python
#时间趋势
def main1():
    new_df = df['数量'].resample('W').sum()
    x_data = [str(n).split(" ")[0] for n in new_df.index]
    y_data = list(new_df.values)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9),dpi=300)
    plt.plot(x_data,y_data,color='#b82410')
    plt.title("时间热度趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/发帖热度时间趋势图.png')
    plt.show()
```

然后再去统计家暴，女性不平等，性别对立这样标签的一个走势情况，来查看在无官方干预的情况下，这些标签数量的一个变化

```python
#查看家暴的时间趋势变化
def main2():
    df1 = df[df['分类结果'] == '家暴']
    new_df = df1['数量'].resample('W').sum()
    new_df = new_df.sort_index()
    x_data = [str(n).split(" ")[0] for n in new_df.index]
    y_data = list(new_df.values)

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9), dpi=300)
    plt.plot(x_data, y_data, color='#873600',label='家暴')
    plt.legend()
    plt.title("家暴-时间热度变化趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/家暴-时间热度变化趋势.png')
```

```python
#查看性别对立的时间趋势变化
def main3():
    df2 = df[df['分类结果'] == '性别对立']
    new_df1 = df2['数量'].resample('W').sum()
    new_df1 = new_df1.sort_index()
    x_data1 = [str(n).split(" ")[0] for n in new_df1.index]
    y_data1 = list(new_df1.values)

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9), dpi=300)
    plt.plot(x_data1, y_data1, color='#117A65', label='性别对立')
    plt.legend()
    plt.title("性别对立-时间热度变化趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/性别对立-时间热度变化趋势.png')
    plt.show()
```

```python
#查看女性不平等的时间趋势变化
def main4():
    df3 = df[df['分类结果'] == '女性不平等']
    new_df2 = df3['数量'].resample('W').sum()
    new_df2 = new_df2.sort_index()
    x_data2 = [str(n).split(" ")[0] for n in new_df2.index]
    y_data2 = list(new_df2.values)

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9), dpi=300)
    plt.plot(x_data2, y_data2, color='#6C3483', label='女性不平等')
    plt.legend()
    plt.title("女性不平等-时间热度变化趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/女性不平等-时间热度变化趋势.png')
    plt.show()
```

接着再去查看人们对这件事情的一个情感走势，查看人们随着时间的变化，情感是如何变化的，是负面情绪越来越严重还是非负情感占主导地位这样

```python
#查看情感变化的时间趋势变化
def main5():
    df3 = df[df['情感分值'] == '非负']
    new_df2 = df3['数量'].resample('W').sum()
    new_df2 = new_df2.sort_index()
    x_data2 = [str(n).split(" ")[0] for n in new_df2.index]
    y_data2 = list(new_df2.values)

    df4 = df[df['情感分值'] == '负面']
    new_df3 = df4['数量'].resample('W').sum()
    new_df3 = new_df3.sort_index()
    x_data3 = [str(n).split(" ")[0] for n in new_df3.index]
    y_data3 = list(new_df3.values)

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9), dpi=300)
    plt.plot(x_data2, y_data2, color='#148F77', label='非负')
    plt.plot(x_data3[1:], y_data3[1:], color='#922B21', label='负面')
    plt.legend()
    plt.title("情感倾向-时间热度变化趋势")
```

接着在做好上面的步骤之后，我们来查看互动量趋势的一个变化情况，来了解人们对这件事件的关心程度如何，

一般这种互动量其实要综合来看，这边建议是把点赞，转发，评论，发文数量，这四张图放在一起来看，这样更能反应出这件事情整体热度的一个变化趋势

```python
#点赞趋势
def main6():
    new_df = df['点赞'].resample('W').sum()
    x_data = [str(n).split(" ")[0] for n in new_df.index]
    y_data = list(new_df.values)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9),dpi=300)
    plt.plot(x_data,y_data,color='#b82410')
    plt.title("点赞热度趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/点赞热度时间趋势图.png')
    plt.show()
```

```python
#评论趋势
def main7():
    new_df = df['评论'].resample('W').sum()
    x_data = [str(n).split(" ")[0] for n in new_df.index]
    y_data = list(new_df.values)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9),dpi=300)
    plt.plot(x_data,y_data,color='#b82410')
    plt.title("评论热度趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/评论热度时间趋势图.png')
    plt.show()
```

```python
#转发趋势
def main8():
    new_df = df['转发'].resample('W').sum()
    x_data = [str(n).split(" ")[0] for n in new_df.index]
    y_data = list(new_df.values)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(16, 9),dpi=300)
    plt.plot(x_data,y_data,color='#b82410')
    plt.title("转发热度趋势")
    plt.xlabel("时间")
    plt.ylabel("发帖数量")
    plt.xticks(rotation=65)
    plt.savefig('./数据可视化-无官方干预/转发热度时间趋势图.png')
    plt.show()
```

接着非官方分析完之后，后面便是全部推文分析，通过这样对比，更能直观反应出官方对这一整件事情的一个干预程度如何，用同一个维度分析图片，对比来看，对比分析，找出它们的数量不同的表现情况