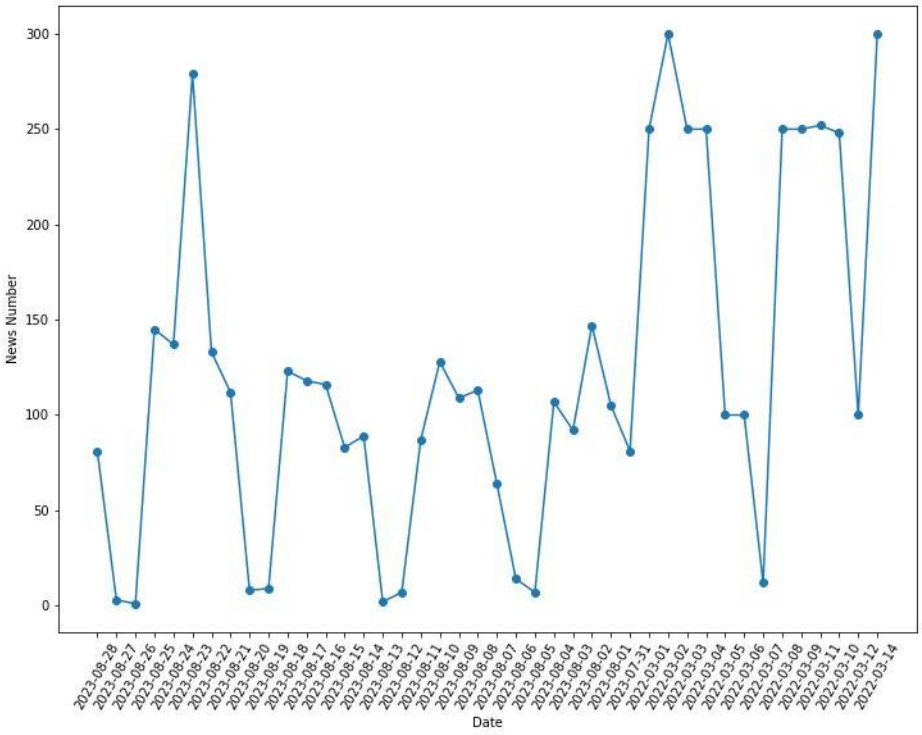
数据分析报告

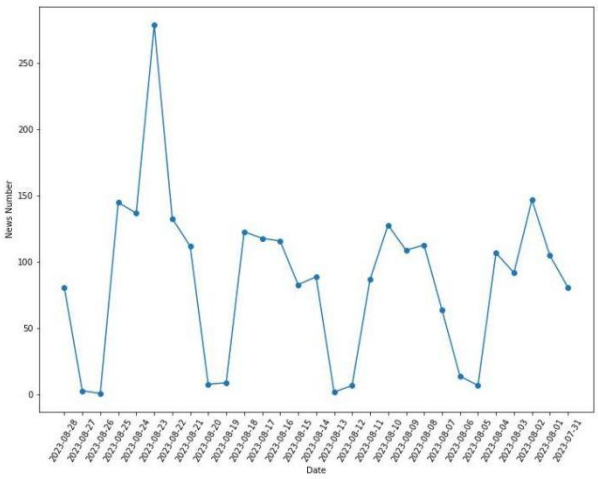
刘雅迪
计 26

注：相关代码见报告最后

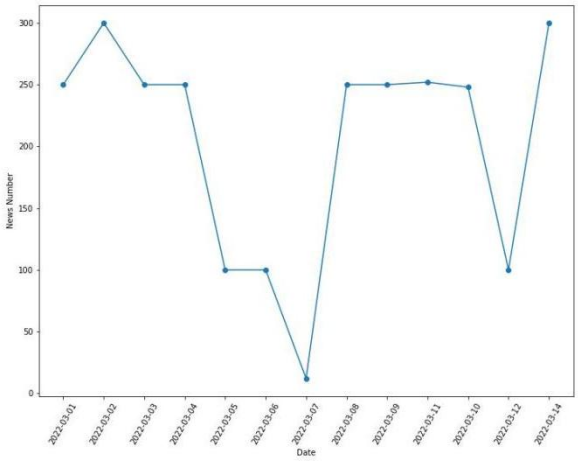
一、新闻数量和趋势分析

1. 全部时间的新闻发布数量（按天来计算）





2023 年新闻发布数量

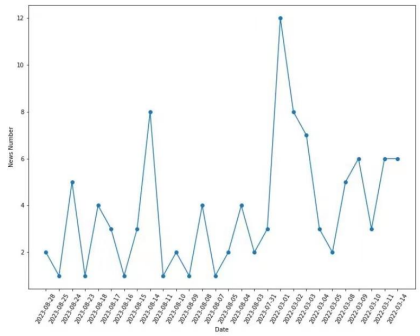


2022 年新闻发布数量

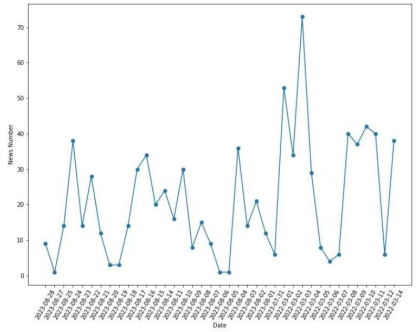
上图所示，除去某些特殊值后，按年份来看，2023 年和 2022 年每天发布的新闻数量较为稳定，且 2022 年每天发布的新闻数量比 2023 年更多。2023 年新闻发布数量日均值在 125 左右，而 2022 年则在 250 左右。

年份的不同导致了新闻发布日均值的不同说明了新闻的发布数量与时间有关，我们可以根据不同年份某月的新闻发布量推测下月的新闻发布量。而每天新闻发布数量的稳定性保证了网站的一定浏览量，同时也保证了新闻发布的时效性。

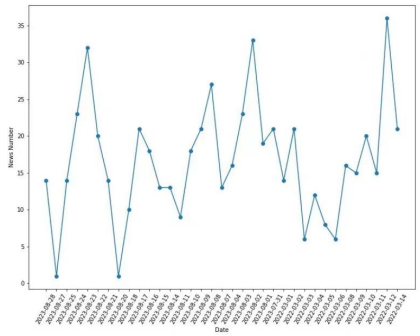
2.每一类别的新闻发布数量（按天来计算）



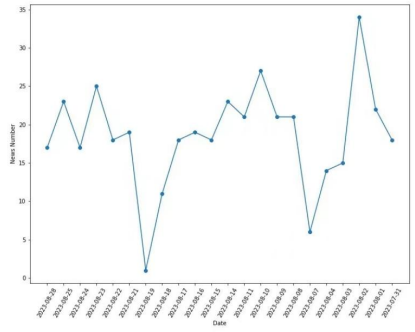
新浪数码



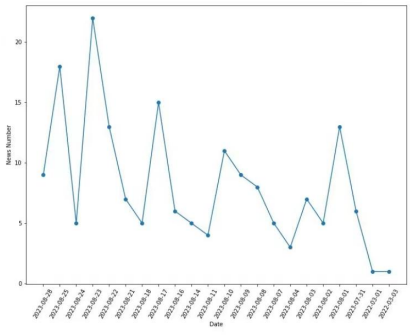
新浪科技



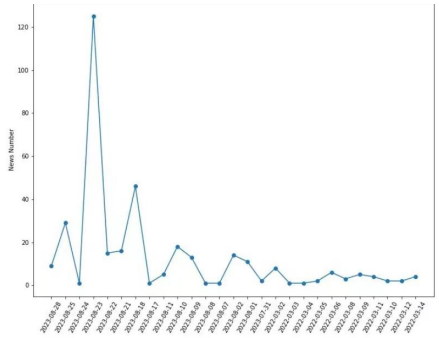
IT 之家



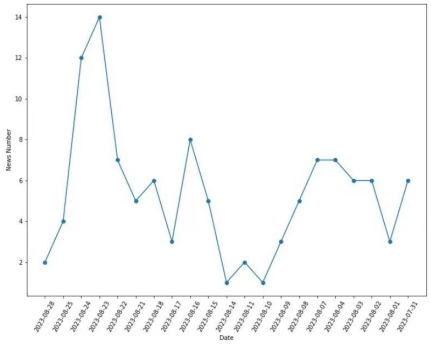
快科技



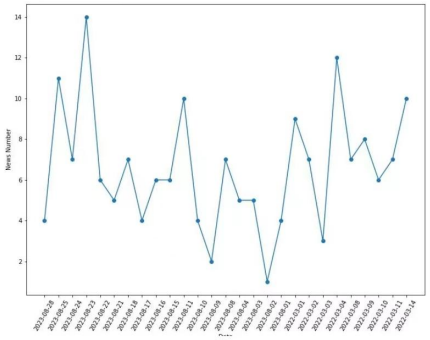
太平洋电脑网



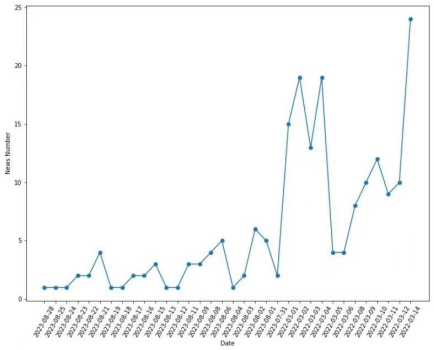
中关村在线



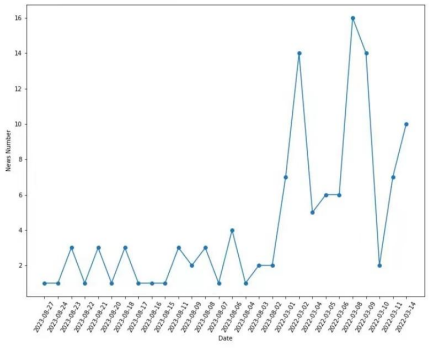
中国家电网



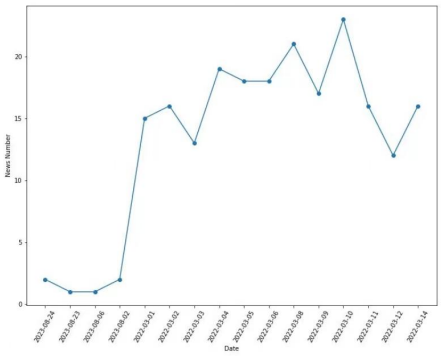
CNMO



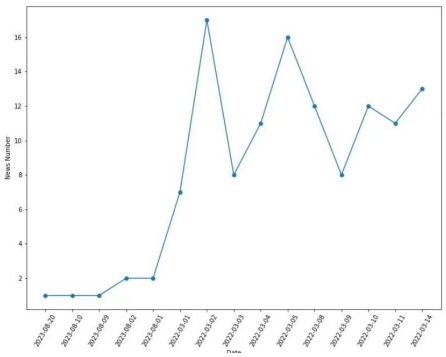
市场资讯



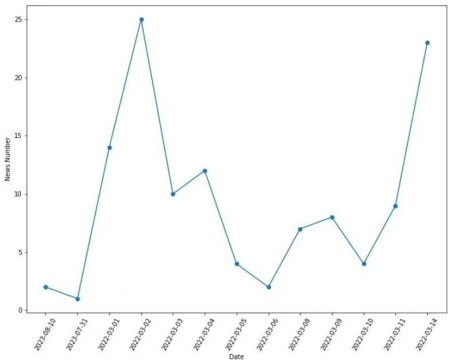
媒体滚动



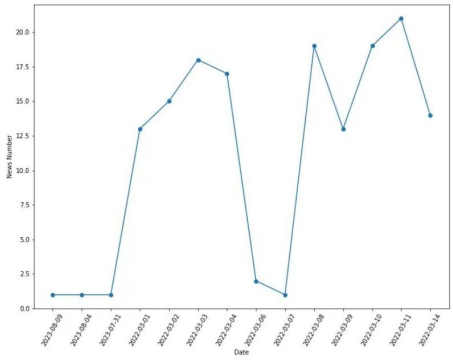
新浪科技综合



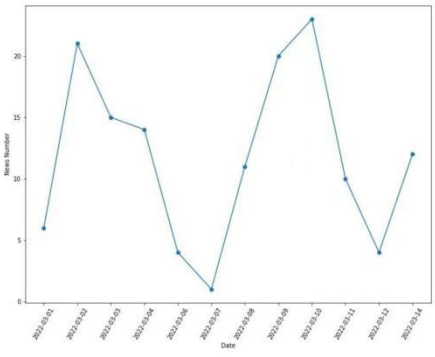
澎湃新闻



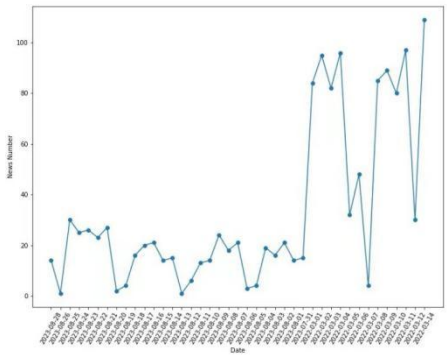
财联社



快科技 2018



界面新闻



其他

从上图中可以看出，“快科技”和“中国家电网”只在 2023 年发布过新闻，“界面新闻”只在 2022 年发布过新闻。此外大部分类别的新闻 2023 年和 2022 年均有发布，但部分类别的发布时间有所侧重。如“太平洋电脑网”和“中关村在线”类别主要在 2023 年发布

新闻，“市场资讯”“媒体滚动”“新浪科技综合”“澎湃新闻”“财联社”“快科技 2018”“其他”类别主要在 2022 年发布新闻。

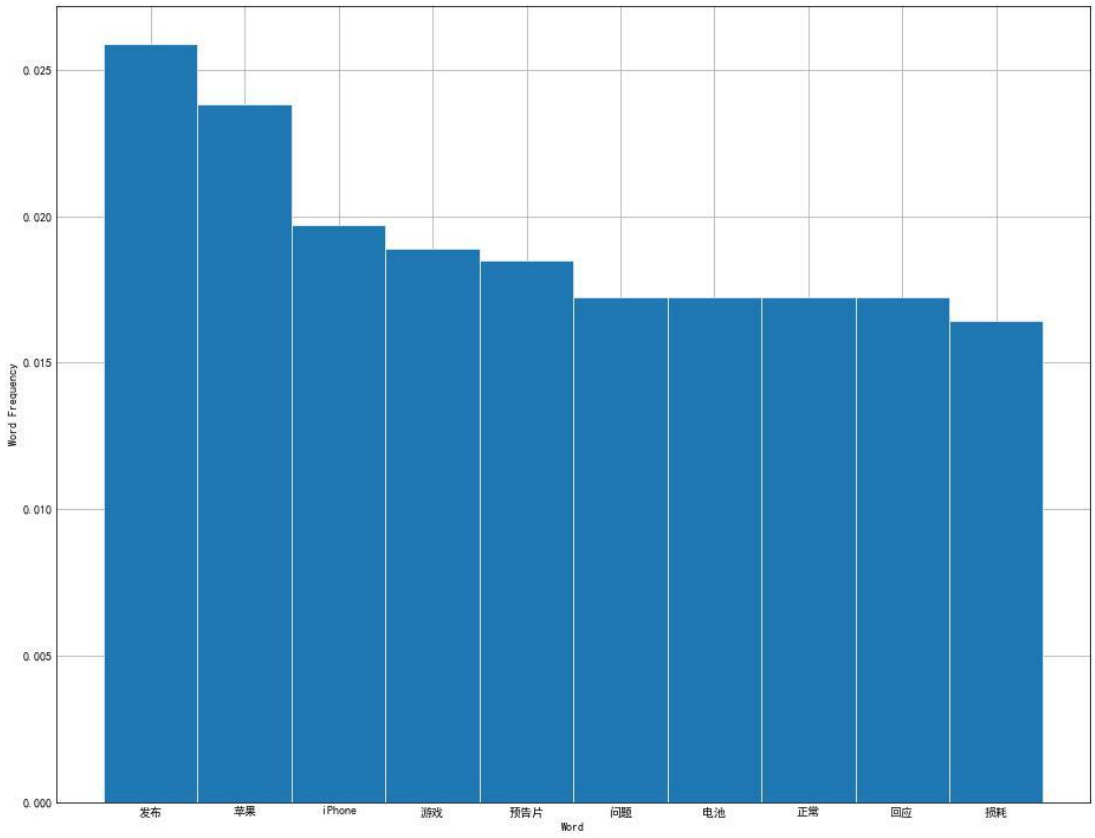
知道新闻类别的大致发布时间后有利于那些偏向于阅读某一类新闻的人查找自己想要阅读的新闻。同时对新闻类别来说，可以适当地均衡一下自己的发布时间，这样有利于获得更多的阅读量。

二、新闻关键词分析

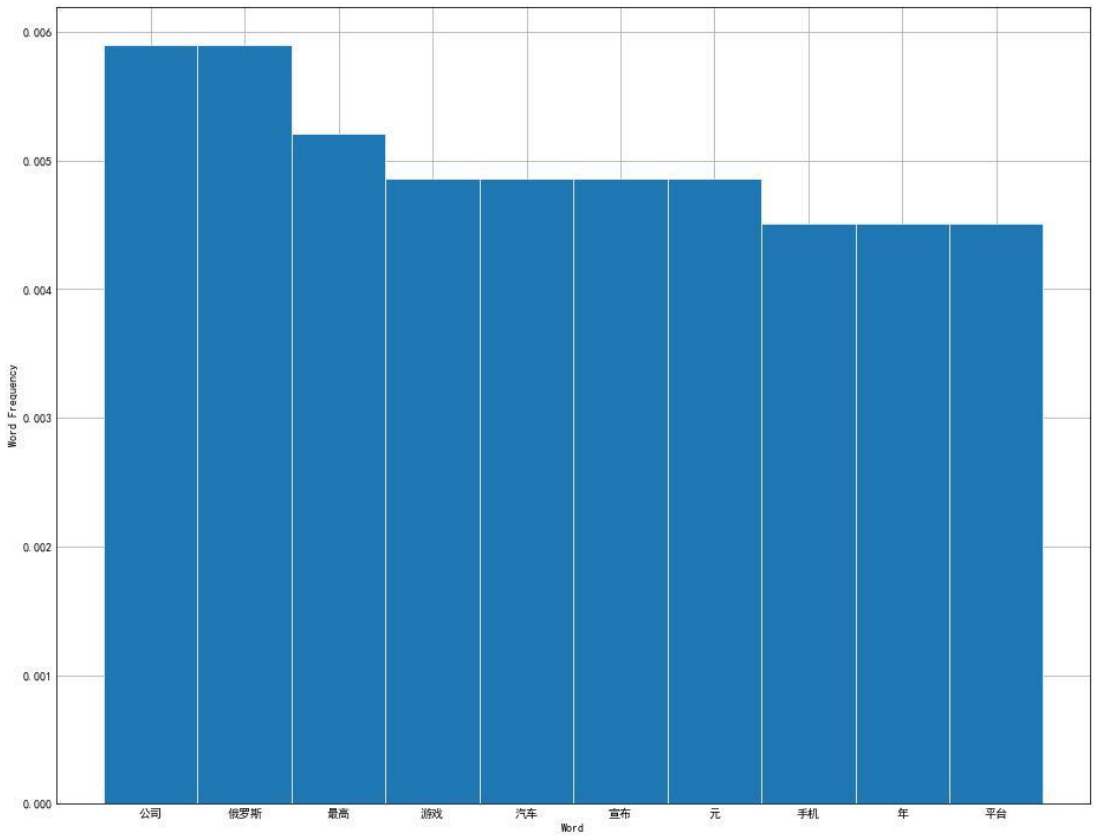
这里分析新闻较多的那天的关键词/主题。

根据总体的新闻发行量趋势，选取 2023-08-23, 2022-03-02 ,2022-03-14 三天，这三天的新闻发行量均在 300 左右，样本数较多。

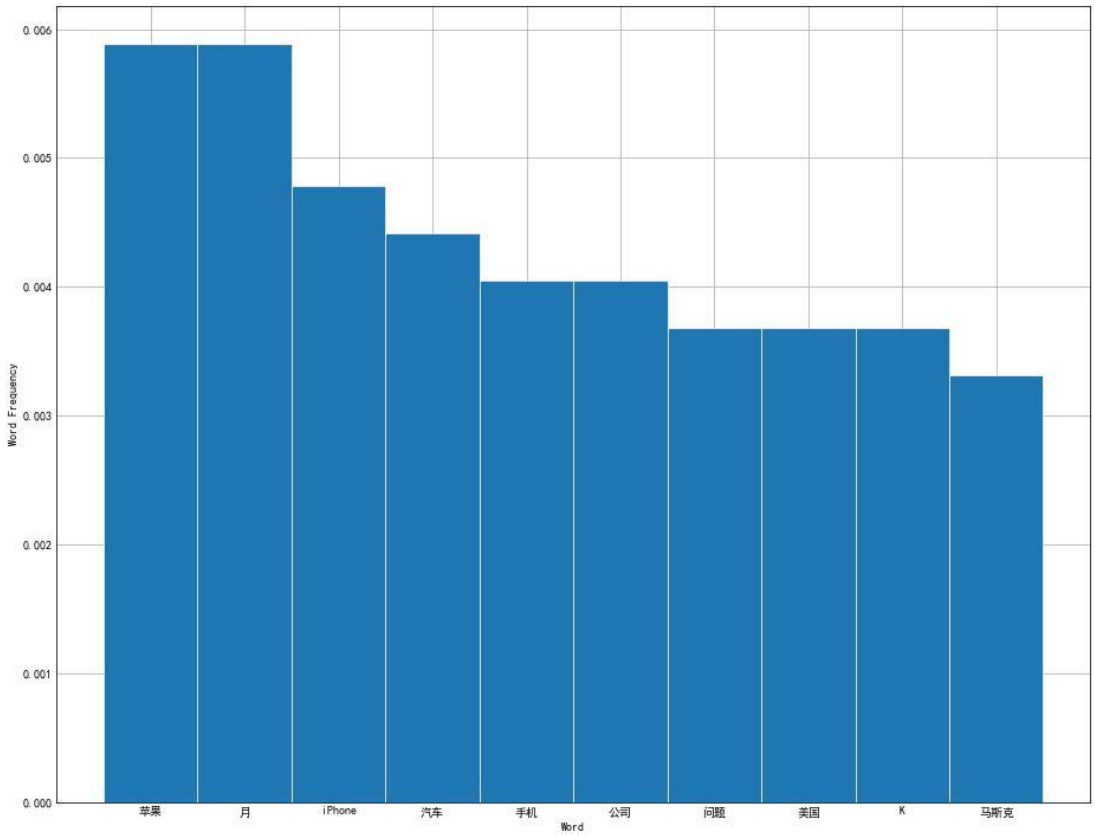
(1) 分析标题关键词



2023-08-23 标题关键词



2022-03-02 标题关键词



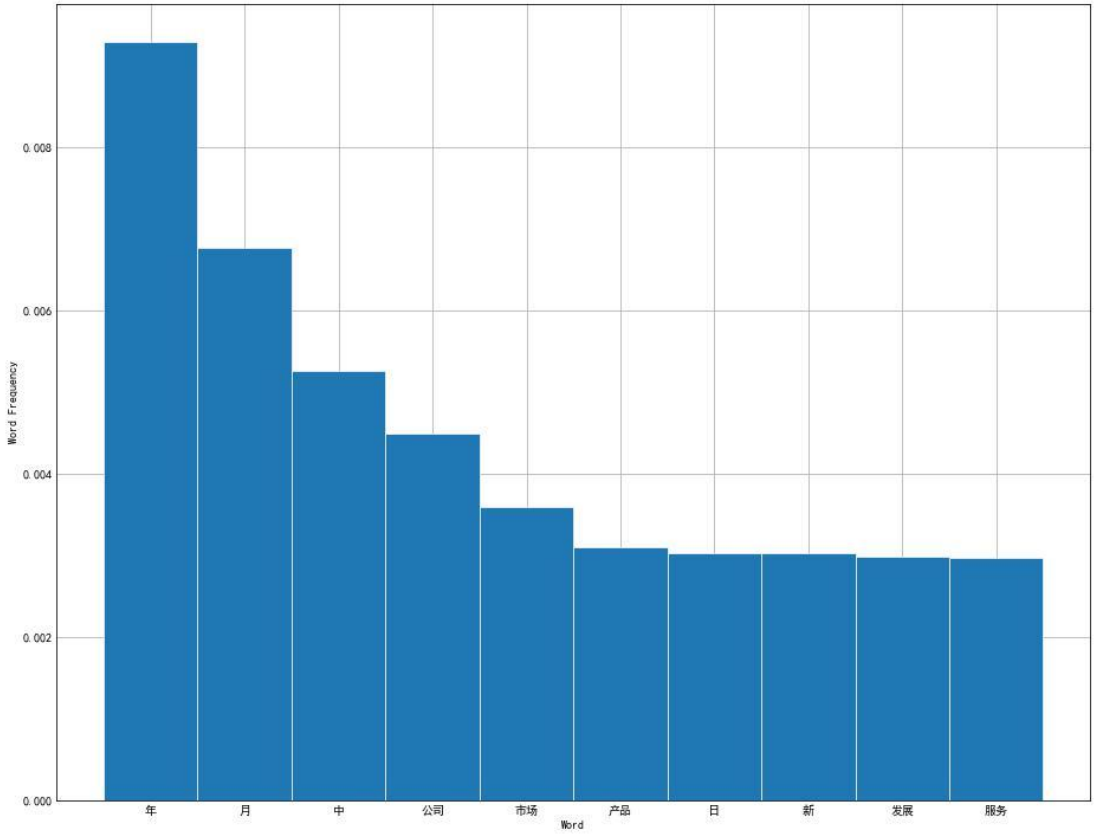
2022-03-14 标题关键词

2023-08-23 词云图

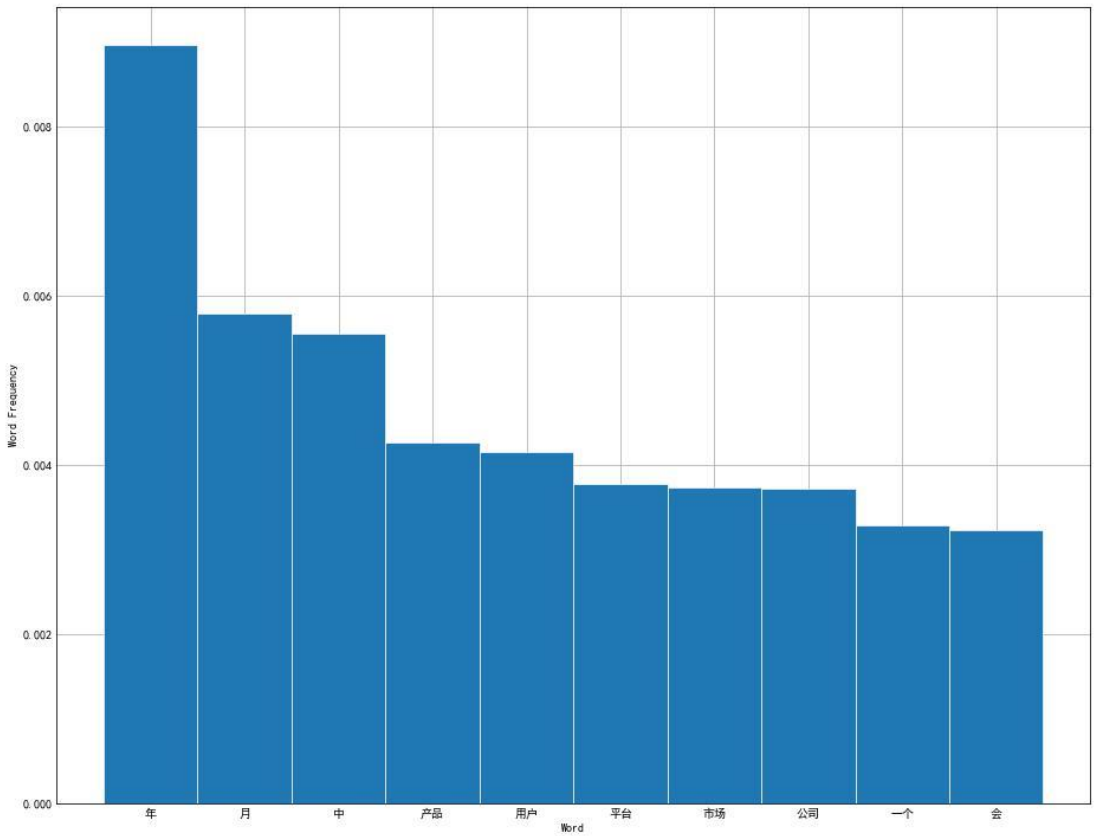
2022-03-02 词云图

2022-03-14 词云图

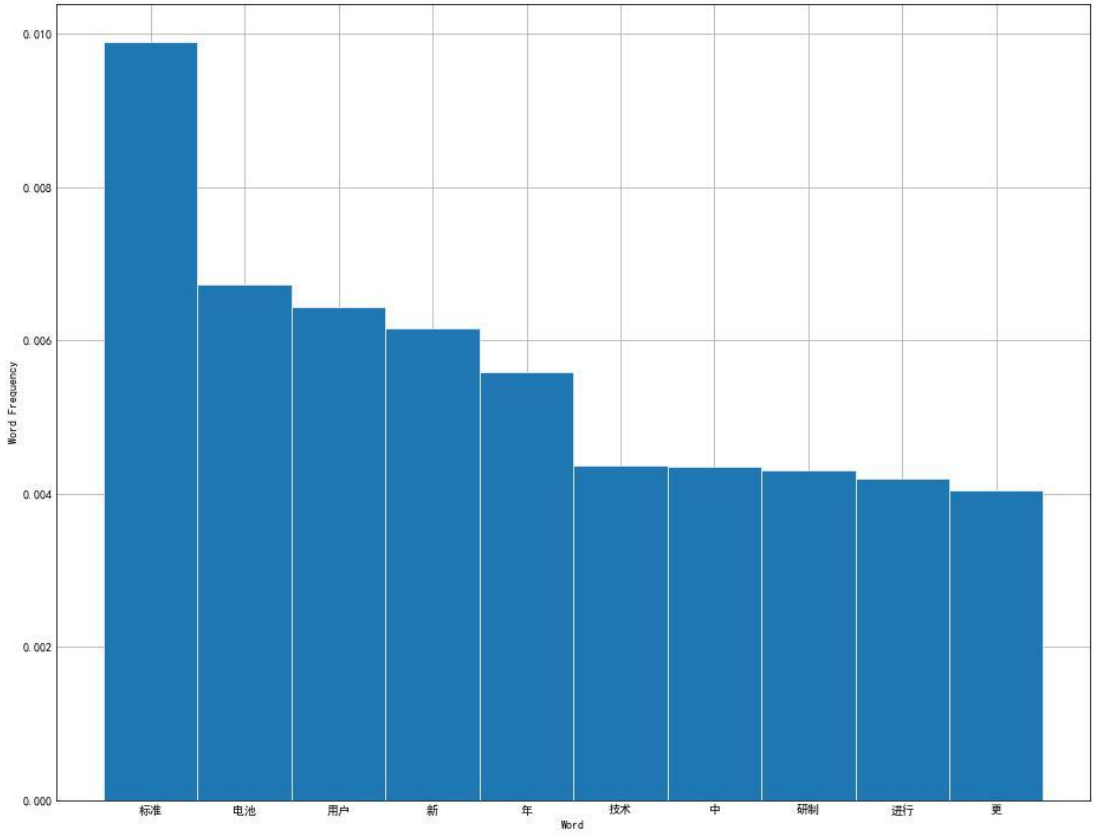
(2) 分析正文关键词



2023-08-23 正文关键词



2022-03-02 正文关键词



2022-03-14 标题关键词



2023-08-23 词云图　　2022-03-02 词云图　　2022-03-14 词云图

从上面的柱状图和词云图来看，这三天的新闻主题均与“苹果手机的发布”有关。可能是这个事件被多家媒体报道，故新闻数较其他日期多。并且新闻的标题就已经基本概括了正文的内容。另外可合理推测，科技板块的新闻大多与新产品的发布或者新技术有关。

三、结论总结

1. 新闻发布数量与发布年份有关，同一年份的新闻发布数量的日均值较稳定，2022 年每天发布的新闻数量比 2023 年更多。

2. 不同类别的新闻的发布时间有所不同，大部分类别的新闻发布时间较为均衡，部分类别的新闻侧重于 2023 年发布或 2022 年发布。

3. 新闻的标题就已经基本概括了正文的内容，科技板块的新闻大多与新产品的发布或者新技术有关。当出现新产品或新技术时，当天的新闻数量会显著增多。

四、相关代码

1. 全部时间的新闻发布数量（按天来计算）

```
#数据处理
import json
from datetime import datetime
res = []
cleaned_res = []

with open('../project1/result.txt','r',encoding='utf-8') as f:
    for line in f:
        t = json.loads(line)
        pub_date = t['date']
        res.append(pub_date)

def clean_date_time(date_str):
    try:
        date_time = datetime.strptime(date_str, '%Y-%m-%d %H:%M:%S')
    except ValueError:
        try:
            date_time = datetime.strptime(date_str, '%Y 年%m 月%d 日 %H:%M')
        except ValueError:
```

```
            try:
                date_time = datetime.strptime(date_str, '%Y 年%m 月%d 日%H:%M')
            except ValueError:
                return date_str

    # 将时间戳转换为 Django 的 DateTimeField 格式
    django_date_time = date_time.strftime('%Y-%m-%d %H:%M:%S')
    return django_date_time

for i in res:
    cleaned_res.append(clean_date_time(i))

date_dict = {} #显示每天的新闻数
def res_append(st):
    if st not in date_dict:
        date_dict[st] = 1
    else: date_dict[st] +=1

for date_time in cleaned_res:
    date = date_time.split(' ')[0]
    res_append(date)

#画图
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

dates = list(date_dict.keys())
news_nums = list(date_dict.values())
fig, ax = plt.subplots(figsize=(12,9))

# 设置 x 轴为日期格式，并设置日期格式和日期分割
fmt = mdates.DateFormatter('%Y-%m-%d')
locator = mdates.DayLocator()
ax.xaxis.set_major_formatter(fmt)
ax.xaxis.set_major_locator(locator)

ax.plot(dates, news_nums, 'o-')
ax.set_xlabel('Date')
ax.set_ylabel(r'News Number')

plt.xticks(range(len(dates)), dates,rotation=60)

plt.savefig('news_num.jpg')
plt.show()
```

2. 每一类别的新闻发布数量（按天来计算）

```
import json
from datetime import datetime
res = []
cleaned_res = []
source = []
cate = ['新浪数码', '新浪科技', 'IT 之家', '快科技', '太平洋电脑网', '中关村在线', '中国家电网', 'CNMO', '市场资讯', '媒体滚动', '新浪科技综合', '澎湃新闻', '财联社', '快科技 2018', '界面新闻', '其他']

with open('../project1/result.txt','r',encoding='utf-8') as f:
    for line in f:
        t = json.loads(line)
        pub_date = t['date']
        if t['author'] !='':
            author = t['author']
        else: author = t['media_name']
        if author in cate:
            source.append(author)
        else: source.append('其他')
        res.append(pub_date)

def clean_date_time(date_str):
    try:
        date_time = datetime.strptime(date_str, '%Y-%m-%d %H:%M:%S')
    except ValueError:
        try:
            date_time = datetime.strptime(date_str, '%Y 年%m 月%d 日 %H:%M
        except ValueError:
            try:
                date_time = datetime.strptime(date_str, '%Y 年%m 月%d 日%H:%M')
            except ValueError:
                return date_str

    # 将时间戳转换为 Django 的 DateTimeField 格式
    django_date_time = date_time.strftime('%Y-%m-%d %H:%M:%S')
    return django_date_time

for i in res:
    cleaned_res.append(clean_date_time(i))

#按分类来构建新字典
dict_1,dict_2,dict_3,dict_4,dict_5,dict_6,dict_7,dict_8,dict_9,dict_10,dict_11,dict_12,dict_13,dict_14,dict_15,dict_16 = [],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
```

```
for i in range(len(source)):
    if source[i]=='新浪数码': dict_1.append(cleaned_res[i])
    elif source[i]=='新浪科技': dict_2.append(cleaned_res[i])
    elif source[i]=='IT 之家': dict_3.append(cleaned_res[i])
    elif source[i]=='快科技': dict_4.append(cleaned_res[i])
    elif source[i]=='太平洋电脑网': dict_5.append(cleaned_res[i])
    elif source[i]=='中关村在线': dict_6.append(cleaned_res[i])
    elif source[i]=='中国家电网': dict_7.append(cleaned_res[i])
    elif source[i]=='CNMO': dict_8.append(cleaned_res[i])
    elif source[i]=='市场资讯': dict_9.append(cleaned_res[i])
    elif source[i]=='媒体滚动': dict_10.append(cleaned_res[i])
    elif source[i]=='新浪科技综合': dict_11.append(cleaned_res[i])
    elif source[i]=='澎湃新闻': dict_12.append(cleaned_res[i])
    elif source[i]=='财联社': dict_13.append(cleaned_res[i])
    elif source[i]=='快科技 2018': dict_14.append(cleaned_res[i])
    elif source[i]=='界面新闻': dict_15.append(cleaned_res[i])
    elif source[i]=='其他': dict_16.append(cleaned_res[i])

date_dict = {} #显示每天的新闻数
def res_append(st):
    if st not in date_dict:
        date_dict[st] = 1
    else: date_dict[st] +=1

#以第一分类为例，其他分类类似
for date_time in dict_1:
    date = date_time.split(' ')[0]
    res_append(date)

import matplotlib.pyplot as plt
import matplotlib.dates as mdates

dates = list(date_dict.keys())
news_nums = list(date_dict.values())

# 创建 figure 和 axes 对象
fig, ax = plt.subplots(figsize=(12,9))

# 设置 x 轴为日期格式，并设置日期格式和日期分割
fmt = mdates.DateFormatter('%Y-%m-%d')
locator = mdates.DayLocator()
ax.xaxis.set_major_formatter(fmt)
```

```
ax.xaxis.set_major_locator(locator)

# 使用 plot 函数绘制数据
ax.plot(dates, news_nums, 'o-')

ax.set_xlabel('Date')
ax.set_ylabel(r'News Number')
plt.xticks(range(len(dates)), dates,rotation=60)

plt.savefig('news_num_cate1.jpg')
plt.show()
```

3. 新闻关键词分析

```
import json
from datetime import datetime
res = []
cleaned_res = []
title_res = []
content_res = []

with open('../project1/result.txt','r',encoding='utf-8') as f:
    for line in f:
        t = json.loads(line)
        pub_date = t['date']
        title = t['title']
        for p in t['content']:
            content +=p
            content+='\n'
        res.append(pub_date)
        title_res.append(title)
        content_res.append(content)

def clean_date_time(date_str):
    try:
        date_time = datetime.strptime(date_str, '%Y-%m-%d %H:%M:%S')
    except ValueError:
        try:
            date_time = datetime.strptime(date_str, '%Y 年%m 月%d 日 %H:%M')
        except ValueError:
            try:
                date_time = datetime.strptime(date_str, '%Y 年%m 月%d 日%H:%M')
            except ValueError:
                return date_str
```

```
        # 将时间戳转换为 Django 的 DateTimeField 格式
        django_date_time = date_time.strftime('%Y-%m-%d %H:%M:%S')
        return django_date_time

for i in res:
        cleaned_res.append(clean_date_time(i))

#得到这三天的 index
date_index_0823 = []
date_index_0302 = []
date_index_0314 = []

for date_time in cleaned_res:
        date = date_time.split(' ')[0]
        if date == '2023-08-23':
                date_index_0823.append(cleaned_res.index(date_time))
        elif date == '2022-03-02':
                date_index_0302.append(cleaned_res.index(date_time))
        elif date == '2022-03-14':
                date_index_0314.append(cleaned_res.index(date_time))

#分析 2023-08-23
tit = []
con = []
for i in date_index_0823:
        tit.append(title_res[i])
        con.append(content_res[i])

import jieba
words = []
for t in tit: #分析正文关键词只用把 tit 换成 con 即可
        devision_t = jieba.cut(t,cut_all=False)
        words.extend(list(devision_t))

import re
stop_list = []
no_stop_words = []
for line in open("cn_stopwords.txt", 'r', encoding='utf-8').readlines():
        stop_list.append(line.strip())

for word in words:
        if word not in stop_list:
                word = re.sub(r'\d', "", word)
                word = re.sub(r'\s', "", word)
```

```
            word = re.sub(r'\W', "", word)
            if word:
                no_stop_words.append(word)

result = {}
def res_append(st):
    if st not in result:
        result[st] = 1
    else: result[st] +=1

for i in no_stop_words: res_append(i)
result = sorted(result.items(),  key=lambda kv:(kv[1], kv[0]), reverse=True)
result = dict(result)
new_a = {}

#输出所用次数最多的前十个
print("所用次数最多的前十个词")

for i, (k, v) in enumerate(result.items()):
    new_a[k] = v
    if i == 9:
        print(new_a)
        break

sum = 0
for i in result.values(): sum+=i
new_a2 = {}
for i in new_a.items():
    new_a2[i[0]] = i[1]/sum
print("所用次数最多的前十个词的词频")
print(new_a2)

#画图
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('_mpl-gallery')

# make data:
x = new_a2.keys()
y = list(new_a2.values())

# plot
fig, ax = plt.subplots(figsize=(13,10))
ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)
```

```
plt.rcParams['font.sans-serif'] = ['SimHei']   # 设置默认字体为SimHei
plt.rcParams['axes.unicode_minus'] = False   # 解决保存图像时负号'-'显示为方块的问题

ax.set_xlabel('Word')
ax.set_ylabel(r'Word Frequency')

fig.tight_layout()

plt.savefig('words_freq_0823.jpg')

plt.show()

#生成词云
import matplotlib.pyplot as plt
from wordcloud import WordCloud
wordcloud = WordCloud(font_path='simkai.ttf', width=800, height=600, background_color='white', max_words=100, max_font_size=200).generate(' '.join(no_stop_words))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')

plt.savefig('words_cloud_0823.jpg')
plt.show()
```