

# 利用搜索指数窥探舆情

原创 闲欢 Python技术 3月5日

新冠病毒全球蔓延，你知道人民最关注的是什么呢？最近股市大爆，你知道人们相关搜索最多的内容是什么呢？特不靠谱天天嘴炮，你知道他最引人注目的特征是什么呢？如果你对这些都感兴趣，那么请跟随我的脚步去看看吧！

## 百度指数

什么是百度指数？

百度指数是以百度海量网民行为数据为基础的数据分享平台。在这里，你可以研究关键词搜索趋势、洞察网民需求变化、监测媒体舆情趋势、定位数字消费者特征；还可以从行业的角度，分析市场特点。

我们可以简单理解为，百度指数是百度官方提供的一个数据参考平台。特别是平台中提供的搜索指数，是以网民对关键词的搜索量作为基础来计算的。搜索指数反应的是网民在过去一个月内，对关键词搜索量的加权和。

我们知道，百度是国内最大的搜索引擎，也是世界最大的中文搜索引擎，所以我们所关心的关键词搜索基本上可以用百度指数体现出来。

这个指数对于运营人员或者市场营销人员应该很有用，精准地捕捉人们关注的事件关键词可以更好地帮助他们进行关键词购买或者投放。

百度指数的首页地址是：<http://index.baidu.com/v2/index.html#/>，首页页面为：

在这里我们可以直接输入关键词进行搜索。例如，我输入“蔡徐坤”进行搜索，结果页面如下：

这里可以看到对应日期的关键词搜索指数。

## 利用百度指数

如果感兴趣的话，大家可以自己研究一下百度指数的功能，这里就不介绍了。

下面进入本文的正题。虽然百度指数页面可以比较直观地看到我们关键词的搜索结果，但是我们看不到这些结果背后的数据。作为一个程序员，我还是觉得我需要看到具体的数据，所谓“数据在手，天下我有”，有了数据，我们可以根据数据自由使用，想画成曲线也好，想做成直方图也罢，都凭自己喜好。

基于这个需求，我的程序步骤分为两步：第一步是获取数据，第二步是将获取到的数据做成酷酷的词云。

## 获取数据

我们先来看看百度指数页面是怎么查询的：

首先选择“需求图谱”栏目，然后输入关键词“新冠病毒”，点击“确定”按钮，下方就展示了结果了。

我们打开开发者工具，找到搜索关键词指数的请求如下图所示：

有了这个请求，我们就可以通过代码来获取数据了,关键代码如下：

```
1 # 搜索指数URL
2     data_url = 'http://index.baidu.com/api/WordGraph/multi?wordlist[]={keyword}'
3
4 headers = {
5     "User-Agent": 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, li
6     "Cookie": 'PSTM=1579955530; BAIDUID=C98F0EF9DCB3FC7E06D3B0FA63695787:FG=1; BIDUPSID=1FB86823
7     "Host": "index.baidu.com",
8     "Referer": "http://index.baidu.com/v2/main/index.html"
9 }
10
11 # 获取指数数据
12 def get_index(self, params):
13     url = self.data_url.format(**params)
14     response = requests.get(url, headers=self.headers)
15
16     data = json.loads(response.text)['data']
17     print(data)
18
```

获取到数据之后，我们打印一下返回结果：

```
1 {'period': '20190303|20200223', 'wordlist': [{'keyword': '新型冠状病毒', 'wordGraph': [{'word': '新冠状病
2
```

结果是 json 格式，从结果中，我们可以获取到我们搜索的关键词以及与之关联的关键词的热度（pv）、搜索变化率（ratio）、相关性（sim）。

这里有一点需要注意，不是我们输入的所有关键词都有结果的，因为百度也是根据大家在搜索框输入的关键词来做统计，所以大家没有输入的关键词在这里是查询不出来的，例如，我查询“新冠状肺炎”，页面返回的结果是这样的：

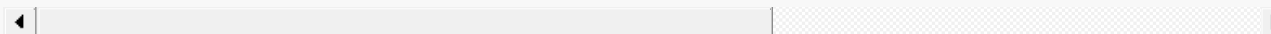
我们再仔细查看页面开发者工具，会发现每次查询关键词前，都会做一个 `checkWordsExists` 的请求，目的是查找关键词是否收录。请看下图：

当关键词存在时，返回的结果是：

```
1 {"status":0,"data":{"result":[]},"logid":2886305955,"message":""}
```

当关键词不存在时，返回的结果是：

```
1 {"status":0,"data":{"result":[{"word":"新冠肺炎","status":10003}],"addWordsNum":0,"addWordsLeft": ""
```



根据这两个结果，我们就可以做一个简单的判断了：

```
1 # 检查关键词是否存在
2 def check_word(self, kw):
3     url = self.check_url % kw
4     response = requests.get(url, headers=self.headers)
5     data = json.loads(response.text)['data']
6     return not len(data['result'])
```

如果存在关键词我们才进行指数请求，不存在直接返回。

## 制作词云

得到百度指数结果后，我们选取热度和搜索变化率分别来制作成词云。代码如下：

```
1 # 获取指数数据
2 def get_index(self, params):
3     url = self.data_url.format(**params)
4     response = requests.get(url, headers=self.headers)
5
6     data = json.loads(response.text)['data']
7     print(data)
8
9     pv_dict = {}
10    ratio_dict = {}
11    for item in data['wordlist'][0]['wordGraph']:
12        pv_dict[item['word']] = item['pv']
13        ratio_dict[item['word']] = item['ratio']
14
15    # 生成词云
16    self.gen_wc_tags(pv_dict)
```

```

16         self.gen_wc_tags(ratio_dict)
17     # 生成词云
18     def gen_wc_tags(self, tags):
19         # 设置一个底图
20         # mask = np.array(Image.open('./bf.jpg'))
21         wordcloud = WordCloud(background_color='black',
22                                mask=None,
23                                max_words=100,
24                                max_font_size=100,
25                                width=800,
26                                height=600,
27                                # 如果不设置中文字体，可能会出现乱码
28                                font_path='/System/Library/Fonts/PingFang.ttc').generate_from_frequencies(tags)
29
30         # 展示词云图
31         plt.imshow(wordcloud, interpolation='bilinear')
32         plt.axis('off')
33         plt.show()
34
35         # 保存词云图
36         wordcloud.to_file('./gzbd_wc.png')
37

```

我们搜索关键词“新冠病毒”，运行程序，得到热度词云图是：

变化率词云图是：

从词云图中，我们可以看到大家搜索“新冠病毒”，相关查询最多的是病毒的特征以及早期症状，这两个是大家最关心的内容。而从变化率的词云图来看，变化最大的还是我们平时看到的一些标题党喜欢用的词语，这类词语一般过一段时间就会出来一个，然后又迅速消退。

我们再来看看关键词“股市”的词云图：

最吸引我注意的是“1万炒股一年最多挣多少”，看来好多人还是想着一夜暴富啊！当然我也想知道呢！所谓“有事问百度”，其他的一些相关关键词完美地提现了韭菜的特质，都是一些股市新手爱问的问题。

最后，我们再看看关键词“特朗普”的词云图：

映入眼帘的大多数是一些人名，应该基本上是跟特朗普有关的政治家吧。所以想知道谁跟特不靠谱打交道最多，从这张图上可以窥见一二吧。

## 总结

本文主要通过爬取百度指数的关键词搜索数据，制作成词云图，来展现关键词相关词的搜索情况，从而找到人们对于某一个热点的关注点集中在哪里。

文中示例代码：<https://github.com/JustDoPython/python-100-day>

**PS:** 公众号内回复「Python」即可进入 Python 新手学习交流群，一起 **100天计划**！

-END-

**Python 技术**  
**关于 Python 都在这里**