孙笑川微博爬虫

前言

这其实是五月初自己学习爬虫时做的项目，中间比较忙一直拖到现在才开始写。本爬虫爬取了孙笑川的 700 条微博，其微博下的约 188 万条评论，以及发表评论的约 25 万名用户。本爬虫基于 WeiboSpider，代码见 https://github.com/lukewys/SunXiaoChuan-spider。

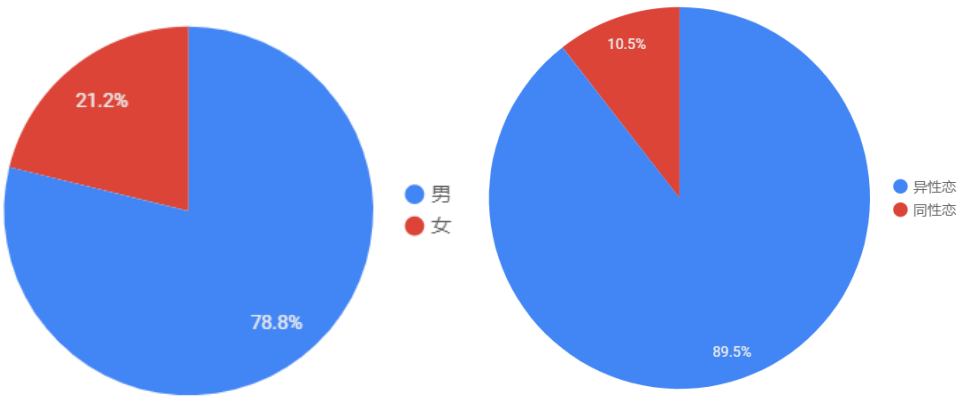
我们首先分析孙笑川微博下评论的用户的组成。在第二节中，将分析评论内容的组成。最后一节将介绍如何使用单台 ADSL 代理服务器进行爬虫。

1-狗粉丝是谁

对于用户信息，本爬虫爬取了性别、性取向、VIP 等级、地理位置、微博数、粉丝数等信息。

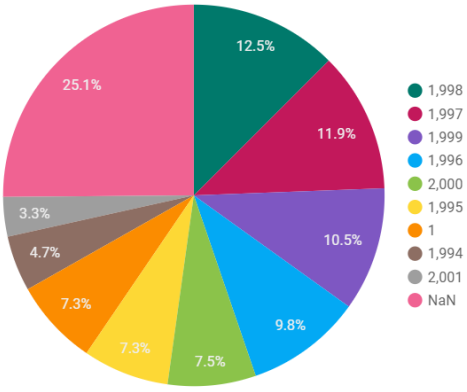
性别、性取向

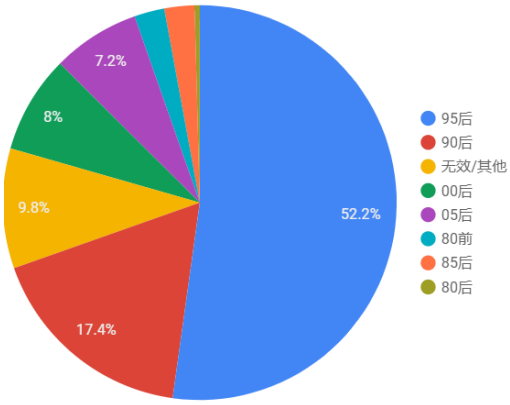
男性、异性恋为主。



年龄

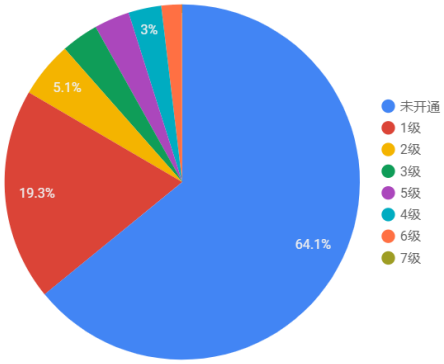
98 年人数最多，多数为 95 后。





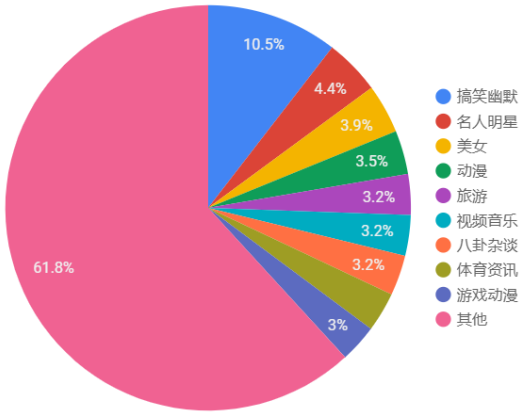
VIP 等级

多数为未开通 VIP。



标签

这里的标签似乎只能在网页端编写，所以填写的比例不大。总体来说，狗粉丝的标签多数为搞笑、二次元等。



地理位置分布

使用百度地图 api 展示如下：

2-狗粉丝说了什么

在本节中，重点分析了评论的内容，展示如下。

@频率

Top5：

@带带大师兄

@蔡徐坤

@共青**央

@紫*阁

@Mr_凡先生

微博表情使用频率

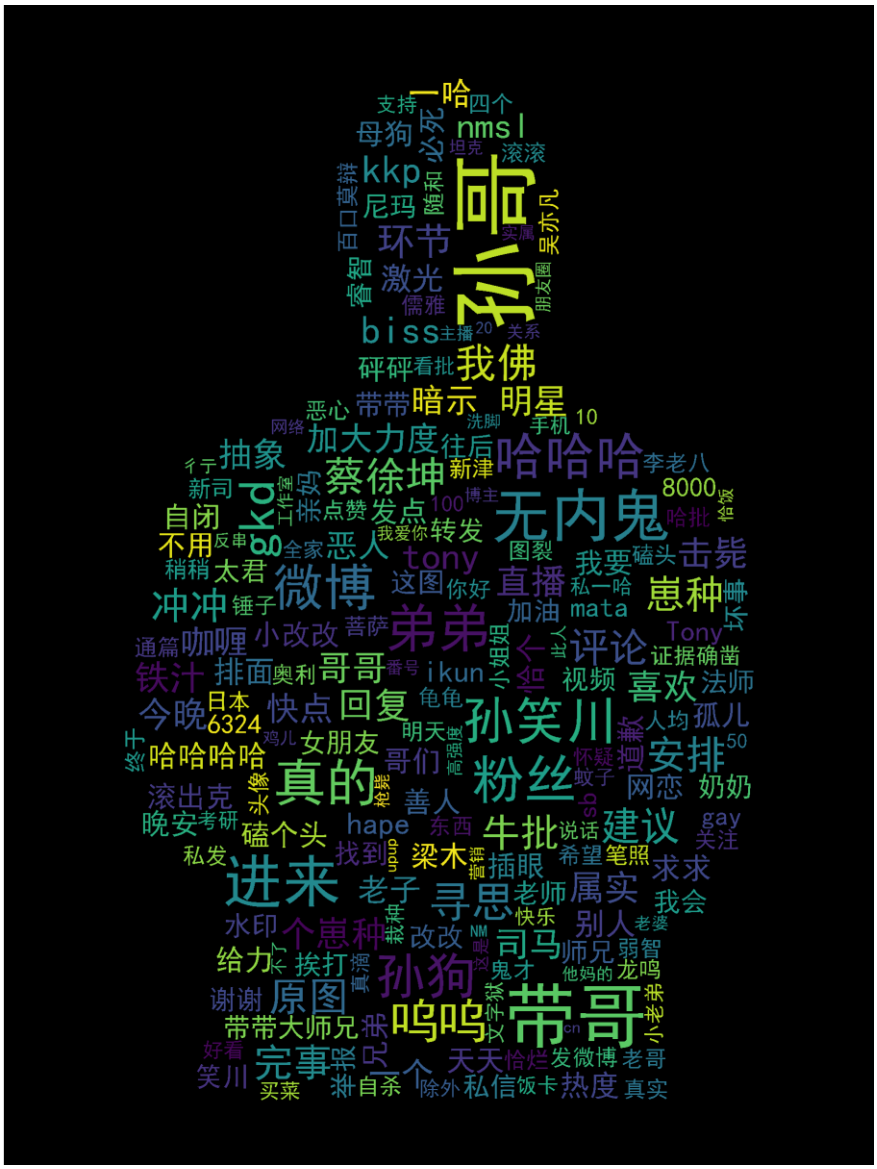
Top5：[马头], [狗], [赛马], [啤酒], [爷爷]

emoji 使用频率

Top5：🍌🐶🏇🍌🐎

分词+词云

试了几个工具，最后的结论还是 jieba 分词最好。



3-单 ADSL 服务器代理

在本节中我将会介绍一些爬虫中遇到的坑以及解决办法。

本项目所有代码见这里。

本项目中爬虫基于[WeiboSpider](https://github.com/nghuyong/WeiboSpider，主要使用了其中 master 分支，即账号池爬虫。

爬虫的基本问题

首先我们需要简单介绍一下爬虫遇到的基本问题。首先，最简单的方式就是从网页返回的数据中提取想要的信息。这种单线程的方式运行速度是相当慢的，因为大部分时间都用在等待网页通信上。因此，我们需要使用多线程爬虫，开多个线程并发进行爬取信息，这就需要用到类似 scrapy 这样的框架。但是网站显然也不能让你疯狂薅羊毛，单个 IP/账号请求过于频繁，网站很快就会将 IP 或账号临时封禁了。因此，我们需要建立 IP 池以及账号池，随机选取账号以及 IP 代理进行访问。这样从网站的角度每个 IP 以及每个账号的访问频率就不是很频繁了。

账号池相对容易搭建，成本也很低，具体在[WeiboSpider](%5Bhttps://github.com/nghuyong/WeiboSpider%5D(https://github.com/nghuyong/WeiboSpider)中有详尽的介绍。这里花成本的地方主要是代理 IP，如果你不在乎成本，那你可以买市面上的爬虫 IP 池，取之不尽的 IP 直接拿来用。但是这样的话你的钱包很快就日渐消瘦。而且，上述那些服务一般针对超高并发、追求速度与稳定的爬虫。如果你只是自己想玩玩，一边做一边学的话，很可能还没等你学明白怎么挂代理，钱就花完了。

穷人的爬虫代理解决方案

因此，我这里介绍一种低成本的代理解决方案——ADSL 服务器。ADSL 服务器是一种动态 IP 的拨号服务器，其连接的网络是 ADSL 网络（也就是普通家用的拨号宽带），其最大的特点就是动态 IP，每拨一次号其公网 IP 就进行改变。在淘宝上，大约 50 元就可以租一个月的 ADSL 服务器。

读到这里，聪明的你大概会想到，我开若干台 ADSL 服务器，每台服务器定时重新拨号，岂不是就拥有了无限容量的 IP 池？

确实是这样，不过这其中有一个问题就是在重新拨号 IP 变化后，如何更新 IP 池中的 IP 地址。一个[方案](https://github.com/Python3WebSpider/AdslProxy 是使用 Redis Server 来储存并更新 IP 池中的 IP 地址，不过这又要求把 Redis Server 开在另外一台固定 IP 服务器上。

那能不能再给力一点呢？

下面介绍只开一台 ADSL 服务器进行代理爬虫的方法。

先说明动机：实际使用发现，在 ADSL 拨出的 IP 地址中，有的 IP 可以高强度爬很长时间不被封号，而有的 IP 则使用时间很短。因此，如果使用上述 Redis Server 方法，开的 ADSL 服务器很少的话，大部分时间都用在了定时拨号上。并且，会出现有的时候 IP 还可以用就拨号了，或者 IP 已经被封禁了还没到拨号的时候。

思路如下：先紧着 ADSL 的当前 IP 地址疯狂撸爬虫，在 IP 被封之后马上令 ADSL 服务器重新拨号，之后撸下一个 IP。那么服务器在重新拨号后公网 IP 已经变了，如何和服务器通信呢？在这里就使用 SSH，因为虽然服务器不拨号不可访问外网，但是服务器的 SSH 的地址是固定的。因此，可以使用 SSH 命令让服务器重新拨号，再返回命令行显示的结果，提取出改变后的 IP 地址。

具体代码如下，完整代码见 GitHub：

```
def get_random_proxy(self):
    global ready
    if ready is False: #  拨号完成的判断
        return None
    error_threshold = 10
    if self.error > error_threshold: #  如果出错数目超过阈值，即 IP 被封，则重新拨号
        ready = False
        self.ssh.connect(hostname='服务器 IP', port=服务器端口, username='用户名',
password='密码')
        stdin, stdout, stderr = self.ssh.exec_command('python3.4 adsl.py')
        result = stdout.read() #  读取代码返回显示的 IP 字符串
        result = result.decode()[:-1] #  去除换行符
        self.proxy = result + ':8888' #  加上端口号
        ready = True
        self.error = 0
    if self.proxy is None:
        return None
    return 'http://' + self.proxy
```

这里用到了 [AdslProxy](%5Bhttps://github.com/Python3WebSpider/AdslProxy%5D(https://github.com/Python3WebSpider/AdslProxy)中的拨号代码，需要把其中的 Redis Server 部分删除。