

**本科生毕业论文（设计）**

题 目：基于Python的新浪微博用户数据

获取并分析技术的设计与实现

院 （系）： 信息科学学院

专 业： 软件工程

学生姓名： 刘棉鑫

学 号： 15053048

指导教师： 谈斯聪(讲师)

二〇一九年五

基于Python的新浪微博用户数据获取并分析技术的设计与实现

[摘要]新浪微博用户数量的急剧增长带来了信息量的剧增，基于微博数据的社交行为分析、数据挖掘等等研究越来受到大众的关注。本文分析如何利用Python爬取微博数据，利用Cookie实现通过用户的模拟登录，以“冷知识bot”为例，抓取了它的全部信息。接着，从微博发布动作偏好、微博信息特征、微博关键词、微博主影响等四个方面，展开了对该微博主的分析。获取了一些有趣的发现，如（1）微博主形成了较为固定的发布微博模式；（2）微博主发布信息的点赞数、评论数、转发数遵从幂律分布；（3）该微博主发布的微博信息的关键词主要有“中国”、“发现”、“知识”、“世界”、“时间”、“喜欢”、“名字”等；（4）该微博主的影响力具备逐步增大的趋向。

[关键词] 新浪微博；爬虫；数据获取；数据分析

**Design and implementation of sina weibo user data acquisition and analysis technology based on Python**

**[Abstract]**The rapid increase in the number of Sina Weibo users has led to a sharp increase in the amount of information. Research on social behavior analysis and data mining based on Weibo data has attracted more and more attention. This article analyzes how to use Python to crawl microblog data, and use cookies to implement the simulated login through the user. Take the "cold knowledge bot" as an example and grab all the information. Then, from the four aspects of microblog publishing action preferences, microblog information features, microblog keywords, and microblog master influence, the analysis of the microblog master was launched. Some interesting findings were obtained, such as (1) microblogging masters formed a relatively fixed microblogging mode; (2) microblog information points, comments, and forwarding numbers obey power law distribution; (3) the micro The keywords of Weibo information published by bloggers mainly include “China”, “Discovery”, “Knowledge”, “World”, “Time”, “Like”, “Name”, etc.; (4) The influence of the Weibo owner The force has a tendency to increase gradually.

**[Key words]** Sina weibo, Crawler, Data acquisition, Data analysis

**目 录**

摘 要 I

ABSTRACT II

[第 1 章 引言 1](#_Toc18158)

[1.1 研究背景 1](#_Toc3398)

[1.2 国内外研究现状 1](#_Toc24970)

[1.3 研究意义 2](#_Toc5896)

[1.4 研究目标 2](#_Toc10014)

[1.5 本文的主要工作与创新点 2](#_Toc20192)

[第 2 章 相关原理与理论 5](#_Toc21117)

[2.1 爬虫的工作原理 5](#_Toc3626)

[2.2 Cookie的作用 6](#_Toc8710)

[2.3 基于Python开发爬虫 6](#_Toc631)

[2.4 本章小结 7](#_Toc31500)

[第 3 章 数据获取的设计与实现 8](#_Toc14789)

[3.1 总体设计 8](#_Toc24992)

[3.2 关键模块实现 9](#_Toc26028)

[3.3 本章小结 13](#_Toc12190)

[第 4 章 爬虫测试与数据展示 14](#_Toc13491)

[4.1 测试环境 14](#_Toc3369)

[4.2 运行状态与数据抓取结果展示 14](#_Toc23387)

[4.3 数据分析 15](#_Toc32248)

[4.4 本章小结 27](#_Toc29368)

[第 5 章 总结与展望 28](#_Toc12837)

[5.1 工作总结 28](#_Toc16518)

[5.2 研究展望 28](#_Toc32349)

[参考文献 2](#_Toc23829)9

[致谢 3](#_Toc23829)0

# 第1章 引言

## 1.1研究背景

随着如今互联网的飞速发展，便捷快速的网络环境使各种网络信息触手可得，面对互联网中的海量数据，如何从海量的数据中迅速找到有价值的信息，成为人们日益关心的话题[1] [2]。

百度、谷歌等搜索引擎产生一定程度上为用户提供了一个方便易用的数据获取途径。虽然有很多方便的搜索引擎，但这并不能满足人们日益增长的需求，搜索引擎所带来的信息也并不全面。许多计算机研究者对于搜索引擎提供的服务感到不满意，他们需要一种可控的明确目标的获取数据的方式。网络爬虫能满足使需求，它[4][5]是一种按照程序设计者所设定的规则，自我驱动抓取网页信息的程序或者脚本。我们可以根据自己想要获取的数据格式来调整爬虫，使其剔除掉无用的部分而保留有效信息。

新浪微博是我国出现时间最早，也是规模最大的一个社交平台。随着微博平台的不断发展与完善，微博用户数量仍在持续稳定增长。

新浪微博用户数量的急剧增长带来了信息量的剧增，基于微博数据的社交行为分析、数据挖掘等等研究越来受到大众的关注，如何从网络上爬取研究所需要的微博数据成为一个关键的问题。

本文使用Python语言，通过模拟用户登录，实现一种针对新浪微博的爬虫，并对获得的数据进行有效地分析。

## 1.2 国内外研究现状

早在20世纪末，国内外研究者就展开了网络爬虫的开发，现在爬虫技术已经相对成熟了[6][7][8]。尤其在国外，推特、脸书等社交平台出现得较早，人们可以通过官方提供的API接口轻松获取其中的数据进行研究。但在国内，新浪虽然也提供了API接口，但在实际使用时限制比较多，所以还需要研究其他的获取数据的方式。

在网络爬虫开发中，Python针对于网络爬虫，有着其他语言无法比拟的优势，它有着丰富的开源库，代码简洁，易于编写，最重要的是代码的维护变得较为简单轻松。而Python的爬虫应用发展至今，也有许多开源的库供我们直接使用，如：Requests。本文也将基于Requests来实现一个爬虫。

目前和多网站出于保护知识产权的原因，不愿意网站的内容被网络爬虫爬取到，的确网站内容应该受到版权的保护。所以很多网站为了防止被人出于商业目的爬取内容，甚至进行盗版侵权的行为，推出了反爬取机制。新浪微博就具有着完备的反爬取小组来应对网络爬虫[10][11]，一旦被认为是爬虫或者程序操控，而不是用户在获取信息，新浪微博就会拒绝访问。当然本文研究的爬虫并不是以商业盈利为目的，只是进行程序设计上的一些研究，但如何应对新浪的反爬机制也是需要解决的问题。

## 1.3 研究意义

1. 深入学习Python和Requests库，在实现用爬虫爬取网络数据的过程中，进一步深化对新技术的掌握，积极提升自主设计程序的能力。

2. 设计出一个简单的程序，具备一定的可改性，可以为作为以后拓展出更高级更具体实用程序的基础。

## 1.4 研究目标

本文的的研究目的主要在于使用基于Python的技术，实现对新浪微博数据的爬取，并将获得的数据可视化。

1.模拟登录新浪微博，利用Python 的Requests和lxml库，抓取数据。

2.选择合适的方式进行数据存取。

3.对获取的数据简单地进行可视化展示，并进行分析。

## 1.5本文的主要工作与创新点

该文设计并实现了新浪微博爬虫程序，以微博主“冷知识bot”为例，对他发布的微博内容进行了全部抓取，并从微博发布动作偏好、微博信息特征、微博关键词、微博主影响等四个方面，进行了数据分析。

（1）综合使用Requesets、lxml等Python第三方库，采用了Cookie模拟登录方式，实现了一款针对新浪微博的爬虫程序；

（2）使用关键词分析手段，实现了对微博主所发微博信息的内容提取与云词化展示；

（3）综合使用点赞数、转发数、评论数等数据，采用加权平均值方法对微博主的影响力进行分析与评价。

# 第2章 相关原理与理论

本章介绍涉及爬虫的相关理论，主要包括爬虫工作原理、Cookie的作用和Python相关第三方库。

## 2.1 爬虫的工作原理

网络爬虫（Web Crawler）[12]，简称为爬虫。通常，网络搜索引擎如百度、谷歌等，会使用网络爬虫，抓取其他网络的有价值信息，来更新其网站内容的索引，以便用户可以更有效地搜索。此外，我们也可以使用网络爬虫抓取网络数据，以展开数据分析，形成有价值的分析报告。

简单的爬虫结构如图2-1所示[13]。其中，爬虫引擎相当于蜘蛛机器人的“大脑”，负责整个爬取数据的调度，控制爬虫的整个工作流程，根据事先设定的动作，完成爬虫事件的触发。URL调度器负责管理爬虫过程中的网址信息，通常爬虫开始工作时，有初始URL，被称为种子（Seeds）；随着爬虫工作的展开，在后续网页中也会爬取许多URL，它们被放入URL调度器中，受到去重复和统一管理。下载器负责访问网站的URL，并下载网页数据。网页解析器在收到下载器传来的网页信息后，解析网页的结构，分析出有价值的信息或拿到下次需爬取的URL，最终形成有用信息。



* 1. 爬虫结构图

网络爬虫的基本工作流程通常为：

1. 爬虫引擎根据用户的爬虫设置，收到种子URL，作为最开始要抓取的网址，交给URL调度器；
2. URL调度器收到URL后，将需要下载的URL发送到下载器；
3. 下载器负责访问URL并下载网页信息再发送到解析器；
4. 解析器开始解析网页信息，获取下次要抓取的URL，交给URL调度器同时分析并获取有用信息；
5. 重复第二步骤，直到爬虫引擎关闭本爬虫。

## 2.2 Cookie的作用

Cookie[14]简单来说就是网站通过加密保存到用户本地终端的一段信息，主要用于验证用户身份等。我们访问网页时，通常使用的时Http协议。它是一种无状态协议，也就是说无法判断用户是否已经访问过。比如，用户A第一次访问了网站W，那么用户A再一次访问网站W时，该网站并不知道这次访问的用户是A。Http协议只是接收用户的访问请求，然后返回响应信息给用户。这就给用户造成了麻烦，当用户多次访问同一网站时，用户不得不多次输入验证信息如用户名，密码等，影响用户体验。而Cookie正是为了解决这一问题，Cookie利用了HTTP中的Headers字段，由网站创建并保存在用户本地终端上，能够有效地跟踪Session状态，进而辨别用户身份。

新浪微博就是这样，需要添加Cookie进行模拟登录才能进一步爬取数据。

## 2.3 基于Python开发爬虫

Python语言是一种胶水性解释语言，具备语法简洁、容易上手、简单方便，受到人们的欢迎，社区活跃人数局开源社区的前列；同时，Python能够在多个操作平台上使用，支持Windows、Linux、Mac 等主流开发环境。众多开发者纷纷投入Python语言的怀抱，贡献了许多源代码，形成了许多开源的、功能强大的第三方库。

### 2.3.1 Requests库

为了实现网页下载，就不得不有HTTP客户端库。Python自带了标准库urllib，在python2.x和python3.x中均有绑定版本。但是它的操作相对比较麻烦，本文使用功能更为强大、使用更为简单的第三方库Requests[15]。

Requests库具有发起Get请求、获取Response响应、自定义网页请求头header、发送POST表单数据、上传文件、自定义cookies等等功能，能够较好地实现本文的需求。同时，采用了Apache2 Licensed开源协议，用户可以遵循开源规则的情况下，免费使用；且基于urllib封装，能轻松地实现对网页信息的操作，非常易用。

### 2.3.2 lxml库

除了获取网页内容之外还需要对这个网页进行解析，也就是通过某种方式获取到你所需要的，有研究价值的部分信息或者是新的URL。

Python常用到的解析库有Re正则表达式，html.parser和lxml。第一种Re正则表达式常用于解析字符串信息，将整个网页也作为字符串，从而根据模式匹配的方法，获取信息；但是存在解析困难，适用性不强的问题。html.parser是python自带的标准库，和第三方库lxml解析原理类似，将整个网页文档作为Document对象，它具有树状结构，可以使用上下级标签，以实现对网页的遍历和数据的获取。但lxml解析速度快，资源库丰富，同时包含了ElementTree、html5lib、BeautfulSoup等库，适合容错能力强的场景。所以本文选择lxml作为解析器的库。

## 2.4本章小结

本章对爬虫有关的理论及知识做了介绍，有助于后续章节对爬虫进行研究与设计。

# 第3章 数据获取的设计与实现

获取新浪微博的数据主要方法有：（1）利用新浪提供的OpenAPI接口；（2）自己编写爬虫。

新浪微博本身提供了约20种API接口。通过这些接口，能对微博的内容，用户、评论以及用户关系进行爬取。从理论上看，这无疑是最简单最直接的方式。但事实上微博API接口存在着某些限制，对于小型研究团队或个人而言不是很方便。比如开发者必须经过新浪网站的授权，且授权有期限；OpenAPI接口访问频次也有限制等，所以本文采用了第二种方法，编写纯爬虫。

新浪微博目前有三个类型的站点：古老的网页端weibo.cn、移动端m.weibo.cn以及现在的网页端weibo.com。这三个站点的复杂程度是逐渐提高的。weibo.cn站点没有加入目前的新技术，用户动态等从html里面就可以获取，爬取相对来说比较简单。但weibo.cn这个站点也完全可以获取一个微博用户的个人资料、粉丝、关注以及他的全部微博等等。因此，本文从简单的weibo.cn站点入手获取数据。

## 3.1 总体设计

本文的主要设计分为三大模块：（1）登陆微博：使用一个已经注册好的微博账号进行登陆，获取该账号的Cookie,并获取待抓取用户的user\_ID；（2）网页下载：该模块访问被抓取用户的各类信息界面，利用Requests库下载待抓取页面的html代码；（3）网页解析：利用Python的lxml库分析html代码，获得待抓取的各类信息。

该爬虫程序的工作流程如图3-1所示，首先登录微博，成功后访问待抓取页面；接着下载待抓取的页面的html代码，使用lxml等python库解析页面，并根据预先定制的规则，抽取有关信息；如若信息内具有下一次待抓取的页面链接，则回至网页下载模块，重新进行下载、解析；待抓取完毕后，将所有信息保存至数据库。



1. 爬虫工作流程示意图

## 3.2 关键模块实现

该部分主页对关键模块进行介绍，包括模拟登陆及获取用户ID、网页下载器、网页解析器和数据保存。

### 3.2.1 模拟登录以及获取用户ID

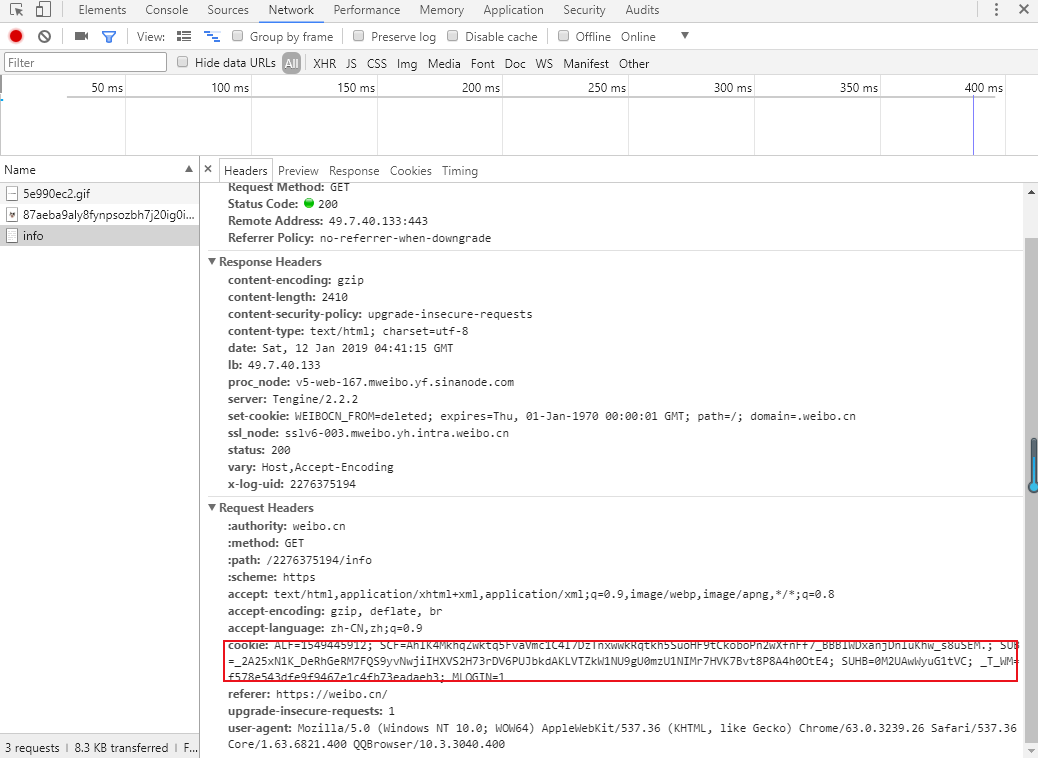
（1）利用Cookie模拟登录

要访问微博数据，必须要先登录，这也是爬取新浪微博的难点。可以用爬虫程序实现模拟微博用户登录，来进一步获取数据。

本文利用Cookie的这一特性，事先通过浏览器的方式，登录新浪微博主页；接着输入用户名、口令，实现登录。当成功登陆后，新浪微博服务器会记录登录的状态，经加密后，使用Response的方式，发送给用户的的客户端，从而存储于本地客户端的Cookie中。

当本文编写的Python爬虫，需要登录新浪微博时，首先读取已存储于本地的Cookie，并发送给新浪微博服务器。而该服务器检查这个Cookie，可以认定客户端的合法身份，从而实现用户对新浪微博的成功登录。

如图3-2所示，使用QQ浏览器登录微博后跳转到weibo.cn站点，F12键打开开发者工具，利用提供的开发者工具可以看到请求的信息。其中，红线所表示的“COOKIE”就是Cookie信息，由新浪微博服务器产生，并保存于本地。



1. Cookie获取示意图

首先需复制这段Cookie文本的内容，然后在Python中定义Cookie对象，这样在进行页面访问时，Request对象中会存在Cookie信息。这样的话，在Cookie的有效期内爬虫就能直接进行访问，最终提取到想要的数据。采用这种做法，只需以下主要代码就可以实现微博的登录并简单获取页面数据 ：

cookie = {"Cookie": "<Cookie文段>"}

url = "https://weibo.cn/u/%d?filter=%d&page=1" % (

self.user\_id, self.filter)

html = requests.get(url, cookies=self.cookie).content

（2）获取用户ID

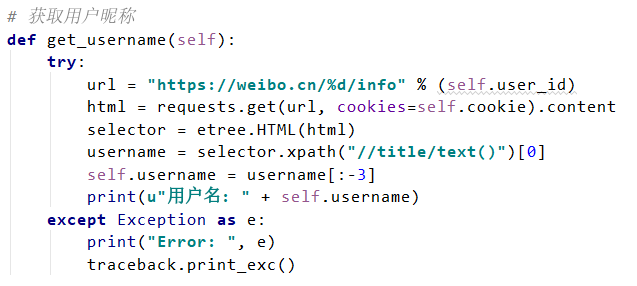
统一资源定位符(URL)，简单来说就是网页的地址。打开浏览器首页的地址也就是一个URL，URL通过一段数字或字母的排列来确定一个能让浏览器识别信息的地址。

本文使用的URL=’https://weibo.cn/<user\_ID>’，所以我们只需要在weibo.cn站点中找到要爬取的某个用户，用户主页的地址栏里就能获取到user\_ID。

### 3.2.2 网页下载器

网页下载器是爬虫的核心组件，它能将Web上URL对应的网页下载到本地 。我们采用Requests库实现网页下载。当访问网站时，经常使用requests.get()函数，它返回HTTPresponse类型的数据

本文为以获取用户昵称为例，来说明Requests库的使用。如图3-3所示，html = requests.get(url, cookies=self.cookie).content，即使用requests的get函数，访问url = "https://weibo.cn/%d/info" % (self.user\_id)的网址；当新浪微博服务器，返回响应信息Response后，使用html获取该网页的二进制数据。接着，经由Python的lxml库的etree．HTML来处置网页源代码，从而获得xpath能够解析的信息。



1. 获取用户昵称函数图

### 3.2.3 网页解析器

爬虫程序成功下载到爬取页面后，就可以对得到的HTML脚本进行解析，进行数据的提取。我们需要提取博主发表的微博内容、发表位置、发表时间、转发评论、点赞数等。除此以外，微博主还有个人的用户昵称、用户ID、关注数、粉丝数等信息。

不难发现微博的页面上的每条博文均是以<div class="c" id

= ⋯> 开始，内部以<div>标签形成的树结构，其主要结构如表3-1所列。

表3-1博文的数据结构

|  |  |
| --- | --- |
| <div class="c" id="xxx"> | 一条博文 |
| 第一个子div：<div><span class="ctt">…..</span></div> | 博文的具体内容 |
| 第二个子div：<div><a>…..</a></div> | 博文的点赞数、转发数、评论数 |
| 第二个子div：<div><span class="ctt">…..</span></div> | 博文的发布日期、发布时间 |

但博文的数据表示并不统一。一条博文的全部内容存在于<div class="c" id="xxx">的标签内；博文的具体内容存在于第一个子div中，其形式为<div><span class="ctt">…..</span></div>；博文点赞数、转发数、评论数存在于第二个子div中，放于多个形如<div><a>…..</a></div>的超链接中；博文的发布日期、发布时间同样存在于第二个子div中，属于<span class="ctt">…..</span>的内容之中。

正是因为存在这些差异，简单使用标签过滤很难有效取得需要的数据。为了方便微博数据的提取，本文使用lxml作为微博页面解析的工具。

通过Python的lxml库的etree函数可以解析网页，从而获取想要的数据。关键代码如下，例如获取长微博文字内容：

def get\_long\_weibo(self, weibo\_link):

try:

html = requests.get(weibo\_link, cookies=self.cookie).content

selector = etree.HTML(html)

info = selector.xpath("//div[@class='c']")[1]

wb\_content = info.xpath("div/span[@class='ctt']")[0].xpath(

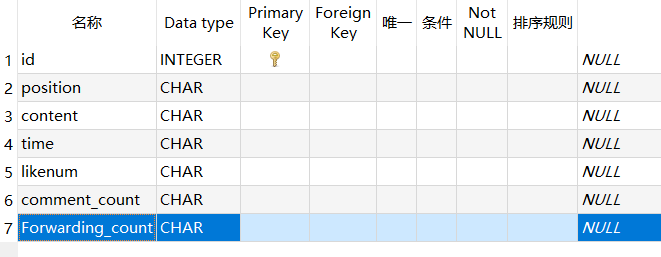
"string(.)").replace(u"\u200b","").encode(sys.stdout.encoding, "ignore").decode(sys.stdout.encoding)

return wb\_content

### 3.2.4 数据保存

本文使用Sqlite数据库[17]，实现数据的保存。

以保存微博信息为例说明数据保存过程，如图3-4所示，本文创建微博信息的保存数据表。该表包括7项，分别为id、position、content、time、likenum、comment\_count\_forwarding\_count，其中id为Int类型，是该表的主键，自动递增；position为char类型，表示微博的发表地点；content是char类型，表示微博内容；time为char类型，表示微博的发表时间；likenum为char类型，表示点赞数；comment\_count为char类型，表示微博评论数；forwarding\_count为char类型，表示转发数。



1. 获取用户昵称函数图

本文使用如下的sql语句创建该数据表：

CREATE TABLE weibo (

id INTEGER PRIMARY KEY AUTOINCREMENT,

position CHAR,

content CHAR,

time CHAR,

likenum CHAR,

comment\_count CHAR,

Forwarding\_count CHAR

);

在Python语言中，首先需要import sqlite3，就是将Sqlite数据库导入，并通过sqlite3.connect('xx.db')连接数据库，接着创建一个例程即用conn.cursor( )。当抓取到微博信息后，使用execute("INSERT INTO WEIBO (position、content、time、likenum、comment\_count\_f-orwarding\_count) VALUES (无, '斑鱼狗（Ceryle rudis）是一种鸟。', '2019-02-15 11:40', '1535', '173', '217',)")，从而实现微博内容的保存。

## 3.3 本章小结

该章主要对微博爬虫进行了总体设计，分为了三大模块，并依次模块的实现进行了关键介绍。

# 第4章 爬虫测试与数据展示

本章介绍爬虫的测试环境、运行状态、抓取结果，并给对结果进行数据分析，以研究微博主的发布动作偏好、微博信息特征、微博信息关键词和微博主影响力。

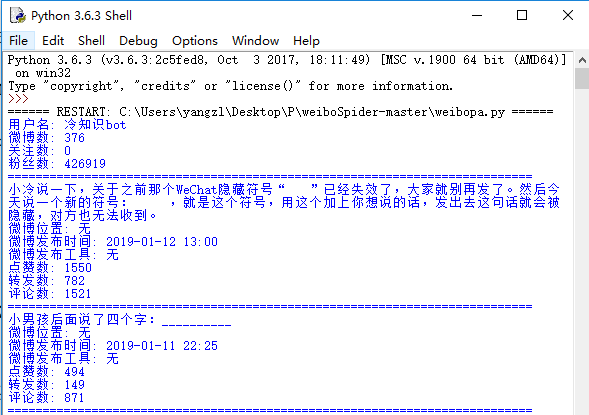
## 4.1 测试环境

爬虫的测试环境：

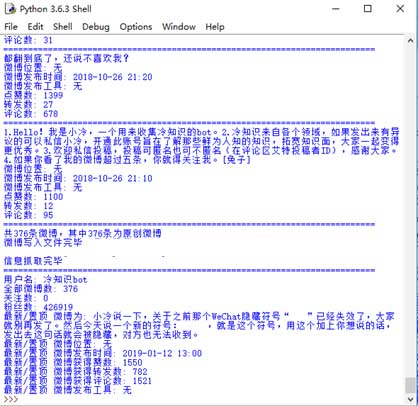
1. 操作系统：windows 10 专业版
2. 硬件配置：Intel Core i5-3317U @1.70GHz 8G RAM
3. 网络环境：校园网
4. 软件环境： Python 3.6

## 4.2 运行状态与数据抓取结果展示

运行时的状态，图4-1所示。每次抓取的微博内容，均能在终端中显示。爬取结束后写入文件。

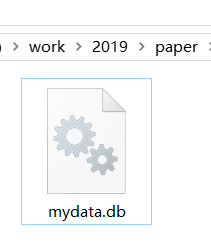


1. 运行状态图

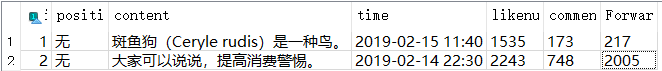


1. 运行结束图

本文对微博ID为3911558393的用户进行抓取后，数据写入到Sqlite数据库中，如图4-3所示。它的内容如4-4图所示，包括微博内容、位置、发布的时间、点赞转发数、评论数和使用工具）。



1. 导出信息图



1. 抓取信息内容

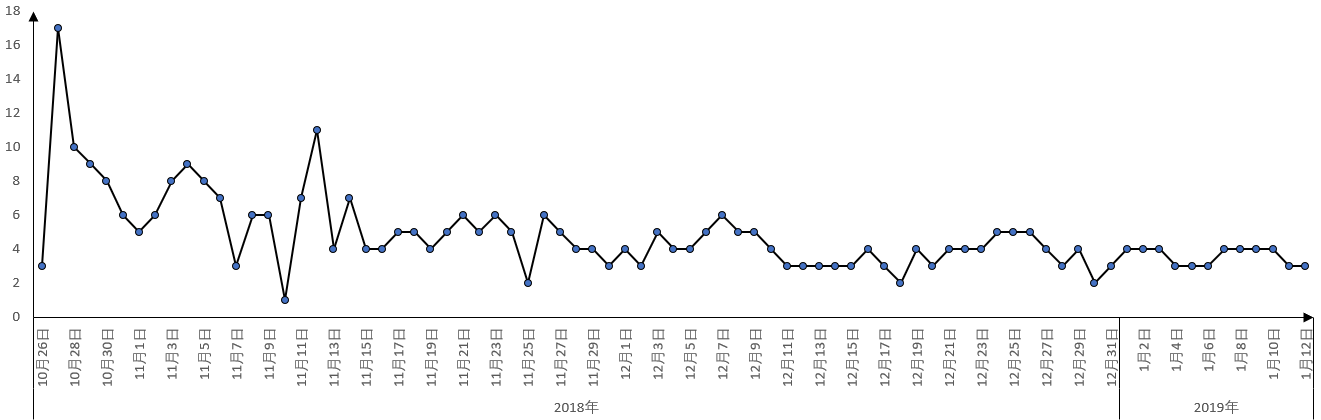
## 4.3 数据分析

本文针对新浪微博“冷知识bot”进行研究分析，抓取了其全部微博内容和其他基本信息。在获取博主的博主号和注册时间同时获取到该博主共发布了396条微博信息，有粉丝426919人，累计获得点赞数522217次、转发数254394次、评论数193948次。

### 4.3.1微博发布动作偏好

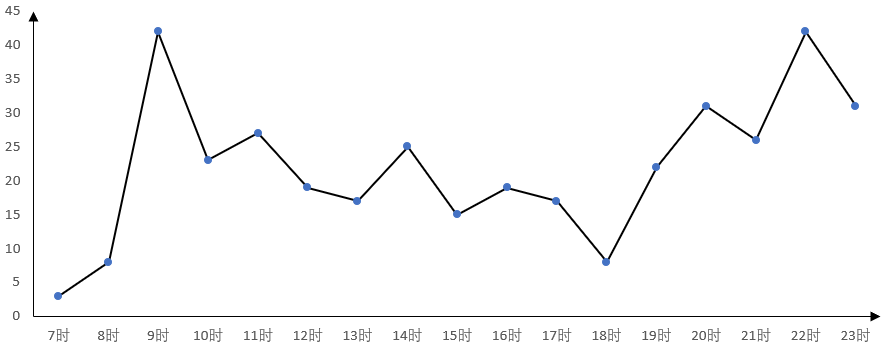
该部分主要研究微博主发布微博信息的次数、发布日期、发布时间等信息，从而了解微博主的信息发布特征。

如图4-5所示，横坐标轴为日期，其范围为2018年10月26日至2019年1月12日，横坐标文本间隔1天显示；纵坐标轴为发布的微博次数。从图中可以看出，该微博主每天均发布微博信息，平均每日发布微博4.7条，其中2018年每日发布微博4.9条，而2019年这一数值降为3.58。同时该微博主2018年11月14日之前，发布微博次数波动较大，平均值为7.05，该段数据在总体数据中的样本方差为11.63；而2018年11月14日以后的微博次数平均值为3.98，该段数据在总体数据中的样本方差为0.98。



1. 微博发布次数图

如图4-6所示，横坐标轴为发表时间，纵坐标轴为微博发布的时间。从图中可以看出，该微博主发布微博信息的时间集中在每日的7时到23时；除2019年1月1日凌晨，微博主发布了一个新年微博外，在23时至次日7时的时间段内没有发表任何内容。同时，早上9时和晚上22时是微博发布的高峰时间，这两个时间均发布了42次，两者占整个发布次数的22.34%，接近1/4；早上7时、早上8时和晚上18时是微博发布的低谷时间，这三个时间分别发布了3次、8次、8次；其余时间段内，微博发布数量在15-32之间。



1. 微博发布时间图

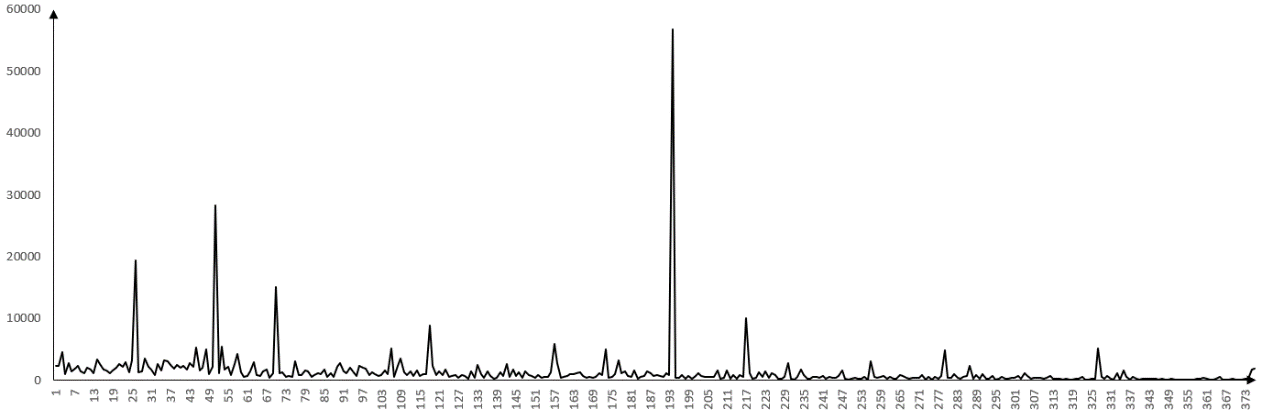
从微博发布日期和时间来看，微博主从刚开始发布次数波动较大，逐渐形成了固定的模式，即每天发布微博信息3-4条，时间较集中在早上9时和晚上23时。

### 4.3.2微博信息特征分析

该部分主要研究微博主发布的微博信息具有的特征，主要通过以下三个因素进行分析：

（1）点赞数分析

如图4-7所示，通过按照微博发布时间的次序，列出每条微博的点赞数。从该图可以看出，个别微博的点赞数极高，而大多数的微博点赞数基本差别不大，小于5000。为了更好地观察，我们按照点赞数大小进行排序，如图4-8所示，其横坐标为排名次序，纵坐标为点赞数。该图显示微博点赞数服从幂律分布[18]，即少数高赞微博受到人们的极大关注，而图中有一个长长的尾巴，说明其他微博受关注度一般。该分布又称长尾理论，另外一个通俗解释就是马太效应，即穷者越穷富者越富。



1. 微博点赞数按时间分布图



1. 微博点赞数按大小排序图

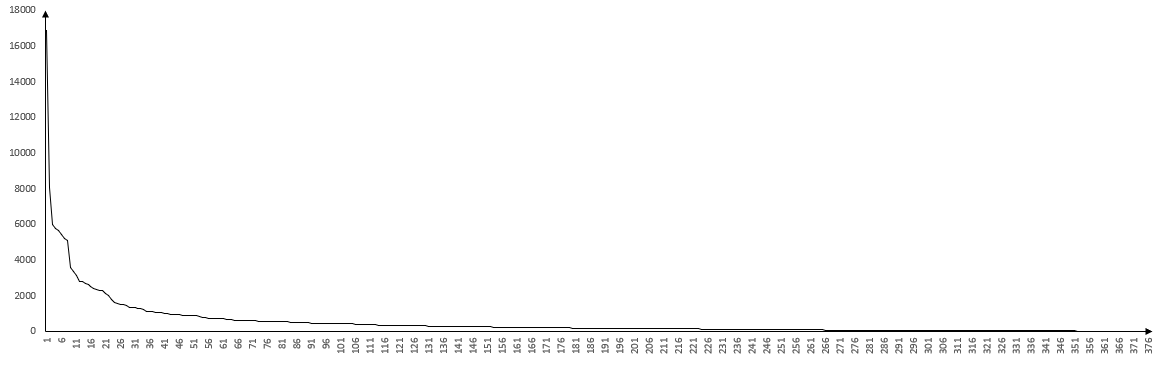
如表4-1所示，本文列举了点赞量排名前十的微博。该微博主共发布微博376个，全部点赞数为522217。排名第一的微博点赞数为56757，占全部微博点赞数的10.87%，这是一个极高的数字。排名第二的微博点赞数为28322，占全部微博点赞数的5.42%，近乎是第一名的一半。排名前四的微博点赞数占全部点赞数的22.88%，而排名前十的微博点赞数占全部点赞数的30.68%。可以看出，少数点赞量高的微博吸引了大量的关注，而众多的其他微博表现相对一般。

1. 点赞量排名前十的微博

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **发布日期** | **发布时间** | **点赞数** | **微博内容** | **比重** |
| 194 | 2018/11/23 | 19:10:00 | 56757 |  | 10.87% |
| 51 | 2018/12/29 | 14:20:00 | 28322 |  | 5.42% |
| 26 | 2019/1/5 | 22:20:00 | 19403 |  | 3.72% |
| 70 | 2018/12/24 | 22:15:00 | 15025 |  | 2.88% |
| 217 | 2018/11/19 | 9:45:00 | 10073 |  | 1.93% |
| 118 | 2018/12/10 | 21:10:00 | 8827 |  | 1.69% |
| 157 | 2018/12/2 | 15:31:00 | 5902 |  | 1.13% |
| 53 | 2018/12/28 | 22:30:00 | 5424 |  | 1.04% |
| 45 | 2018/12/31 | 13:40:00 | 5271 |  | 1.01% |
| 327 | 2018/10/31 | 14:54:00 | 5220 |  | 1.00% |
| 合计 | | | | | 30.68% |

（2）评论数分析

图4-9横坐标为排名次序，纵坐标为每条微博的评论数。从该图可以看出，它同样服从幂律分布，即存在少说评论数高的微博，而众多的其他微博的评论数相对差异不大，在图中形成一个长尾。



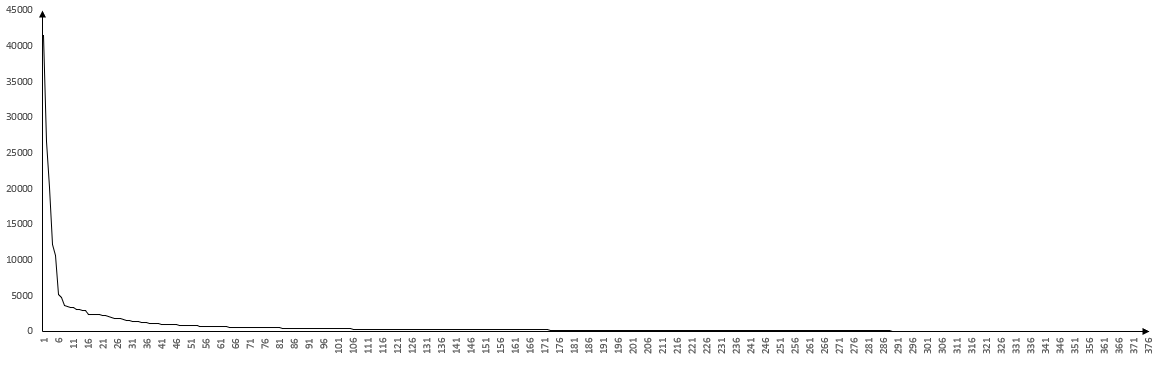
1. 微博评论数按大小排序图

如表4-2所示，本文列举了评论数排名前十的微博。该微博主共发布微博376个，全部评论数为193948。排名第一的微博评论数为16840，占全部微博点赞数的8.68%。排名第二的微博评论数为8084，占全部微博点赞数的4.17%，比第一名的一半还少。排名前十的微博评论数占全部评论数的35.09%，表现出了比微博点赞数更加强的聚集性。比较表4-1与4-2，我们可以看出，评论数最高的微博并不是点赞数最高的微博，两者并不是必然的联系；但是评论数排名前十的微博，有5个进入微博点赞量前十。

1. 评论数排名前十的微博

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **发布日期** | **发布时间** | **评论数** | **微博内容** | **比重** |
| 26 | 2019/1/5 | 22:20:00 | 16840 |  | 8.68% |
| 51 | 2018/12/29 | 14:20:00 | 8084 |  | 4.17% |
| 198 | 2018/11/22 | 23:00:00 | 5976 |  | 3.08% |
| 217 | 2018/11/19 | 9:45:00 | 5789 |  | 2.98% |
| 3 | 2019/1/12 | 13:00:00 | 5625 |  | 2.90% |
| 209 | 2018/11/20 | 23:00:00 | 5403 | 北方人问一个问题，南方人来回答\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | 2.79% |
| 194 | 2018/11/23 | 19:10:00 | 5213 |  | 2.69% |
| 214 | 2018/11/19 | 23:00:00 | 5100 |  | 2.63% |
| 53 | 2018/12/28 | 22:30:00 | 3584 |  | 1.85% |
| 11 | 2019/1/9 | 22:40:00 | 3343 |  | 1.72% |
| 185 | 2018/11/25 | 23:00:00 | 3109 |  | 1.60% |
| 合计 | | | | | 35.09% |

（3）转发数分析



1. 微博转发数按大小排序图

图4-10横坐标为排名次序，纵坐标为每条微博的转发数。从该图可以看出，它同样服从幂律分布，即存在少数转发数高的微博，而后转发数极速下降，而众多的其他微博的转发数相对差异不大，在有些微博转发数为0，最终在图中形成一个长尾。

如表4-2所示，本文列举了转发数排名前十的微博。该微博主共发布微博376个，全部转发数为254394。排名第一的微博转发数为41492，占全部微博点赞数的16.31%；排名第二的微博转发数为26533，占全部微博点赞数的10.43%，比重下降了1/3多。排名前三的微博转发数占全部转发总数的34.65%，而排名前十的微博转发数占全部转发总数的51.50%，表现出了比微博点赞数、微博评论数更强的聚集性。同时，转发数排名前十的微博，有7个进入微博点赞量前十，有5个进入微博评论量前十。

1. 转发数排名前十的微博

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **发布日期** | **发布时间** | **转发数** | **微博内容** | **转发比重** |
| 194 | 2018/11/23 | 19:10:00 | 41492 |  | 16.31% |
| 70 | 2018/12/24 | 22:15:00 | 26533 |  | 10.43% |
| 26 | 2019/1/5 | 22:20:00 | 20118 |  | 7.91% |
| 217 | 2018/11/19 | 9:45:00 | 12178 |  | 4.79% |
| 51 | 2018/12/29 | 14:20:00 | 10565 |  | 4.15% |
| 177 | 2018/11/27 | 10:20:00 | 5122 |  | 2.01% |
| 157 | 2018/12/2 | 15:31:00 | 4670 |  | 1.84% |
| 173 | 2018/11/28 | 10:00:00 | 3562 |  | 1.40% |
| 53 | 2018/12/28 | 22:30:00 | 3459 |  | 1.36% |
| 287 | 2018/11/6 | 9:52:00 | 3319 |  | 1.30% |
| 合计 | | | | | 51.50% |

### 4.3.3微博关键词分析

为了更好地分析微博内容，本文对抓取到的微博内容进行分词，并制作了词云，如图4-11所示。在众多关键词中，“中国”、“发现”、“知识”、“世界”、“时间”、“喜欢”、“名字”等词占据着重要位置。



1. 微博信息内容词云

微博内容关键词分析的主要步骤：

（1）抓取所有微博内容：主要利用第三章设计实现的爬虫完成；

（2）中文分词：将整个微博内容的文本通过某种方法切分为多个有意义的词语，方便下一步的数据分析；这部分功能的实现使用了python中的jieba库，jieba库是一个优秀的中文分词的第三方库。主要依赖中文词库，确定汉字之间的关联概率，从而形成分词结果；

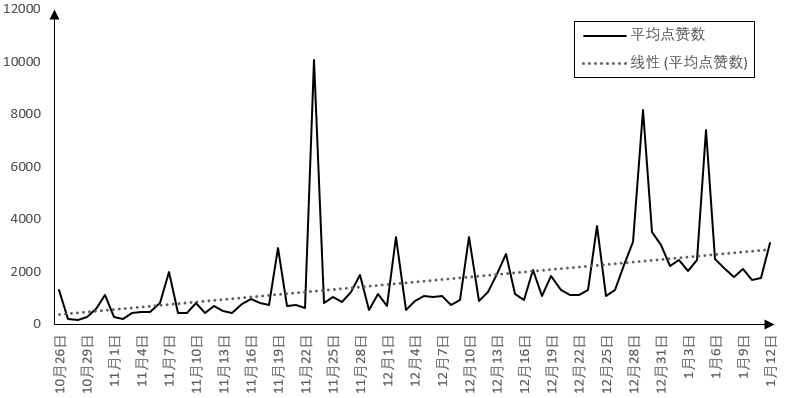
（3）文本清洗：文本切分之后会出现一些特殊符号甚至一些没有意义的词语，这些词语会对之后分析造成一定的影响。所以我们使用用清洗文本库stopwords.txt，将切分出来的符合条件的分词剔除掉；

（4）绘制词云：该部分利用python中的词云库WordCloud来完成，它是一个优秀的关键词云展示库，得到了广泛应用。

### 4.3.4微博主影响力分析

为了评价微博主“冷知识bot”的影响力，本文认为每日每条微博信息的点赞数、评论数和转发数等参数，能够在一定程度上反应微博主的受欢迎程度。接下来，我们分别对这三个参数分别进行研究，接着提出了一种综合评价的方法。

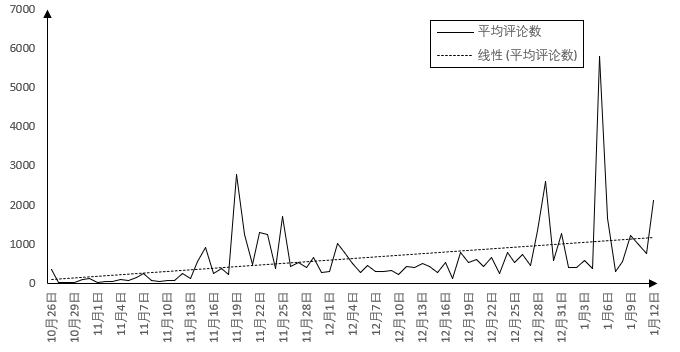
（1）平均点赞数

本文将每日每条微博信息的点赞数称之为平均点赞数，由（当日微博总点赞数/微博个数）计算而来。从图4-12可以看出，平均点赞数在某些微博上会出现极高值，而这些微博就是节4.3.2讨论过的高赞微博；同时，图中增加了线性趋势线，用点线来表示；它的值随着时间的推移而逐渐增大。因此可以说，微博主发布的微博得到更多受众的称赞。

1. 平均点赞数按照日期分布图

（2）平均评论数

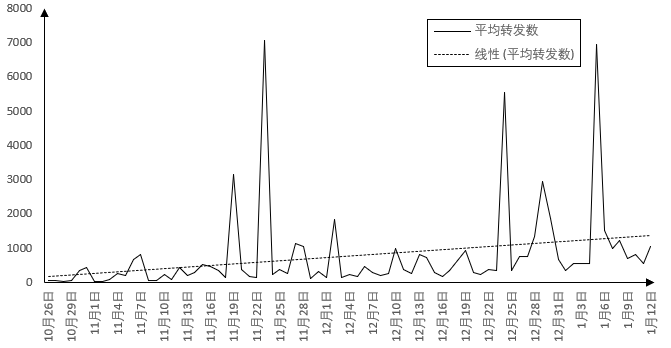
本文将每日每条微博信息的评论数称之为平均评论数，由（当日微博总评论数/微博个数）计算而来。从图4-13可以看出，平均评论数会在个别些微博上会出现极高值；从线性趋势线的增长情况来看，微博的评论数随着时间的推移而逐渐增大。所以，微博主发布的微博得到更多人的评论。



1. 平均评论数按照日期分布图

（3）平均转发数

本文将每日每条微博信息的转发数称之为平均转发数，由（当日微博总转发数/微博个数）计算而来。与图4-12和图4-12类似，图4-14中出现了少数微博具有超高平均转发数；且其线性趋势线成增长状态。可以看出，微博主发布的微博得到更多人的转发，具备了更大的影响力。

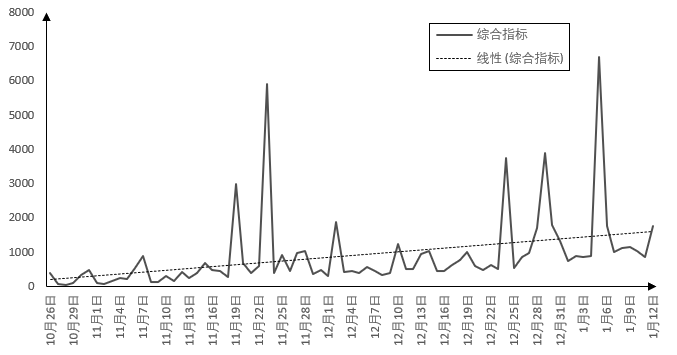


1. 平均点赞数按照日期分布图

（4）综合指标

为综合评价微博主的影响力，本文综合平均点赞数、平均评论数和平均转发数等指标，采用加权平均值方法进行综合评价。综合指标=0.5\*平均转发数+0.3\*平均评论数+0.2\*平均点赞数。这里的加权值主要考虑到，平均转发数反映了微博的传播能力，影响较为深远；同时一般而言，转发微博的难度>评论的难度>点赞的难度。随着日期的增长，综合指标如图4-15所示，少数微博具有远超其他微博的超高值，且其线性趋势线成增长状态。

这反映出微博主的每天微博得了更多人的关注和传播，其影响力仍处于逐渐增长的状态。



1. 综合指标按照日期分布图

## 4.4 本章小结

本文实现的微博网络爬虫能有效获取到新浪微博的数据，实现方法也较为简单，代码稳定也能较好的的避免反爬虫机制的检测；接着，针对微博发布动作偏好、微博信息特征、微博关键词、微博主影响等四个方面展开了分析。

# 第5章 总结与展望

该文设计并实现了新浪微博爬虫程序，对微博主“冷知识bot”发布的微博内容进行了全部抓取，并从微博发布动作偏好、微博信息特征、微博关键词、微博主影响等四个方面，进行了数据分析。

## 5.1 工作总结

该爬虫能够直接获取已登录用户的Cookie，避免使用其他繁琐的模拟登录方法；基于lxml解析微博数据，简化了数据筛选的操作。经过数据分析，本文得到如下信息：

（1）微博主发布的微博模式较为固定；

（2）博主的微博的点赞数、评论数以及转发数服从幂律分布；

（3）博主发布的微博内容的关键词主要有“中国”、“发现”、“知识”、“世界”、“时间”、“喜欢”、“名字”等；

（4）该微博主的影响力具有逐渐增大的趋势。

但该爬虫程序的主要缺陷是爬取效率不高，不适用于海量数据爬取，可以通过在Scrapy框架的基础上使用多线程爬取，用户可根据需要进行适当改动。数据分析方面，由于新浪微博对粉丝200人数抓取的限制，未能大规模地抓取“冷知识bot”的粉丝信息，缺乏对粉丝的深入分析与研究。

## 5.2 研究展望

为了更好地进行微博数据分析，未来还需要在数据抓取和数据分析上下功夫：

（1）结合Scrapy爬虫框架[19]，设计分布式爬虫，能够进行多线程同时抓取；

（2）适当地使用新浪Api实现对微博主的粉丝信息进行获取；

（3）采用代理IP等多种方式，实现新浪微博的防屏蔽方式，规避新浪微博的拒绝爬取行为；

（4）可结合Topsis等方法[20]，对微博主发布微博进行影响力排序，以更深入研究微博的影响力；

（5）可结合复杂网络等知识，分析研究微博主的分析相互关注信息，从网络的角度研究微博主的影响。

参考文献

1. 基于移动云的社交网络可视分析关键技术的研究[D]. 武汉理工大学, 2015.
2. 张爱国. 基于大数据的数据挖掘技术与应用[J]. 现代工业经济和信息化, 2017, 7(3):86-87.
3. Halavais, Alexander. Search engine society[M]. John Wiley & Sons, 2017.
4. 周德懋, 李舟军. 高性能网络爬虫:研究综述[J]. 计算机科学, 2009, 36(8):26-29.
5. 刘金红, 陆余良. 主题网络爬虫研究综述[J]. 计算机应用研究, 2007, 24(10):26-29.
6. 魏冬梅,何忠秀等 基于python的web信息方法获取研究[J].软件导刊,2018. 安子建.
7. 吴睿, 张俊丽. 基于 R 语言的网络爬虫技术研究. 科技资讯. 2016;14(34):35-6.
8. 基于Scrapy框架的网络爬虫实现与数据抓取分析[D].
9. 罗咪. 基于Python的新浪微博用户数据获取技术[J]. 电子世界, 2018.
10. 陈智, 梁娟, 谢兵, et al. 新浪微博数据爬取研究[J]. 物联网技术, 2016(12).
11. 陈琳, 任芳. 基于Python的新浪微博数据爬虫程序设计[J]. 信息系统工程, 2016(9):97-99.
12. Heydon A, Najork M. Mercator: A scalable, extensible web crawler.[J]. World Wide Web, 1999, 2(4):219-29.
13. 胡松涛. Python 网络爬虫实战[M]. 清华大学出版社, 2017.
14. Cole RA, Goldberg LG, White LJ. Cookie cutter vs. character: The micro structure of small business lending by large and small banks. Journal of financial and quantitative analysis. 2004 Jun;39(2):227-51.
15. Chandra RV, Varanasi BS. Python requests essentials[M] . Packt Publishing Ltd; 2015.
16. Lawson R. Web scraping with Python[M] . Packt Publishing Ltd; 2015.
17. Owens M. The definitive guide to SQLite[M] . Apress, 2006.
18. 罗斌, 陈翔. 幂律特性在新浪微博个性化推荐中的应用研究[J]. 计算机工程与科学, 2018, v.40；No.280(04):165-173.
19. 安子建. 基于Scrapy框架的网络爬虫实现与数据抓取分析[D]. 吉林大学, 2017.
20. Behzadian M, Khanmohammadi Otaghsara S, Yazdani M, et al. A state-of the-art survey of TOPSIS applications[J]. Expert Systems with Applications, 2012, 39(17):13051-13069.

# 致 谢

在本次论文撰写与作品设计过程中，谈斯聪老师从论文选题到最后的定稿每个环节的细心指导，使我顺利完成最后的毕业设计。不但提高了我个人的技术能力也让我感受到了导师高度的敬业精神和责任感。

同时我也要感谢学校和学校的领导给我们创造的良好的学习环境，让我们能够在良好的氛围中完成论文。感谢同学和家人在这期间的大力支持。