微博热门话题分析

《计算机网络技术与实践》开放性实验

罗惠文（2013011057）　单萌（2013011065）　李智涛（2013011053）

1 简介

1.1 背景

在新浪微博中，每天都有许多热门话题引发微博用户的广泛讨论。分析这些热门话题的特征（如粉丝在地理上的分布，粉丝间组成的社交网络等），对于发现新浪微博中有相似兴趣的用户群体，理解信息在社交网络中传播的行为有重要意义。

使用网页爬虫技术对新浪微博进行爬取，得到热门话题的相关信息。利用这些信息分析该热门话题的特征。

1.2 组内分工

李智涛：爬虫的编写与信息的提取

罗惠文：用户信息分布的统计及分析

单　萌：用户社交网络的统计及分析

2 实验方案设计

2.1 实验内容

使用爬虫获取新浪微博某一热门话题下的讨论用户和粉丝的信息，并做统计分析。

2.2 数据结构

参与讨论的用户信息以如下数据结构进行存储：

每行表示一个参与了热门话题讨论的用户的信息，行内相邻两列之间用4个空格分隔，第一列为该用户发表讨论的时间，第二列表示该用户是否为微博认证用户（是为Y，不是为N），第三列为该用户的uid（用户的唯一标识），第四列为用户的昵称，第五列为用户的所在地，第六列为用户的性别，第七列为用户的出生日期，第八列起到末尾由一对中括号括起来，其中每一项均为该用户关注的一个用户的uid。

关注话题的用户信息存储方式除无第一列发表讨论时间外与讨论用户相同。

2.3 分析内容

统计关注话题的粉丝和讨论用户的地点分布、年龄分布、性别分布和大V的比例。

分析关注用户和讨论用户各自组成的社交网络（每个用户为一个节点，用户间的关注关系为有向边），统计节点的出入度的分布，社交网络中的孤岛个数（内部连通且与外部无连接），平均每个孤岛的节点数等数据特征。同时统计上述特征随时间的变化。

分析大V在热门话题传播中的作用，观察大V参与话题前后关注量增长曲线的斜率，分析前后关注者中为大V粉丝的比例变化。

3 实验内容

3.1 爬虫设计

爬虫的实质，是模拟一个浏览器向服务器发送和接受消息，通过对收到的消息中信息的提取，实现在网页之间的爬动和相应信息的抓取。在本实验中，为简化起见，省略了爬虫模拟登陆新浪微博的步骤，采用浏览器登陆新浪微博后，将微博账号存储在本地的cookie直接填入程序中再进行爬取的方法。

本次实验中爬虫需要完成的功能有：获取一个热门话题下的参与讨论用户列表及讨论时间；获取关注某一热门话题的用户列表；对用户列表中的每一用户，获取其个人信息（所在地、年龄等）与其关注的用户列表。

根据上述描述，设计爬虫结构如表1所示。

函数名	功能描述
get_hot_topic_discuss_information	获取热门话题下讨论用户的所有信息
get_hot_topic_fans_information	获取关注热门话题的用户的所有信息
get_user_info_by_id	获取指定uid的用户个人信息与关注用户
get_user_personal_info	获取指定uid的用户的个人信息
get_user_friends_by_page_id	获取指定uid的用户的所有关注用户

表 1

运行爬虫时，首先将账号的cookie填入，并在get_hot_topic_discuss_information或get_hot_topic_fans_information中修改url为目标热门话题的url即可。

在实际实现过程中，在确定待爬取网页的url后，生成一个urllib2模块的Request类的实例，将浏览器请求页面时发送的http头部信息填入到该类中，比如“Host”、“User-Agent”、“cookie”等信息。然后调用urllib2模块的函数urlopen来建立与服务器的连接，连接建立后，就可以通过read()操作获取服务器返回的消息了。

获取到服务器的应答消息后，需要对其进行处理，提取出其中我们所关注的信息。以请求用户个人信息的页面为例，其请求的页面url是http://m.weibo.cn/users/%s/?，其中%s项为目标用户的uid。根据上述步骤获取该页面的html后，通过观察可以发现，其基本信息都会有固定的首尾标识，如性别以形式“性别 男 ”标识，所在地以形式“所在地 地点 ”标识等。在消息中查找所有上述信息并提取其中的关键字，即可获得目标信息。在获取用户的关注用户列表时，由于爬取的是移动版微博，因此其加载多页信息不是通过翻页方式，而是通过页面滚动来自动请求一个新的url的资源来实现的，其url的一般形式为http://m.weibo.cn/page/json?containerid=%s_-_FOLLOWERS&page=%d，其中%s项为目标用户的pageid，在其用户首页对应的消息中可以获得该项信息，%d项标识当前的页码。在请求热门话题的所有粉丝时，其请求的url中包含有nextcursor一项，根据实际测试可知，该项可忽略，不影响获取效果。

3.2 社交网络分析设计

首先利用 C 语言进行数据整理。首先进行去重，见 quchong.cpp。定义一个类 info，包含自己的 uid，关注人的 id 向量 vector<int>guid，和讨论时间（粉丝部分没有该项）及是否为大 V。使用流读入数据到 info 数组 vector<info> user，然后使用 sort 函数按时间顺序进行排序（粉丝那部分没有这项），最后调用函数做出有向图 G，G[i][j] = 1 代表了第 i 个人的关注列表中有 j，见 sort_info.h 和 sort_info.cpp。还同时得到了是否为大 V 的列表，都保存在 TXT 文件中供 MATLAB 调用，见 getV.cpp。

接下来是 MATLAB 处理的步骤（以更复杂的 discuss 讨论为例）。首先读入刚才产生的文件，然后进行时间分布的处理。调用 time.m 函数，以一小时为间隔进行划分，可以观察每小时内产生的讨论数量和大 V 加入讨论的时间点，据此可以分析大 V 在话题讨论产生中的带动作用。再进行关系网随时间分布的分析。使用 MATLAB 自带的 graph 工具箱，我们可以轻易地获得刚刚所做有向图的一些信息，如出入度，孤岛个数/大小等，具体实现见 get_relation_graph.m，实验中实现了图像的即时保存，更加方便。

4 实验结果及分析

4.1 爬虫结果及分析

通过观察爬虫输出的用户信息文件可知，爬虫功能正常。实验过程中遭遇的问题就是爬取频率过快导致封号、cookie的改变或者网络不稳定导致爬取过程中断。对于前一个问

题，在实验中采用的解决方法是设置爬取频率，通过实际测试，每7秒爬取一个网页可防止封号情况的发生。对于后者，在爬取过程中断后需要重启爬虫，并视情况修改写入在程序中的cookie，在这种情况下，事先将用户列表保存在本地就显得尤为重要。

在爬虫编写过程中，起初不清楚移动版微博滚动加载页面时是如何请求下一页用户信息的，因此我尝试了爬取电脑版微博的方法。但是电脑版微博反而有一个限制，就是对于没有被我所关注的用户，其关注用户列表我只能浏览前5页，这就导致用户关注信息获取不全的问题。后经过助教指导，了解了移动版微博请求下一页用户信息时是通过pageJson对应的一个url来进行的，由此顺利解决了这一问题。

最终由于时间有限，最终爬得的数据与话题的对应关系如表2所示。

文件名	对应的热门话题	类型
discuss_information3.txt	#530网红节#	讨论用户信息
discuss_information4.txt	#六一儿童节#	讨论用户信息
discuss_information5.txt	#女不强大天不容#	讨论用户信息
discuss_information6.txt	#你好六月#	讨论用户信息
fans_information1.txt	#开张票一小时#	关注用户信息
fans_information2.txt	#奔跑吧兄弟#	关注用户信息
fans_information3.txt	#530网红节#	关注用户信息
fans_information6.txt	#你好六月#	关注用户信息

表 2

4.2 信息分布统计结果及分析

4.2.1 话题#530网红节#粉丝的年龄分布、性别分布、大V分布与地理分布

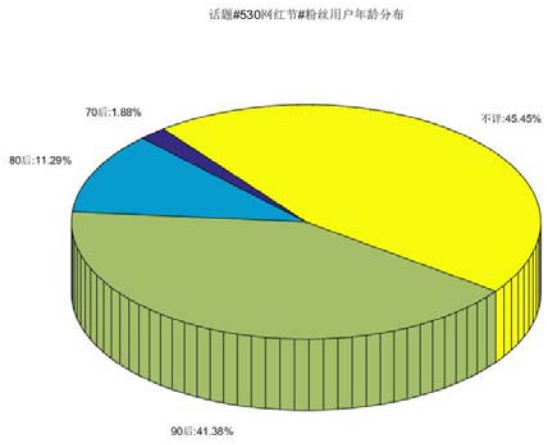

图 1 #530网红节#粉丝年龄分布图

图 2　　#530网红节#粉丝性别分布图与大V比例图

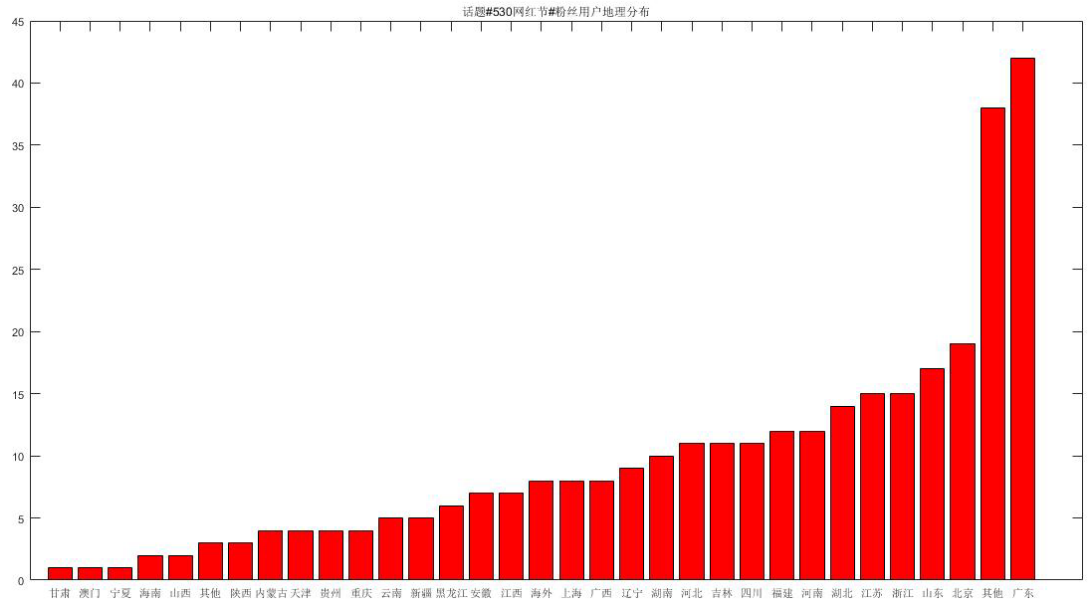

图 3　　#530网红节#粉丝地点分布图

分析：从图 1 中可以看到，除了约 45.45%的粉丝用户没有明确写明出生年月日（星座或者未知）以外，剩下的粉丝为 90 后、80 后和 70 后，其中 90 后与 80 后为主体，可见 90 后和 80 后这些比较年轻的群体喜欢参与这个话题的讨论，猜测是因为关于“网红”这样的话题应该更吸引年轻人。另外，从图 2 性别分布中看出，女性在粉丝群体中更为活跃，说明女性对网红的关注度或者女性对微博的热衷程度要高于男性。图 3 显示广东、其他和北京三个地方的粉丝数为前三位，除去许多用户没有填写地理信息而归入的其他类，可以看到粉丝人数比较靠前的几个省份均是经济较为发达的省份，中西部省份的人数相对较少，这一方面是由于经济发达省份人口数量较其他省份更为庞大，另一方面也是由于其智能手机普及较广，使用人数较多所致。

4.2.2 统计话题#你好六月#讨论用户的年龄分布、性别分布、大V分布与地理分布

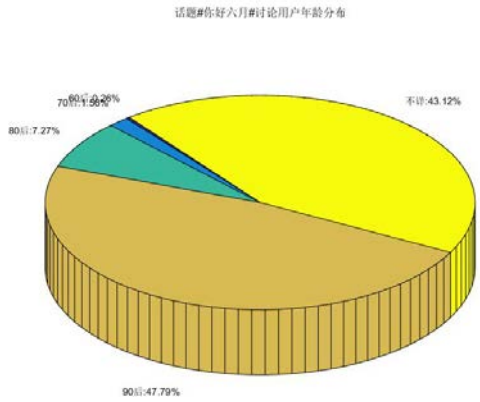

图 4　　#你好六月#讨论用户年龄分布图

图 5　　#你好六月#下讨论用户性别分布图与大V比例图

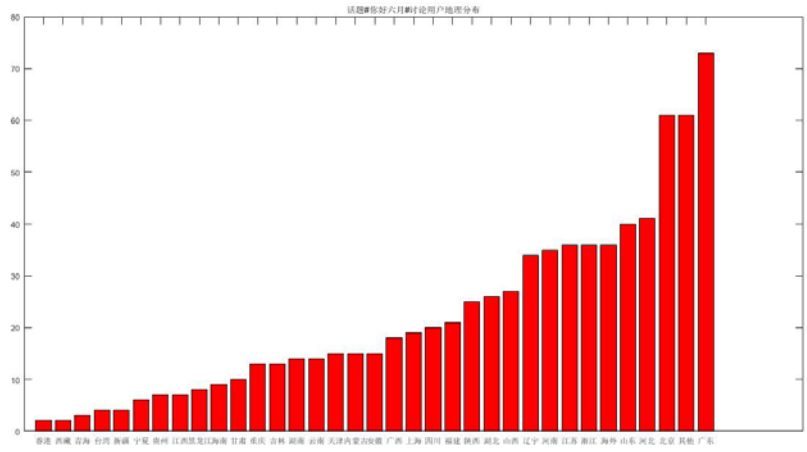

图 6　　#你好六月#讨论用户地点分布图

分析：与#530 网红节#这个话题相比较，参与讨论#你好六月#这个话题的群体年龄分布更多样化，可能是因为这个话题比起前一个话题更适合于大部分年龄段的人，但 90 后仍然非常活跃。另外，参与讨论的人群中，大部分都是女性，大 V 参与讨论话题的人数也比

较大。此外，仍然是广东、北京、其他这三个地方最为活跃，也不排除这三个地方的微博用户本来就比较多的可能。

4.3 社交网络统计结果及分析

在实验过程中，社交网络统计过程中，出现的问题主要集中在利用 C 语言进行数据整理这部分。起初使用二维 string 数组来保存关注人信息，这时候发现当数组过大时会出错导致程序无法执行。因此修改代码，开始使用 vector 的方法来动态分配空间。然而由于最后数据还是过多（800+条微博，很多人的关注人数在 1000 以上），导致程序在初始化 info 数组 user 时突然崩溃。经过单步调试和查询，得知 vector 分配的内存空间是连续的，然后 string 的大小是 32 个字节。结合上学期操作系统的知识，想到了内存是分成了一个一个的块，大小 1MB 左右。这么一算好像确实一个块装不下这些数据所以导致了错误和程序的突然退出。因此将 info 类中所有的 string 改为 int 类型后，有效的减少了所占空间，也在比较时减少了所费时间。

对#530网红节#的粉丝信息进行分析得到的数据如下图7至图9所示。

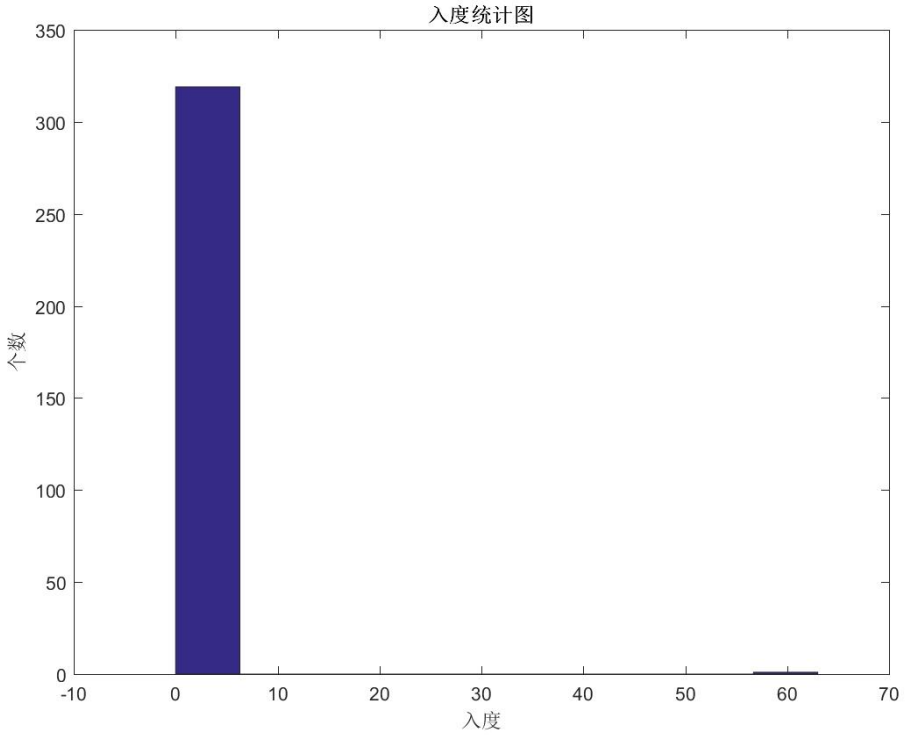

图 7　　#530网红节#粉丝社交网络入度直方图

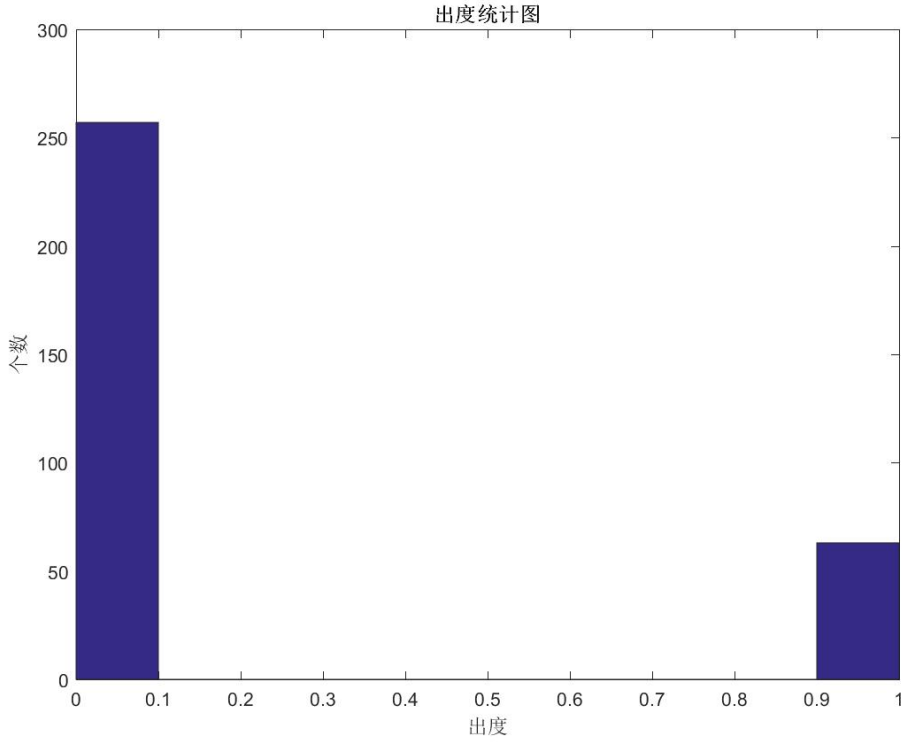

图 8　　#530网红节#粉丝社交网络出度直方图

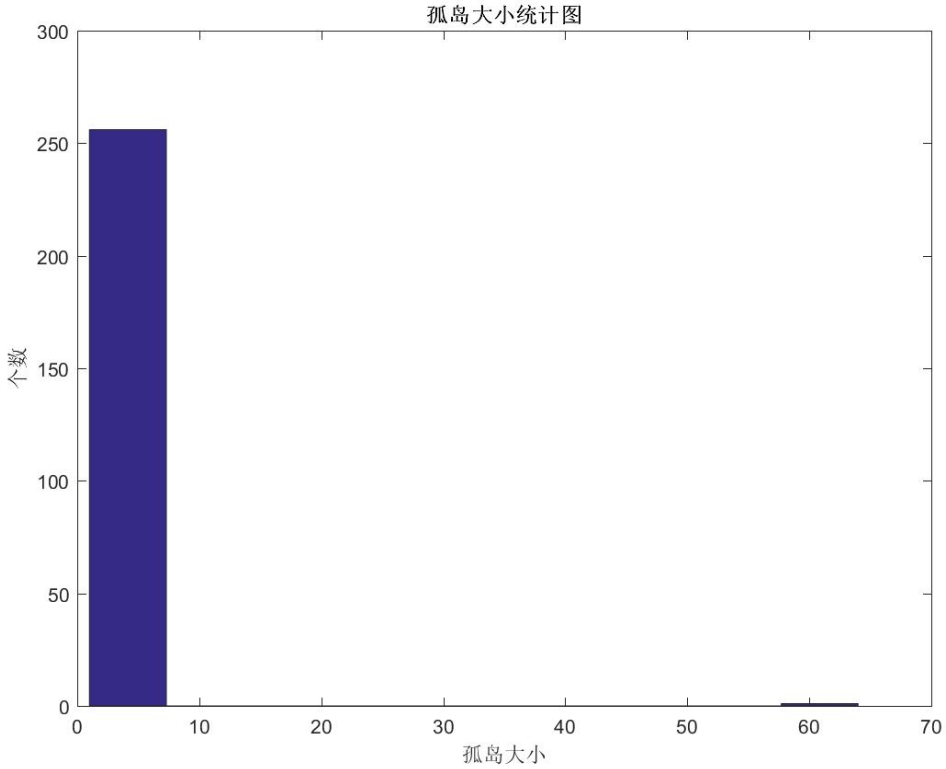

图 9　　#530网红节#社交网络孤岛大小直方图

分析：由于话题粉丝关注的时间无法获得，因此此处不进行随时间变化的分析。但是从这些结果我们可以发现，大多数的人关注的是大 V，由于抓到的大 V 就那么一个，因此大多数人的出度为 0（关注的人没在这里）、一部分人出度为 1（关注的那个大 V）；大多数人入度为 0（没人关注），一个人入度很高（这就是那个大 V）。然后关注大 V 的那些人和大 V 形成了孤岛，大小 60 左右，很多人自己就是孤岛，大小为 1，形单影只。

对#你好六月#话题中参与讨论的用户数量随时间的变化曲线如图 10 所示。

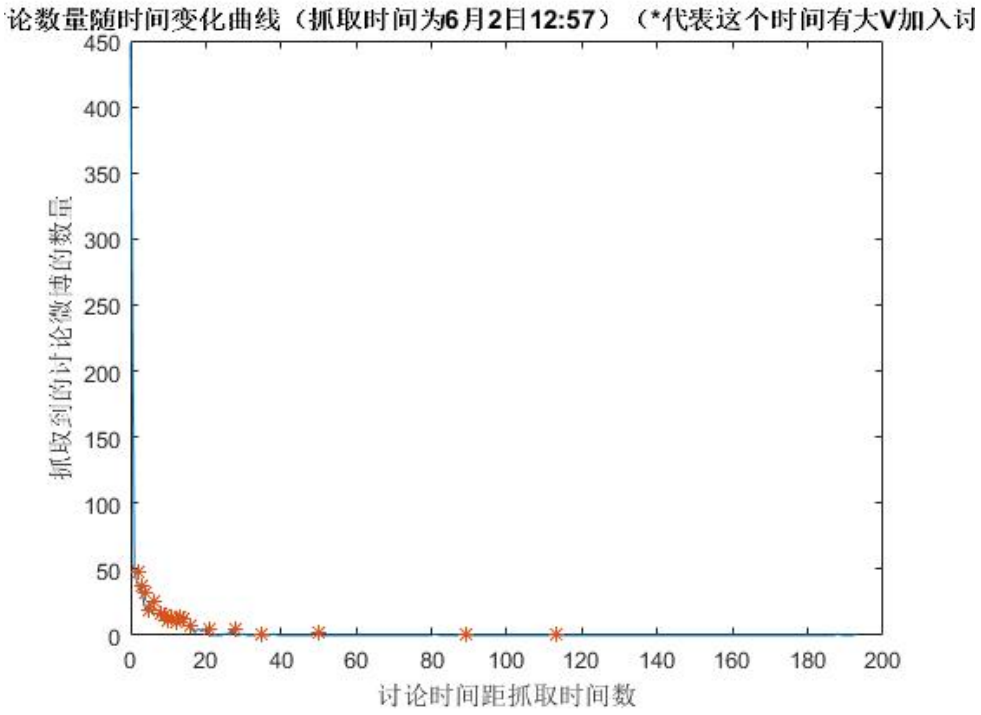

图 10　#你好六月#讨论用户数量变化曲线

对图10中局部进行放大后，如图11。

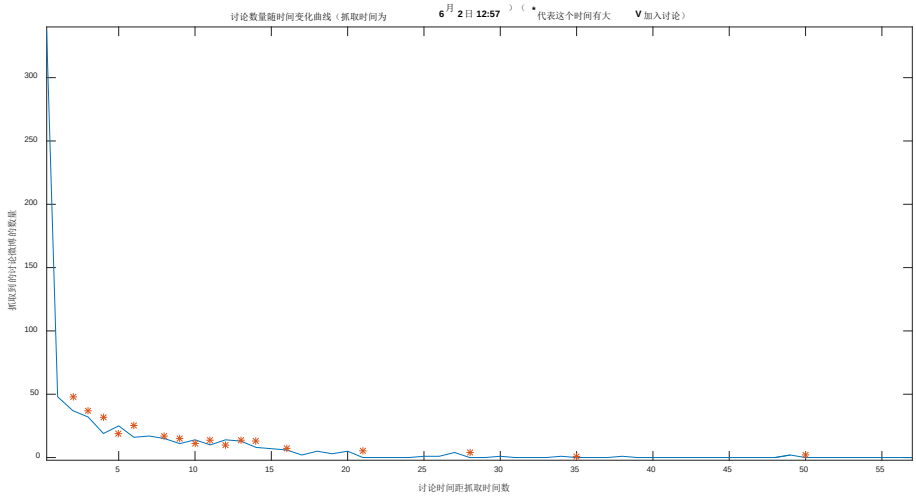

图 11　#你好六月#讨论用户数量随时间变化曲线局部图

分析：可以发现，该热门话题在最近的一小段时间快速火了起来，大概是从 50*20 分钟前有大 V 加入讨论（此处以 20 分钟为间隔划分区间），在 30*20 分钟即 10 个小时前快速火了起来，接下来就以指数增长到抓取时间。

下面分析社交关系网随时间变化的情况。以一小时为间隔进行统计，共得到图片13组，每组3张，由于数量过多，因此此处仅展示一部分，剩余图片见附件。

以发表讨论最早的一个小时内的情况进行统计，得到结果如图12至图14所示。

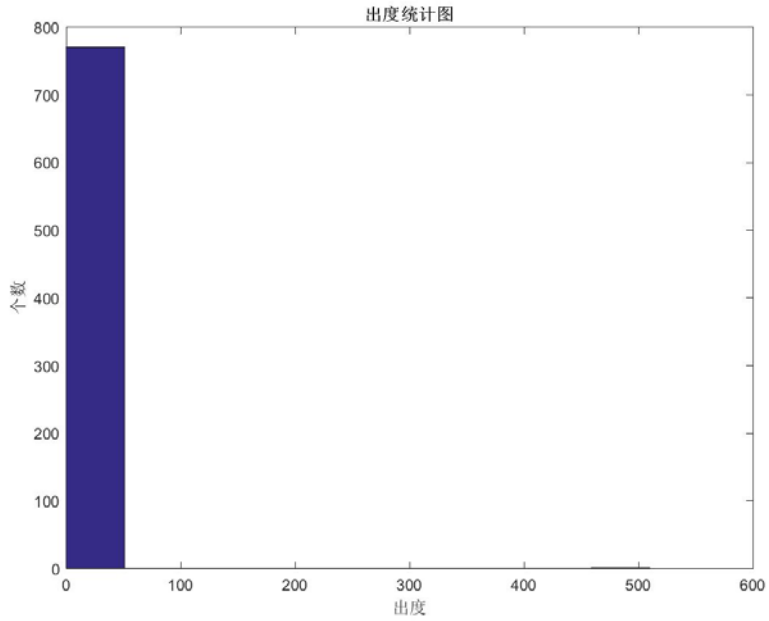

图 12　#你好六月#最初时讨论用户出度数量直方图

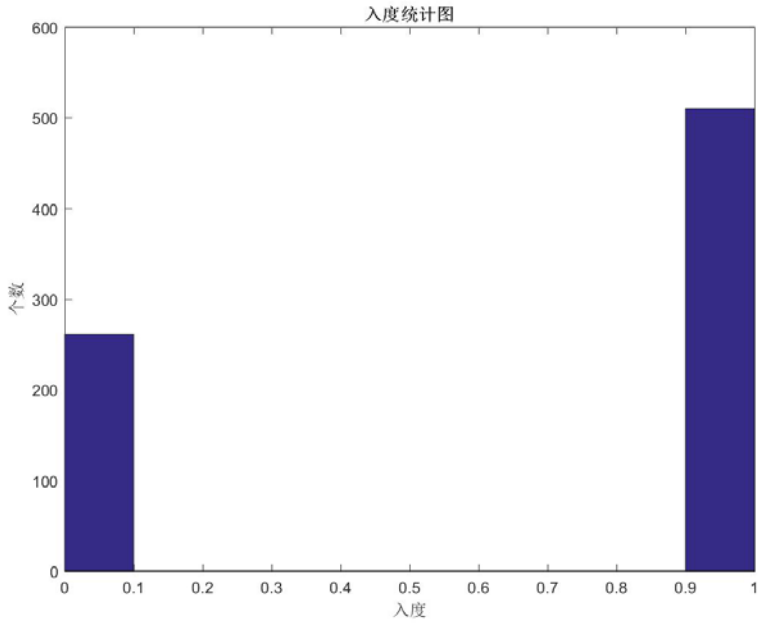

图 13　#你好六月#最初时讨论用户入度数量直方图

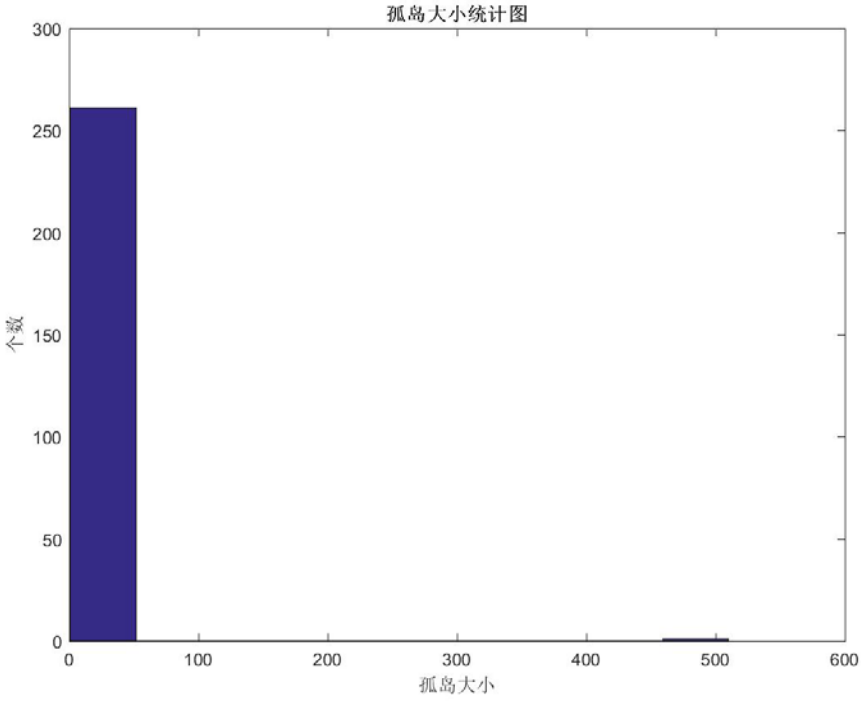

图 14　#你好六月#最初时讨论用户孤岛大小直方图

截止到抓取时间的13个小时前，再次每个结点的出入度分布和孤岛大小，得到结果如图15至图17所示。

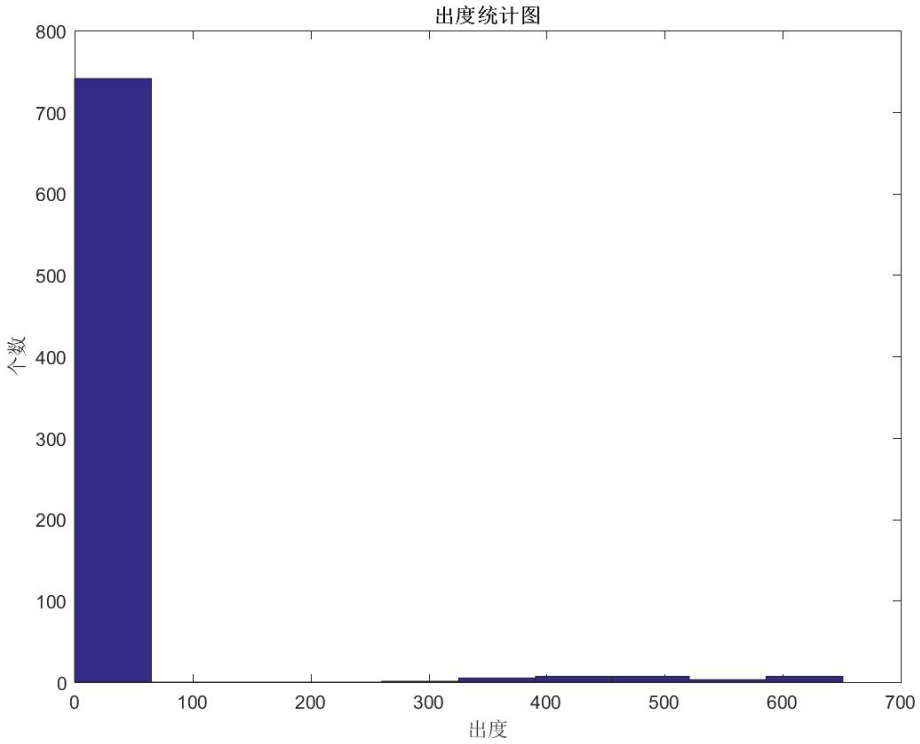

图 15　#你好六月#13小时前讨论用户出度数量直方图

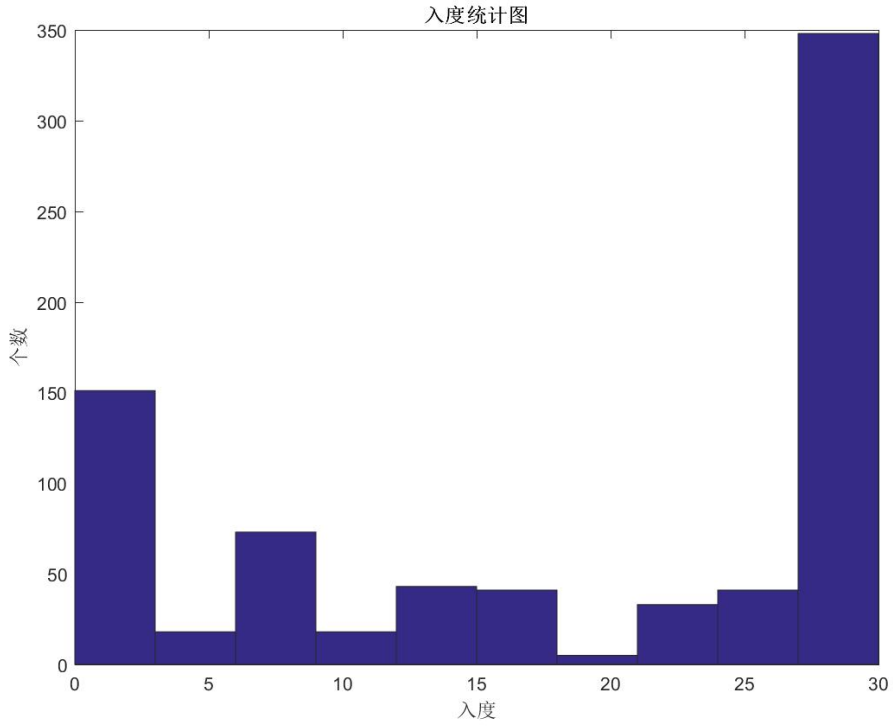

图 16　#你好六月#13小时前讨论用户入度数量直方图

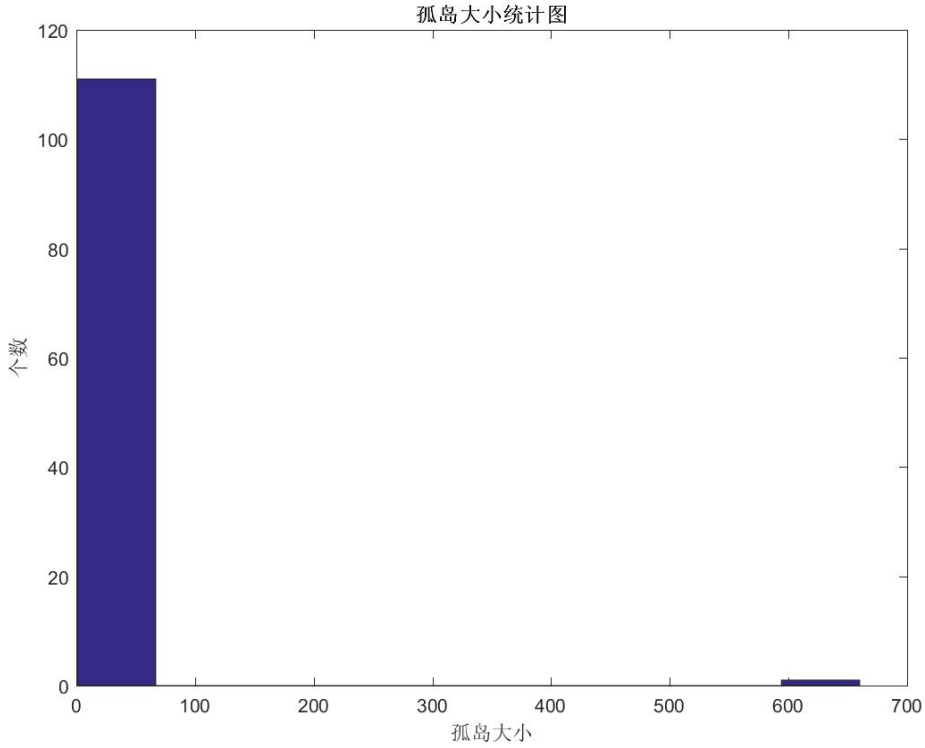

图 17　#你好六月#13小时前讨论用户孤岛大小直方图

可以发现，随着参与讨论用户数量的增多，关系网变得紧密起来了。最后再看看到抓取时间时的关系网，如图 18 至图 20 所示。

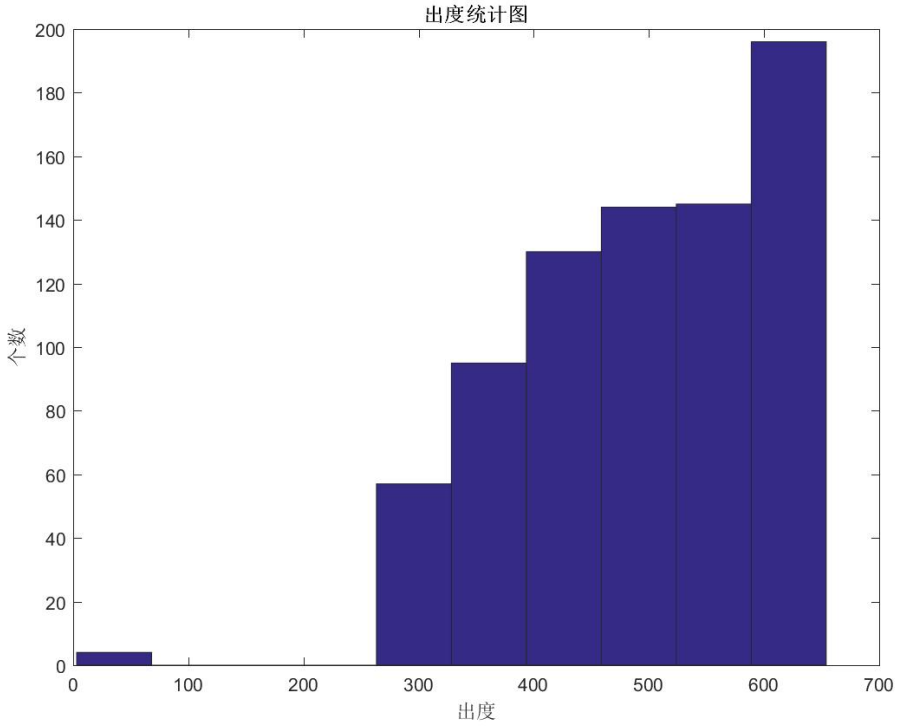

图 18　#你好六月#抓取时间时讨论用户出度数量直方图

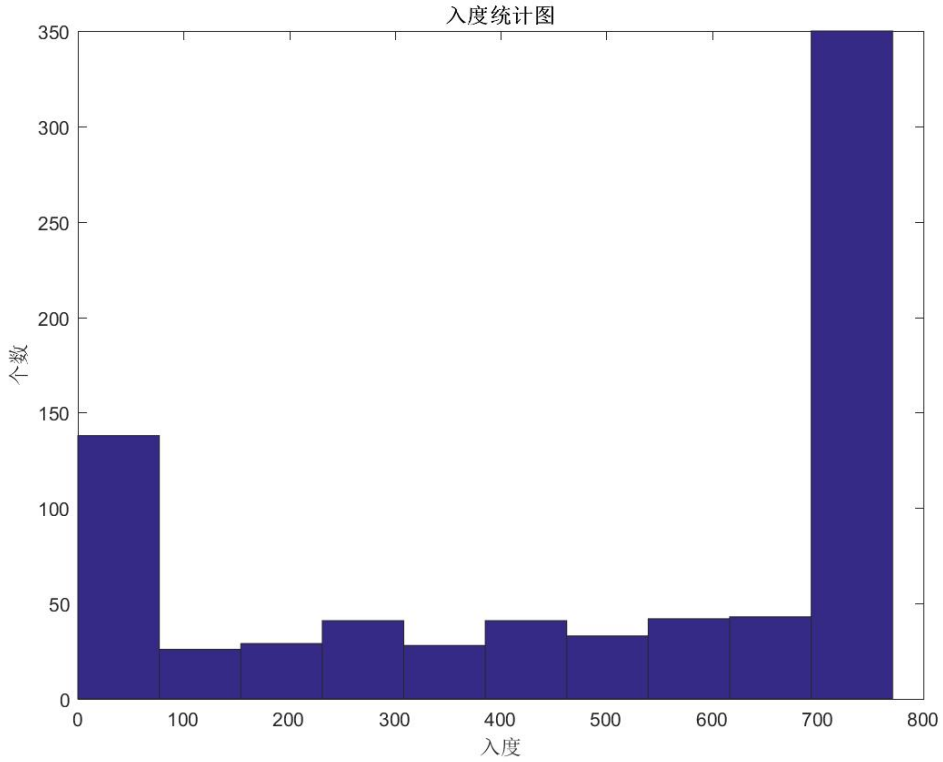

图 19　#你好六月#抓取时间时讨论用户入度数量直方图

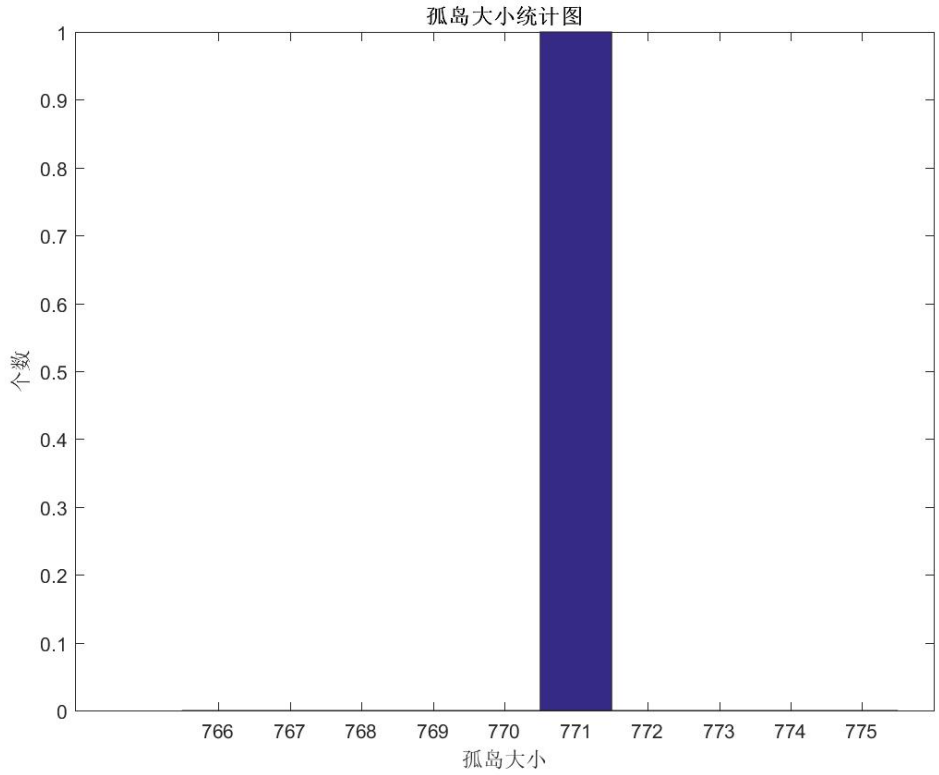

图 20　#你好六月#抓取时间时讨论用户孤岛大小直方图

分析：此时可以发现，仅有一个孤岛，即当前抓取的所有用户之间都存在联系。由于抓取的数据有限，分析并不是十分精确，然而通过这些分析，还是可以得到如下结论：

1、 大 V 在话题的讨论及传播中起到了关键作用，他们的加入提高了话题的传播速度、影响力等，使得话题快速升温。但是此处进行说明：抓取时是根据微博的 V 标志判断是否为大 V，但是现在微博认证泛滥，只有千余人关注的博主就会有微博认证标识，而有的有千余万人关注的博主却没有（比如微博搞笑排行榜），因此这个并不是很准确，可以考虑使用粉丝人数来判定是否属于大 V。

2、 随着话题参与讨论人数的增多，讨论者们的关系更为密切。这也符合大多数话题是从自己的好友和关注那里看来的这个常识。

3、 通过 c++处理处有向图的时候，可以发现一个现象，即置为 1 的操作（说明这个人关注了另一个人）往往出现在关注列表的前列。这说明很可能是这些人发表讨论时顺便浏览到了其他人的讨论时顺手关注的。这也符合事实。

5 实验总结

在本学期之前的实验中，我们曾使用新浪 API 来进行新浪微博数据的获取，而在本次实验中，我们使用爬虫来模拟浏览器的行为，消除了新浪 API 的限制（如只能获取关注用户的 30%等），可以获得更加完整和全面的数据。通过爬虫的编写，我们进一步理解了浏览器请求页面的过程，也对使用爬虫获取网页信息有了深入的体会。

而处理获取的信息则使我们重新复习了 c++的使用方法，尤其是文件读写部分，有了与大一时不同的认识，收获很大。在这过程中，关于程序内存管理部分，遇到了很多问题，也引发了我们很多思考，得到了很多收获。在提高程序运行效率方面，我们也进行了努力，使得数据处理程序的耗时与最初相比减少了 70%有余。

最后，衷心感谢老师和助教们一个学期以来的悉心指导，使我们在顺利完成所有实验的同时，真正有所收获有所感悟，对计算机网络有了更深入的体会和更浓厚的兴趣。