Social Network Analysis of Zhihu

Group 9

Gu Zhijing	Huang Xianghai	Lyu Zishen	Zhao Che
1155073548	1155055168	1155071037	1155066626
364148828@qq.com	172233634@qq.com	lvzshen@outlook.com	simonzhao1024@gmail.com

ABSTRACT

In this project, we have covered most of work we plan in the proposal, including: data crawling, basic statistics and graph analysis of both following relationship and topic related patterns of Zhihu. There are many interesting conclusions we have drew in our explorations.

Keywords

Social network analysis, data mining, social graph, topic extraction

1. INTRODUCTION

The internet have developed in the recent decades extremely fast, it contains a large number of information and functions, has become an epitome and collection of human civilization, and these information are still in growth and interchanges. Because of the open of the internet, its functions are no more restricted in the original "information exchanges in inner network", but so many functions have be added in by users all over the world, and thus becomes an indispensable, irreplaceable component of daily life.

Social network is a very significant application of the modern internet, given an example with our project, a very important goal of using the social network is the "Asking-Answering" function. People get thus references for their studies, works and daily life. The reason is that the social network can provide the users with an extremely large amount of "counselors". They are in all sorts and in all ways of life, which are impossible in reality.

"Zhihu" is a very typical example, as the biggest Chinese online community of "Asking-Answering", till to October 2015, there are already more than 3000 thousand registered users in Zhihu, and there are more than 7000 thousand questions and 23 million answers generated. In such many users and information, there are certainly some inner links, which can be for us very valuable.

In our project, we have tried to find out what kind of form of social network may exists in Zhihu, by analyzing the connections of its users. We have also tried to analyze whether the knowledge areas (topics of questions) have any contact with the interpersonal relationships, which is a special issue of Zhihu compared to common social websites like Twitter or Facebook.

2. RELATED WORK

There remains very few public research works of Zhihu. Here are two reports that involve analysis of Zhihu's users and their interpersonal relationships:

- http://zhuanlan.zhihu.com/sulian/19781120
- http://zhuanlan.zhihu.com/sulian/19907234

3. DATA COLLECTION

3.1 Features and Database

There are many kinds of relationships and data of Zhihu that can be analyzed. Every user of Zhihu can have followees and followers. This following relationship is the most important directed relationship. Every user can ask or answer questions; for a particular question, every user can only have one answer. Every question can have several topic wiki-tagged. One answer can be agreed and thanked and the counts of the two behaviors will be added together as important references to a user's answers quality. Besides, the topics that a user usually have answers about can be a signal of their personal interests or professional knowledge areas.

In this project, we are most interested in the user following relationship, the topics of the answer and some other features. We design and implement a SQLite database to store these data. Here we list the schema of the database to show what kind of data we actually want:

- User (user_url, user_id, answer_num, followee_num, follower_num, agree_num, thanks_num, layer)
- Following (user_url, followee_url)
- Question (question_id, topic)
- UserQuestion (user_url, question_id)
- UserTopic (user_url, topic)

3.2 Crawling

3.2.1 Tools for Crawler Implementation

Our crawler is implemented in Python language. Currently, there is no official API for requesting information on Zhihu. In order to avoid all the troublesome processing with html content and speed up our implementation, we adopt an open source package called "zhihu-python". This library is developed based on python. It has classes of all the data features of Zhihu we want.

Zhihu-python imports some important requirements to implement, such as the Beautiful Soup, Requests, html2text, termcolor and so on. Different from before, the Beautiful Soap make it much more easily to get the information from HTML files than analysis the files by people themselves and the usage of Requests is much more humanized than the urllib3 or the versions before. So here comes the introductions of them.

Beautiful Soap is a python library to abstract data from HTML and XML file and transform the format into the type which fits on the requirements of the navigating, searching and so on. After the handling of the Beautiful Soap, the files from the webpage have been formatted into the object of soap which has quit a lot of attribute including all the information of the webpage. Generally, the soap can get the title of the HTML or the XML and output as the text format, it can also get all the tags easily and store them into an array which ordered by the sequent of in the original webpage, so people can get all the tags circularly by using the function of find all or use the attributes of the tags to get the specified tag which is needed by the zhihu-python.

Actually, the Requests inherits all the features from the urllib3, though the urllib3 offers most of the functions which is needed, but the API of it is not quit convenient enough to meet the nowadays' requirements. The Requests support using the cookies to keep the session, which is quit important to be used into the zhihu-python because the Zhihu website hides most of the information before logging into the website, and the verification of the website is just to check the cookies included in the packages which is transferred in the session.

3.2.2 Crawling Strategy

The zhihu-python package is built for users to conveniently gather specific information from Zhihu. However, the package is implemented in single thread and also has some issue on the Windows platform. So we read the source code and modify it to our needs.

The data we need includes user information, the people that a user is following, the questions that a user answers and the topics of questions. After some digging, we found that it is more robust to crawl all those data in a two-stage manner. The first stage is to crawl user information, the people that a user is following, ID of questions that a user answered, while the second stage is to crawl the topics of questions based on the question ids which we obtained in the first stage.

Because we have no access to computer cluster, we have to just run our crawler on a single machine. Using single thread to crawl is too slow for obtaining large amount of data. Therefore, we apply multiprocessing trying to maximize the crawling speed on a single machine. **Figure 1** shows how our multiprocessing works. Note that for the first stage, in the beginning, only the URL of the seed user is in the task queue, when crawling for a user, the URLs of people that this user is following are later put into the task queue. For the second stage, all the task items (i.e. the question ids) are already in the task queue at the beginning of crawling.

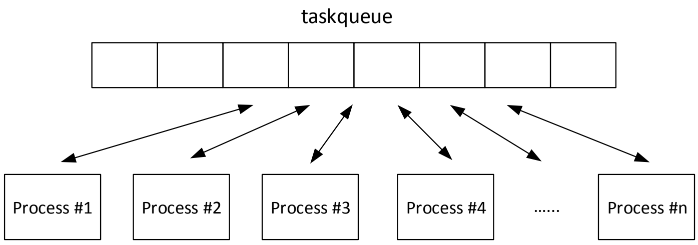


Figure 1. Multiprocessing Mechanics of Crawler

We build our automated crawler with a feature of automatic error handling, which is very important because in such ways we can avoid crawling all over again if an error occurs. Data crawled are stored in text files in the first place, and are transferred to database afterwards for later processing.

In the first stage, our crawler applies breadth-first principle. We crawl two layers from the seed user, seed user as layer 0. The layer structure is shown in **Figure 2**.

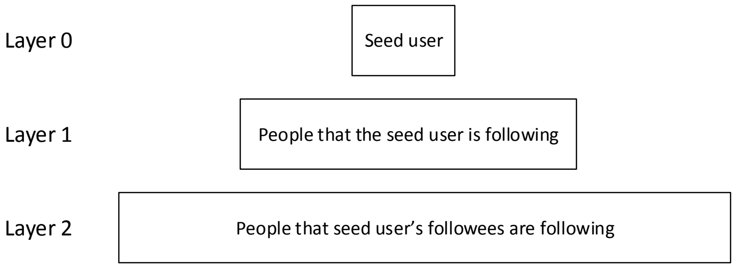

Layer 0
Seed user
Layer 1
People that the seed user is following
Layer 2
People that seed user’s followees are following

Figure 2. Layer Structure of Collected User Data

We first put the seed user as the first item in the task queue in the beginning. Then for every process, get an item from the task queue, process it, add the people he or she follows as new items to the queue. Keep doing this until there is no item in the task queue. The detailed processing procedure of one process is shown in **Figure 3**.

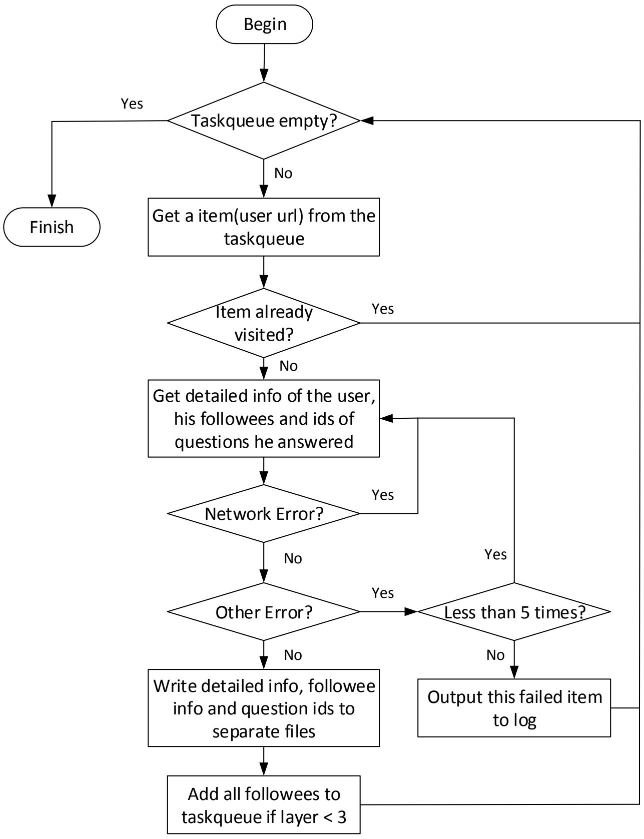

Begin
Taskqueue empty?
Yes
No
Finish
Get a item(user url) from the taskqueue
Item already visited?
Yes
No
Get detailed info of the user, his followees and ids of questions he answered
Network Error?
Yes
No
Other Error?
Yes
Less than 5 times?
Yes
No
No
Write detailed info, followee info and question ids to separate files
Output this failed item to log
Add all followees to taskqueue if layer < 3

Figure 3. Crawling Procedure I

In the second stage, all the question ids are put in the task queue in the beginning. Then for every process, it execute the following procedure until there is no item in the task queue. The detailed processing procedure is shown in **Figure 4.**

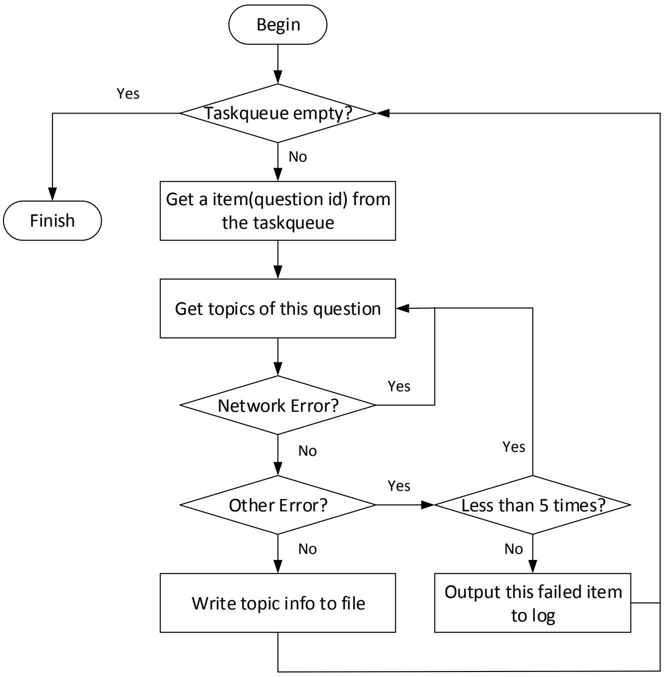


Figure 4. Crawling Procedure II

3.3 Collected Data

In our project, we choose Zhao Che (https://www.zhihu.com/people/zhao-che) as the seed user in crawling. Data crawled are stored in csv files in the first place, and are transferred to the database.

Here is the summary of data we collect:

- SQLite .db file: 688MB
- Number of Database Records: User: 26,000, Following: 4.6 million, Question: 2.2 million, UserQuestion: 1.7 million, UserTopic: 5.4 million

Figure 5 shows a more intuitive overview of the collected user data:

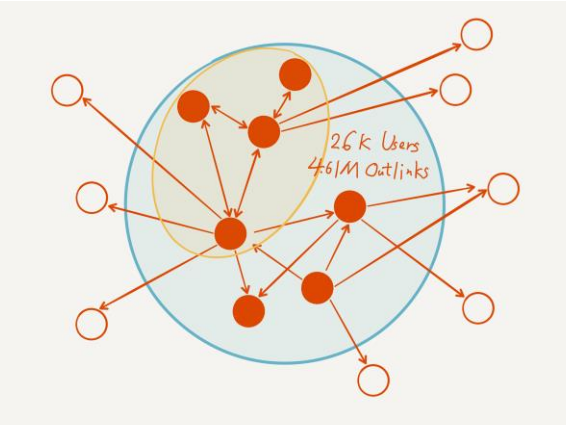


Figure 5. Overview of Collected Data

From the collected data, we know all the following and followed relationship and user information of users within the blue circle. We also know all the other followees of these users out of the circle, but we only know these followees' ID. So the largest following network we can have from the data, which contains complete information , will be the blue circle. These 26 thousand users have 4.61 million followees (out-links) in all.

4. BASIC STATISTICS

Using the collected data mentioned in **3.3**, we do some basic statistics on the features of users in order to grasp some overall characteristics of these users in the data.

4.1 User Average Characteristics

Firstly we calculate the mean, the median and the standard deviation of number of followees, followers, answers, agrees and thanks of users. We use the seed user's features values to be a comparison. This is showed in **Table 1**.

Table 1. User Average Characteristics

	Mean	**Median**	**Standard Deviation**	**zhao-che**
Followee	176.3	67	565.9	149
Follower	3620.0	112	22978.5	306
Answer	68.9	17	225.9	120
Agree	3858.4	96	21951.4	837
Thanks	865.3	28	4627.6	293

We can see that the standard deviation values for all the five feature are all really large. Besides, the mean value and the median value of the same feature also has a huge difference. This leads to a hypothesis that the features cannot follow uniform or normal distributions.

4.2 Log-Log Distributions

We plot log-log distributions of the five user features to verify the hypothesis in **4.1**, as showed from **Figure 6** to **Figure 10** (the order is followee, follower, answer, agree, thanks).

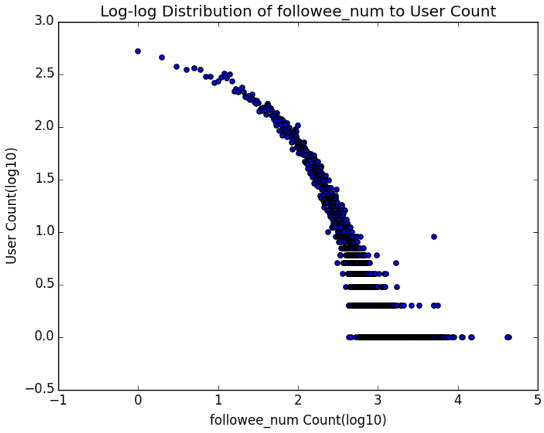


Figure 6. Log-Log Distribution of Followee Number

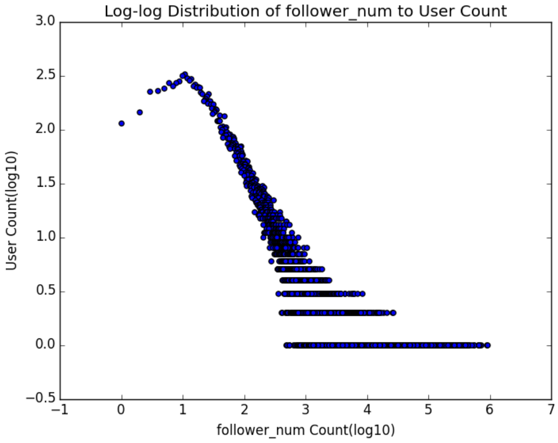


Figure 7. Log-Log Distribution of Follower Number

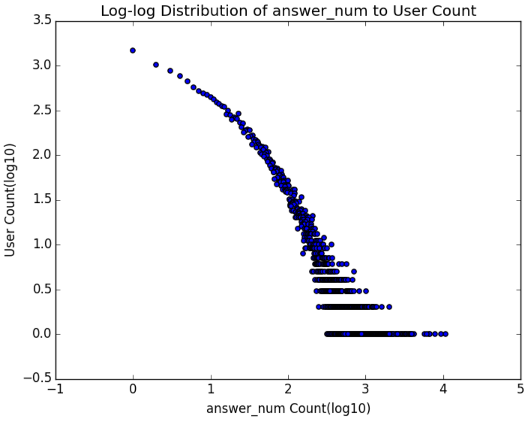


Figure 8. Log-Log Distribution of Answer Number

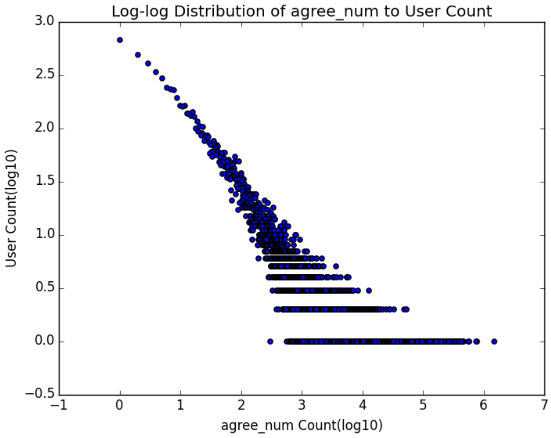


Figure 9. Log-Log Distribution of Agree Number

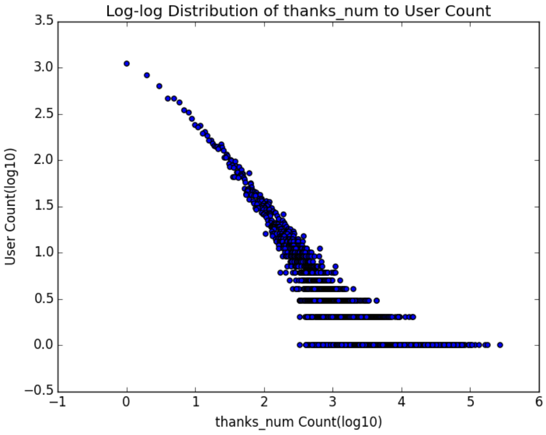
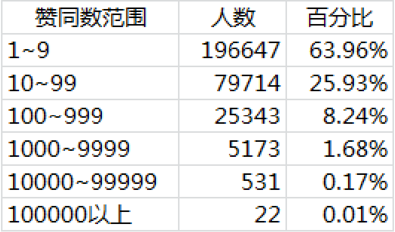


Figure 10. Log-Log Distribution of Thanks Number

Log-log Distributions of all five features shows a significant power law. This proves our hypothesis to be true. In addition, the distributions of followees and followers are actually the out-degree distribution and the in-degree distribution of the following relationship. This may means that the overall social network structure of Zhihu is similar with the typical structure of social websites like Twitter.

A similar statistics of the agree number can be found in the report we mentioned in **2**, which involved 3.5 million users, as showed in **Table 2**:

Table 2. Agree Statistics of 3.5 Million User



赞同数范围	人数	百分比
1~9	196647	63.96%
10~99	79714	25.93%
100~999	25343	8.24%
1000~9999	5173	1.68%
10000~99999	531	0.17%
100000以上	22	0.01%

4.3 Agree and Follower Correlation

We also plot a distribution of agree number to follower number in order to see the correlation between them as **Figure 11**. The plotting result shows that there is merely a positive correlation. This means the more agrees a user has obtained via his/her answers, the more likely he/she will get more followers, and vice versa.

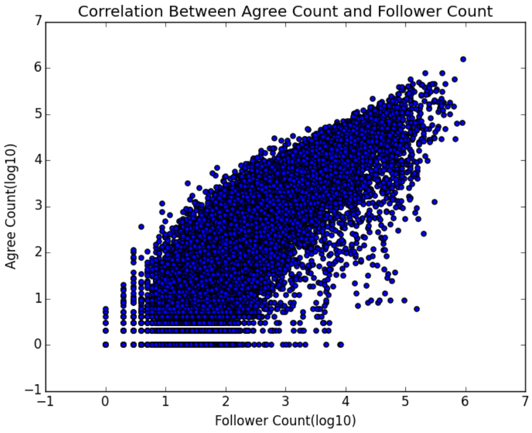


Figure 11. Correlation Between Agree and Follower Number

5. NETWORK ANALYSIS

Next we begin network analysis works on two subsets of users we have collected. The main tool we use in this section is Networkx, a popular python graph computing library.

5.1 User Subset Selection

Due to the limitation of time and machines, we choose to pick two user subsets from the collected data for the network analysis. They are: users who have agree number of more than 10 thousand, users who have agree number of more than 50 thousand. We name them **Net10k** and **Net50k** for convenience.

When picking the subsets, we should avoid using those out links to users outside the subsets. We use SQL commands like the example below to handle this kind of selections:

```
select user_url, followee_url from Following where
        followee_url in (select user_url from User where agree_num > 50000) and
        user_url in (select user_url from User where agree_num > 50000)
```

5.2 Basic Graph Properties

Using the subset data, we firstly calculate some basic graph properties of the user following network. Because Net10k and Net50k are directed, we cannot calculate the diameter and the radius. As for the average shortest path length, networkx do have implemented method for this directed graph, but we find there may be bugs which would lead to wrong results. For example, for a directed graph: [(1,2), (2,4), (3,4)], networkx will have an average shortest path length value of 0.42, which is less than 1. We do not think this is right. But for a strongly connected directed graph, there would be no error.

The densities of Net10k and Net50k are 0.064 and 0.195. This may means there is a larger tendency of clustering for users who have more agree number. Note that 0.195 is a quite high value for graph density. We will discuss about this phenomenon further in **5.4**.

We then plot the vertex closeness and the vertex betweenness distributions of Net10k as showed in **Figure 12** and **Figure 13**.

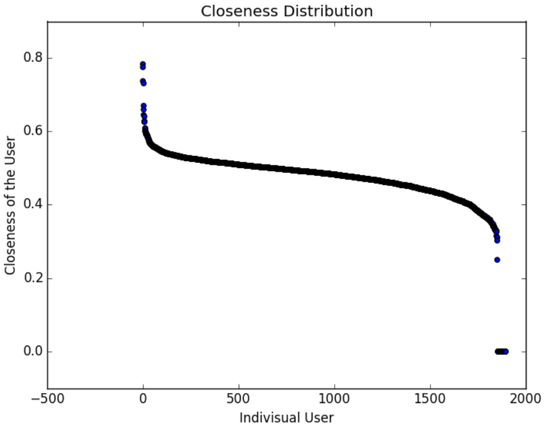


Figure 12. Closeness Distribution of Net10k

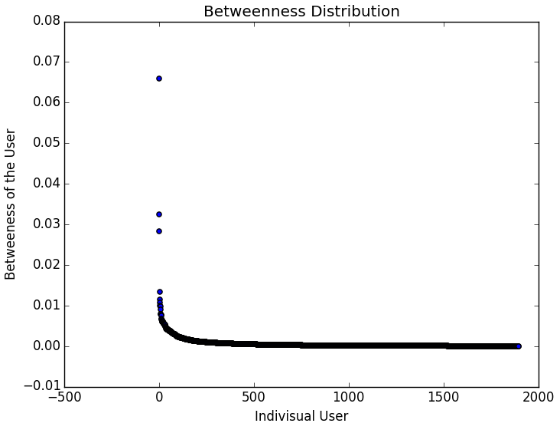


Figure 13. Betweenness Distribution of Net10k

The closeness distribution curve is almost horizontal (centered in around 0.5) in general, which may means users in Net10k are equally important, and 0.5 means they are quite close to each other. However, there are several special user nodes which have 0 closeness, which means they do not have out-links to other users, that is, they are dangling nodes. We will discuss this interesting phenomenon in **5.4**, too.

The betweeness distribution showed another aspect of the property. Most users have 0 betweeness, and even the largest betweenness value is quite small (around 0.066). This means that if you don't have to reach another user through any of other users. There are always lots of paths you can choose.

Figure 14 and **Figure 15** are the similar plots of Net50k, the conclusion we can draw from them are similar. But there are slight differences, such as the closeness curve of Net50k is more horizontal, meaning larger equality among users in Net50k.

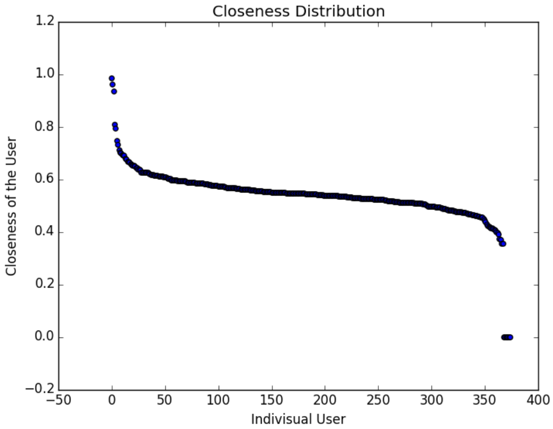


Figure 14. Closeness Distribution of Net50k

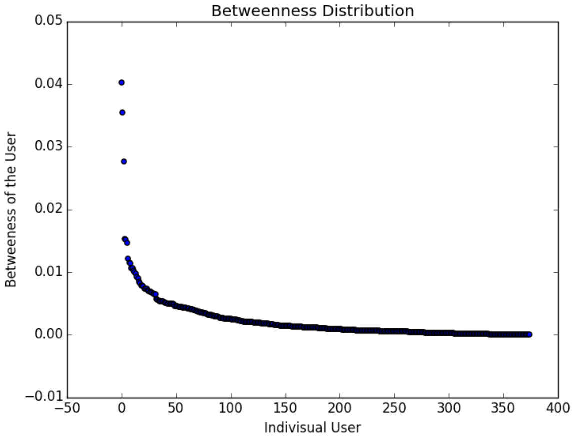


Figure 15. Betweenness Distribution of Net50k

5.3 Link Analysis

Next we do link analysis on Net10k and Net50k separately using PageRank and HITS. **Table 3** and **Table 4** shows the top 5 of PageRank users in Net10k. **Table 5** and **Table 6** shows results in Net50k.

Table 3. PageRank Top 5 of Net10k

PageRank	**PR Score**
jixin	0.00736
ma-bo-yong	0.00560
gejinyuban	0.00551
zhouyuan	0.00510
raymond-wang	0.00503

Table 4. HITS Top 5 of Net10k

Hub	**Hub Score**	**Authority**	**Auth Score**
zhounuo	0.00344	zhang-jia-wei	0.00345
Namoamitabhaya	0.00338	liangbianyao	0.00339

jun-mo-52	0.00336	gejinyuban	0.00324
qisini	0.00325	ma-bo-yong	0.00319
wang-wang-wang-08-18	0.00290	jixin	0.00315

Table 5. PageRank Top 5 of Net50k

PageRank	PR Score
jixin	0.0112
ma-bo-yong	0.0103
zhang-jia-wei	0.0102
liangbianyao	0.0095
commando	0.0090

Table 6. HITS Top 5 of Net50k

Hub	Hub Score	Authority	Auth Score
Namoamitabhaya	0.00976	liangbianyao	0.00749
jun-mo-52	0.00972	zhang-jia-wei	0.00721
qisini	0.00953	ma-bo-yong	0.00683
edison-chen-8612	0.00885	cai-tong	0.00677
miaomiaomiao	0.00798	xiepanda	0.00670

This is the first time we focus on some individual users. We can draw a lot of interesting conclusions from the results above, such as:

- The highest PageRank scores of Net50k are much smaller than Net10k although it contains less users (which means if everyone gets share same proportion of the sum of PageRank scores 1, the score must be higher). This means a larger equality in Net50k than in Net10k. What is surprising is that this is exactly consistent with the conclusion we obtain in **5.2**.
- "Jixin" is a co-founder of Zhihu, who has lots of followers and also lots of followees, so it is not a surprise that he gets the highest PageRank scores in both Net10k and Net50k.
- "zhang-jia-wei", "liangbianyao" and "ma-bo-yong" are all famous writers having many fans in the Chinese Internet world. So it is reasonable that they have highest Auth scores.
- We can see quite a few same names in the top 5 of Pagerank or HITS, but the actual ranks are not identical. This is an interesting phenomena, which may means the levels of popularity of one user are different in different user subsets.

5.4 Strongly Connected Components

In the previous sections, we have seen that Net10k and Net50k are both intensively dense and clustered networks. Can we get a more clear illustration for this? The answer is yes. We calculate the strongly connected component for the two networks and visualize them with networkx and pylab (**Figure 16** and **Figure 17**).

Net10k has 231416 links in all. It has 43 strongly connected components; the nodes distribution of them is [1853, 1, 1, 1, 1, 1, 1, 1, 1, …, 1]. It means instead of 42 dangling nodes, 1853 users are all connected to each other. This distribution is quite interesting, just like islands surrounding the continent. For the 1853 component, we calculate and get that its average shortest path length is 2.11, diameter is 5.0, and radius is 2.0, which is more connected compared to a Six Degree of Seperation.

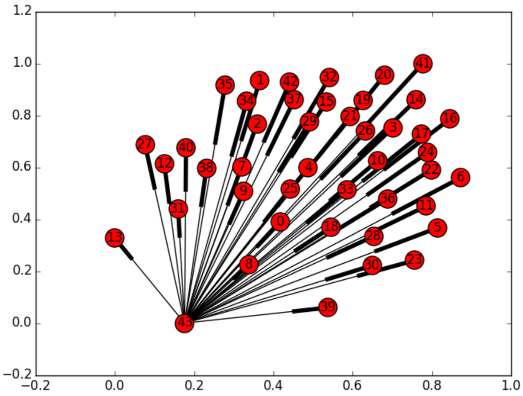


Figure 16. Closeness Distribution of Net50k

Net50k has 27324 links in all. It has 8 strongly connected components; the nodes distribution of them is [368, 1, 1, 1, 1, 1, 1, 1]. It means instead of 7 dangling nodes, 368 users are all connected to each other. For the 1853 component, we calculate and get that its average shortest path length is 1.85, diameter is 4.0, and radius is 2.0, which is even more connected than Net10k.

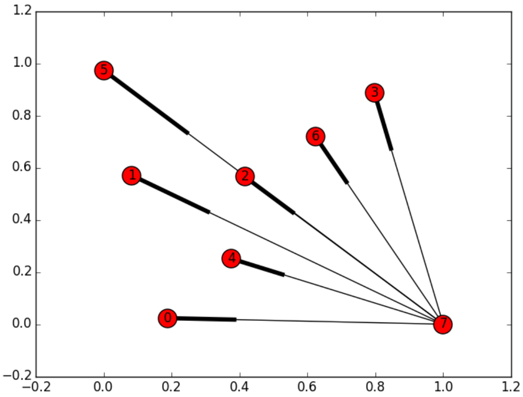
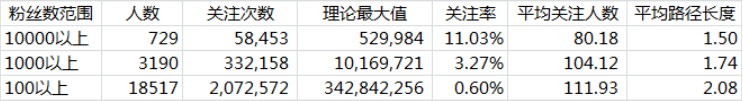


Figure 17. Closeness Distribution of Net50k

Let us rethink the Net10k and Net50k. They both contain the most agreed users that are more famous than other users on Zhihu. They may have different professional areas, but they seem to really like following each other. This is really interesting and informative.

Table 7 shows a similar analysis from other work we mentioned in **2**.

Table 7. Similar Network Analysis from Other Work



粉丝数范围	人数	关注次数	理论最大值	关注率	平均关注人数	平均路径长度
10000以上	729	58,453	529,984	11.03%	80.18	1.50
1000以上	3190	332,158	10,169,721	3.27%	104.12	1.74
100以上	18517	2,072,572	342,842,256	0.60%	111.93	2.08

5.5 Popular Topics Extraction

Finally, we consider extracting the most popular topics within users of Net10k and Net50k. If we can assume that the topics of questions frequently answered by Net10k and Net50k's users are popular topics in the whole platform of Zhihu, we may find an easy way to rank popular topics on Zhihu. But how to extract popular topics from the two user set? Here is what we do: Firstly we calculate the dominant sets of Net10k/Net50k, then we count the frequency of topics from questions answered by users from the dominant set. For Net10k, the dominant set contain 220 in 1896 users, and for Net50k 45 in 375. Here are the top 20 topics we get:

Top 20 from Net10k:
调查类问题 3792, 生活 3096, 历史 1713, 恋爱 1464, 心理学 1432
电影 1419, 人际交往 1404, 社会 1332, 互联网 1214, 情感 1197
政治 1028, 两性关系 994, 教育 897, 中国 823, 人生 815
游戏 805, 文学 772, 知乎 772, 法律 750, 音乐 738
爱情 699, 文化 659,创业 628, 大学 621, 程序员 619
心理 617, 你如何评价 X 609, 女性 604, 编程 585, X 是种怎样的体验 582

Top 20 from Net50k:
生活 1435, 调查类问题 1365, 政治 1285, 历史 1204, 电影 1084
健康 996, 社会 984, 医学 941, 恋爱 717, 中国 695
两性关系 688, 英语 678, 人际交往 640, 心理学 634, 互联网 595
法律 587, 微软（Microsoft） 555, 美国 552, 健身 538, 编程 511

Note that topics like *调查类问题* (Survey-like question) and *你如何评价* X (How do you think about X) are tags promoted by Zhihu to label those questions are asking people’s own opinions. It cannot be a specified to one particular area of interests or professions.

Combining two results, we may conclude that three most popular topics are *生活* (life), *历史* (history) and *电影* (movie).

6. SUMMARY

In this project, we have covered most of work we plan in the proposal, including: data crawling, basic statistics and graph analysis of both following relationship and topic related patterns of Zhihu. There are many interesting conclusions we have drew in our explorations. These interesting results we get can be given back to users, recommending new friends and similar topics to the users. Operators of Zhihu will also be glab to use these information to improve the service accuracy and quality.

The main limitation of our work is from the data we used. It is just a tiny subset of the whole Zhihu platform, and it may contains a strong bias from the pre-defined seed user.

There remains loads of interesting issues of such as:

- Analysis of the complete Zhihu data (may need months to crawl, users who do not have in-links are not considered)
- Analysis combining user following and user-topic networks
- Topic extraction from the corpus of answers

For us, this project has been full of surprises and excites. We really appreciate what we have learnt during such limited time and wish we could crawl more data and dig deeper in the future.

7. CONTRIBUTION

Table 8. Contribution of team members

Student Name	Contribution
Gu Zhijing	Network Analysis, Proposal, Report
Huang Xianghai	Depth-first Crawling, Network Analysis, Report
Lyu Zishen	Width-First Crawling, Network Analysis, Report
Zhao Che	Database, Basic Statistics, Network Analysis, Proposal, Presentation, Report