Group 8: Mapping Mental Health Discussions in a Blog Forum

Xin Shu
2627520
ITEE
mr_comfort@163.com

Sihan Zhu (Group Leader)
2627724
ITEE
sihanzhu22@gmail.com

Jiani Wang
2627821
ITEE
JacquelineWJN@outlook.com

***Abstract*—This research scraps dataset which contains threads obtaining more than 10 replies on Bipolar forum from the UK mental health forum. We use web crawler to extract thread initiators, thread titles, thread repliers and reply times and conclude the distribution features and topic features based on the 5 pages sample. Then construct unweighted network, weighted network based on interaction times as well as weighted network based on reply time intervals separately to study their attributes. Finally, we calculate different centrality values to find the most influential user based on the 1 page sample.**

***Index Terms*—Mental Health, Web crawler, NetworkX, Weight, Centrality Value**

Github: https://github.com/Mr-Sushi/SNA_group8/.

I. Introduction

This document is aiming to investigate the mental health discussion taking part in UK health forum. About this Mental Health Forum, which on this forum you can share experiences, ask questions or vent your emotions with people who know what's it's like to experience mental health difficulties and everything that goes alongside them aiming to be the friendliest forum for support with mental health issues. As there are many theme forums in this forum, this document mainly focuses on the analysis of bipolar forum.

This document consists of three parts: collecting the data of this forum, and analyzing the close relationship between the posters and respondents of this forum by constructing a network.

A. Bipolar Disorder

Bipolar disorder (formerly called manic-depressive illness or manic depression) is a mental disorder that causes unusual shifts in mood, energy, activity levels, concentration, and the ability to carry out day-to-day tasks.

There are three types of bipolar disorder. All three types involve clear changes in mood, energy, and activity levels. These moods range from periods of extremely "up," elated, irritable, or energized behavior (known as manic episodes) to very "down," sad, indifferent, or hopeless periods (known as depressive episodes). Less severe manic periods are known as hypomanic episodes.

Bipolar I Disorder— defined by manic episodes that last at least 7 days, or by manic symptoms that are so severe that the person needs immediate hospital care. Usually, depressive episodes occur as well, typically lasting at least 2 weeks. Episodes of depression with mixed features (having depressive symptoms and manic symptoms at the same time) are also possible.

Bipolar II Disorder— defined by a pattern of depressive episodes and hypomanic episodes, but not the full-blown manic episodes that are typical of Bipolar I Disorder.

Cyclothymic Disorder (also called Cyclothymia)— defined by periods of hypomanic symptoms as well as periods of depressive symptoms lasting for at least 2 years (1 year in children and adolescents). However, the symptoms do not meet the diagnostic requirements for a hypomanic episode and a depressive episode.

B. Web Crawler

A web crawler, spider, or search engine bot downloads and indexes content from all over the Internet. The goal of such a bot is to learn what (almost) every webpage on the web is about, so that the information can be retrieved when it's needed. They're called ”web crawlers” because crawling is the technical term for automatically accessing a website and obtaining data via a software program.

These bots are almost always operated by search engines. By applying a search algorithm to the data collected by web crawlers, search engines can provide relevant links in response to user search queries, generating the list of webpages that show up after a user types a search into Google or Bing (or another search engine). [7]

A web crawler bot is like someone who goes through all the books in a disorganized library and puts together a card catalog so that anyone who visits the library can quickly and easily find the information they need. To help categorize and sort the library's books by topic, the organizer will read the title, summary, and some of the internal text of each book to figure out what it's about.

C. NetworkX

NetworkX is a Python language package for exploration and analysis of networks and network algorithms. The core package provides data structures for representing many types of networks, or graphs, including simple graphs, directed graphs, and graphs with parallel edges and self loops. The nodes in

NetworkX graphs can be any (hashable) Python object and edges can contain arbitrary data; this flexibility mades NetworkX ideal for representing networks found in many different scientific fields. In addition to the basic data structures many graph algorithms are implemented for calculating network properties and structure measures: shortest paths, betweenness centrality, clustering, and degree distribution and many more. NetworkX can read and write various graph formats for eash exchange with existing data, and provides generators for many classic graphs and popular graph models, such as the Erdoes-Renyi, Small World, and Barabasi-Albert models, are included. [8]The ease-of-use and flexibility of the Python programming language together with connection to the SciPy tools make NetworkX a powerful tool for scientific computations. We discuss some of our recent work studying synchronization of coupled oscillators to demonstrate how NetworkX enables research in the field of computational networks.

II. PROBLEM DESCRIPTION FROM THE PROJECT SPECIFICATION

Our project is based on the investigation of a bipolar forum taking part in UK health forum. The efficient dataset is collected from threads which contain more than 10 replies.

A. Scrap the useful dataset

We are required to use either API or manual download some structured database, which includes thread messages and their initiated user IDs, associated replies with their user IDs and time of posts. For this part, web crawler is needed when selecting required information.

B. Test the distribution regulation of threads per user

Firstly we need to trace the plot showing the number of threads per user ID to find out some basic regulations of their distribution. We plot its histogram to check whether it appears long-tail appearance or not. Then we need to test whether the distribution of the number of threads per user follows Power-Law distribution and approve it with appropriate statistical test such as mean squared error and confidence level.

C. Conclude how the topics vary for top 10 active users

We need to build up a manual labeling list to describe the categories of common topics, such as suicide, medical treatment, and experiences sharing. The we need to sum up the number of all the replies and all the posts from the same user, and then compare the find out the biggest ten numbers and their corresponding user IDs, which are also the ten most active users. By inquiring the thread titles they posted and replies, we use our own manual labeling list to conclude some regulations of their topics features.

D. Construct the first unweighted network

We are required to build up a network showing whether two users have interactions or not. If they show up within the same thread, they are regarded as having interactions and an edge between their nodes is established. We need to firstly traverse all the replies in one single thread and secondly traverse all the threads.

E. Construct the second weighted network based on the number of replies

We are required to build up the second network showing the influence per edge on the first network. The influence is decided by the number of replies initiated by the same user with the same thread, and whether the node has initiated a thread or is only involved in the replies of another thread. The weight follows the following rules: (1) an edge between node A and node B is established with a maximum weight 10 if either A or B is the user ID who initiated the thread; (2) If both A and B occur as replies in the same threat, then the weight of edge between node A and node B is equal to the total number of replies issued by A and B. Also, we set the weight as 10 if the calculated weight in rule (2) overcomes 10. Therefore, the weight range is from 1 to 10.

F. Construct the third weighted network based on the time difference of reply

We are required to build up the third network showing the influence per edge on the first network. The influence is decided by the time response taken by A and B respond to the last message. It also requires as to design the weight distribution on our own.

G. Summarize the attributes of three network graph separately

We need to calculate the attributes of the constructed graph, including number of nodes, number of edges overall clustering coefficient, average path length, size of giant component, diameter, maximum degree, average degree, number of communities using Girvan-Newman algorithm and adjacent metrics as implemented in NetworkX. Then summarize the compare their results.

H. Compare the most influential nodes of three network graph separately

We are supposed to calculate different centrality value to find out the most influential nodes in each graph. In-betweenness centrality and closeness centrality will be used to measure their influence.

I. Discuss and comment on the interpretations

We need to use appropriate health literature to reinforce our interpretations.

III. DATASET DESCRIPTION

Because the data of this forum is too large, here we select the sample data on page 50 of this forum as an example. The data of the first table is the title, publication time, publisher and website of the post on page 50 of this forum respectively.

The second table is the ID of the person who posted and replied to the first post on page 50, and the time of the post and reply.Through the reply time of the poster and the reply, we build the whole network structure and analyze the data.

TABLE I
THREAD INFORMATION

Post user	**Post time**	**Thread title**	**Thread link**
tiltawhirl	2010-12-15T22:46:27+0000	Do not do this at home!	/forum/threads/do-not-do-this-at-home.18371/
lauli-ann	2010-12-15T22:52:41+0000	and so...	/forum/threads/and-so.18372/
cethalopod	2017-05-10T02:40:28+0100	confused about diagnosis	/forum/threads/confused-about-diagnosis.164808/
Spaceman	2010-11-30T22:56:06+0000	I don’t want to go mad on my own!	/forum/threads/i-dont-want-to-go-mad-on-my-own.17869/
baby_dolly_face	2011-02-16T13:57:30+0000	5th day running anxiety has woken me up	/forum/threads/5th-day-running-anxiety-has-woken-me-up.20686/
magicman2002	2013-12-14T00:34:17+0000	things you do when your manic or hypomanic	/forum/threads/things-you-do-when-your-manic-or-hypomanic.77264/
calypso	2013-03-24T17:50:18+0000	God, why is life so stressful!!	/forum/threads/god-why-is-life-so-stressful.57552/
munchie	2010-12-16T14:11:08+0000	How do you have a norma relationship and be BP as well?	/forum/threads/how-do-you-have-a-norma-relationship-and-be-bp-as-well.18385/
Topcat	2013-01-03T11:20:37+0000	Stuck	/forum/threads/stuck.53205/
letmein	2017-02-13T16:00:05+0000	skint.	/forum/threads/skint.155609/
GoghTardis	2013-09-15T01:03:47+0100	Heading downhill...	/forum/threads/heading-downhill.70621/
baby_dolly_face	2011-10-02T15:52:29+0100	Can any of you help me please? xxxx	/forum/threads/can-any-of-you-help-me-please-xxxx.29919/
Apotheosis	2011-08-23T05:20:21+0100	The rise in bipolar disorder is a myth	/forum/threads/the-rise-in-bipolar-disorder-is-a-myth.28387/
baby_dolly_face	2011-01-14T16:29:50+0000	A tumultuous patch in the tranquility of the ocean....	/forum/threads/a-tumultuous-patch-in-the-tranquility-of-the-ocean.19429/
firemonkee57	2011-08-09T14:51:48+0100	Questioning Whether Bipolar Disorder Is an Illness	/forum/threads/questioning-whether-bipolar-disorder-is-an-illness.27877/
TOONAFISH	2012-06-06T14:47:43+0100	Are any of you premmie babies?	/forum/threads/are-any-of-you-premmie-babies.41962/
Twylight	2008-10-23T20:27:28+0100	Mood swings and bipolar	/forum/threads/mood-swings-and-bipolar.2540/
bobshocker	2013-05-25T21:41:21+0100	***Trigger Warning***i reckon bullshit works	/forum/threads/trigger-warning-i-reckon-bullshit-works.61423/
living in the fast lane	2011-12-22T17:15:13+0000	I am sorry	/forum/threads/i-am-sorry.33522/
GoghTardis	2013-08-26T13:39:20+0100	Advice on working with a clouded mind?	/forum/threads/advice-on-working-with-a-clouded-mind.68853/
damnmouse	2019-02-13T16:49:49+0000	Do you have a ”manic uniform?”	/forum/threads/do-you-have-a-manic-uniform.195063/
Funkygirl89	2013-08-11T18:19:10+0100	How do you cope with bipolar???	/forum/threads/how-do-you-cope-with-bipolar.67577/
mckeo5514	2012-07-07T20:20:29+0100	getting sillier?	/forum/threads/getting-sillier.43770/
happyhappy	2008-06-09T18:23:10+0100	Started to stop my meds last night	/forum/threads/started-to-stop-my-meds-last-night.1276/
Desperado	2010-08-01T21:40:21+0100	OMG - help! Please not another manic/mixed episode!	/forum/threads/omg-help-please-not-another-manic-mixed-episode.14337/

TABLE II
POST USER ID AND REPLY TIME

reply user ID	reply time
’26344’	’2013-07-18T22:19:36+0100’
’2658’	’2013-07-18T23:07:14+0100’
’26344’	’2013-07-18T23:19:38+0100’
’10687’	’2013-07-19T00:30:44+0100’
’26344’	’2013-07-19T01:33:07+0100’
’26344’	’2013-07-19T01:37:44+0100’
’26344’	’2013-07-19T01:43:06+0100’
’26344’	’2013-07-19T01:46:54+0100’
’10687’	’2013-07-19T02:05:32+0100’
’26344’	’2013-07-19T02:50:42+0100’
’26344’	’2013-07-19T02:53:16+0100’
’26344’	’2013-07-19T02:55:31+0100’
’26344’	’2013-07-18T22:19:36+0100’
’2658’	’2013-07-18T23:07:14+0100’
’26344’	’2013-07-18T23:19:38+0100’
’10687’	’2013-07-19T00:30:44+0100’
’26344’	’2013-07-19T01:33:07+0100’
’26344’	’2013-07-19T01:37:44+0100’
’26344’	’2013-07-19T01:43:06+0100’
’26344’	’2013-07-19T01:46:54+0100’
’10687’	’2013-07-19T02:05:32+0100’
’26344’	’2013-07-19T02:50:42+0100’
’26344’	’2013-07-19T02:53:16+0100’
’26344’	’2013-07-19T02:55:31+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’
’57702’	’2013-07-19T04:53:25+0100’
’26344’	’2013-07-19T11:41:47+0100’
’26344’	’2013-07-19T11:42:29+0100’
’26344’	’2013-07-19T11:43:34+0100’
’10687’	’2013-07-19T12:24:43+0100’

input and output;

block diagram; the relationship between

IV. General methodology

The methodology mainly consists of three parts: data collection, network construction and analysis of results. The primal technologies are Beautiful Soup and NetworkX, which we will discuss further later.

A. Data collection

We scrap the dataset and utilize threads containing more than 10 replies. Using web crawler, we downloaded the dataset including thread title, post user ID, reply user ID, post time, and reply time. The process of data acquisition is:

- Impersonate the client to make a request to the server.
- Find the right data by using developer tools.
- Use regular expressions to select the appropriate data and store it.

This part is actually the most challenging and time-consuming of all, as we didn't expect that the Mental Health Forum UK is somehow strange, the semantics of html code was not very similar to a normal website. The "tourists" without an user ID can post and reply freely as well. Their user IDs are set to 0. Small issues as this are what slows the process down.

B. Network construction

The plan is to building different networks with different weight assignment methods.

- Network 1: basic network
- Network 2: network with weights decided by interaction between users
- Network 3: network with weights decided by time interval between two replies

There are two steps taken to solve the problem:

- Step 1: Build different networks in the same thread
- Step 2: Based on step 1, loop over all the threads and build larger networks. In the end, the networks with different weight assginment methods contains all threads.

As NetworkX is a powerful tool with many built-in functions, finishing step 1 was relatively simple. Network 1 and 2 is constructed following a written script, while for network 3 we designed the script from scratch. The challenge lies in the second step. As a matter of fact, the NetworkX is not perfect, because we found some illogical issues in the process. There were originally three plans to build the network, and only one work efficiently. We are going to discuss why the other two plans fail in details.

C. Results analysis

- Display the number of threads per user ID and see if it follows Power-Law distribution or not.
- Find the 10 most active users and label their topics
- Calculate the attributes of the networks including average path length, size of giant component, diameter, max degree, number of communities...
- Calculate centrality value (degree centrality, in-betwennes centrality, closeness centrality and PageRank centrality)

into 3 parts.

V. Detailed methodology

A. *urllib.request*

The urllib.request module defines functions and classes which help in opening URLs (mostly HTTP) in a complex world — basic and digest authentication, redirections, cookies and more.

B. *BeautifulSoup*

Once urllib.request has pulled in the content from the URL, we use the power of BeautifulSoup to extract and work with the data within it. Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

C. *Regular Expression*

A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

Regular expressions can be concatenated to form new regular expressions; if A and B are both regular expressions, then AB is also a regular expression. In general, if a string p matches A and another string q matches B, the string pq will match AB. This holds unless A or B contain low precedence operations; boundary conditions between A and B; or have numbered group references. Thus, complex expressions can easily be constructed from simpler primitive expressions like the ones described here. For details of the theory and implementation of regular expressions, consult the Friedl book [Frie09], or almost any textbook about compiler construction. [9]

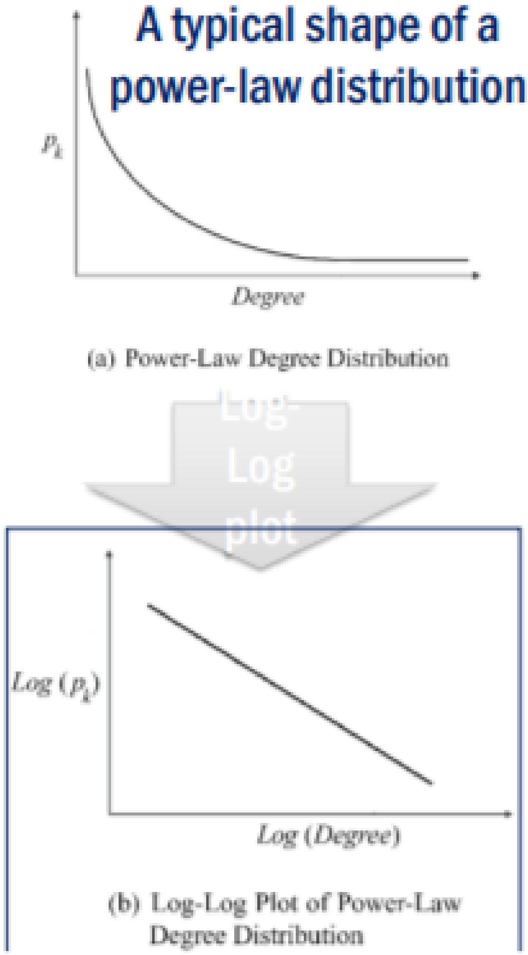
When get the string by using BeautifulSoup, use the re to match the related information and then the data is sorted and stored.

D. *Test the distribution of the number of threads per user*

After plotting the the number of threads per user ID, which shows the long-tail appearance and can be regarded as Power-Law distribution preliminarily, we test it in details and justify it further as follows:

1) Power-Law distribution: In statistics, a power law is a functional relationship between two quantities, where a relative change in one quantity results in a proportional relative change in the other quantity, independent of the initial size of those quantities: one quantity varies as a power of another. In empirical contexts, an approximation to a power-law often includes a deviation term, [13]which can represent uncertainty in the observed values (perhaps measurement or sampling errors) or provide a simple way for observations to deviate from the power-law function (perhaps for stochastic reasons):

$$y = ax^k + \varepsilon \tag{1}$$



(a) Power-Law Degree Distribution

(b) Log-Log Plot of Power-Law Degree Distribution

Fig. 1. transformation of Power-Law Distribution

We find that the equation can be transformed as a linear formula by adopting logarithm. Then the equation will be expressed as:

$$\lg y = k \lg x + \lg a \tag{2}$$

Where k is the line slope. The Fig.1 shows the transformation. Then we can test the numbers after logarithm whether follows linear distribution.

2) Linear Regression: For test and justify our answer, we use linear regression in machine learning to fit the numbers. In statistics, linear regression is a regression analysis that uses the least squares function called linear regression equation to model the relationship between one or more independent and dependent variables. This function is a linear combination of one or more model parameters called regression coefficients. The case with only one independent variable is called simple regression, and the case with more than one independent variable is called multivariable linear regression. [10] In linear regression, data is modeled using linear prediction functions, and unknown model parameters are also estimated from the data. These models are called linear models. [11] The most commonly used linear regression modeling is that the conditional mean of y given the value of X is the affine function of X. We are going to use it and measure the fitness with least square method and justify it with mean squared error. Least Square Method The method of least squares is a standard approach in regression analysis to approximate the solution of overdetermined systems (sets of equations in which there are more equations than unknowns) by minimizing the sum of the squares of the residuals made in the results of every single equation. The most important application is in data fitting. The best fit in the least-squares sense minimizes the sum of squared residuals (a residual being: the difference between an observed value, and the fitted value provided by a model). Mean Squared Error In statistics, the mean squared error (MSE) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss. [12]It can be expressed as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3}$$

Usually, the smaller MSE is, the better performance the regression achieves. To reduce the calculation time, we use 5 pages to test briefly. If the random and small amount of data follows Power-Law distribution, then the total data can be recognized as following power-law distribution.

E. NetworkX

”NetworkX is a Python package for exploration and analysis of networks and network algorithms.” [1]

Network is powerful and particularly suitable for our work, because:

- NetworkX depends on a pure-Python ”dictionary of dictionary” data structure. It’s a reasonably efficient, highly scalable and very portable framework for social network analysis.
- Network provides classes to represent directed and underected graphs, including optional weights and self loops, and a special representation for mutigraphs which allows multiple edges between pairs of nodes..
- Once a network is represented as a NetworkX object, users are able to find degree distributions, clustering coefficients, shortest paths, and spectral measures. [1]
- Networks is useful for operating on large real-world graphs, such as graphs of more than 10 million nodes and 100 million edges. [2]

Some built-in functions of NetworkX were useful:

- nx.graph(): An empty graph is created with no nodes or edges.
- add_nodes_from: Add multiple nodes from a list, graph or string.
- add_edges_from: Add multiple edges from a list, graph or string.

- add_weighted__from(ebunch, weight='weight'):Add all the edges in ebunch as weighted edges with specified weights.
- nx.draw():Draw the graph with Matplotlib.

F. Building the networks

1) Network 1: At the beginning we constructed a graph without any weight. The nodes are users. If two users are under the same thread, there is a connection between them. The number of edges can be large since every pair of people under the same thread has a connection. In fact, if there are N threads, each including M users in the thread, the total number of edges will be N*M*(M-1)/2. To implement a large network as this, the methodology is simple:

- Get an array of user IDs of a thread. A common mistake would be getting a list of users from each reply, possibly including multiple records of the same user. Therefore the list of user IDs are set to unique using numpy.unique().
- Create an empty graph using nx.Graph() function.
- Every node has edges with all of the other nodes in the thread graph. For that, “itertools” is needed.

Use the built-in adding edges function (add_edges_from) of Network X. And a simple line of code allows all the edges to be added at once: G.add_edges_from(itertools.combinations(unique_users, 2))

2) Network 2: Adding weight to the original graph gives us a new graph. The maximum weight is 10.

- If one user of the edge initiated the thread, the weight is 10.
- If both the users only replied in the thread, the weight is the number of total replies.

Consider two nodes a, b and the edge between them. If the edge has no weight, the weight can be set by:

H[a][b]['weight'] += new_weight

Otherwise, the weight is set using:

H.add_edge(a, b, weight=new_weight)

What if a pair of users appear together in multiple threads? In this case, the combined weight is likely to be more than 10. Therefore, if the edge already has weight, and the sum of existing weight and new weight is more than 10, the weight will be set to 10 again.

3) Network 3: In network 3 we take time into accounts.We designed the script below:

- Given a pair of nodes A,B. If A or B initialized the thread, add 10 to the weight.
- Otherwise, the added weight is the sum of time scores of both A and B: w = np.sum(np.average(np.array(w_a)) + np.average(np.array(w_b))).
- The idea is, the sooner a person sent a reply after last message, the higher the score is.
- If the edge already has weight, and we increase by no more than 10, the new weight could be easily more than 10. So in the end if the weight is larger than 10, it is set to 10.

TABLE III
TIME SCORE FOR EACH USER

time score	the time interval between user's rely and the last reply
5	Less than 6 hours
4	6-12 hours
3	12-18 hours
2	18-24 hours
1	More than 24 hours

The main issue during the process is the setting of time interval. The rule showed in the figure is a rather successful case. But in the beginning of the process, we set the interval too big. In fact, if a person replied within a day, the time score was 5. As a result almost every edge has a weight of 10, which is pointless.

4) Building large networks: In the beginning, we only considered the case of one thread, thinking that if the network of one thread is clear, then all we need to do is adding a for loop. The first idea is to build three networks separately. Building network1 is easy: At first, create an empty graph using nx.Graph(). For each thread of the dataset, add the nodes according to a list of unique user ID, and add edges between every two nodes. Once the for loop is finished, the network is built perfectly, and the graph can be plotted. The problem is with network 2 and 3. Since the two networks are based on network 1, we need to utilize the graph mentioned before.There are some flaws:

- Whether should we use a unique array of users, or just a list of users for all posts and replies that could contains multiple instances of the same user? For the former, it's impossible to track how many times they appeared in a thread. For the latter, it could build strange relations. For example, node A would have edges with itself because of multiple occurrence, which is wrong. The only rational method would be combine both, but it would be easier if we ignore the finished network 1 and build network 2 and 3 from an empty graph.
- Now that the graph already has all the threads, we need to find all the edges using user IDs, and then add the weight. That's a complicated process, meaning that there could many possible problems in the process. Even if we can run the model smoothly, each time adding the weight, we need to search the whole network,which requires higher computational ability and a longer period of time to finish. The effort would be unnecessary.

The second idea is to implement 3 networks all at once. For each thread, we create H1, H2, H3 corresponding to network 1,2,3. H1 is built using an array of unique users. H2 and H3 are constructed on H1. Then we add H1, H2, H3 to the full graph G1, G2, G3 respectively. The idea is valid in theory. But as we proceed, we realized:

- The NetworkX make mistakes when building three networks at once. For example, if we print weights of H2 right after it's built, the output is correct. And after a few lines of codes about constructing H3, we printed weights

again, the weights were all 10.
- Also there are other illogical errors like this. But if we process 2 networks at once instead of 3, the errors are eliminated.

Here's a series of steps, and the detailed methodology that we took in the end:

- Create two empty graphs G1, G2, corresponding to network 1, 2
- Inside each thread, build graph H1 from an array of unique users. Based on H1, add weights as the number of replies to be H2.
- Add H1 to G1 using G1.add_edges_from(H1.edges)
- Add H2 to G2 using G2 .add_weighted_edge_from (H2_edges)
- Loop over the edges of G2 to make sure that no weight is higher than 10.
- Create two empty graphs G1, G3, corresponding to network 1, 3
- Inside each thread, build graph H1 from an array of unique users. Based on H1, add weights as the sum of average time scores to be H3.
- Add H1 to G1 using G1.add_edges_from(H1.edges)
- Add H3 to G3 using G3. add_weighted_edge_from (H3_edges)
- Loop over the edges of G2 to make sure that no weight is higher than 10.

G. Attributes of network graph

We are going to calculate required attributes of each network graph. For each one, we need to calculate the following numbers:

1) Number of nodes: The nodes represent users in the network. The number of nodes means the number of user in the network. The calculation can be achieved by the function in NetworkX package as *nx.number_of_nodes()*.

2) Number of edges: In a social setting, where nodes represent social entities such as people, edges indicate internode relationships and are therefore known as relationships or (social) ties. The edges represent connections between two users in the network. The number of nodes means the number of connected user pairs in the network. The calculation can be achieved by the function in NetworkX package as *nx.number_of_edges()*. It is denoted as

$$|E| = m \tag{4}$$

3) Overall clustering coefficient: Clustering coefficient is used to capture partial transitivity. The average clustering coefficient can be calculated using NetworkX package. The overall clustering coefficient can be obtained by multiplying the number of nodes. It is defined as:

$$C = \frac{\text{number of closed paths of length two}}{\text{number of paths of length two}} \tag{5}$$

4) Average path length: A walk is a sequence of incident edges visited one after another. A walk where nodes and edges are distinct is called a path. The length of a path is the number of edges visited in the path or cycle. It is calculated in a network graph that is connected. In our model, nodes appear to be three groups so that it is not connected. To solve this problem, we calculate the total path length for each component separately. Then divide it with the number of nodes in the network.

5) Size of giant component: A component in an undirected graph is a connected sub graph. We find out all the component in the network. Then find out the biggest one and measure its size.

6) Diameter: The diameter of a graph is the length of the longest shortest path between any pair of nodes between any pairs of nodes in the graph. It can be expressed as:

$$diameter_G = max_(v_i, v_j) v_i, v_j \in V \times V^l_{i,j} \tag{6}$$

Here the diameter means the biggest component one. The calculation can be achieved by the function in NetworkX package as *nx.diameter()*.

7) Maximum degree: The number of edges connected to one node is the degree of that node, which is also the size of its neighborhood. We sort the number of degree in a list and find out the biggest one.

8) Average degree: We sum all the number of degree up and then divide it by the number of nodes.

9) Number of communities: We get the number of communities by using Girvan-Newman algorithm. The steps is following: *Step 1.* Calculate edge betweenness for all edges in the graph; *Step 2.* Remove the edge with the highest betweenness; *Step 3.* Recalculate betweenness for all edges affected by the edge removal; *Step 4.* Repeat until all edges are removed. Here we assume to get the first 10 tuples of communities.

10) Adjcency matrix: Social media networks tend to have vert sparse adjacency matrix. It represents whether two nodes have connections. Two nodes are adjacent if they are connected via an edge. It is established as follows:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between nodes } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Among all the attributes, overall clustering coefficient, average path length, size of giant component and adjacency matrix will be influenced by weight.

H. Centrality value

Centrality addresses the equation that who is the most important or central person in this network. Many centrality measures have been proposed. We are going to compare the most influential nodes in terms of centrality value including the following:

1) Degree centrality: It is simply the degree of node. It ranks nodes with more connections higher in terms of centrality. In can be calculated as follows:

$$C_d(v_i) = d_i \tag{8}$$

Where d_i is the degree for node v_i.

改图

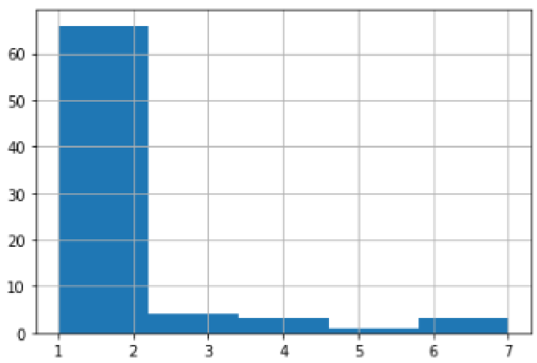


Fig. 2. distribution of the number of threads per user

2) Page-Rank centrality: We can divide the value of passed centrality by the number of links. Each connected neighbor gets a fraction of the source node's centrality. It can be described as:

$$C_p = \beta(I - \alpha A^T D^{-1})^{-1} \tag{9}$$

Where:

$$D = diag(d_1^{out}, d_2^{out}, \cdots, d_n^{out}) \tag{10}$$

3) In-betweenness centrality: Betweenness centrality captures how much a vertex falls between others, in contrast with other centrality measures we have seen, which capture how well connected the node is. It is formulated as:

$$C_b(v_I) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \tag{11}$$

Where: σ_{st} represents the number of shortest paths from vertex s to t, and $\sigma_{st}(v_i)$ represents the number of shortest paths from s to t that pass through v_i.

4) Closeness centrality: Closeness centrality indicates the closeness of a given node to all other nodes in terms of shortest path. Typically, the more central a node is, the closer it is to all other nodes. It is reciprocal of the shortest path distances from v to all $(n-1)$ nodes. It can be expressed as follows:

$$C_v(v) = \frac{1}{\sum_{i=1}^{n} d(v, v_i)} \tag{12}$$

In all the above centrality values, only in-betweenness centrality and page-rank centrality will be influenced by weight.

VI. Results and Discussions

A. Test the distribution

1) Plot the distribution of the number of thread per user: We select the page 46 to page 50 to perform the test. In total there are 125 threads. By plotting its histogram, the distribution can be seen in Fig.2: It shows obvious long-tail effect and can be preliminary regarded as following power-law distribution.

2) Fit linear regression to the data: Firstly we adopt logarithm to the data to get new data. Also, we can see that the number of thread per user ranges from 1 to 7. Considering its small size, we just select the first five pairs of data as train data and the last two pairs as test data. Then by fitting linear regression model and predict the result based on test data. We calculated coefficient of this model and its mean squared error, the results are:

$$\begin{cases} coefficient = -0.53265 \\ mse = 0.12 \end{cases} \tag{13}$$

The mean squared error is relative small. Then we can believe it follows the distribution of Power-Law.

B. Top 10 active users

1) Labeling list: After searching the forum and viewing for the labels on the forum, we generally divide the topics in TABLE IV and describe their contents separately:

TABLE IV
MANUAL LABELING LIST

Labels	Description
Romance	Related with sex or love;
Symptoms	Describe symptoms when experiencing an illness, after eating some medicine, or just some uncommon feelings or appearance;
Depressed	Express depressed feelings or sadness;
Need advice	Ask for help or look for advice;
Experiences	Share experiences or stories happening around them;
Medication	Information about medicine;
Positive	Express positive feelings or attitude;
Suicide	Want to suicide or look for suicide ways;
Mood	Express discernible change in opinion.

2) Select top 10 active users and match their topics: After selecting the ten most active users and collecting his/her involved topics separately, we get the table showing the results and show the most active users list as an example in the appendix. Then we match the topics with our labeling list in the TABLE V: By counting the times each label shows up, we know that depressed appears 8 times, need advice appears 8 times, experiences appears 6 times, and suicide appears 4

TABLE V
TOPICS OF TOP 10 ACTIVE USERS

Rank	User ID	Topic Labels
1	24395	need advice; depressed; experiences; suicide; mood
2	10457	depressed; mood
3	11254	need advice; experiences; suicide
4	11882	romance; need advice; suicide
5	12260	romance; experiences; depressed
6	1379	depressed; need advice; experiences
7	5526	symptoms; depressed; need advice; medication; suicide
8	11318	depressed; experiences; need advice
9	10470	need advice; depressed
10	5579	need advice; experiences; depressed

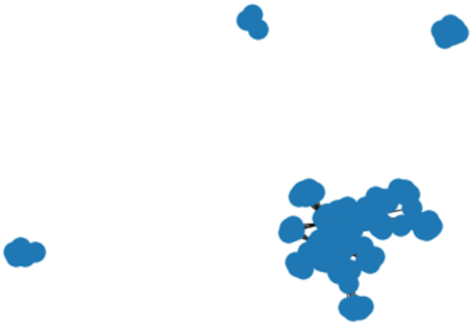


Fig. 3. 25 threads-Network 1

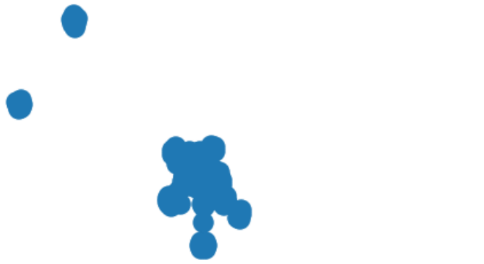


Fig. 4. 25 threads-Network 2

times. Romance, mood seldom appear and symptoms, medication just appear once separately. We can conclude that most people on this bipolar forum tend to experience depressed feelings and ask for help.

C. Construct networks

1) Constructed network graph: We constructed the three network following the detailed methodology as required. And the graphs of page 50 as well as pages from 46 to 50 are shown separately from Fig.3 to Fig.8:

2) Weights for constructed networks: Here we show part of weights of network 2 and network 3 in TABLE VI: The weights in network 2 vary from 1 to 10 and in network 3 vary from 1 to 10 as well. Active users tend to reply quickly.

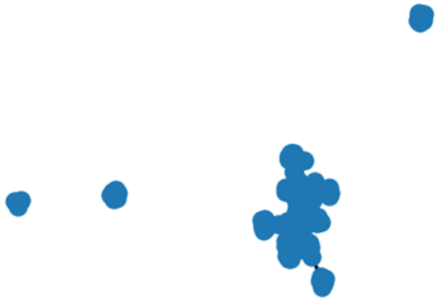


Fig. 5. 25 threads-Network 3

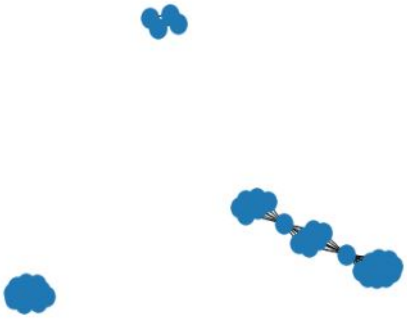


Fig. 6. 125 threads-Network 1

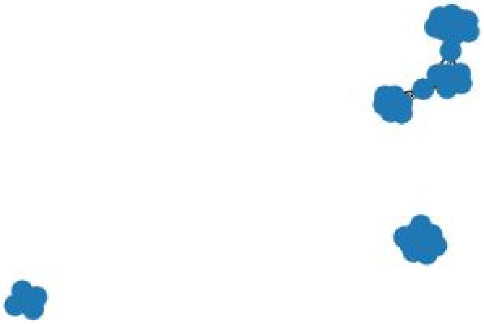


Fig. 7. 125 threads-Network 2

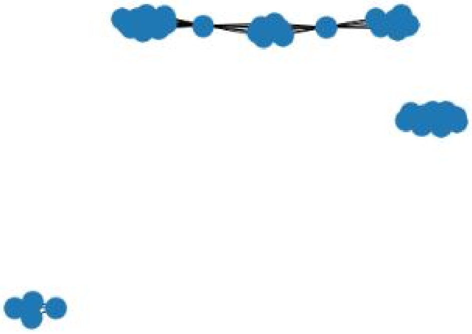
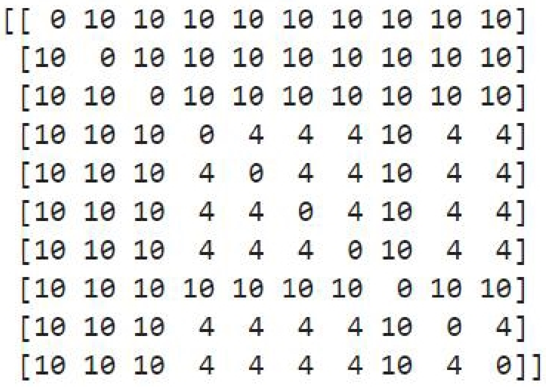


Fig. 8. 125 threads-Network 3

TABLE VI
SOME WEIGHTS OF NETWORK 2NETWORK 3

user pairs	N2 weight	N3 weight
('10457', '10676')	10	10
('10701', '10662')	10	10
('10701', '10745')	4	10
('10701', '2792')	6	10
('1379', '10910')	8	10
('1379', '19669')	10	7
('11254', '10687')	6	10
('15511', '5816')	4	9
('10687', '29450')	8	8



```
[[ 0 10 10 10 10 10 10 10 10 10]
 [10  0 10 10 10 10 10 10 10 10]
 [10 10  0 10 10 10 10 10 10 10]
 [10 10 10  0  4  4  4 10  4  4]
 [10 10 10  4  0  4  4 10  4  4]
 [10 10 10  4  4  0  4 10  4  4]
 [10 10 10  4  4  4  0 10  4  4]
 [10 10 10 10 10 10 10  0 10 10]
 [10 10 10  4  4  4  4 10  0  4]
 [10 10 10  4  4  4  4 10  4  0]]
```

Fig. 9. part of adjacent matrix of network 2

D. Attributes of each network

The following results are obtained based on the data from page 50, which has the mediate number of the replies. We calculate the attributes of each network in terms of number of nodes, number of edges overall clustering coefficient, average path length, size of giant component, diameter, maximum degree, average degree, number of communities using Girvan-Newman algorithm and adjacent metrics as implemented in NetworkX shown in the table. We can see that overall clustering coefficient, average path length and size of giant component vary according to different weight. For these 25 threads, 124 users and totally 616 edges are obtained. Therefore, the users on this forum appear and reply a lot. The average degree is 9.935, which indicates that one person will have connections with approximately 10 persons. Also, generally as the weight increases, the average path length and size of giant component increase while the overall clustering coefficient decrease. Part of adjacent matrix of network 2 is shown in Fig.9 as an example.

TABLE VII
ATTRIBUTES OF CONSTRUCTED NETWORK GRAPH

Attributes	**Network1**	**Network2**	**Network3**
number of nodes	124	124	124
number of edges	616	616	616
overall clustering coefficient	110.743	93.399	100.711
average path length	2.448	20.214	21.686
size of giant component	577	4856	5307
diameter	5	5	5
maximum degree	44	44	44
average degree	9.935	9.935	9.935
number of communities	14	14	14

E. Centrality value

We calculate the different centrality values for each network. Then find out the highest value and its corresponding node. The result is shown in the table.

For these 125 threads, these nodes are nearly all the same one and the centrality measures related with weight show different in-betweenness centrality values and page rank centrality values.

TABLE VIII
CENTRALITY VALUES FOR CONSTRUCTED GRAPH

		N1	**N2**	**N3**
degree centrality	nodes	10457	10457	10457
	value	0.357	0.357	0.357
in-betweenness centrality	nodes	10457	57702	10457
	value	0.175	0.171	0.278
closeness centrality	nodes	10457	10457	10457
	value	0.495	0.495	0.495
page-rank centrality	nodes	10457	10457	10457
	value	0.028	0.033	0.026

VII. CONCLUSION AND PERSPECTIVES

In the passage, we discuss the process of mapping the discussions of bipolar forum of the Mental Health Forum. First we scrap the dataset from web pages despite of very challenging conditions. Then we built three different networks: one with no weight, one with the number of replies as weight, and another one with reply time as weight. Finally we obtained some analytic results, which are very insightful.

People with long-term conditions of bipolar may feel isolated if they don't know other people who has the same problem. [3] That's why having a forum as this one is very helpful. In the bipolar forum, people share experience, ask questions, give advice and support each other. We believe that they can form strong connections online, and in the forum there are a number of communities. As a matter of fact, our dataset is only a very small portion of the forum, and there are 13 communities formed.

Health information on Internet has an empowering effect, partially because patients and care givers play an active role in managing health and receiving peer support. [4] As mentioned before, the 10 most active members of the bipolar forum have posted many messages related to giving advice, sharing stories, depression and suicide, which are really beneficial for other members of the forum.

Online forums may be a cost-effective and pragmatic option for enhancing peer support for people with bipolar disorder. [5] From the project, we can conclude that the bipolar forum is a large social network, including many communities and important nodes. It provides a really positive and powerful influence on the bipolar patients, as well as their family and friends.

REFERENCES

[1] Aric A. Hagberg, Daniel A. Schult, Pieter J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX", Proceedings of the 7th Python in Science conference (SciPy 2008), G. Varoquaux, T. Vaught, J. Millman (Eds.), pp. 11–15.

[2] Aric Hagberg, Drew Conway, "Hacking social networks using the Python programming language (Module II – Why do SNA in NetworkX)", Sunbelt 2010: International Network for Social Network Analysis.

[3] Bauer R, Bauer M, Spiessl H, Kagerbauer T. Cyber-support: An analysis of online self-help forums (online self-help forums in bipolar disorder). Nord J Psychiatry 2013 Jun;67(3):185-190.

[4] Davies M. NM Incite. USA: NM Incite; 2011. Healthcare social media by the numbers. URL: http://www.slideshare.net/NMIncite/healthcare-social-media-by-the-numbers-sxsh-2011 [accessed 2015-06-01]

[5] Poole R, Smith D, Simpson S, ”How Patients Contribute to an Online Psychoeducation Forum for Bipolar Disorder: A Virtual Participant Observation Study”, JMIR Ment Health 2015;2(3):e21

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] Hagberg, Aric Swart, Pieter Chult, Daniel. (2008). Exploring Network Structure, Dynamics, and Function Using NetworkX. Proceedings of the 7th Python in Science Conference.

[9] Chhetri, Kamal. (2019). Regular Expression.

[10] Rencher, Alvin C.; Christensen, William F., Chapter 10, Multivariate regression – Section 10.1, Introduction, Methods of Multivariate Analysis, Wiley Series in Probability and Statistics 709 3rd, John Wiley Sons: 19, 2012.

[11] Hilary L. Seal. The historical development of the Gauss linear model. Biometrika. 1967, 54 (1/2): 1–24.

[12] Lehmann, E. L.; Casella, George (1998). Theory of Point Estimation (2nd ed.). New York: Springer

[13] Newman, M. E. J.; Reggiani, Aura; Nijkamp, Peter (2005). ”Power laws, Pareto distributions and Zipf’s law”. Cities. 30 (2005): 323–351.

VIII. APPENDIX

TABLE IX
THE THREAD TITLES THAT ’24395’ INVOLVED IN

no.	**thread title**
3	very high, very low
4	Questions?
10	Quetiapine (Seroquel) Experiences
20	Struggling to cope
21	Very Very low
27	I Love To Sleep
35	How many zoppies can I safely take? Rough Night.
42	MANIA: The craziest thing I did. What about you?
49	hello everyone sunday 5th december, how are we...
53	goodbye
61	i dunno who i am any more
64	My Cat has died
65	Bad fall and anxiety attack
70	Free from medication, couldn’t be more po...
71	Back to the circus
72	i just want to die
73	I’m bored!! Anyone about?
75	a step in the right direction...
76	what a shocker. Im a really bad person - long ...
80	a to the r to the g to the h!
82	Over Medicated and Under Motivated
83	I’m furious! Need to let off steam
86	political correctness
91	If i was a dog theyd put me down.TORTURE
96	Still Throwing Up
98	adn so...
100	I don’t want to go mad on my own!
103	God, why is life so stressful!!
110	A tumultuous patch in the tranquility of the o...
112	Are any of you premmie babies?
124	Want to die