Fan Guo

A20473828

Project I - Social Media Data Analysis for Twitter

Project Github Link: https://github.com/fguo10/SocialMediaDataAnalysis

In this project, I use twitter token and tweepy package to crawl twitter friends data, as well as process and report some analysis on my extracted data.

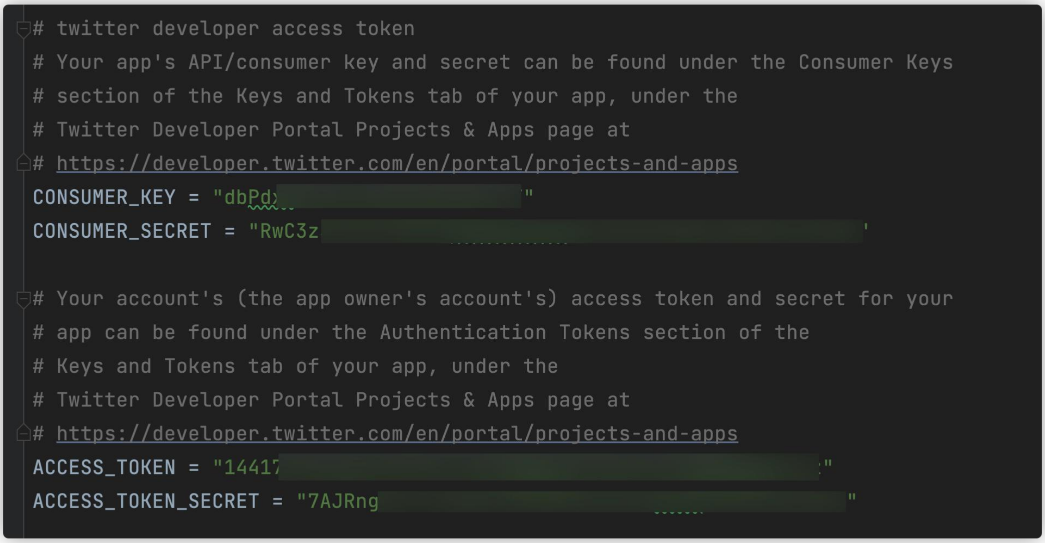
It contains 3 steps here:

1) Data Collection
2) Data Visualization
3) Network Measures Calculation

1. Data Collection

Create an account on Twitter, then visit the link: https://developer.twitter.com/en.

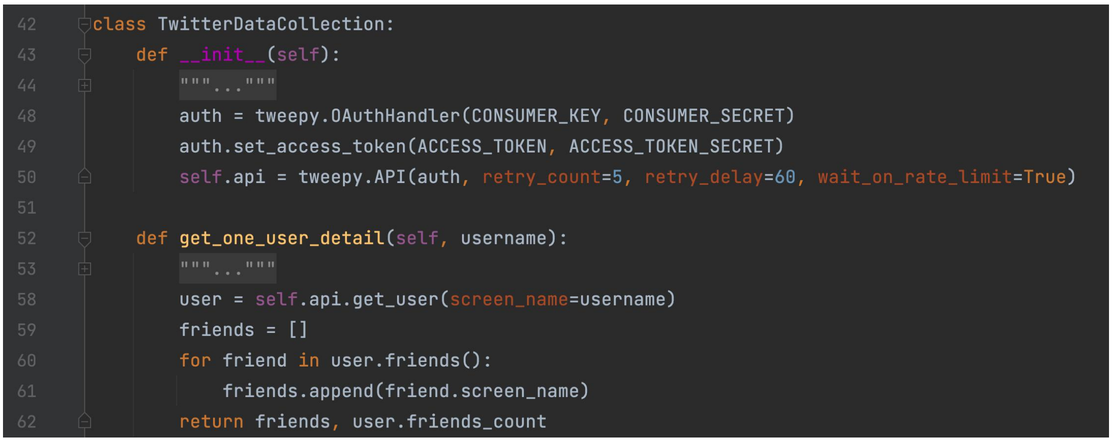
Then click on Apply for a Developer account, we can select Specify the type of application, and Specify the type of application, put in why you want to apply for the developer account. When the API developer is approved, we can get token details:



```
# twitter developer access token
# Your app's API/consumer key and secret can be found under the Consumer Keys
# section of the Keys and Tokens tab of your app, under the
# Twitter Developer Portal Projects & Apps page at
# https://developer.twitter.com/en/portal/projects-and-apps
CONSUMER_KEY = "dbPd[redacted]"
CONSUMER_SECRET = "RwC3z[redacted]"

# Your account's (the app owner's account's) access token and secret for your
# app can be found under the Authentication Tokens section of the
# Keys and Tokens tab of your app, under the
# Twitter Developer Portal Projects & Apps page at
# https://developer.twitter.com/en/portal/projects-and-apps
ACCESS_TOKEN = "14417[redacted]"
ACCESS_TOKEN_SECRET = "7AJRng[redacted]"
```

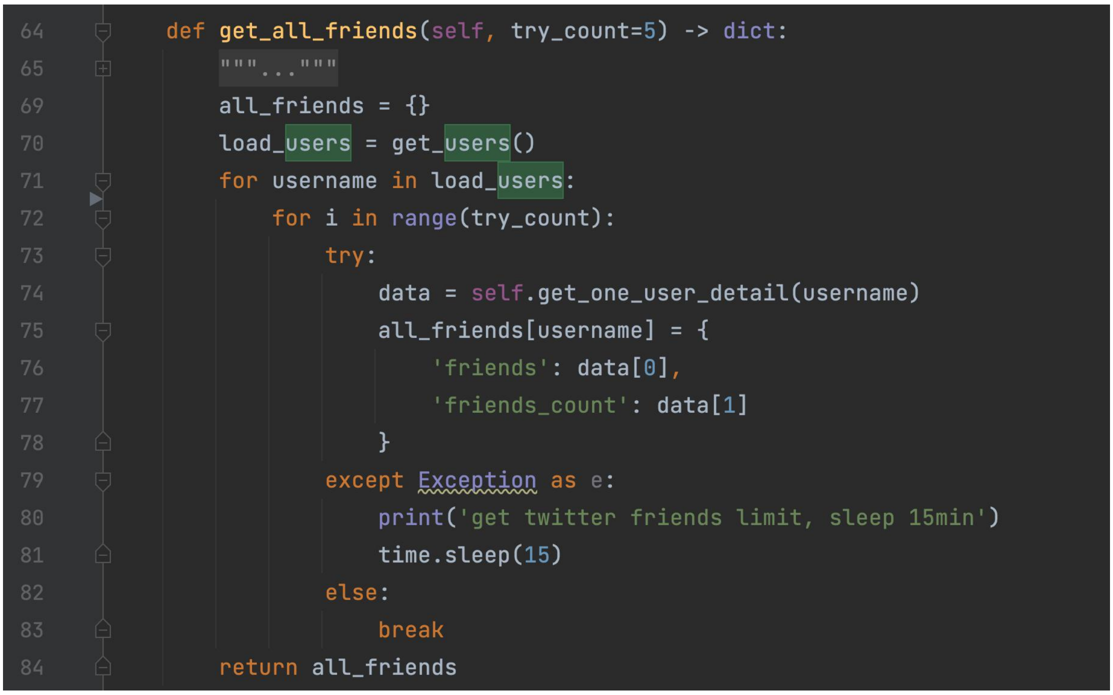
Base on the API token, we can use tweepy python package esaily get twitter user friends.As we know, Tweepy is an open source Python package that makes it easy to access Twitter and get information about friends for developers to use.

Firstly, Setting the authentication credentials.



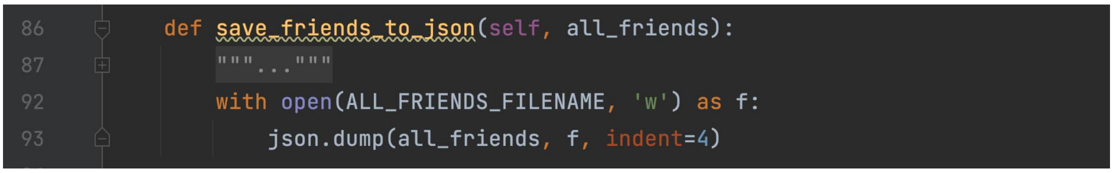
```python
class TwitterDataCollection:
    def __init__(self):
        """..."""
        auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
        auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
        self.api = tweepy.API(auth, retry_count=5, retry_delay=60, wait_on_rate_limit=True)

    def get_one_user_detail(self, username):
        """..."""
        user = self.api.get_user(screen_name=username)
        friends = []
        for friend in user.friends():
            friends.append(friend.screen_name)
        return friends, user.friends_count
```

However, Twitter has a rate limit on the number of requests made to the Twitter API. Allow 900 requests /15 minutes. If get the request error because of too many request, we need wait 15 minutes and try again, the code as below:



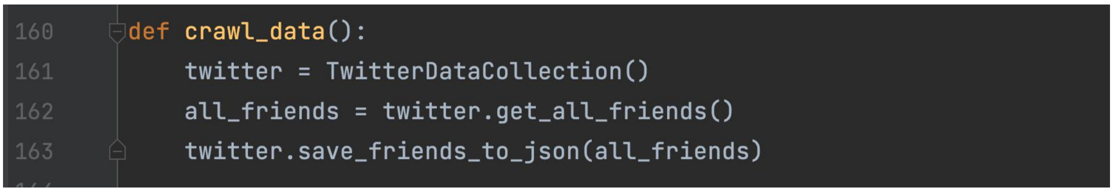
```python
    def get_all_friends(self, try_count=5) -> dict:
        """..."""
        all_friends = {}
        load_users = get_users()
        for username in load_users:
            for i in range(try_count):
                try:
                    data = self.get_one_user_detail(username)
                    all_friends[username] = {
                        'friends': data[0],
                        'friends_count': data[1]
                    }
                except Exception as e:
                    print('get twitter friends limit, sleep 15min')
                    time.sleep(15)
                else:
                    break
        return all_friends
```

Meanwhile, to reduce the number of Twitter requests, I save the data into files doc/all_friends.txt. When analyzing data, we don't have to collect friends of a given user on Twitter every time. The code details as follows:



```python
    def save_friends_to_json(self, all_friends):
        """..."""
        with open(ALL_FRIENDS_FILENAME, 'w') as f:
            json.dump(all_friends, f, indent=4)
```

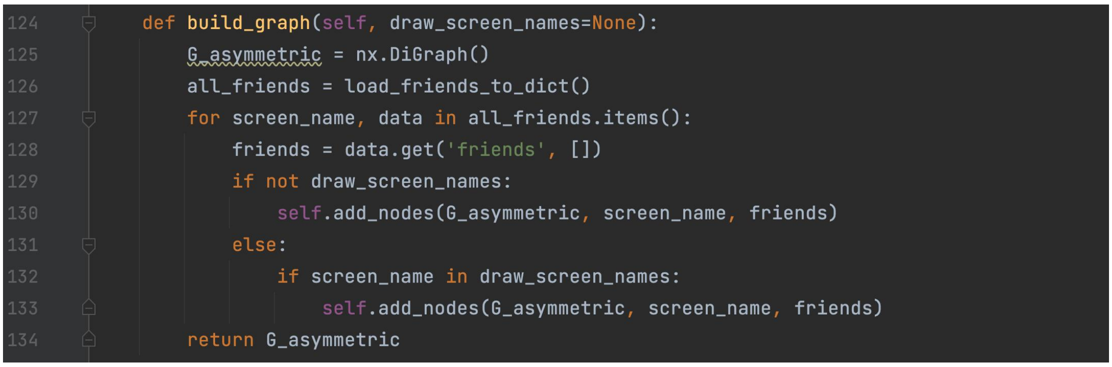
The part about data collection has been completed, the steps executed are stored in the function crawl_data, and the file doc/all_friends.txt will be generated after the execution, the details are as follows:



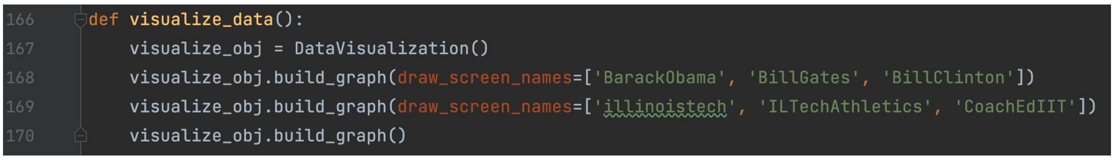
```python
def crawl_data():
    twitter = TwitterDataCollection()
    all_friends = twitter.get_all_friends()
    twitter.save_friends_to_json(all_friends)
```

2. Data Visualization

I have fetched the data, in this step, I use networkx to visualize network as graph. The core code here:

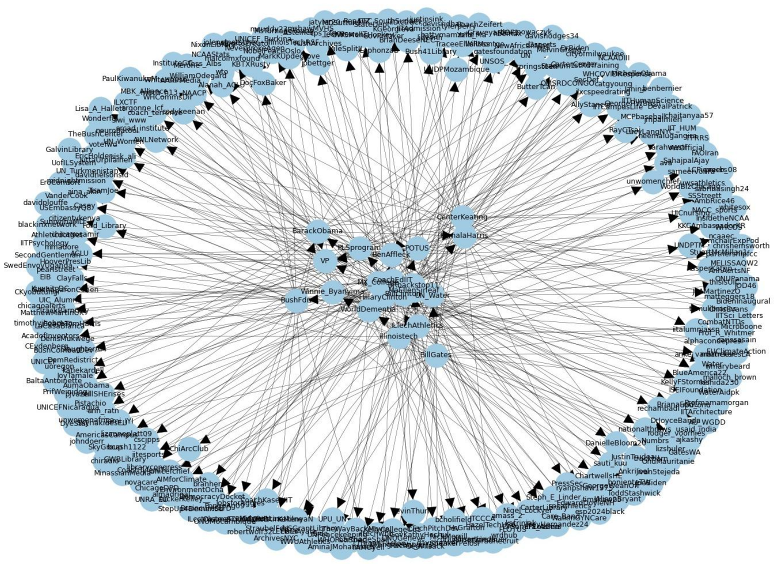


```python
    def build_graph(self, draw_screen_names=None):
        G_asymmetric = nx.DiGraph()
        all_friends = load_friends_to_dict()
        for screen_name, data in all_friends.items():
            friends = data.get('friends', [])
            if not draw_screen_names:
                self.add_nodes(G_asymmetric, screen_name, friends)
            else:
                if screen_name in draw_screen_names:
                    self.add_nodes(G_asymmetric, screen_name, friends)
        return G_asymmetric
```

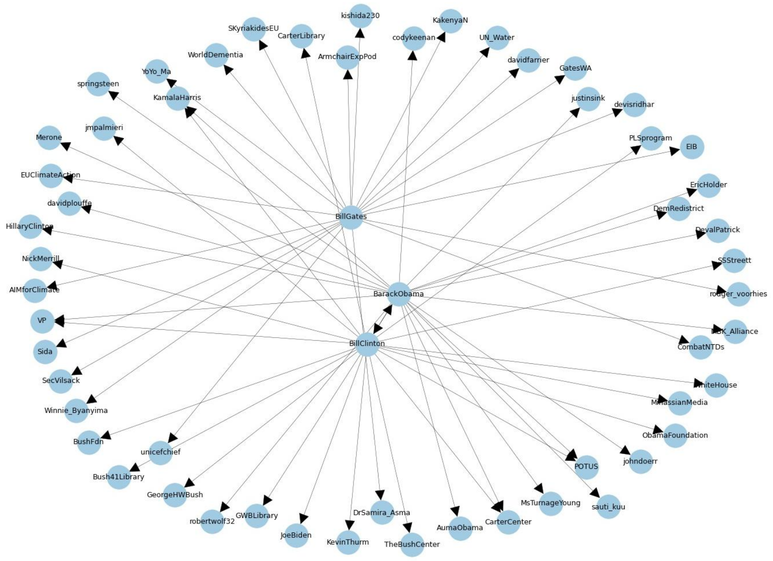


```python
def visualize_data():
    visualize_obj = DataVisualization()
    visualize_obj.build_graph(draw_screen_names=['BarackObama', 'BillGates', 'BillClinton'])
    visualize_obj.build_graph(draw_screen_names=['illinoistech', 'ILTechAthletics', 'CoachEdIIT'])
    visualize_obj.build_graph()
```

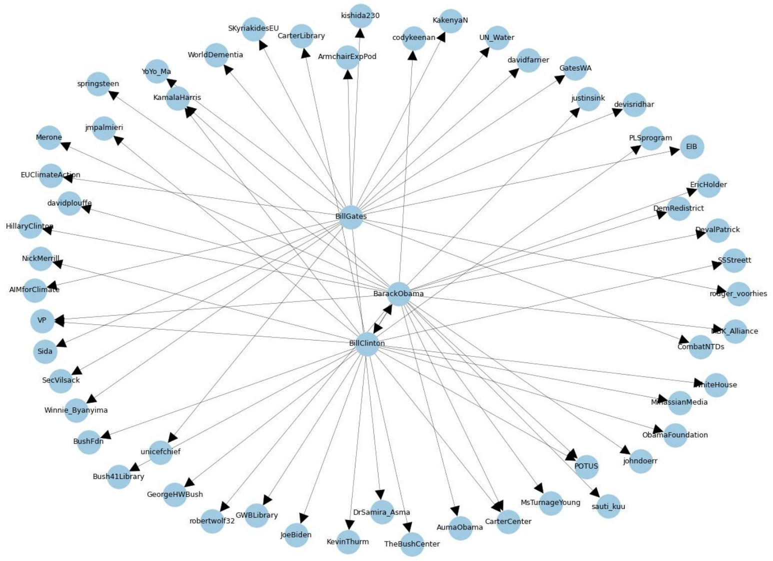
When I use all the data(400+ nodes) here, I got a complex graph:



When I choose screen_users: ['BarackObama', 'BillGates', 'BillClinton'], get a simple graph here:

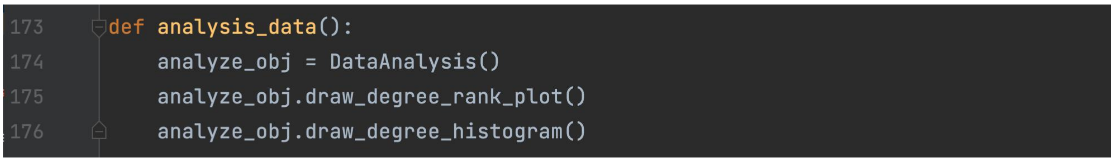


Then I choose screen_users: ['illinoistech', 'ILTechAthletics', 'CoachEdIIT'] , get a simple graph here:



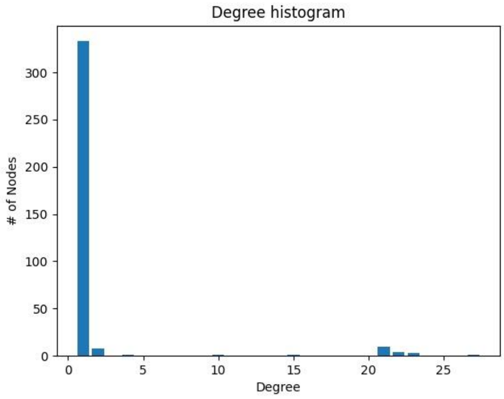
3. Network Measures Calculation

In this step, I use networkx and matplotlib to caculation Degree Distribution and Pagerank, and plot it as histogram. The core code here:

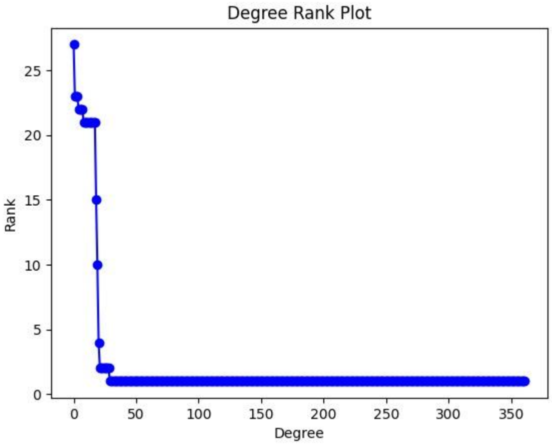


```
def analysis_data():
    analyze_obj = DataAnalysis()
    analyze_obj.draw_degree_rank_plot()
    analyze_obj.draw_degree_histogram()
```

Then the degree histogram as the picture shows:



The degree-rank plot for the Graph as the picture shows:



4. Conclusion

In this project, Firstly, get twitter data from API and save data into file. Secondly, we do data visualize about 100+ nodes and 400+nodes, At the end , we do various type of statistical analysis on the tweets. After the completion of the project, I learned a lot about online network analysis

Project Github Link: https://github.com/fguo10/SocialMediaDataAnalysis