# CSE 472 – SOCIAL MEDIA MINING

# PROJECT – I

## Word Co-occurrence Network using Telethon

Submitted by -
Vamsi Krishna Kanagala ( ASU ID - 1218608781 )

# IMPLEMENTATION

## 1. Obtaining credentials and Scraping data:

For this project, I was assigned the platform Telegram to scrap the data and to build the network. The network I chose to develop is Word Co-occurrence network.

And the first step of the implementation process is to scrape the data from the platform. Telegram provides an API, Telethon, to scrawl through the platform and provides an access to read through the data. For getting the access, we need to create a Telegram by providing the phone number. Once an account is created, we can establish a connection via Telegram API by creating a basic application to generate the api_id and api_hash values.

This project is implemented in Python 3 using the libraries telethon, networkx, nltk, matplotlib etc., which are used to develop and execute scripts in the project.

In this project, I have scraped the messages from the Telegram channel, “One America News Network” (group id: OANNTV) by a keeping a total messages count (total_messages) of 100000 reading 2000 messages at a

time and saving all the messages in a list. Upon reaching the total messages limit, this data is stored in a json file named, “messages.json”.

2. Preprocessing the data:

Once completing data scraping, the next step in this project is to preprocess the obtained data.

In this step, I have loaded the data from the json file into pandas dataframe by selecting the columns, id, title, description of each message record. Here I used the python library, nltk (Natural Language Tool kit), which contains a module named “stopwords”. This contains a set of words which are very frequently used in sentences in English (example: the, he, I, you, what, where, how etc.,).

For each record in the dataframe, I excluded all the stop words in both the columns, title and description, and stored the remaining words. By doing so, we can get the key words present in the title and description of each message record and can reduce the denseness of our word connection graph.

From the updated data, the pair of words that occurred frequently were retrieved by using nltk’s module “Bigram”. This function will take a list of strings as input and will return a list of pairs of words that occurred

frequently. This frequency is then obtained by using “Counter” from “collections” library in Python.

Once after completing this step, we have a list of pairs of words and their frequencies. For plotting the graphs, the top 250 commonly occurring pairs of words were considered. Note that, these pairs of words can have a common word but with different combinations and hence the total number of nodes in the graph might be less than 250.

3. Data Visualization:

For plotting the network, I made use of the python’s libraries matplotlib and networkx. Matplotlib library is used to generate the network measures such as degree distribution, betweenness and closeness. Networkx library is used to plot the Word connection network using the most frequent bigrams obtained from the earlier steps.

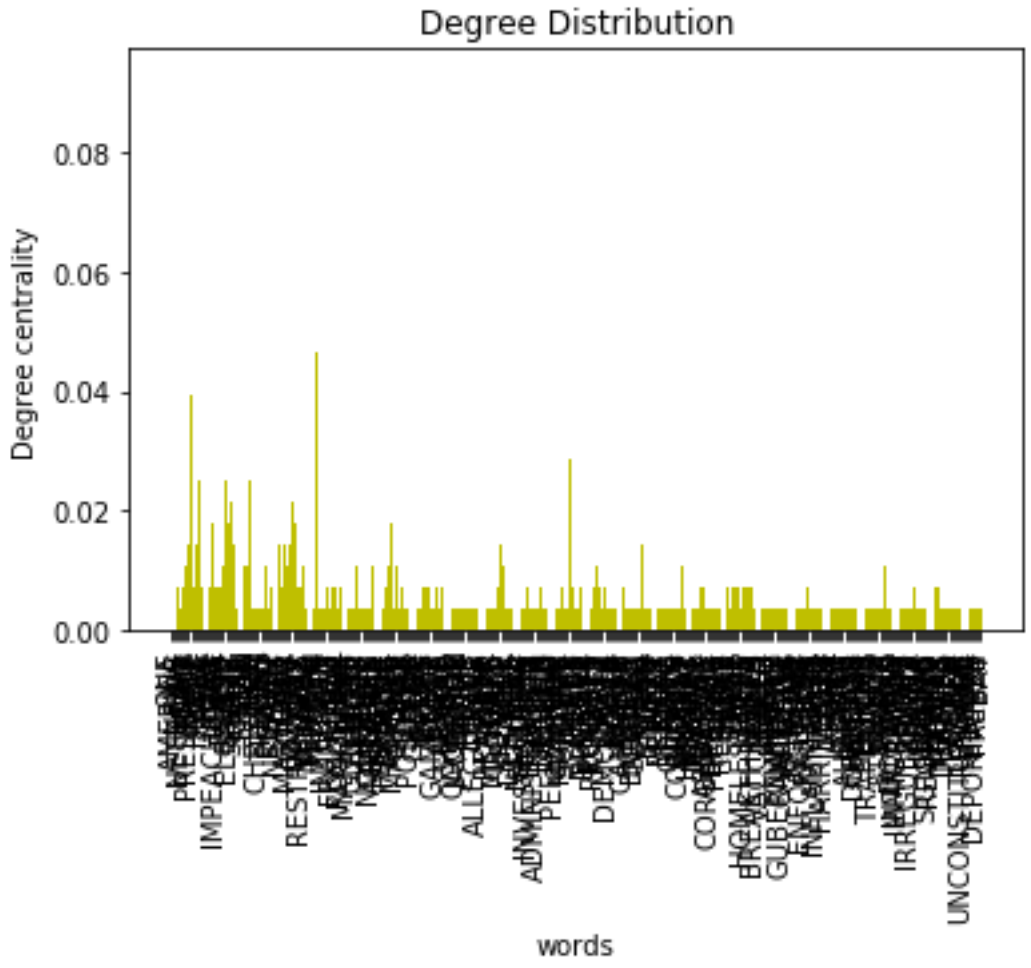
The files used in the project are as follows,
a. code.py – which establishes as connection to the telethon api and scrape the data from the telegram channel mentioned earlier.

b. graph.py – which preprocess the data and generate the different visualizations mentioned above.

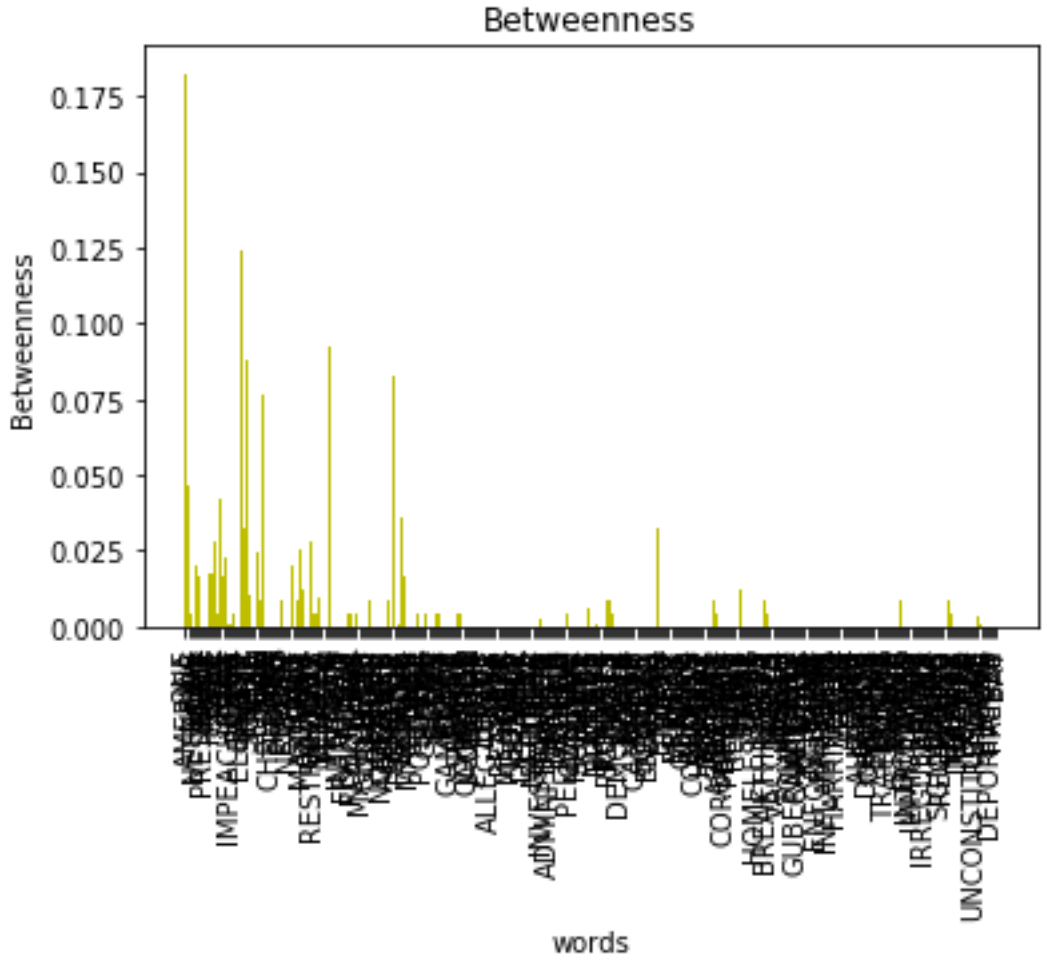
c. messages.json – which contains the messages that are scraped from the telegram channel mentioned earlier.

## VISUALIZATIONS

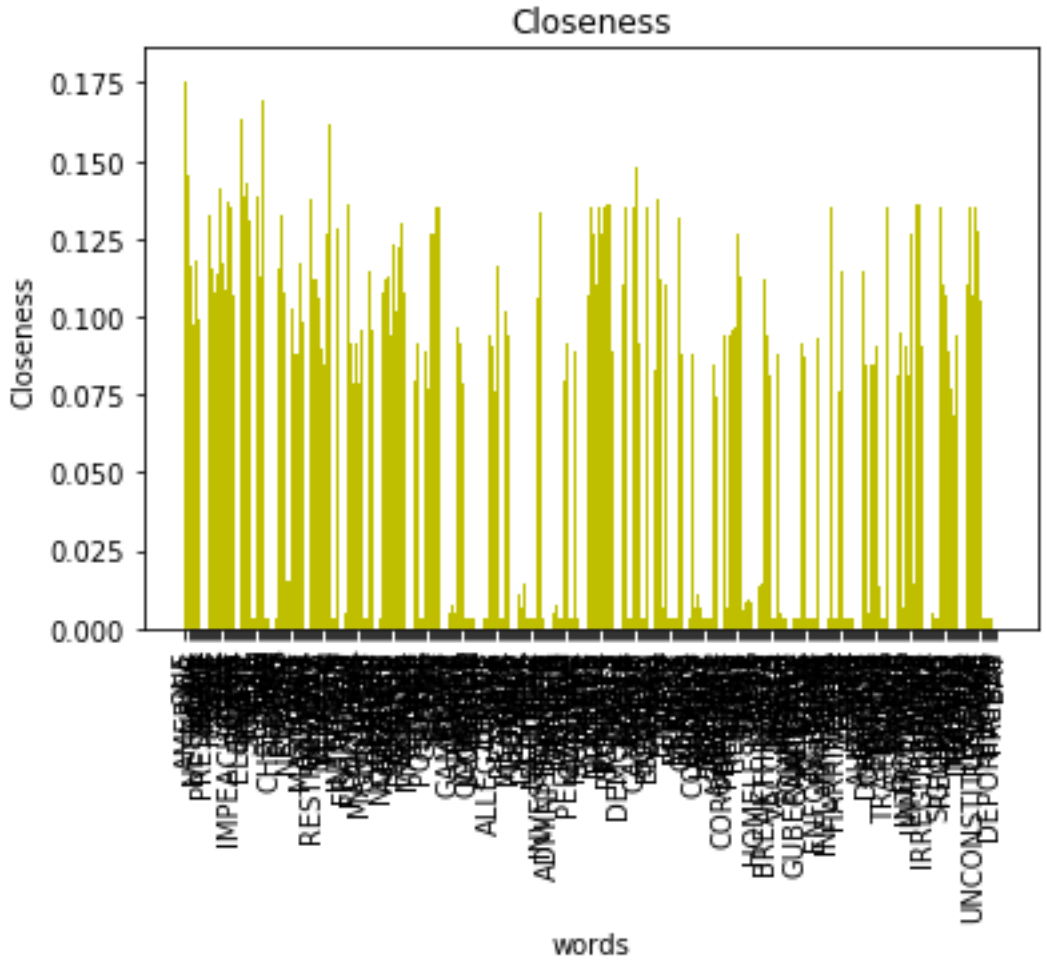
## A. DEGREE DISTRIBUTION



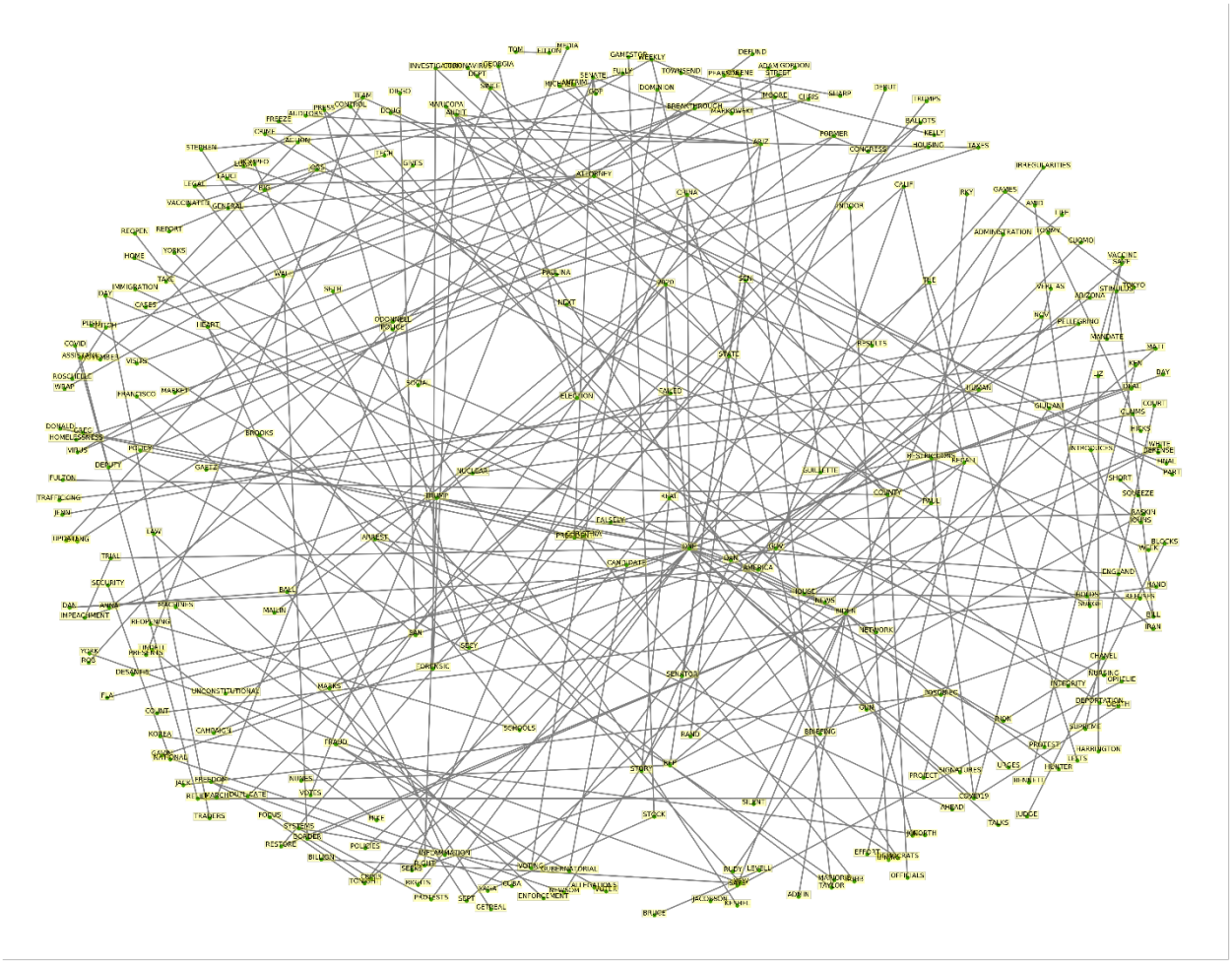
# B. BETWENNESS



# C. CLOSENESS


Closeness
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Closeness
words

# D. WORD CONNECTION NETWORK



All these visualizations are labelled as Figure_1.png, Figure_2.png, Figure_3.png, Figure_4.png

## 4. References

Below are the URL links for the resources that helped us to achieve the results of this project.


- https://docs.telethon.dev/en/latest/
- https://docs.telethon.dev/en/latest/concepts/asyncio.html
- https://www.nltk.org/api/nltk.html
- https://gist.github.com/sebleier/554280
- https://www.nltk.org/api/nltk.tokenize.html
- https://matplotlib.org/stable/gallery/statistics/hist.html
- https://networkx.org/documentation/stable/tutorial.html