Mark Gameng

CS 579

**Project 1**

For this project, I decided to use Twitter to make a follower and followee relationship using IIT’s president, Raj Echambadi. His twitter is @rechambadi with 445 followers. This is not that much followers but due to Twitter’s API rate limiting, going through all of his followers would take half a day. So, instead I decided to randomly choose 50 of his followers and get their following/followers for the network.

To get the Twitter followers and information, I used the Twitter API and Tweepy, a python library for that API. Essentially, I used my credentials such as my bearer token and then found a user’s following and followers through the API. I then saved the “links” or edges, such as whether a user is following a person, or getting followed by a person. Here are some screenshots below to show the code that I used to get the data.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

I also saved all the edges in a json file so you can use it without having to use Twitter API since it requires a bearer token. Also, take note, that this took a few hours, due to rate limitations. Twitter API only allows 15 follower lookups in 15 minutes, so essentially, 1 follower lookup every minute. Here are some examples of the edges from @rechambadi.

Text, letter

Description automatically generated

These edges or tuples are essentially (follower, user). So, in this case, @drcurstisodom and @Handyside are following @rechambadi. @drcurstisodom -> @rechambadi

At the end, I got **8226** total nodes and **9066** edges. I will clean this graph later in the paper to have a better visualization. But without any edits, and filters, this is what the default graph looks like. A higher resolution png file will be included in the submission as well.

A close-up of a cactus

Description automatically generated with low confidence

With this graph, the **degree distribution** is as such.

Chart, bar chart, histogram

Description automatically generated

As you can see, most nodes, or users, have a degree of less than 25. Specifically, **7452** nodes have a degree of **1**, and **668** nodes have a degree of **2**. The average degree of this network is **2.2** and the max degree is **208** from NU\_Business. This is interesting because that is Northeastern School of Business, and this network was built from IIT’s President’s followers. Here is the total degree distribution.



The next other thing I looked at was **PageRank**. The average was **0.0001** and the max is **0.007** from @LenM\_Ops**.** Here is the histogram, which is very similar to the degree distribution histogram.

Chart, bar chart

Description automatically generated

The next thing that I looked at was **closeness centrality**. The average was **0.021** and the max is **0.269** from @rechambadi which makes sense. Here is the histogram.

Chart, histogram

Description automatically generated

To clean up the graph, I tried various methods. One method was removing nodes with low closeness and a degree of 1. Which resulted in this which is much better than the previous one. This however only shows 287 nodes.

A picture containing day

Description automatically generated

Another method was removing nodes with an in-degree of 1 which resulted in this. This one has a total of 3942 nodes.

Scatter chart

Description automatically generated with medium confidence

I found this visualization to be better, while the layout is random, it is much easier to see the “important” nodes.

The graph below has less emphasis on the lines with added labels as well. The labeled nodes, or users, have in-degrees greater than 80 which means that those users have 80 or more followers in the network.

A picture containing text

Description automatically generated