Sumanth Nallamotu
GWU Research 2022

Leveraging Graph DB's to Trace Mis/Disinformation Across Social Media

Motivation:

With the growth of the Internet and applications facilitating public discourse like Twitter, Reddit, and Facebook, the spread of information has been revolutionized. To many, this is great; people from across the world can discuss topics they're interested in like never before. We have unprecedented knowledge about other cities, states, and countries and their political, economical, and societal conditions. It could be argued that online discourse is one of the most important developments in modern history.

However, one major pain point is poisoning this alleged progress. Misinformation and disinformation have been rampant issues in the past couple years. Whether it is bending the truth to fit a political agenda or flat out disseminating lies, credibility of public figures and news networks has taken a massive hit. Recognizing this, many have called on companies like Twitter and Facebook to look into the deceptive content on their platforms and mitigate it. Even after a year or so of their efforts, taming misinformation and disinformation has proven to be a difficult endeavor.

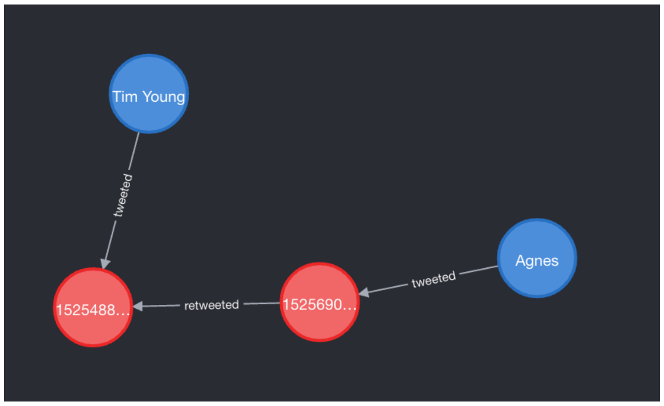
Goal:

Stopping mis/disinformation is difficult because there can be endless debates about what the "truth" really is. In addition, public discourse, whether online or in-person, is crucial and putting a muzzle on any particular crowd is surely asking for trouble. This is why an alternative approach is necessary: detection.

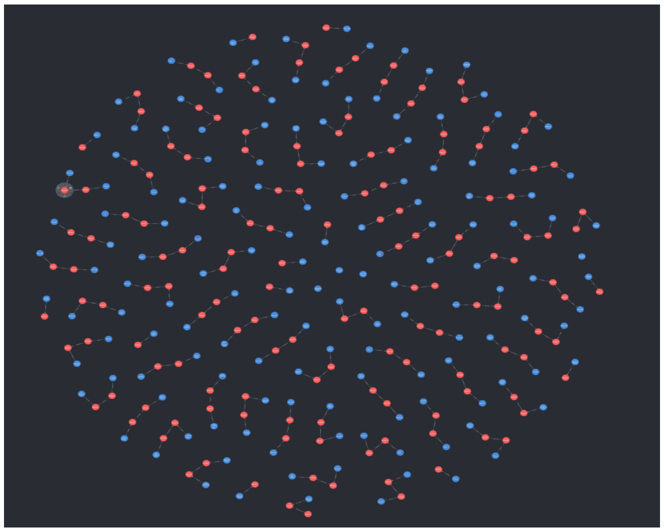
With an influx of bot accounts and spam created with ulterior motives (recent example: Russian Disinformation Effort Through Twitter), identifying suspicious account behavior patterns can be essential for mitigating this particular aspect of mis/disinformation control. But how can we do this? With the minds behind such accounts possessing technical aptitude of their own, how can we look past their deception, flag malicious accounts, and track the spread of mis/disinformation across the Internet?

Implementation:

The approach I felt that was best-suited for accomplishing this goal was to create a graph database containing posts and detailing connections such as who authored it and how they interacted with other authors' posts. I decided to start with Twitter, as that was the platform whose API I was most familiar with.



Using Python and Neo4j, I was able to create a sample graph that contained 456 nodes representing individual tweets with a number of attributes (i.e. tweet text, author ID). If a tweet referenced another tweet (i.e. retweeted, quoted, or replied to), a node for the referenced tweet would be created. In addition, the authors along with a number of attributes (i.e. time zone, language, followers, account age) had their own nodes with relationships created between them and the tweets they authored. This is demonstrated by the screenshot below. Each blue node represents an author while each red node represents a tweet. The intention here is to be able to run queries to identify patterns between different types of accounts and the content they are posting. For example, if a particular account with a low account age is interacting with tweets from another account with a low account age, is that cause for suspicion? Are they bots spreading potential mis/disinformation?



Looking at the graph as a whole, I believe that it can allow us to find interesting interactions between different accounts. If we are able to create a map detailing all these interactions, can we use each node's attributes to identify any red flags that will point us to bots or bad actors? What I am hoping for with this approach is that it lays the groundwork for further investigation. Simply having access to different social media platforms APIs is not enough. We must have a means for tracing information not only within one platform but also across multiple platforms.

Future Work

I must admit that my code implementation is a little crude. I simply wanted to see if the graph database approach would be as promising as I hoped. With more time, I will be able to make it more modular and efficient as well as expanding the attributes per tweet and author. I also want to include an element of traceability: if a tweet contains a link, where does it lead? Do tweets from a particular user always lead to a certain website? If so, are there other users following the same pattern? I believe the graph approach will provide a means through which we can form research questions and investigate patterns.

For whatever topic we'd like to look into, generating a graph of posts and querying it with whatever we'd like to investigate is how I envision this approach panning out. An example query in Neo4j could be:

```
MATCH (a:author) WHERE a.account_age < 10 AND a.total_posts > 10 RETURN a
```

where we’re looking for author’s who have an account age over 100 days and over 100 followers in an effort to find recently created accounts who have been posting very frequently- behavior that could be seen as bot-like.

Other features I’d like to include are graph functionality for other platforms to create one unified graph to enable tracing of potential mis/disinformation across different platforms (i.e. posts about the Ukraine-Russia conflict from Twitter, Reddit, and Facebook), NLP to determine sentiment of a tweet by itself or in relation to its referenced tweets, leveraging of AWS Neptune to enable crowdsourcing of tweet/post-mining to one or multiple unified graph databases (i.e. different users can pool their mined tweets to an AWS Neptune graph database regarding the Russia-Ukraine conflict while other users send their mined tweets to a graph database about the 2024 election), and dashboard views of graph databases and charts depicting query results. There are many directions we can take this approach, and I am optimistic that we can move further towards our goal by leveraging the functionality of graph databases.