CSE 472: Social Media Mining

Amey Bhilegaonkar
Ira Fulton School of Engineering
Arizona State University
Tempe, Arizona
abhilega@asu.edu

I. Introduction

In the age of digital interconnectedness, this research delves into the realm of social media mining, with a specific focus on the contentious Canada-India controversy as a case study. It explores the intricate web of social media networks, investigating data collection techniques, network construction, and large language model (LLM) classification to decipher diverse perspectives within this digital discourse. By leveraging Mastodon's REST APIs, we gather a wealth of data, construct Friendship Networks to visualize user relationships, and employ LLMs to classify users into pro, neutral, or anti-categories. This research contributes to our understanding of online discourse, network structures, and sentiment analysis, offering valuable insights into the dynamics of contemporary digital conversations.

II. Topic Selection and Data Crawling

In this section, I will provide a detailed overview of the data crawling process carried out to build a network of at least 300 nodes for the controversial topic of the Canada-India controversy. The objective of this data crawling exercise was to collect data from Mastodon using specific search keywords or hashtags and to create a network that incorporates diverse perspectives on the topic.

A. Topic Selection

The chosen topic for this project is the Canada-India controversy. This topic was selected due to its relevance and the presence of varying opinions and perspectives within the Mastodon platform. To ensure a comprehensive view, I aimed to include pro-topic, neutral, and anti-topic perspectives in the network.

B. Data Collection Methodology

To collect relevant data for our network, we utilized specific search keywords and hashtags associated with the chosen topic. These included hashtags such as:

1) #Canada
2) #India

These keywords and hashtags were carefully selected to cover a wide range of discussions and opinions related to the Canada-India controversy.

C. Data Crawling Process

We used the Mastodon REST APIs to retrieve data from the platform. Our data crawling process involved the following steps:

1) **Authentication**: Set up authentication to access the Mastodon APIs securely. This included obtaining the necessary access tokens.
2) **API Queries**: Crafted API queries using the selected search keywords and hashtags. These queries allowed us to retrieve posts and user profiles related to our topic. Specifically used **timeline_hashtag** API with **pagination** to retrieve data.
3) **Data Retrieval**: Collected data in JSON format, which included information such as post_id, account_id, account_notes, and post content.
4) **Data Cleaning**: We performed data cleaning to remove any irrelevant or duplicate entries, links, and mentions, and converted the HTML response of the content body into plain text using Python libraries. Translated non-English content into English language using Google Translator library. This ensured the quality and consistency of the collected data.
5) **Data Storage**: The cleaned data was saved in JSON format for further analysis and network construction.

III. Friendship Network Construction

In this section, I will provide a detailed overview of the construction and visualization of a Friendship Network. This network represents user relationships on the Mastodon platform, where nodes represent users, and edges indicate friendship relationships.

A. Network Construction

Identified friendship relationships between users by analyzing the "follower" and "followee" attributes. If user A follows user B, establish a directed edge from node A to node B using account_id field.

NetworkX Library:
We chose the NetworkX library for network construction due to its flexibility and comprehensive features for network analysis. The following steps were taken to build the Friendship Network:

1) **Node Creation**: Each user profile was represented as a node in the network. I used the "account_id" as a unique identifier for nodes.

2) **Edge Creation**: Friendship relationships between users were represented as directed edges between nodes. If user A followed user B, we added an edge from node A to node B.
3) **Graph Creation**: Created a directed graph using NetworkX, where nodes represented users, and edges represented friendship relationships.
4) **Graph Visualization**: The constructed Friendship Network was visualized using NetworkX's visualization capabilities. The nodes and edges were displayed to provide a clear representation of user relationships.

B. Snapshot of the Friendship Network

Below is a snapshot of the Friendship Network, which includes nodes representing users and directed edges indicating friendship relationships:

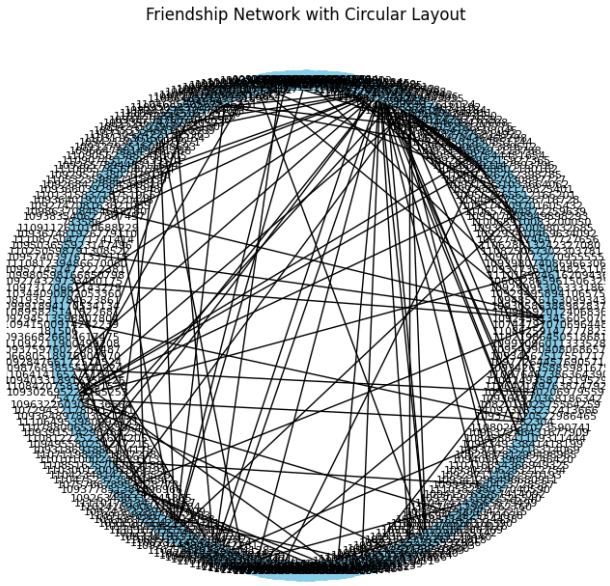

Fig. 1. Friendship Network

IV. Data Storage

To preserve the constructed Friendship Network for future analysis and visualization, we used the following data structure for node attributes:

Node: Each node (representing a user) had two key attributes:

1) "account_id": A unique identifier for the user.
2) "post_content": A list of post content associated with the user.

This data structure allowed us to store user-specific information while retaining the network structure. The Friendship Network and associated node attributes were saved in a suitable format for future use.

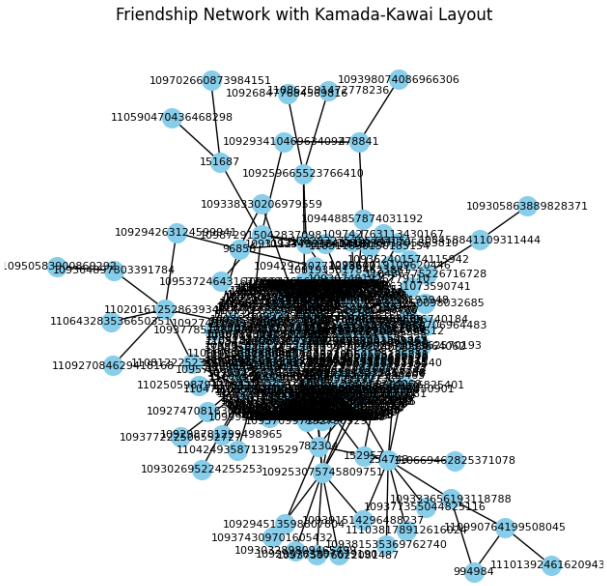

Fig. 2. Friendship Network

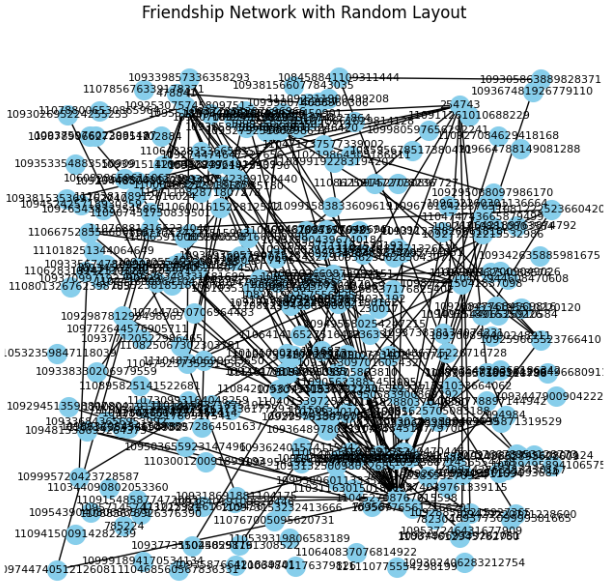

Fig. 3. Friendship Network

V. Node Classification

The primary focus of our project was node classification. I aimed to categorize users into one of three groups: pro-Canadian, neutral, or anti-Canadian. Large Language Models, specifically Alpaca-7B, played a crucial role in this classification process.

A. Classification Prompts

Prompts served as essential guiding mechanisms for the LLMs. We designed prompts that incorporated user account

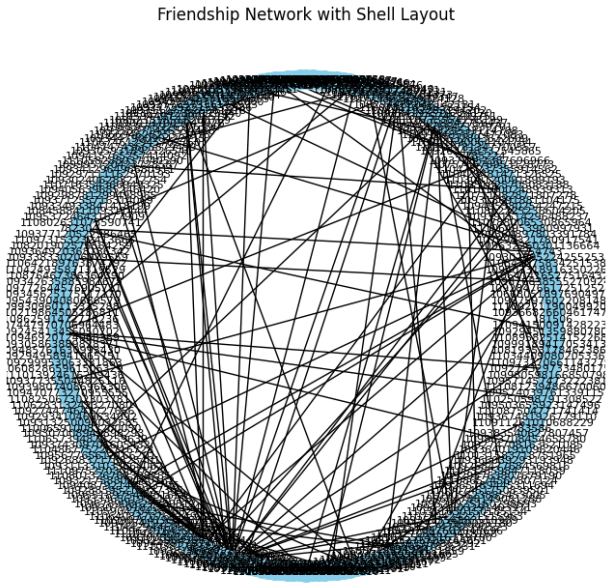

Fig. 4. Friendship Network

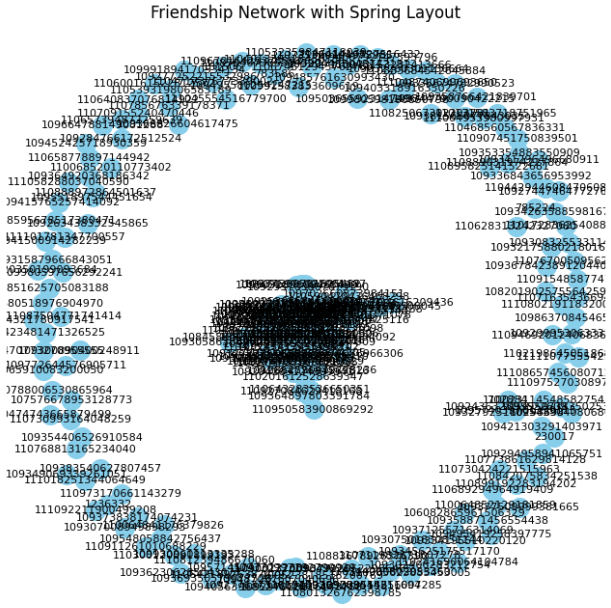

Fig. 5. Friendship Network

notes and post content, structured in the following prompt:

```
Below is an instruction that describes
a task. Write a response that
appropriately completes the request.

### Instruction:
By analyzing input in the following
format:
post by the user: {}

Please classify this user's post
sentiment as either
1. pro
2. Neutral or
3. Anti
based on their post by the user.

Give only ONE WORD answer.

### Response:
```

These prompts instructed the LLM to classify users based on their expressed sentiments regarding the Canada-India issue.

B. Classification Methodology

Our classification methodology relied on the responses generated by the LLMs. We formulated prompts that aimed to elicit responses indicative of users' sentiments. By instructing the LLMs to classify individuals as pro-Canadian, neutral, or anti-Canadian, we gained insights into user opinions and affiliations within the network.

C. Visual Representation

To provide a visual representation of the classified network, we employed NetworkX, a Python library for network analysis. Nodes and their classifications were added to the graph, with different colors assigned to each category (e.g., green for pro-Canadian, blue for neutral, and red for anti-Canadian). This visualization offered a clear depiction of the network's composition and node classifications.

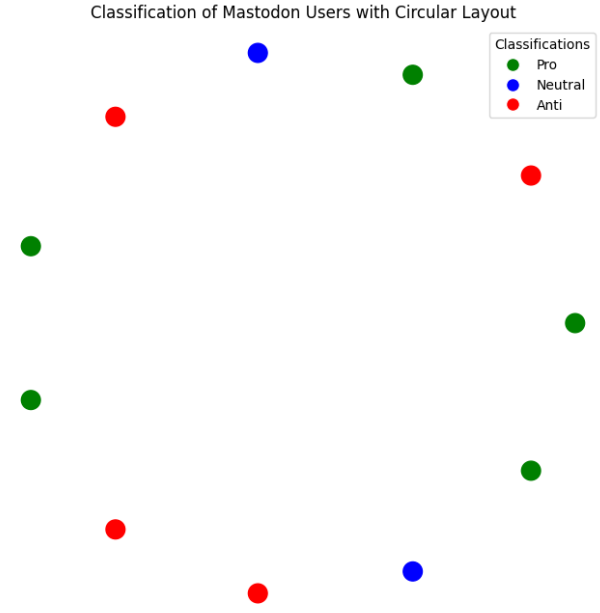

Fig. 6. Node Classification

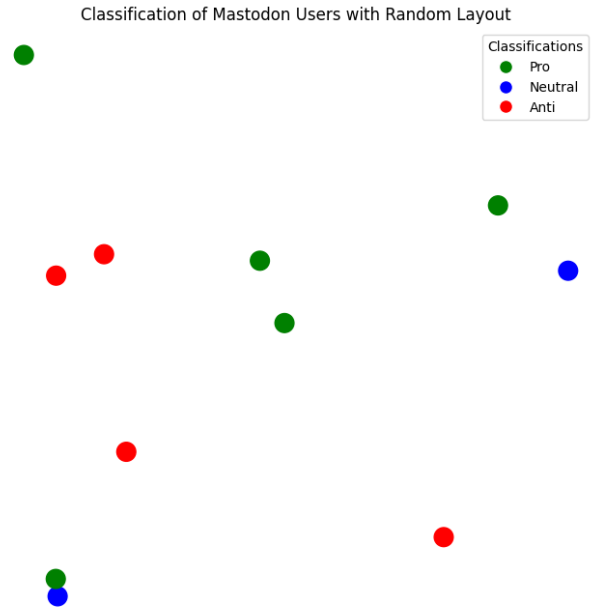

Fig. 7. Node Classification

VI. Challenges

Node classification in social media networks comes with its challenges. These include dealing with unstructured textual data, designing effective prompts, and ensuring accurate model responses. The main challenges were out-of-memory errors and longer responses for a single query to LLM. It took 2 hours for my machine to create a graph and another 2 hours to query to LLM and fail miserably with out-of-memory error.

I then shifted to Google Collab where it also failed after running for 3+ hours with the prompts. To resolve it, I limited my results to 70 Nodes only for classifying the dataset. Below is the classification of 70 Nodes. I will be creating a config file where we can modify this number if we have extended hardware for the same.

VII. Findings

Degree Distribution Histogram for the entire network. The x-axis represents the node degrees, while the y-axis shows the frequency of nodes with a specific degree.

Classification of Nodes doesn't have an edge between them, I suspect this because I am limiting results to 70 nodes while classifying nodes with no common edges might have been selected. I selected fewer nodes because the LLM response was taking a lot of time and was giving out-of-memory issues with larger nodes. If the case would have been otherwise, we might have seen an excellent network and derived wonderful results.

As observed in Figure 8, the Degree Distribution for the 300 Node Network follows a power-law distribution. This indicates that a few nodes have a disproportionately large number of connections, reflecting a network's skewed and highly connected structure, with implications for robustness and information flow.

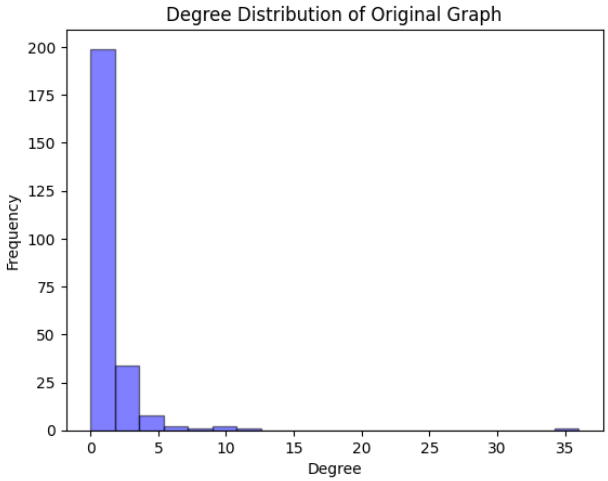

Fig. 8. Degree Distribution Histogram for 300 Node Network

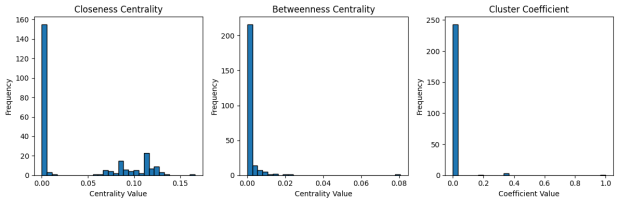

Fig. 9. Closeness, Betweenness and Cluster Coefficient for 300 Node Network

References

[1] Mastodon API Documentation: Link.

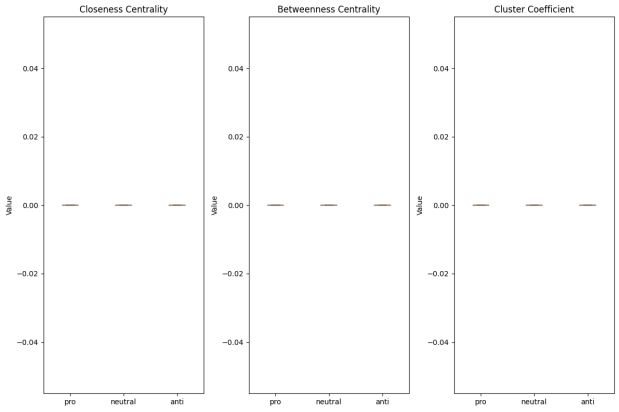

Fig. 10. Closeness, Betweenness, and Cluster Coefficient for classification Node Network