CSE 472: Social Media Mining

Project I - Social Media Data Analysis

Prof. Huan Liu
TA: Alimohammad Beigi
abeigi@asu.edu
Due at 2023 September 22^{nd}, 11:59PM

This is an ***individual*** project assignment. Please refer to the Academic Integrity information at the end of the document before you start this project. You will work on the steps described in the project guideline to create the dataset, codes, and report (See the "Submission" section for more details). We highly recommend you finish reading the whole document before starting your implementation. To ensure you can submit on time, we encourage you to begin working ***ASAP***!

Project Objectives

Through this project, you will develop skills in social media data crawling, how to operate an open-source large language model while crafting compelling prompts, and performing exploratory analysis on the gathered data. For this project, you are required to crawl data from the **Mastodon** social media platform. Additionally, you have the freedom to choose a topic of your choice. However, it is crucial to select a controversial topic that fosters the emergence of contrasting viewpoints, such as Climate Change (pro-climate change, neutral, anti-climate change), Threads (pro-Threads, neutral, anti-Threads), Twitter (anti-Elon, neutral, pro-Elon), AI tech industry (AI tech optimist, neutral, AI tech pessimist), or other similarly engaging issues.

Project Guideline

Step 1. Obtaining Mastodon API credentials.

The initial step is to acquire the necessary credentials (i.e., API key) to scrape data from Mastodon. The process for obtaining these credentials on the Mastodon platform is described in the appendix, and we strongly recommend initiating the request promptly. Once you have obtained the credentials, you will need to familiarize yourself with using the API. You can refer to the official documentation or use existing libraries such as Mastodon.py, which is a Python library specifically designed for interacting with the Mastodon API. Additionally, you have the freedom to select any other Python-based library or tool that suits your preferences.

Step 2. Data Collection.

In this step, you are asked to crawl the data based on your preferred topic. For the selected controversial topic, you need to build a network with at least 300 nodes. For example, to examine climate change, you will need to build a network that consists of perspectives from pro-climate change, neutral and anti-climate change. To explore different perspectives, you will crawl the Mastodon data using specific *search keywords* or *hashtags*. For example, hashtags like *"#Climate"*, *"#ClimateChange"*,

"#ClimateCrisis", and *"#CleanEnergy"* are few of the hashtags related to the topic of climate change. This approach will allow you to build a network that includes contrasting perspectives. It is important to save the collected data in the **JSON** format, and please provide a brief description in your report explaining your topic selection and how you crawled your data.

Step 3. Network Construction and Visualization.

Now you need to construct and visualize your data as a network. There are multiple packages and software available for network analysis such as, networkx (recommended), snap, NodeXL, etc. Choose one and read the instruction on how to build your graph. Each package requires a certain graph data format, such as edge list, adjacency matrix, or adjacency list.

Following are two examples of constructing a network. You can choose one of the following network types or define your own network type, as long as you can justify it.

1. Friendship Network: A user's friendship network can be represented as a graph in which the nodes represent the users, and the edges indicate the presence of a friendship relationship between them. The graph depicts the users and their follower and followee relationships in a directed manner.
2. Diffusion Network: Each node in the network represents a user who can publish, receive, and propagate information. A directed edge between the nodes indicates the direction of information propagation. Example: toots propagation where the nodes are users and the edges are boosts (reposts)/comments/mentions.

You need to provide a description of the data collection and network construction process in your report. Please note that in addition to the crawled mastodon content, different types of networks may require collecting various features and additional information about toots and users. In your report, describe the features you used to construct the graphs and include a snapshot of your generated graphs.

Step 4. Classification.

Now, it's time to use the capabilities of Large Language Models (LLMs) for performing classification. By providing suitable prompts, you can classify nodes into different groups. The quality of your prompts plays an important role in correctly classifying nodes. As an illustration, if you were to classify the climate change network, you would end up with three separate node categories: pro-climate change, neutral, and anti-climate change.

The prompt serves as an initial input or text that guides the model in generating coherent and contextually relevant responses. It can range from a few words to a sentence or even a paragraph, providing the necessary context or specifying the desired output. You can refer to the following resources for more information such as An Introduction to Prompt Engineering and How to construct prompts. Keep in mind that you will need to design the suitable prompts which align with both the LLM itself and the task at hand. In this project, you can choose one of the following LLMs for performing classification: **Llama2-7B** or **Alpaca-7B**. The Appendix provides a comprehensive guide on how to use Llama2-7B and Alpaca-7B, with easy-to-follow steps.

There are multiple options available for classifying individuals. One way is to use the text content of user profile bio or posts that are associated with the users. Prompt the LLM you have chosen to analyze the given text content. The objective is to identify contrasting perspectives expressed in the

texts and then determine the appropriate classifications. For instance, in a climate change network, you may instruct the LLM to classify individuals as either pro-climate change, neutral, or anti-climate change. Subsequently, link these classification attributes to the appropriate nodes in the network. It's crucial to understand that you have the flexibility to devise your own classification method, as long as you can provide a valid justification for it.

Regardless of the option you choose, it is essential to include a detailed description of the process in your report. In the report, write down the steps you followed to classify the network and capture a snapshot of the classified graph, incorporating them as visual references.

Step 5. Metrics.

In the course, you will explore various network measures such as Degree Distribution, Clustering Coefficient, PageRank, Diameter, Closeness, and Betweenness. For the network you have constructed, it is mandatory to implement three distinct network measures for each classified group (e.g. we have three classified groups for climate change such as pro-climate change, neutral, and anti-climate change) and report the differences in the network measures. Utilize your preferred software or package from step 2 to acquire the "Degree Distribution" and visualize it as a histogram. In addition to this, choose two other measures that you learned in the class and plot them accordingly, either as distributions or as single values. Include the generated results in your report for reference.

Submission

We will run your code to see if it works for all steps. The final submission should include data, source codes, and a report (preferred to be in PDF format). Submit everything on "Gradescope". You can decide whether to upload your submission as a zip file or not because it will be automatically uncompressed once you upload the zip file. Most importantly, you are only allowed to use **Python** as the programming language. Make sure your source code has the **".py"** extension. i.e., If you used iPython, then you have to convert it into ".py" extension before you submit your codes.

In summary, your final submission should contain the following items:

1. The project report that satisfies the requirements suggested in each step.
2. The dataset generated by your code (The dataset should be in the format of a ".json" file).
3. Source codes with ".py" extension and related files (e.g., config file, environmental file, etc).

Grading Criteria

pts	Description
3	Data Collection
3	Network Construction and Visualization
3	Classification
1	Metrics
10	

Table 1: Grading Rubric

Academic Integrity

- To prevent any potential plagiarism, we will randomly select students for each phase of the project and ask them to talk with the TA.
- Your submission codes will be automatically checked by the similarity detection tools.
- For Step 2, you have to develop your own code for data scraping. It is NOT permissible to use publicly available datasets.
- For all the steps, you can only *refer* to others' code and use libraries, software, and packages but it is NOT permissible to copy any existing code from others.
- Use a "Reference" section and cite all the tutorials, packages, software, and libraries you used in your data.

APPENDIX: Instructions on how to obtain API-keys

Mastodon:

1. Visit https://joinmastodon.org/servers
2. Choose a Mastodon server:
   (a) Decide on the specific Mastodon server you want to access. Mastodon is a decentralized platform, and each server operates independently with its own API.
   (b) Sign into your Mastodon account or Sign up for a new one.
3. Create a new application:
   (a) Click the gear icon to access your Mastodon preferences.
   (b) In the bottom left corner of your home page, click the "Developers' link.
   (c) On the "Your applications" page, click the blue "NEW APPLICATION' button.
   (d) Give your application a name (e.g., "Data Crawler"), and select all the access that is provide there. You can always change this later.
   (e) At the bottom of the page, click the blue "SUBMIT" button.
4. Obtain API credentials:
   (a) After creating the application, you will receive your API credentials, which typically include a client ID and a client secret. These credentials are unique to your application and are required for authentication and access to the Mastodon API.
5. Authenticate your requests:
   (a) After creating the application, you will receive your API credentials, which typically include a client ID and a client secret. These credentials are unique to your application and are required for authentication and access to the Mastodon API.
6. Test your API access:
   (a) Once you have obtained your access token or completed the authentication process, you can test your API access by making a simple API request, such as retrieving the timeline or user information. This will confirm that your credentials are valid and that you can successfully communicate with the Mastodon API.

APPENDIX: Instructions on how to work with Llama2-7B

Llama2-7B

1. Begin by initiating a request for access to the "Llama2". Provide your name and ASU email, followed by indicating your country as "United States" and your organization as "Arizona State University". Typically, access approval is granted within 1-2 days.

2. Once you've submitted your request, proceed to create an account on "Hugging Face" using the same ASU email address employed in the previous step.

3. After you have been granted access from Meta for Llama2, use the provided form to enable access to Llama2-7B on Hugging Face. Please note that the email address associated with the Hugging Face account **MUST** match the email you provide on the Meta website at step 1, or your request will not be approved. Typically, access approval is granted within 1-2 days.

4. Upon completing step 3, navigate to your Hugging Face account. In the upper-right corner, click on the "Profile Icon". Next, click on the "Settings". Proceed to the "Access Tokens" section by clicking on "Access Tokens". Then select "New Token", and set the name as "Llama2". Choose the role as "Read", then finalize the process by clicking on "Generate a Token". Now you can see the generated token for Llama2.

5. To start working with Llama2-7B, create a copy of the Google Colab code provided in the link and save it to your Google Drive. Then under the section "Loading Llama 2" on the Google Colab code, insert the generated token in the "INSERT YOUR TOKEN HERE".

6. Customize the generation configuration by adjusting the following parameters to align with your preferences:

   (a) Temperature: it refers to a parameter that controls the randomness or diversity of the generated output. A higher temperature value, such as 1.0, leads to more varied and creative responses, as it introduces more randomness into the model's output. Conversely, a lower temperature value, such as 0.5, produces more focused and deterministic responses, favoring more common or probable choices.

   (b) Top p: it restricts the model's output to a subset of the most likely next words based on their cumulative probability. Instead of considering all possible words, the top p sampling selects from the top p fraction of words with the highest probabilities. This technique allows for better control over the model's output and helps avoid improbable or non-sensical completions.

   (c) Repetition penalty: it discourages repetitive or redundant phrases in the generated output. It helps the model avoid producing the same or similar phrases multiple times within a single response. By applying a repetition penalty, the model is encouraged to generate more diverse and coherent responses, enhancing the overall quality of the generated text. The penalty can be set to a value greater than 1 to increase the discouragement of repetitions.

   Modify these values according to your desired level of creativity, coherence, and avoidance of repetitive responses.

7. Once the parameters are initialized, refer to the provided examples that demonstrate how to create instructions that describe your specific task. These examples serve as helpful references to help you effectively frame your desired instructions.

By following these instructions, you will be able to work efficiently with Llama2-7B.

APPENDIX: Instructions on how to work with Alpaca-7B

Alpaca-7B

1. Begin by watching the "Running Alpaca-7B in Colab" video, which provides a comprehensive overview of how to effectively utilize Alpaca-7B.

2. To start, create a copy of the Google Colab code provided in the link and save it to your Google Drive. This allows you to customize the code and leverage the capabilities of Alpaca-7B.

3. Customize the generation configuration by adjusting the following parameters to align with your preferences:

   (a) Temperature: it refers to a parameter that controls the randomness or diversity of the generated output. A higher temperature value, such as 1.0, leads to more varied and creative responses, as it introduces more randomness into the model's output. Conversely, a lower temperature value, such as 0.5, produces more focused and deterministic responses, favoring more common or probable choices.

   (b) Top p: it restricts the model's output to a subset of the most likely next words based on their cumulative probability. Instead of considering all possible words, the top p sampling selects from the top p fraction of words with the highest probabilities. This technique allows for better control over the model's output and helps avoid improbable or non-sensical completions.

   (c) Repetition penalty: it discourages repetitive or redundant phrases in the generated output. It helps the model avoid producing the same or similar phrases multiple times within a single response. By applying a repetition penalty, the model is encouraged to generate more diverse and coherent responses, enhancing the overall quality of the generated text. The penalty can be set to a value greater than 1 to increase the discouragement of repetitions.

   Modify these values according to your desired level of creativity, coherence, and avoidance of repetitive responses.

4. Once the parameters are initialized, refer to the provided examples that demonstrate how to create instructions that describe your specific task. These examples serve as helpful references to help you effectively frame your desired instructions.

By following these instructions, you will be able to work efficiently with Alpaca-7B.