WEEK 9

Progress Report

# Introduction

Our project aims to provide a tool for analyzing the sentiment of users in a specific community or connected subgraph, based on various constraints specified by the user. The tool allows users to search for keywords and specify time intervals, and then view the sentiment analysis of the users within the selected community or subgraph.

We understand the importance of sentiment analysis in understanding public opinion and making data-driven decisions. Our tool utilizes advanced natural language processing techniques to accurately identify the sentiment of users' posts or comments within the specified community or subgraph.

In the following demo, we will showcase the functionality of our tool and demonstrate how it can be used to gain valuable insights into public opinion.

# Objective:

The main aim of our project is to contribute to the field of sentiment analysis and provide a useful tool for analyzing social media data. By leveraging the power of natural language processing, we aim to develop a model that can classify tweets as positive, negative, or neutral based on the language used and the overall sentiment expressed. We believe that such a tool will be valuable for businesses, researchers, and individuals looking to gain insights from social media data.

# Overview:

This week, we made significant progress in implementing a feature that allows users to specify constraints such as keywords and time intervals, and then see the sentiment of users in a connected subgraph close to a query user. Specifically, we were able to complete the following tasks:

1. **Data preprocessing:** We preprocessed the dataset by cleaning and formatting the raw data to ensure that it can be properly analyzed by our algorithms. This involved removing duplicates, handling missing data, and converting data types where necessary.
2. **Keyword extraction**: We implemented a keyword extraction algorithm that allows users to input a set of keywords and retrieves all posts that contain those keywords. This algorithm uses natural language processing techniques such as part-of-speech tagging and lemmatization to identify relevant posts.
3. **Time interval selection:** We added a feature that allows users to specify a time range and retrieve all posts that fall within that range. This feature is based on a simple filtering algorithm that compares the timestamps of each post to the specified time interval.
4. **Sentiment analysis:** We integrated a sentiment analysis algorithm that evaluates the sentiment of each post in the connected subgraph. This algorithm uses a pre-trained machine learning model to classify each post as positive, negative, or neutral based on its content.
5. **Subgraph generation:** Finally, we implemented a subgraph generation algorithm that creates a connected subgraph consisting of all users who have interacted with the query user. This subgraph is used to evaluate the sentiment of users in the community close to the query user.

# Conclusion:

Overall, we made significant progress this week towards implementing a feature that allows users to see the sentiment of users in a community close to a query user. We were able to complete all the necessary preprocessing steps and implement the key algorithms required for this feature. We plan to continue refining and testing these algorithms over the coming weeks to ensure that they are accurate and reliable.

# Final week

In the final week, we should focus on wrapping up our project and preparing it for submission. Here are some suggestions on what we can do:

* Finalize our code: We should make sure our code is clean and well-organized. We should remove any unnecessary code, comments, or print statements. We should also make sure all functions and modules are working correctly.
* Write documentation: We should document our code thoroughly so that others can understand it. We should include instructions on how to install and run our code, as well as a detailed explanation of each function and module.
* Test our code: We should test our code rigorously to ensure that it works as expected. We should try different inputs and edge cases to make sure our code can handle unexpected situations.
* Write a summary: We should write a summary of our project, highlighting the problem we were trying to solve, our approach, and our results. We should also include any limitations or future directions for our project.
* Prepare for presentation: We should prepare a presentation to showcase our project. We should summarize our project in a clear and concise manner, highlighting the most important aspects of our work.
* Submit our project: We should submit our project before the deadline. We should double-check that we have included all the necessary files and documentation.

Top of Form

FFFF

Top of Form