DS210 Final Project

The data I considered is Reddit and Subreddit embeddings. This dataset comprises two types of files: user embeddings and subreddit embeddings, both derived from Reddit data spanning 2.5 years from January 2014 to April 2017. The embeddings are vector representations, each with 300 dimensions. User embeddings correspond to individual users, with similarity between vectors indicating users posting in similar subreddits. Subreddit embeddings represent communities on Reddit, with similarity between these vectors reflecting subreddits shared by similar users.

I found this particularly interesting as it offers insights into the dynamics of online communities, particularly on Reddit. By analyzing these embeddings, researchers can understand how communities (subreddits) are related and how users engage across different topics. This research aslo assist in providing a method to quantitatively measure relationships and interactions within a large online network.

This Rust code implements a *Graph* structure to represent a directed graph using *HashMap* and *HashSet* from the Rust standard library (lines 1-4). The graph struct *Graph* is defined with a single field *adj*, representing adjacency relationships (line 7-9). The *Graph* implementation includes methods for initializing an empty graph, adding nodes using *add_node*, and creating directed edges using *add_directed_edge* and *add_edge* (lines 12-39). The method *to_dot* generates a graph in DOT format for visualization (lines 41-51). Nodes and edges can be loaded from CSV files using the *add_from_csv* method, integrating external data into the graph (lines 73-86). The main functionality of calculating the degrees of separation (shortest path) between nodes is provided by the *degrees_of_separation* method (lines 53-71). This capability is extended in the main function (lines 88-130) to handle multiple node pairs, demonstrating the graph's application in analyzing a network of Reddit users and subreddits. The code also includes a unit test to validate the creation of an empty graph using *empty_graph* test, to ensure that it runs as intended (lines 132-138).

In the output there is a long list in the beginning, this shows all the reddit usernames and the subreddits we considered in this dataset. I am looking at the degrees of separation, it is generally used to describe the number of intermediate steps or connections it takes to link one person or entity to another within a network or social graph. It is often used to measure the distance or level of separation between individuals within a social or professional context. Two variables might have a 1st degree of separation, meaning they know each each other directly. They can also have 2nd degree of separation, where they don't have a direct connection but are connected through a mutual link. In my analysis I found the following 1st degree of separation between reddit users and the subreddits they are active on:

- *fireteams* is connected to *amici_ursi*
- *funny* is connected to *unremovable*
- *askreddit* is connected to *rotoreuters*

- *globaloffensivetrade* is connected to *fiplefip*
- *the_donald* is connected to *CDRE_64*

Note: All of them in the following format, subreddit name → reddit username

The graph contains several nodes (vertices) represented by usernames, and directed edges connecting these nodes. Each line of the form A→B indicates a directed edge from node A to node B. This means there is a relationship or connection between A and B, where A points to B.

Initially I wanted to find degrees of separation for the whole set of nodes between subreddits and the reddit users but was not able to as my dataset has too many vertices and my terminal could not handle running the code and I kept getting the error *"cargo run terminated"*. So, I decided to only find the degrees of separation between specific variables;

I intially tried finding the shortest path between two users, namely rotoreuters and amici_ursi and this is the message I found in the output: *"No path found between rotoreuters and amici_ursi"*

This suggests that there is no direct connection or link between the usernames "rotoreuters" and "amici_ursi" in the graph. In other words, there is no way to move from "rotoreuters" to "amici_ursi" by following the existing connections between usernames.

I then tried finding the shortest path between a user and a subreddit, and this is the message I found in my output: *"Degrees of separation between rotoreuters and askreddit: 1"*

A degree of separation of 1 means that there is a direct edge connecting 'rotoreuters' to 'askreddit' or vice versa, this implies an immediate connection between these two nodes in the graph. Specifically in reddit this implies that the user 'rotoreuters' directly interacts with the 'askreddit' subreddit, perhaps through posts, comments, or other forms of engagement.

I also included lines of code that convert the output to help visualize the graph. For that I used *"graph.dot"* and this file an be used for further analysis in the future. The section that starts with "digraph G {" is the actual representation of the graph structure using the DOT language. DOT is a simple text-based language for describing graphs.

Overall, this graph represents relationships or connections between various usernames. Each arrow represents a directed relationship from one username to another.

Sources:

Dataset: Reddit User and Subreddit Embeddings

S. Kumar, X. Zhang, J. Leskovec. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. ACM SIGKDD 2019.

S. Kumar, W.L. Hamilton, J. Leskovec, D. Jurafsky. Community Interaction and Conflict on the Web. World Wide Web Conference 2018.

The Rust Programming Language, 2023. The Rust Standard Library. https://doc.rust-lang.org/std/

B. Burns, 2023. CSV Crate for Rust. https://docs.rs/csv/latest/csv/

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, 2009. Introduction to Algorithms, Third Edition. MIT Press.

Steve Klabnik and Carol Nichols, with contributions from the Rust Community, 2023. The Rust Programming Language.

E. R. Gansner and S. C. North., 2023. An Open Graph Visualization System and its Applications to Software Engineering. Software - Practice and Experience, 30(11), 1203–1233.