Collin Hoffman

Dr. Shao

ILS 29000

April 21, 2022

PURR Project Spring 2022 Semester Report

My work during the Spring 2022 semester mainly focused on gaining familiarity working with large datasets, natural language processing techniques and toolkits, and data visualization modules such as Pyvis, Networkx, and Streamlit. Specifically, I was tasked with cleaning and visualizing data obtained from the Purdue University Research Repository (PURR), specifically data concerning specific tags and abstracts that were assigned to datasets uploaded to PURR.

As a prelude to this, I was tasked with applying preprocessing techniques to the data, such as tokenization, lowercasing, removing stop words, lemmatization, and finally rejoining the data to its original form. The data was originally contained as a Pandas DataFrame, which I had never used before, so I learned a lot with how to work with DataFrames and some of the associated methods that can be used. Some improvements had to be made to the preprocessing methods, especially for the proper lemmatization of words.

An odd error occurs where hyphenated words, such as “two-phase”, are improperly split up when adding underscores during the bag of words creation. I conducted some debugging and determined that the vectorizer and matrix calculations functions likely caused these improper splits. Tested conducted with sample cases returned the expected output, so it is likely that the error occurs during vectorization and matrix multiplication.

Next, to get comfortable with creating Streamlit networks and data visualization, I created a small demonstration network of my friend group. This helped me gain familiarity with the process of creating data nodes and edges before conducting the PURR data visualization. First, I created a Bag of Words by doing matrix multiplication, which produced a table that counts how many times a specific tag is used in the same dataset as another tag. Then, by looking at example code of a Game of Thrones network, I was able to recreate for loops to create the nodes and then add edges between nodes.

However, generating all possible nodes is not conducive for research use, especially considering the number of edges between nodes. Originally, a node edge was always generated, even if two tags had no relation. This was a significant task for my computer, so I implemented a function to determine the 300 most frequently occurring tags, created the nodes relating to those tags, and then created node edges that related two tags together. This created a network of 300 nodes, however, the visual would not load if nodes and edges were created at the same time. Thus, an HTML file was created to store the network for quicker loading when running the Streamlit app.

In this version, the node labels are far too small to read, which makes the network difficult to use. However, node labels can be changed using streamlit.markdown and adding in a font-size label using CSS, although implementation dependent on node size and frequency may be difficult.

Finally, a repository on Github was created, which includes this report and all relevant Python and CSV files. Throughout the semester, I’ve learned a lot about how to use data visualization software and natural language processing techniques, of which my work will be documented on the Github repository.