NewsMaker's Network

Technical Report

Glen Colletti | Shubhkirti Prasad | Pete Stewart

Application URL:

Our application is hosted on pthonanywhere.com and can be accessed at the link below:
URL: http://peeves.pythonanywhere.com/network

Full GitHub URL:

The application source code is hosted on github.iu.edu at the below URL. It includes the entire code repo which we maintained to track and implement all the changes throughout the project.
URL: https://github.iu.edu/shubpras/Fall-2023--NLP-Group-3.git

Project Summary

The NewsMaker's Network provides a network of co-mentions of prominent people mentioned in the news articles taken from Kaggle. The dataset comprises more than 9,000 BBC articles with topics ranging from business to sports.

The web application aims to enrich the user with information about average prevailing sentiment of the selected personality along with the top personalities he/she is co-mentioned with. The thickness of the traces represent the number of co-mentions, depicting the strength of connection between the listed people. The user can also see the Top 10 other entities the selected personality is related to.

Kaggle URL : https://www.kaggle.com/datasets/hadasu92/cnn-articles/

Project Objectives and Usefulness:

Through this project we aimed to gauge the 'Public perception' value of the selected entity. News Articles are strongly representative of how a person is perceived and discussed in the public sphere, often reflecting a combination of factual reporting, public opinion, and media framing.

Through this project, we have achieved the following:
- Gauged the Average sentiment value of the selected personality mentioned in the news articles.

- Developed a co-mention network so that the user can analyze the Top 30 people the selected personality is related to.
- The thickness of the traces in the network graph represent the number of co-mentions.
- Each node represents a personality in the graph, with the color of the node depicting the sentiment value between the people.
- The app also displays the other Top 10 entities that the selected person is associated with.

Technical Description:

Data:

The dataset is hosted on Kaggle. It mainly comprises about 10,000 news articles from the year 2013 to 2020, published by CNN News. The dataset includes various columns with the information about the author, date of publication, category of news, the URL source, small description about the article and the full text of the articles.

Process:

1. We cleaned the data and did an initial EDA with text preprocessing, stop word removal, text tokenization and lemmatization, using NLTK and other basic python libraries.
2. Since our main aim was Named Entity Recognition, we decided on using multiple trained models available to check which model suited our data the best.
3. As per the course we initially used Spacy's small model implementation, then went on to using Google's BERT and RoBERTA models for NER.
4. After much exploration and comparison we found that Spacy's Transformer model (Spacy-TRF) model worked the best for our dataset, using which we extracted the list of named entities from each of the articles.
5. We then cleaned the names and extracted their sentiment value from the articles.
6. After researching visualizations we implemented a complex network graph to create and store our association data and display it using network X, matplotlib and plotly.
7. We also used a similar technique to create a list of associated entities with a given entity as a network seed.
8. We then worked on creating a simple webpage interface using Flask and html to implement functionalities that would enable the user to search for a named entity within our dataset, and plot the network graph from the data.
9. All the final cleaned CSVs are available on the given GitHub, along with our code in the jupyter notebooks.

Deployment Platform:

Python Anywhere was our platform of choice which supports built and deployable web applications.

Tools Used:

Front End: Flask/HTML/Chart-JS

Back End: Python

Libraries : NumPy, Pandas, Matplotlib, networkX, Spacy (Transformer), Spacy Textblob, BERT, RoBERTA

Tools: Visual Studio Code, Jupyter Notebook, MS Excel

User Functionalities:

1. The user gets to select the option of creating the network graph with the 'Create Network tab'

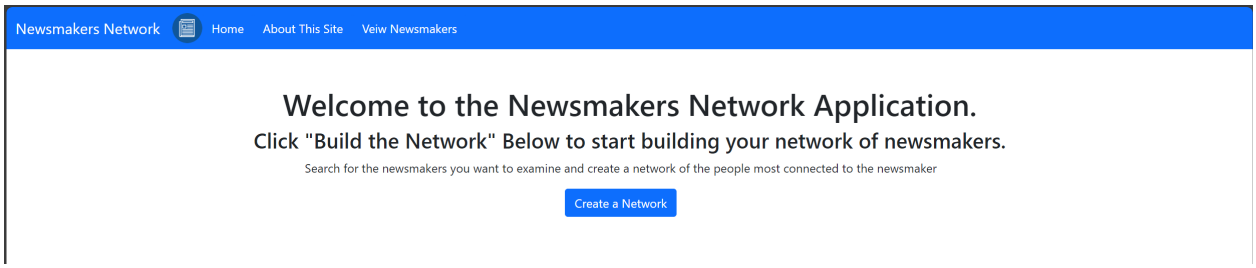

Figure 1 : The home page

2. The user then enters the name of the person and clicks on the 'Submit' button.

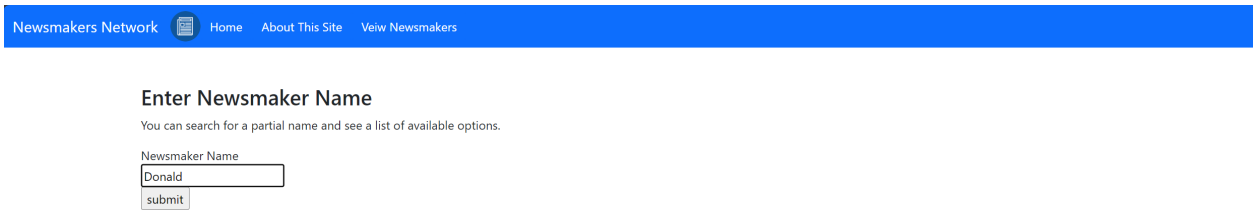

Figure 2: Entering the name of the person.

3. The entered name is then best matched to the names of the entities we have in the database. Here the user can select the name of the entity. For example, we chose ‘Donald Trump’ for the list.

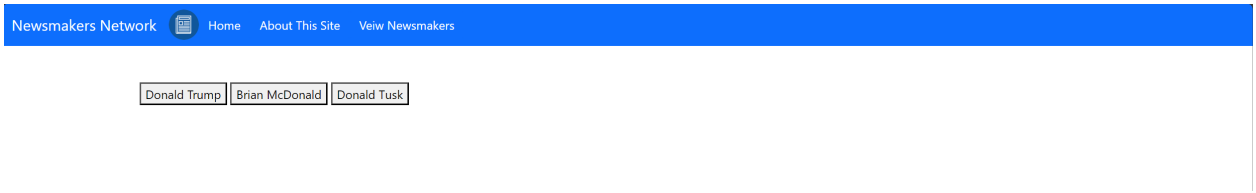

Figure 3: Entity selection

4. The app then creates a network of the Top-30 entities (people) our selected entity is co-mentioned with, with the color of nodes depicting the average sentiment of the entity, and the traces representing the number of co-mentions.

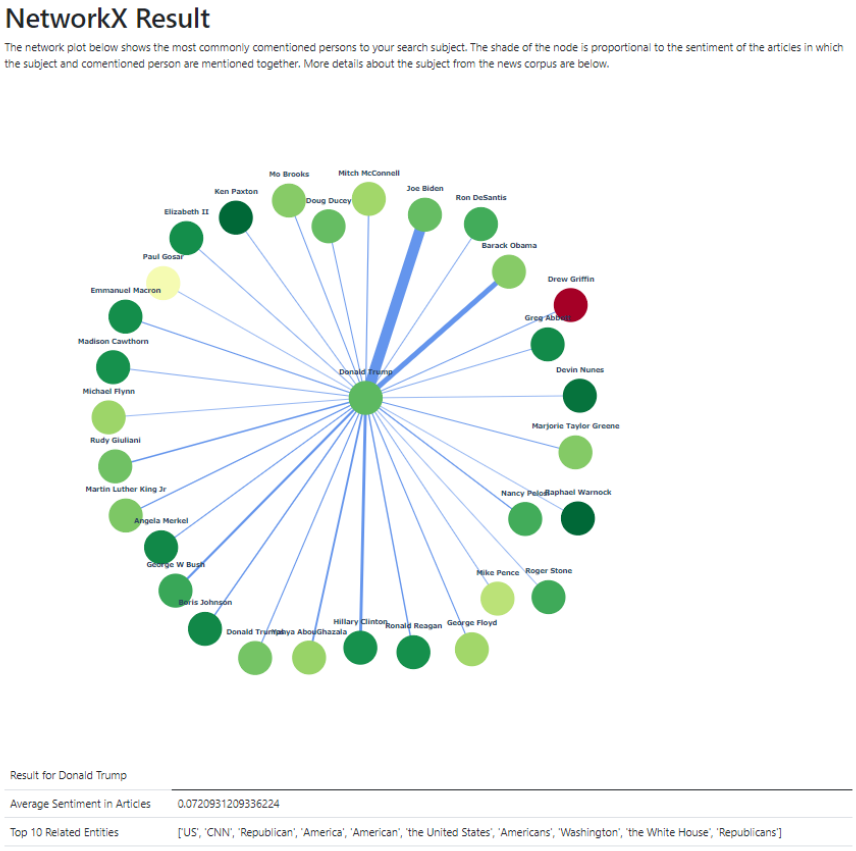

Figure 4: Network X Graph of Donald Trump

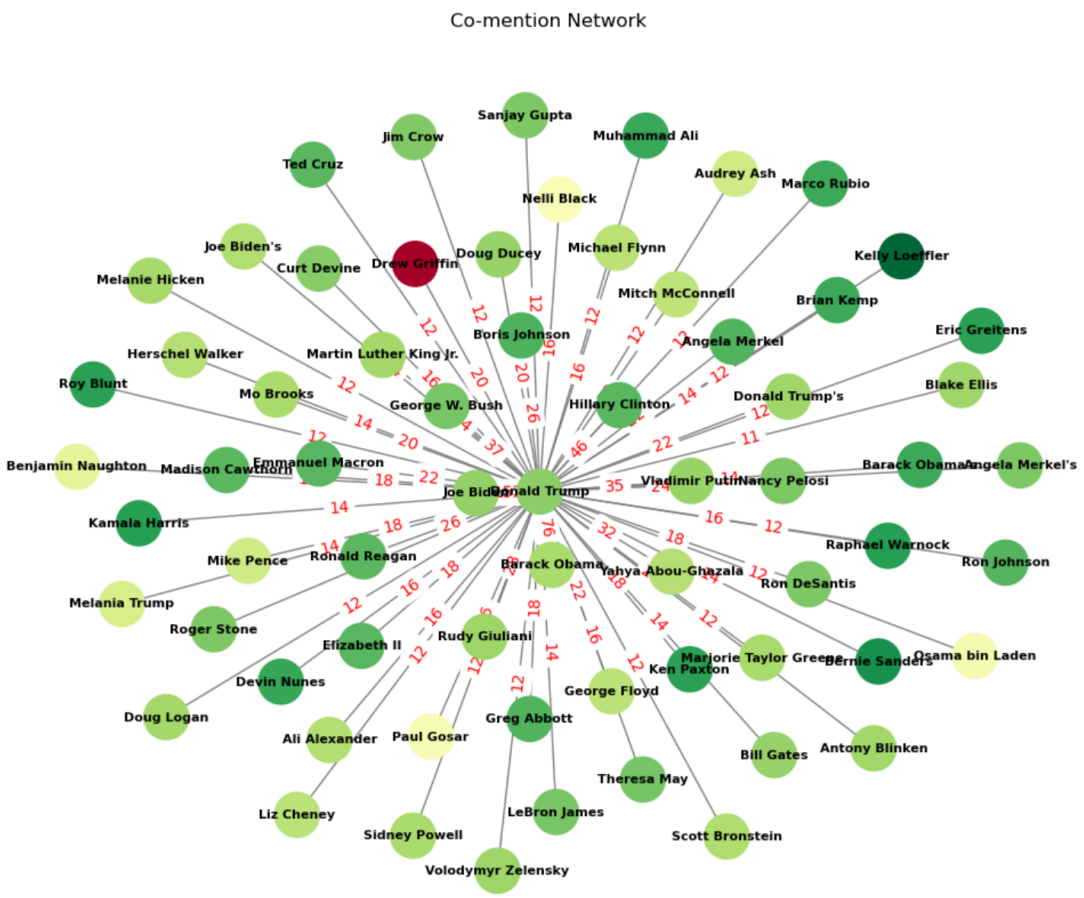

Figure 5: One of our initial implementations of Network X Graph of Donald Trump

Observation:

- Donald Trump is frequently co-mentioned with Joe Biden and Barack Obama, with a slight positive sentiment as represented by the thicker traces and the light green color.
- The dark green nodes represent a highly positive sentiment value, with yellow representing the most neutral sentiment and red (Drew Griffin) representing a negative sentiment.
- The average sentiment value of Donald Trump is slightly positive as shown by the numeric value of about (0.072) on a scale of -1 to 1.
- ‘US’, ’CNN’, ’Republican’ are amongst the top 10 entities most associated with Donald Trump from the given News articles.

5. The app also has a link to our github, as well a short quirky information paragraph about each of the developers.

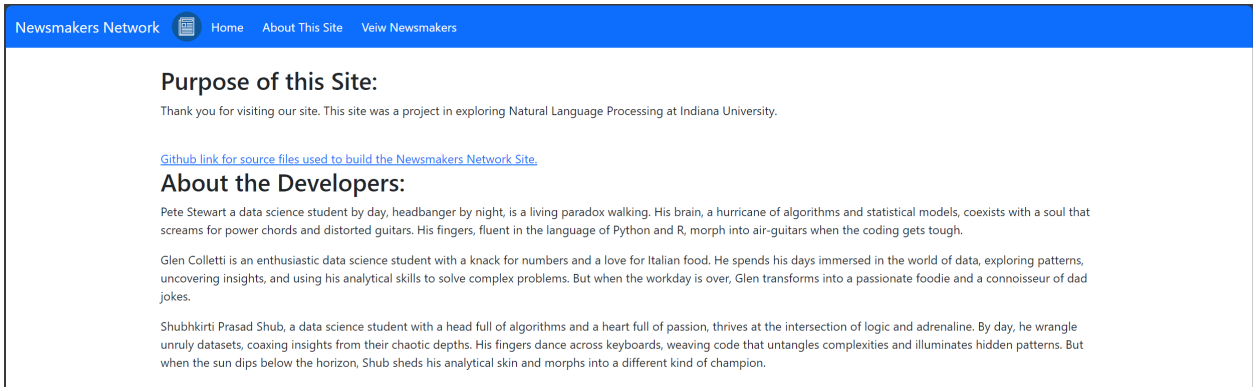

Figure 6 : About the developers

Issues Faced:

1. Getting a relevant dataset for the implementation of our New's Network. All 3 of us spent a lot of time initially to find the apt dataset for our network.
2. Finding the best implementation for NER, which works well on our dataset. We went through a lot of models to find the perfect fit.
3. Final implementation of the network. It took alot of research and debugging to get the final visualization working with all the data variables in our dataset.
4. The integration with flask to correctly display and connect with our database was also troublesome.

- The tasks were divided between the three of us and we worked on them separately at first, to understand the issues we were facing. That helped us solve most of our problems when we got the final code together.
- Meeting at regular intervals to gauge the status of the problems also helped in keeping the project on track.
- In the end, we were able to implement all the functionalities that we planned, and created an app that could actually help the user during the upcoming elections!

Teamwork:

1. Glen Colletti

- Cleaned the data we got from Google's Bert implementation of NER.
- Implemented the search list functionality in Flask for the web application.
- Debugged Flask front end.
- Presented during the demo video.

2. Shubhkirti Prasad

- Responsible for the final implementation, from cleaning the data to getting the final CSVs of all the entities for the frontend.
- Implemented the function for final visualization of the nodes with the sentiment values.

3. Pete Stewart

- Was responsible for using BERT's model on the dataset.
- Created the co-mention network which was further used by Shubh in the final implementation.
- Implemented the Flask frontend along with Glen with all the final functionalities, and deployed the app on PythonAnywhere