GEOG582 | FA24
Chaska Kentish
December 14, 2024

Project Background

American politics–especially when it comes to presidential elections–have become extremely polarizing over time, and 2024 was no exception. As politics becomes more polarizing, it is important to adequately prepare for changes in federal power including the president and their cabinet, Congress, and the Supreme Court. For many Americans, these controls of power can significantly impact different rights ranging from citizenship to abortions to general civil rights. It can also affect different federal programs and states in regards to powers and financial support. Thus, it is important to effectively determine the expectations of elected officials and how different states lean for these expectations. By efficiently forecasting expected outcomes, candidates can better address their potential voters (as seen with Obama's campaign in 2012) and other government officials can better prepare their policy enactment efforts, spending allocations, among other things in preparation for potential limitations or losses with these changes.

The 2024 presidential election was a very non-traditional process compared to prior elections. There was a change in Democratic candidates in July 2024 from Joe Biden to Kamala Harris, and there were various controversial discourses occurring within and across the Democratic and Republican parties regarding candidates and efforts. Thus, there may be greater noise and uncertainty in available social media and political data leading up to the actual election in response to these issues. Inter-party criticism of candidates and high discourse domestically and abroad is highly reflected in Americans' opinions on social media.

Problem Statement

This project seeks to develop a natural language processing (NLP) model to accurately classify and label tweets as supportive of left or right political ideology and use these results to determine the effectiveness of tweets as a predictor of voting outcomes in the 2024 election. This application will be conducted on a state level, with winners being selected based on the majority vote (based on support of Democratic or Republican candidates) as per typical electoral college procedure. An interactive mapping tool will compare the X (or Twitter) based election results to the actual results of the 2024 presidential election, with key information for the percentage of tweets favoring which candidate versus the actual vote breakdown as well as the top 5 hashtags per state.

Program Design

Three datasets were utilized for this project, and all three were Tweets related to U.S. presidential elections. The dataset information can be seen in *Table 1*.

Name	File Path	Description	URL
USC X 24 US Election Twitter/X Dataset	usc-x-24-us-election-main/	This dataset is broken up into separate parts with 20 CSV files per part containing 50,000 tweets related to the 2024 U.S. election. This project utilizes tweets from May through July. *Note: The creators of this data archive are continuing to update it, and as of now it now contains data up through September.*	https://github.com/sinking8/usc-x-24-us-election/tree/main
Knowledge Enhance Masked Language Model for Stance Detection	training_data/manual_train.csv	This dataset contains 1,250 labeled tweets per candidate indicating if they are in favor or opposition of Joe Biden and Donald Trump respectively.	https://portals.mdi.georgetown.edu/public/stance-detection-KE-MLM
US Election 2020 Tweets	training_data/hashtag_donaldtrump.csv	This dataset contains around 1.7 million tweets related to Joe	https://www.kaggle.com/datasets/manchunhui/us-electi

	training_data/hashtag_joebiden.csv	Biden and Donald Trump for the 2020 U.S. election. It is split into a Joe Biden and Donald Trump CSV.	on-2020-tweets
States Shapefile (500K)	state_shape/	Shape file for state geometries.	https://www.census.gov/geographies/mapping-files/time-series/geo/carto-boundary-file.html
actual_results.json	final_data/actual_results.json	The actual 2024 U.S. presidential election outcomes and vote breakdowns for Democrats and Republicans.	https://www.reuters.com/graphics/USA-ELECTION/RESULTS/zjpqnemxwvx/

Table 1. Datasets used for this project. They all contain Twitter/X data related to Presidential elections. The total size of all the data was over 33 GB.

The last two datasets were utilized for training data in addition to a set of 1,439 manually labeled tweets from the USC x 24 US Election dataset. To prepare the US Election 2020 Tweets dataset, a composite sentiment analysis was performed. Three separate sentiment models (VADER, TextBlob, SentiWordNet) were employed on both the Biden and Trump tweet sets to determine a composite sentiment score (positive, negative, or neutral). If a model labeled a tweet as positive (score > 0) it was mapped to a value of 1, if it was negative (score < 0) it was mapped to a value of -1, and if it was neutral (score = 0) it was mapped to a value of 0. Then, the composite sentiment score was: $max(VADER\ Score,\ TextBlob\ Score, SentiWordNet\ Score)$ and mapped to one of three labels:: (1) 0 = Pro-Biden (or Pro-Democrat), (2) 1 = Pro-Trump (or Pro-Republican), and (3) 2 = Neutral and/or Irrelevant tweets respectively. This provided an efficient baseline for the training set and model development. The workflow can be seen in *Figure 1*.

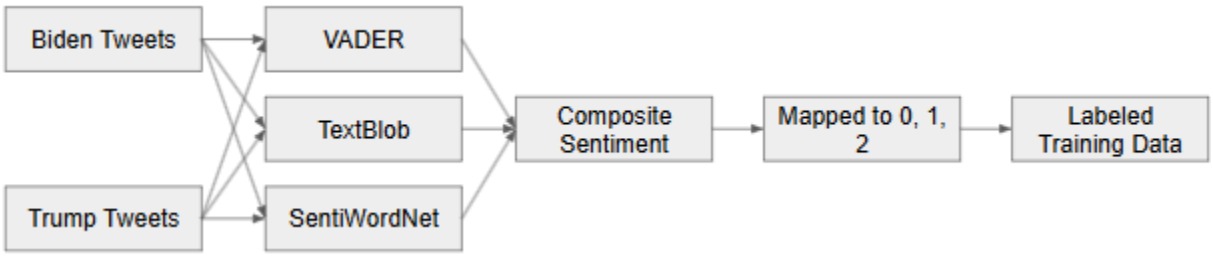

Figure 1. Architecture of the composite sentiment analysis model. Biden and Trump tweets were labeled by all three sentiment models (VADER, TextBlob, SentiWordNet) to get a composite score and mapped for labels.

For model development, conventional machine learning algorithms such as Random Forest, SVM, and Logistic Regression were initially explored with limited success. Because of this, the main efforts for model development focused on neural networks, since they proved to have more potential in the goal of this project. A simple neural network ended up being the final model produced for this project. It includes three types of layers: (1) Linear layers to transform and map data, (2) ReLU layers to introduce non-linearity (due to the non-linear fashion of the data), and (3) Dropout layers to drop some portion of neurons to prevent overfitting. The final architecture of this neural network can be seen in *Figure 2*.

```
class TextClassifier(nn.Module):
    def __init__(self, input_size):
        super().__init__()
        self.model = nn.Sequential(
```

```
            nn.Linear(input_size, 256),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(256, 128),
            nn.ReLU(),
            nn.Dropout(0.2),
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Dropout(0.1),
            nn.Linear(64, 3)
        )

    def forward(self, x):
        return self.model(x)
```

Figure 2. Overview of the neural network architecture.

The above neural network achieved an 83% testing accuracy rate, which while not perfect, is a substantially successful model (as random guessing of labels would be 33% accuracy). The confusion matrix for the model can be seen in *Figure 3*. Most tweets were correctly labeled.

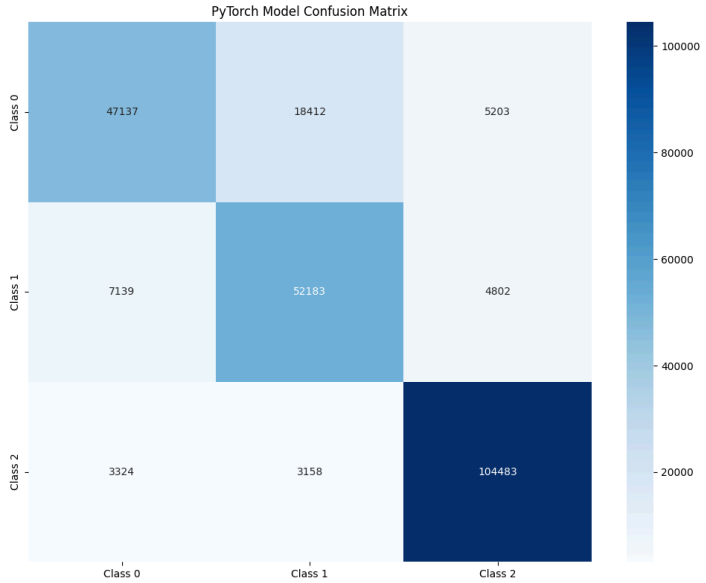

Figure 3. Confusion matrix for the neural network model used for classification. The darker blue squares down the diagonal indicate correctly labeled tweets based on the composite sentiment score. A majority of tweets were properly labeled.

Finally, an interactive map tool was created with two layers: (1) predicted results and (2) actual results for the 2024 election, with each state colored by the party who won as per Electoral College standards. A tooltip was implemented for each state to show the percentage of pro-Democrats and pro-Republicans, with the top five hashtags being displayed. Sample images can be seen in *Figure 4*.

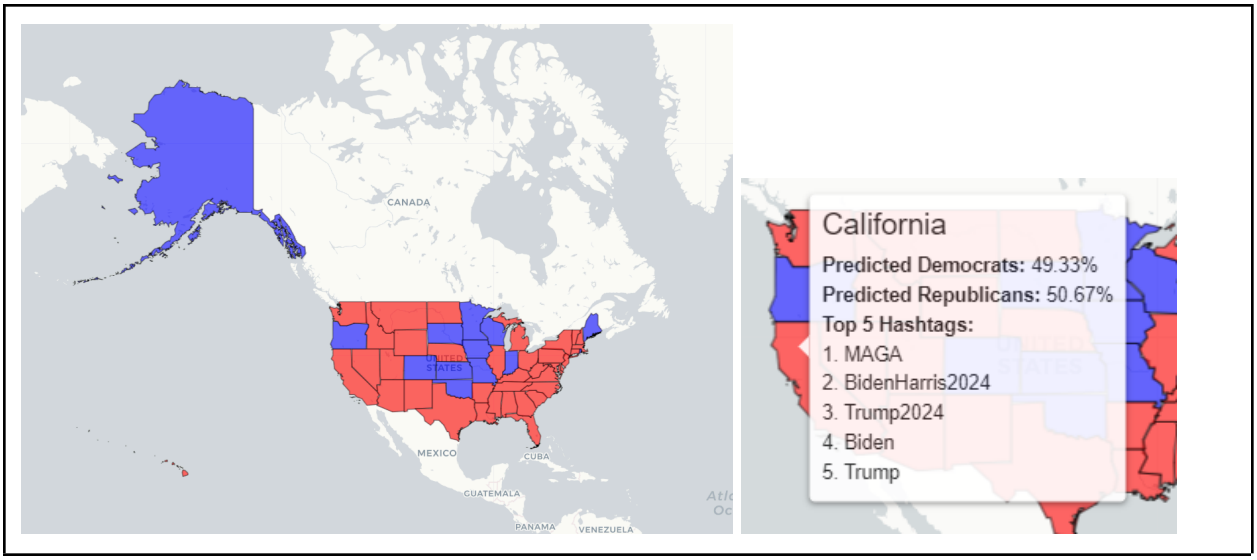

Figure 4. Snapshots of the interactive map tool developed. The output of the tweet predicted outcomes significantly differ from actual election outcomes (Left). A tooltip shows the percent of Democrats and Republicans for majority outcome and the top 5 hashtags was implemented on this map (Right).

Code Implementation

To achieve all of the above, four separate Python files were created to be used in a consecutive order numbered one to four. The information for these files can be found in *Table 2*.

Order	Name	Description
1	1-data-processing.py	Processes the raw USC X 24 US Election Twitter/X Dataset data to extract files, clean the data, and output a single *processed_data/combined_election_data.csv* file for later use.
2	2-training-processing.py	Performs composite sentiment analysis and mapping for the US Election 2020 Tweets data and merges it with Knowledge Enhance Masked Language Model for Stance Detection data and manually labeled data *training_data/training.csv*.
3	3-neural-network.py	Creates, trains, and checks model performance of the neural network classifier. It applies the model to the *combined_election_data.csv* and outputs it with the labels *final_data/labeled_election_data.csv*.
4	4-data-analysis.py	Takes *labeled_election_data.csv* and extracts top 5 hashtags per state, calculates winner of each state, and produces the interactive map and results. The interactive map was outputted to HTML *state_predictions_vs_actual_map.html*.

Table 2. Overview of code implementation and python file relationships.

Major Accomplishments

This project succeeded in developing a strong basis for a NLP classifier model with moderately strong accuracy for U.S. presidential-based political tweets. It also effectively demonstrated why there are major limitations in only using social media for the basis of complex behavioral topics such as politics, due to discrepancies in representation and capturing all determinants of behavior. It also led to the use of more advanced techniques such as composite sentiment analysis and neural network classification. Finally, this process provided a comprehensive and effective pipeline of data collection, processing, and analysis of a large set of data (initially over 33 GB of data).

Limitations

There were three major limitations of this project. Firstly, the data available was limited to the USC repository and Kaggle repository, as the Twitter/X API has become an expensive tool to scrape data. Thus, the tweets used for both training and actual analysis were dependent on the data they collected and/or deemed relevant to the topic of presidential elections. Second, there was a limited amount of computational resources available. The local machine had issues with running some more complicated deep learning based algorithms, so alternative resources such as Kaggle were used. Kaggle provides over 30 hours of access to their GPUs and TPUs weekly, so it was used to run some of the deep learning models. However, even with their resources, preferred models such as BERT were not able to be run due to insufficient resources and time to run. With available resources, using more advanced NLP models such as BERT would be preferred in the future for this type of research. Finally, political behavior and social media are extremely complex topics and mediums to work with. Behavior in general, let alone political behavior, is constantly changing based on real-time events (labeled as success or disaster, unrest, etc.) that can easily change political perception. In addition, social media can contain lots of noise or loud minorities that sway outcomes based on just this type of data and/or how people perceive outlooks within different states.