Predicting the Political Ideological Leanings of U.S. Senators: A Twitter Sentiment Analysis

Aaron Han
Department of Computer Science
Stanford University
than21@stanford.edu

Mary Zhu
Department of Computer Science
Stanford University
maryzhu@stanford.edu

I. Introduction

While political sentiment analysis on Tweets or Reddit comments has been done previously, prior studies generally have focused on classifying in terms of party affiliation as opposed to political ideals. In this project, we plan to classify based on the spectrum described by The Political Compass, a website that uses an individual's responses to a set of 62 propositions to rate their ideology [1]. The site creates a 2-dimensional space where one's ideology can land on based on their social and economic beliefs.

The objective of this project is to utilize natural language processing to classify politically opinionated Tweets from 100 U.S. Senators and 435 U.S. Representatives serving in the current electoral cycle into four distinct categories: libertarian right, libertarian left, authoritarian right, and authoritarian left.

II. Related Works

In developing the model, several related works are used to help understand similar implementations in this field. Bakliwal1 et al. (2013) performs a 3-class sentiment classification analysis on a set of 2,624 Tweets produced during the 2011 Irish General Elections. Tweets considered sarcastic were omitted from the set, and, using a feature set comprised of subjectivity-lexicon-based scores, the model was able to achieve an accuracy of 61.6% [2]. Meanwhile, Caetano, Lima, Santos & Marques-Neto (2018) conduct a sentiment analysis on 4.9 million Tweets during the 2016 U.S. Presidential Election in an effort to measure the degree of homophily among the 18,450 users authoring the Tweets. By analyzing the Follow, Mention, ReTweet, friendship, and multiplex connections, they found that negative, Trump-supporting, and Clinton-supporting users all expressed homophilic tendencies [3]. Finally, Roy (2018) conducts a sentiment analysis on Twitter data during the 2017 Gujarat Legislative Assembly election to predict the chances of winning. He uses an NRC Emotion Lexicon to classify Tweets into eight emotions, and the ParallelDots AI APIs to further distinguish the Tweets between positive, negative and neutral sentiment [4].

Unlike the studies we perused in our literature review, our research differs because our objective is to classify the ideological leanings of specific politicians on two axes. Further, our methodology relies on testing the effects of a range of different classifiers and features on performance.

III. Dataset

The data fed into our model comes from a large dataset of Tweets from various politicians ranging from House Representatives, Senators, Governers, and others [5]. The dataset includes up to roughly 3,200 of the most recent Tweets from each politician as of 2 years ago. This means that the data may encompass the entire Tweet history for some politicians that don't post often (e.g. former Senator Bill Nelson has 601 Tweets in the dataset), while others that post very often (like President Trump) might only have their most recent year's worth of Tweets in the dataset.

Each Congressman was classified in terms of his political leanings based on the traditional spectrum (i.e. liberal vs. conservative economically and socially). The economic leanings were based on a scorecard from Club for Growth, an economically right organization. If someone received a score of 50% or higher on their lifetime score from Club for Growth, they were classified as economically right (and left otherwise). For social leanings, we used the scorecard from American Civil Liberty Union, a socially left special interest group. If someone received a score of 50% or higher on this scorecard, they were classified as socially left (and right otherwise).

There were no scorecards available for socially libertarian vs. authoritarian classifications. This is most likely due to the difficulty of clearly distinguishing socially authoritarian-leaning and libertarian-leaning politicians as most of them had diverging opinions on the role of the government for different social issues such as abortion, immigration, gun control, LGBT rights, and others. The exact economic left/right, social libertarian/authoritarian spectrum from the Political Compass was forgone as a result.

The Tweet dataset included two JSON files: one for the profiles of the Twitter users of the Tweets, and one for the Tweets themselves. The users dataset contained information regarding the users' profiles, including their Twitter screen names and actual names, while the Tweet dataset included details about each Tweet, such as its text, number of ReTweets, timestamp, etc. The Politicians dataset included the names of each Senator and Representative, as well

as their political leanings across the four distinct political leanings categories.

In an effort to bootstrap these separate datasets together, we preprocessed our data by linking all the information together in a Python ordered dictionary. For this data structure, the keys were each Congressman's unique Twitter screen name, while each value was a vector containing the word vector produced by our Bag-of-Words implementation, the real name of the Senator corresponding to the screen name, and the Congressman's political leaning information. Regarding each Congressman's political leaning, we decided to represent left/authoritarian ideology as -1 and right/libertarian ideology as 1 in both economic and social categories.

Senators and Representatives that were not included in the Tweets dataset were excluded from the model.

IV. Model Implementation

The input of our model is comprised of features based on the corpus of a Congressman's Tweet history, while the output is the classification of the Congressman. For example, former Sen. Rand Paul is classified as libertarian-right given his pro-market stances and support of a smaller government. On the other hand, Sen. Dianne Feinstein is classified as authoritarian-left based on her support for substance and firearm regulations as well as her track record of voting for bailouts in 2008 and voting against cuts to taxes and Medicare.

A. Baseline

We implemented the Bag-of-Words method to extract features from the Politicians' Tweets. We concatenated the Tweets of each Congressman together to form a String per politician. The Strings were tokenized to develop a corpus containing the words of all Tweets from all Congressmen, excluding punctuation and stop words as classified by the Python NLTK library. The corpus for each Congressman was normalized into percentages of each word's appearance frequencies.

To serve as a simple baseline, we implemented linear regression from the "word vectors" of each politician as described above. The linear regressions utilized stochastic gradient descent to update the prediction weights after observing features from each Congressman. We developed two separate linear regressions for the social leanings and the economic leanings to keep the baseline relatively simple. However, this approach fails to take into account that economic and social leanings aren't necessarily independent.

B. Main Approach

1) Features: For the implementation of our main approach, we upgraded our simple Bag-of-Words model to a Bag-of-N-Grams Model as well as a TF-IDF Model.

Our motivation for introducing a Bag-of-N-Grams Model was due to the traditional Bag-of-Words model failing to account for the order of words in text sequences. With a Bag-of-N-Grams model, we are now able to account for phrases and contiguous tokens. We experimented with n-grams of different sizes, noting their social, economic, and total accuracy with each test.

Similarly, our motivation for introducing a Term Frequency-Inverse Document Frequency (TF-IDF) Model was due to the potential issues resulting from using a Bag-of-Words model on a large corpus. When solely using the Bag-of-Words model, absolute term frequencies dictate the resulting feature vectors. Consequently, some words that appear across the Tweets of all politicians frequently may dim out the presence of other words. The TF-IDF Model resolves this problem by utilizing a scaling and normalizing factor:

$$tfidf(w,D) = tf(w,D) * idf(w,D) = tf(w,D) * log(\frac{C}{df(w)}) \quad (1)$$

where $tfidf(w,D)$ is the TF-IDF score for word w in document D (or in this case, Tweets), $tf(w,D)$ is the word frequency of w in D as obtained from the Bag-of-Words model, and $idf(w,D)$ is the inverse document frequency for w, computed as the log transform of the total number of documents in the corpus C divided by the document frequency of w [7].

2) Classifiers: We implemented regression and Naive Bayes classifiers for both single and multiclass classifications using Python's scikit library. In theory, using multiclass classification should be the superior classifier since social and economic leanings are not independent, and single-class classification methods that predict social and economic leanings separately would not be able to utilize the dependence. However, given that only 3.9% of Congress members showed difference in their social and economic leanings, multiclass classifications may actually perform worse than single-class classifications.

As for Naive Bayes vs. regression, Naive Bayes would assume that the features are conditionally independent, which is certainly not the case in our problem. This would lead to higher bias, which would lead to higher variance but also higher accuracy if the dataset follows the bias.

3) Limitations: Due to computing limitations, we were unable to run larger feature vectors that were created by using n-grams with $n \geq 4$ as well as spanning ranges (e.g. range(1,3)).

V. Evaluation Metric

For single-class classifications, accuracy is based on the total amount of points available. Each classification is worth 1 point where an entirely correct classification earns 1 point, a half-correct (e.g. predicted left-authoritarian when right-authoritarian in reality) classification earns 0.5 points, and an incorrect classification earns 0 points. No such method was necessary for multiclass predictions.

VI. Results & Analysis

A. Baseline

Using our evaluation metric, the model is 58.8% accurate. The relatively low success rate could be due to the aforementioned inability of the model to combine social and economic leanings as correlated variables. In fact, while the model was able to correctly predict a Senator's social leanings 70.6% of the time, it was only able to predict economic leanings with 47.0% accuracy. A possible reason behind this disparity could be that while a Tweet containing social keywords is a clearer indicator of a Senator's leanings than economic keywords. For instance, a significant proportion of Tweets that contained the word "abortion" was actually from authoritarian Senators. On the other hand, a word like "tax" could appear in either economically left or right contexts, obscuring the weight for the use of the word in predicting economic leanings.

When using the traditional political scale (liberal vs. conservative for both economic and social leanings) as opposed to Political Compass' scale, the model was able to predict with 64.7% accuracy. Accuracy for social predictions were higher as well with 82.4%, while economic predictions remained roughly the same at 47.1%. As stated in the Dataset section, determining authoritarian vs. libertarian leanings socially is difficult with the current political atmosphere. As such, we were able to see that using a more traditional political scale led to higher accuracy in predictions.

B. Regression: n-grams

We first used n-grams as new features on a dataset limited to Senators to test our base single-class linear classification model. Results are as follows:

TABLE I
Results of baseline model and preliminary implementation of main approach.

N-Gram Range	Social Accuracy (%)	Economic Accuracy (%)	Total Accuracy (%)
Baseline	82.4	47.1	64.7
(1, 4)	70.6	47.1	58.9
(2, 2)	47.1	47.1	47.1
(3, 3)	82.4	47.1	64.7
(4, 4)	94.1	47.1	47.1
(5, 5)	94.1	52.9	73.5
(6, 6)	94.1	70.6	82.4
(7, 7)	94.1	76.4	85.3

We could see that while social accuracy increased as n increased while $n \leq 4$, economic accuracy stayed constant. This is most likely due to the smaller n-grams still failing to account for economic sentiment as stated earlier. Once $n \geq 5$, we could see that economic accuracy started increasing and social accuracy stayed constant. This is most likely due to the fact that n-grams with $n \geq 5$ begins to capture economic sentiment while not adding additional value in capturing social sentiment.

We then extended the dataset to include members of the House of Representatives:

TABLE II
Results of baseline model and preliminary implementation of main approach, extended to all Congressmen.

N-Gram Range	Social Accuracy (%)	Economic Accuracy (%)	Total Accuracy (%)
(1, 1)	50.8	65.6	58.2
(2, 2)	47.1	47.1	47.1
(3, 3)	82.4	47.1	64.7

We could see that performance of the single-class n-grams model drop with the House added to the dataset. While this may be an indicator of overfitting in using a smaller dataset previously, it is unclear whether the model overfitted as we could not run predictions for the larger dataset using the same β vector given that more n-grams could have been introduced in the corpus by the edition of new data. The performance increase with larger n-grams was also seen with the larger dataset.

The following are the results from multiclass logistic regression using n-grams features:

TABLE III
Results of multiclass logistic regression with N-grams features.

N-Gram Range	Accuracy (%)
(1, 1)	93.4
(2, 2)	95.1
(3, 3)	82.0

We could see that multiclass classification outperformed single-class classification for regression. This is most likely due to the high percentage of Congress members being aligned "left/left" or "right/right." Another reason for the higher performance can be that the features used can utilize the correlation between social and political leanings in multiclass classification while single-class cannot.

As for performance with increasing n-gram sizes, we could see that the performance increased for $n = 2$ while it dropped for $n = 3$. Because we were unable to test with larger n-grams than $n = 3$, we are not able to deduce that increasing n-gram sizes past 3 would also decrease the accuracy. That being said, a possible explanation for the performance drop could be that smaller n-grams already capture enough information regarding a Congress member's overall political leanings while increasing the size cloud the features to perform worse.

C. Naive Bayes: n-grams

We tested Naive Bayes to check for its accuracy as well as bias in our overall data. The following is using Naive Bayes for classifying social and economic leanings separately:

We could see that Naive Bayes outperformed regression in single class classifications using n-grams as features. This is most likely due to the bias in the data as previously

TABLE IV
RESULTS OF ECON-SOCIAL SEPARATE NAIVE-BAYES WITH N-GRAMS FEATURES.

N-Gram Range	Social Accuracy (%)	Economic Accuracy (%)	Total Accuracy (%)
(1, 1)	90.2	85.3	87.7
(2, 2)	88.5	85.3	86.9
(3, 3)	91.8	91.8	91.8

described, which can be found in both the training set and the test set.

Here are the results from multiclass Naive Bayes classification:

TABLE V
RESULTS OF MULTICLASS LOGISTIC REGRESSION WITH N-GRAMS FEATURES.

N-Gram Range	Accuracy (%)
(1, 1)	85.3
(2, 2)	85.3
(3, 3)	90.2

We could see that multiclass and separate Naive Bayes classification showed similar behavior with different n-gram sizes with similar prediction accuracy increasing as the n-gram size increased. As for the small drop in accuracy in the multiclass model, this is most likely due to the dataset not including a significant portion of right/left and left/right Congress members. The small size of these data points create difficulties in training a model to predict them correctly but also do not significantly weight down the accuracy of the model.

D. Single class classification: TF-IDF

The following are the results from linear regression and econ-social separate Naive-Bayes using TF-IDF features:

TABLE VI
PREDICTION RESULTS USING TF-IDF FEATURES.

Classifier	Social Accuracy	Economic Accuracy	Total Accuracy	Multi-class
Regression	73.8%	50.8%	62.3%	93.4%
Naive-Bayes	85.3%	90.2%	87.7%	85.3%

In Table 6, we see that, consistent with the results presented above, Naive Bayes still outperforms linear regression even when using TF-IDF features when separating social and economic predictions. We then compare the performance of using TF-IDF features with n-grams features in Table 4. We observe that although higher accuracies are achieved with TF-IDF features over n-grams features of smaller ranges (such $n = 1$ and $n = 2$), setting $n = 3$ generates the best results. A possible explanation for this is that the TF-IDF strategy of weighing down the most frequent terms while scaling up the rarer ones is not the most appropriate for Tweets. Since Tweets have limits of 140 characters, the authors of the Tweets probably made sure that all words appearing in their Tweets are essential to getting their concise messages across; thus, the most frequent words are unlikely to be strictly the least meaningful. Further, as mentioned above, trigrams may simply be the best option for examining typical collocations in Tweets.

However, when comparing the econ-social separate predictions with the multiclass results using TF-IDF features, we see that multiclass logistic regression achieves significantly higher performance than single-class regression. This is consistent with the results displayed in Table 3, and the analysis following it.

VII. FUTURE WORK

Next steps would include further testing on different n-gram ranges with the larger dataset. While using a smaller dataset of just Senators, we were able to see that larger n-grams led to more accurate predictions, at least for the separate regression model. Testing the same n-gram ranges with the larger dataset would tell us whether the regression model was a good prediction model or simply overfitted.

We could also look to use unsupervised learning methods to forego the data-labeling process. Two-dimensional k-means or long short-term memory (LSTM) recurrent neural networks can be utilized for clustering Congress members into their ideological categories. However, this may only be appropriate once a more significant portion of the dataset can be assigned to the four possible clusters instead of being mostly in two (right/right, left/left). This will require expanding the dataset beyond Congress members.

REFERENCES

[1] The Political Compass (2020). Retrieved from https://www.politicalcompass.org/usstates?ak=on&az=on&il=on&ny=on.

[2] Bakliwal, Akshat, Foster, Jennifer , van der Puil, Jennifer, O'Brien, Ron, Tounsi, Lamia and Hughes, Mark (2013) Sentiment analysis of political tweets: towards an accurate classifier. In: NAACL Workshop on Language Analysis in Social Media, 13 June 2013, Atlanta, GA.

[3] Caetano, J., Lima, H., Santos, M. et al. Using sentiment analysis to define twitter political users' classes and their homophily during the 2016 American presidential election. J Internet Serv Appl 9, 18 (2018). https://doi.org/10.1186/s13174-018-0089-0

[4] Roy, Sandip. (2018). Analyzing Political Sentiment using Twitter Data.

[5] Reddit (2018). Over one million tweets collected from US Politicians (President, Congress and Governors). Retrieved from https://www.reddit.com/r/datasets/comments/6fniik.

[6] Bhand, M., Robinson, D., & Sathi, C. (2009). Text Classifiers for Political Ideologies. Retrieved from https://nlp.stanford.edu/courses/cs224n/2009/fp/7.pdf.

[7] Sarkar, D. (2018). Traditional Methods for Text Data. Retrieved from https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41