Eshil Patel
NLP Project Write-up
Professor Stone

Predicting the 2018 Election Outcome using Machine Learning

Predicting election results is a huge business in today's age, especially with respect to US congressional races. National parties pour millions of dollars into the races, hoping to win a majority for the next congressional cycle. Websites like 538 have dedicated sections to political predictions. Of particular interest in recent times is the use of twitter in politics, and what, if any, insights one can gain by analyzing the tweets of the senators. The problem that this project aims to explore is the following: using only the tweets of the current senators in the US senate, can we recreate the margin of victory in the 2018 House races (overall in each state, not individual races)?

RELATED WORK:

The literature on the use of NLP in the political domain is vast. Of specific note are two papers that establish methods of using a RNN to classify political text (speeches, tweets) as conservative or liberal. The first is a paper by a group of researchers at UMD [1]. Their paper established that other representations other than bag of words can be used to model political text and more accurately capture the semantics and context. The authors had the sentences annotated in a tree structure. Each individual token was converted to a word embedding. Additionally, the phrases at higher levels were also converted to word embeddings and used in the model. This paper's success motivated the use of an RNN for this task, as the RNN would be ideal to remember previous words/phrases in a tweet.

The second paper of note is a group at Stanford [2]. The authors also used an RNN like the group at UMD, but they applied the RNN specifically to tweets. Additionally, their representation of tweets was simply a word embedding for each token in the tweet (using GloVe), instead of a word embedding at the token and phrase level. The authors also experimented with using LSTM and GRU in the RNN to see how the accuracy was affected. This paper in particular gave the motivation of the tweet representation that was used in this project.

In terms of using NLP/ML methods in election predictions, the most common approach explored in recent years was using sentiment analysis. This involved collecting tweets of users around the election and determining if they described the candidate positively and negatively. This method showed promising results, with one paper showing 66.7% accuracy in predicting the winner of a state [6] using SentiWordNet. Another paper applied the same concept (but different implementation) to the 2016 election in Ohio, Florida, and North Carolina. This yielded good predictive results (who won), although the margins predicted left much to be desired [8]. This paper used a Naïve Bayes classifier to make the final decision on whether the tweet was positive or negative for a candidate. Other papers also used sentiment analysis, but added in tweet volume into the decision making process (how many people were tweeting about a particular candidate).

HYPOTHESIS:

The underlying hypothesis is that states that vote republican will have senators that tweet more conservative things than states that vote democrat (and vice versa). Sentiment analysis, which is used heavily in election prediction in the literature, would not be appropriate for this context for two reasons. One reason is that the papers mentioned above used average twitter users who were talking about specific politicians (oftentimes just two candidates). Because the tweets collected in this project were US senators (particularly US senator tweets after the 2018 election), it is expected that the tweets do not talk about specific candidates very much (especially considering that the tweets were collected in April). The second reason is that capturing sentiment in these tweets would not give potentially useful information on its own (without knowing the context of who the positive/negative sentiment is directed towards). Instead, the stances they have on various legislation/political issues would constitute the majority of the content of the tweets.

DATA

There are a myriad of data sources that could be used for a project in this domain. In previous work in this domain, the favorite data sources for training were the IBC and the Convote dataset. While both give political statements and their labels, the syntax and language used in tweets is distinct from both of these datasets, which are written words or Senate speeches. Therefore, the data that was used in this project was exclusively tweets, in the hopes of accurately capturing how language is used on Twitter in the political sphere.

The tweets that were collected and used are the following: 50 tweets from each current sitting senator (for a total of 5000 tweets) as the test dataset, 50 tweets from each senator in the previous senate (again for a total of 5000 tweets) as a training dataset for the multiclass classifier, and another Kaggle dataset of 86.5k tweets from each party from 2018 (as a training dataset for the linear classifier/RNN-LSTM). The tweets from the senators were obtained via Twitter's API (tweepy) in python.

PREPROCESSING/INPUT:

Every tweet that was collected/downloaded was tokenized into individual words. These tokens were subsequently stemmed, and all of the stopwords/punctuation was removed. From here, there were different transformations done so that the data could be used in the appropriate ML method. For the linear classifier in the baseline method, all tokens were converted to a bag of words model and sent through the linear classifier. For the RNN, all tokens were converted into word embeddings. The word embedding used is a google word2vec model pre-trained on the Google News Dataset. The word embedding vectors were of length 300.

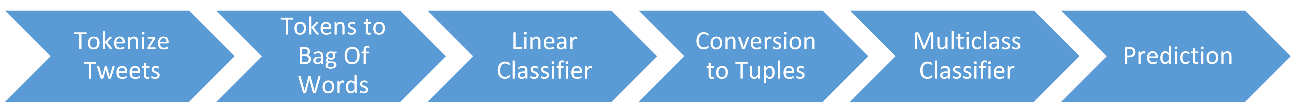
ML PIPELINE EXPLAINED:

The baseline method was a linear classifier and a multiclass classifier. The first linear classifier decided if the tweet was liberal/conservative. This would return a vector of 5000 predictions. Each state would have 100 classifications associated with it. From there, a tuple (number of conservative tweets, number of liberal tweets) was made for each state (e.g. (40, 60)) for a total of 50 tuples. These 50 tuples were inputted into the multiclass classifier, which determined the margin of victory for the winning party. Both the linear classifier and the multiclass classifier were optimized using stochastic gradient descent.

The augmented method utilized an RNN-LSTM, which decided whether the tweet is liberal/conservative. As with the baseline, a tuple was created per state, and then fed into a multiclass classifier that determined the margin of victory for the winning party. The expectation is that the RNN will do a better job at the tweet classification because it can remember previous inputs. Therefore, the multiclass classifier should perform better than the baseline. The project utilized 2 layers of LSTM for the RNN, as well as one dense layer. The RNN-LSTM was optimized using the Adam optimization technique, and the multiclass classifier was optimized with stochastic gradient descent.

For both methods, the Kaggle dataset tweets were used to train the first classifier (linear/RNN). To train the second multiclass classifier, the senator tweets from the 2016-2018 congress were sent through the first classifier to get a vector of classified tweets. Then, tuples were created per state and were then used to train the second multiclass classifier.

Baseline Pipeline



RNN Pipeline

Eshil Patel
NLP Project Write-up
Professor Stone

DISCUSSION/RESULTS:

Baseline Results table: Accuracy 86% F-1 macro – 0.454

	State	Tuples (R,D)	Predicted	Actual	Agreement
0	Alabama	[66, 34]	Solid R (+5)	Solid R (+5)	TRUE
1	Alaska	[55, 45]	Solid R (+5)	Solid R (+5)	TRUE
2	Arizona	[47, 53]	Solid R (+5)	Leans R (+1-5)	FALSE
3	Arkansas	[66, 34]	Solid R (+5)	Solid R (+5)	TRUE
4	California	[13, 87]	Solid D (+5)	Solid D (+5)	TRUE
5	Colorado	[42, 58]	Solid R (+5)	Solid D (+5)	FALSE
6	Connecticut	[13, 87]	Solid D (+5)	Solid D (+5)	TRUE
7	Delaware	[31, 69]	Solid D (+5)	Solid D (+5)	TRUE
8	Florida	[68, 32]	Solid R (+5)	Leans D (+1-5)	FALSE
9	Georgia	[70, 30]	Solid R (+5)	Solid R (+5)	TRUE
10	Hawaii	[20, 80]	Solid D (+5)	Solid D (+5)	TRUE
11	Idaho	[69, 31]	Solid R (+5)	Solid R (+5)	TRUE
12	Illinois	[14, 86]	Solid D (+5)	Solid D (+5)	TRUE
13	Indiana	[68, 32]	Solid R (+5)	Solid R (+5)	TRUE
14	Iowa	[54, 46]	Solid R (+5)	Leans D (+1-5)	FALSE
15	Kansas	[67, 33]	Solid R (+5)	Solid R (+5)	TRUE
16	Kentucky	[46, 54]	Solid R (+5)	Solid R (+5)	TRUE
17	Louisiana	[62, 38]	Solid R (+5)	Solid R (+5)	TRUE
18	Maine	[35, 65]	Solid D (+5)	Solid D (+5)	TRUE
19	Maryland	[21, 79]	Solid D (+5)	Solid D (+5)	TRUE
20	Massachusetts	[17, 83]	Solid D (+5)	Solid D (+5)	TRUE
21	Michigan	[23, 77]	Solid D (+5)	Solid D (+5)	TRUE
22	Minnesota	[19, 81]	Solid D (+5)	Solid D (+5)	TRUE
23	Mississippi	[47, 53]	Solid R (+5)	Solid R (+5)	TRUE
24	Missouri	[52, 48]	Solid R (+5)	Solid R (+5)	TRUE
25	Montana	[74, 26]	Solid R (+5)	Solid R (+5)	TRUE
26	Nebraska	[76, 24]	Solid R (+5)	Solid R (+5)	TRUE
27	Nevada	[20, 80]	Solid D (+5)	Solid D (+5)	TRUE
28	New Hampshire	[17, 83]	Solid D (+5)	Solid D (+5)	TRUE
29	New Jersey	[27, 73]	Solid D (+5)	Solid D (+5)	TRUE
30	New Mexico	[19, 81]	Solid D (+5)	Solid D (+5)	TRUE
31	New York	[11, 89]	Solid D (+5)	Solid D (+5)	TRUE
32	North Carolina	[56, 44]	Solid R (+5)	Leans R (+1-5)	FALSE
33	North Dakota	[72, 28]	Solid R (+5)	Solid R (+5)	TRUE
34	Ohio	[43, 57]	Solid R (+5)	Solid R (+5)	TRUE
35	Oklahoma	[74, 26]	Solid R (+5)	Solid R (+5)	TRUE
36	Oregon	[12, 88]	Solid D (+5)	Solid D (+5)	TRUE
37	Pennsylvania	[34, 66]	Solid D (+5)	Solid D (+5)	TRUE
38	Rhode Island	[22, 78]	Solid D (+5)	Solid D (+5)	TRUE
39	South Carolina	[60, 40]	Solid R (+5)	Solid R (+5)	TRUE
40	South Dakota	[75, 25]	Solid R (+5)	Solid R (+5)	TRUE
41	Tennessee	[63, 37]	Solid R (+5)	Solid R (+5)	TRUE
42	Texas	[45, 55]	Solid R (+5)	Leans R (+1-5)	FALSE
43	Utah	[50, 50]	Solid R (+5)	Solid R (+5)	TRUE
44	Vermont	[28, 72]	Solid D (+5)	Solid D (+5)	TRUE
45	Virginia	[22, 78]	Solid D (+5)	Solid D (+5)	TRUE
46	Washington	[33, 67]	Solid D (+5)	Solid D (+5)	TRUE
47	West Virginia	[54, 46]	Solid R (+5)	Solid R (+5)	TRUE
48	Wisconsin	[46, 54]	Solid R (+5)	Solid D (+5)	FALSE
49	Wyoming	[48, 52]	Solid R (+5)	Solid R (+5)	TRUE

Eshil Patel
NLP Project Write-up
Professor Stone

RNN Results Table: Accuracy 82% F-1 macro -0.432

	State	Tuples (R,D)	Predicted	Actual	Agreement
0	Alabama	[73, 27]	Solid R (+5)	Solid R (+5)	TRUE
1	Alaska	[46, 54]	Solid R (+5)	Solid R (+5)	TRUE
2	Arizona	[48, 52]	Solid R (+5)	Leans R (+1-5)	FALSE
3	Arkansas	[61, 39]	Solid R (+5)	Solid R (+5)	TRUE
4	California	[23, 77]	Solid D (+5)	Solid D (+5)	TRUE
5	Colorado	[47, 53]	Solid R (+5)	Solid D (+5)	FALSE
6	Connecticut	[23, 77]	Solid D (+5)	Solid D (+5)	TRUE
7	Delaware	[34, 66]	Solid D (+5)	Solid D (+5)	TRUE
8	Florida	[46, 54]	Solid R (+5)	Leans D (+1-5)	FALSE
9	Georgia	[66, 34]	Solid R (+5)	Solid R (+5)	TRUE
10	Hawaii	[25, 75]	Solid D (+5)	Solid D (+5)	TRUE
11	Idaho	[66, 34]	Solid R (+5)	Solid R (+5)	TRUE
12	Illinois	[25, 75]	Solid D (+5)	Solid D (+5)	TRUE
13	Indiana	[65, 35]	Solid R (+5)	Solid R (+5)	TRUE
14	Iowa	[61, 39]	Solid R (+5)	Leans D (+1-5)	FALSE
15	Kansas	[78, 22]	Solid R (+5)	Solid R (+5)	TRUE
16	Kentucky	[51, 49]	Solid R (+5)	Solid R (+5)	TRUE
17	Louisiana	[57, 43]	Solid R (+5)	Solid R (+5)	TRUE
18	Maine	[48, 52]	Solid R (+5)	Solid D (+5)	FALSE
19	Maryland	[30, 70]	Solid D (+5)	Solid D (+5)	TRUE
20	Massachusetts	[23, 77]	Solid D (+5)	Solid D (+5)	TRUE
21	Michigan	[40, 60]	Solid D (+5)	Solid D (+5)	TRUE
22	Minnesota	[38, 62]	Solid D (+5)	Solid D (+5)	TRUE
23	Mississippi	[44, 56]	Solid R (+5)	Solid R (+5)	TRUE
24	Missouri	[54, 46]	Solid R (+5)	Solid R (+5)	TRUE
25	Montana	[60, 40]	Solid R (+5)	Solid R (+5)	TRUE
26	Nebraska	[73, 27]	Solid R (+5)	Solid R (+5)	TRUE
27	Nevada	[25, 75]	Solid D (+5)	Solid D (+5)	TRUE
28	New Hampshire	[34, 66]	Solid D (+5)	Solid D (+5)	TRUE
29	New Jersey	[29, 71]	Solid D (+5)	Solid D (+5)	TRUE
30	New Mexico	[30, 70]	Solid D (+5)	Solid D (+5)	TRUE
31	New York	[28, 72]	Solid D (+5)	Solid D (+5)	TRUE
32	North Carolina	[63, 37]	Solid R (+5)	Leans R (+1-5)	FALSE
33	North Dakota	[67, 33]	Solid R (+5)	Solid R (+5)	TRUE
34	Ohio	[52, 48]	Solid R (+5)	Solid R (+5)	TRUE
35	Oklahoma	[63, 37]	Solid R (+5)	Solid R (+5)	TRUE
36	Oregon	[30, 70]	Solid D (+5)	Solid D (+5)	TRUE
37	Pennsylvania	[42, 58]	Solid R (+5)	Solid D (+5)	FALSE
38	Rhode Island	[22, 78]	Solid D (+5)	Solid D (+5)	TRUE
39	South Carolina	[66, 34]	Solid R (+5)	Solid R (+5)	TRUE
40	South Dakota	[70, 30]	Solid R (+5)	Solid R (+5)	TRUE
41	Tennessee	[60, 40]	Solid R (+5)	Solid R (+5)	TRUE
42	Texas	[38, 62]	Solid D (+5)	Leans R (+1-5)	FALSE
43	Utah	[60, 40]	Solid R (+5)	Solid R (+5)	TRUE
44	Vermont	[24, 76]	Solid D (+5)	Solid D (+5)	TRUE
45	Virginia	[38, 62]	Solid D (+5)	Solid D (+5)	TRUE
46	Washington	[32, 68]	Solid D (+5)	Solid D (+5)	TRUE
47	West Virginia	[68, 32]	Solid R (+5)	Solid R (+5)	TRUE
48	Wisconsin	[48, 52]	Solid R (+5)	Solid D (+5)	FALSE
49	Wyoming	[59, 41]	Solid R (+5)	Solid R (+5)	TRUE

Eshil Patel
NLP Project Write-up
Professor Stone

These results show some rather interesting and unexpected results when comparing both methods. At first glance, it seems that the baseline method performs better than the RNN-LSTM method. However, after running an approximate permutation test, it turns out that this discrepancy can be accounted for by chance (p-value of 0.49). In other words, the baseline method and the RNN have the same performance on the data. This is surprising considering that the goal of the RNN-LSTM was to capture more of the previous context of the tweet.

The linear classifier/RNN seem to capture the ideology of the tweets effectively. On closer inspection of the tuples created by each method, however, there are some notable errors. Texas got a liberal leaning tweet tuple from both classifiers (the RNN-LSTM's performance was poorer than the baseline), indicating that some tuning needs to be done to really captures the intricacies of the tweet. Other notable errors are Iowa and Florida, which had conservative leaning tuples, but went Democratic in 2018.

Another interesting result is the overall accuracy of each of the methods, which is surprisingly solid. This lends credence to the idea that each party is either talking about different issues, or are framing issues in a different, distinct way. This also indicates that a lot of states are already set in their voting patterns and have politicians that reflect that. These insights are backed up especially with the baseline classifier, which only used a bag of words representation. The baseline was able to perform well with just the words used by each politician, which suggests that the politicians are using different words to describe their positions. However, the errors that each of these methods made is that they could not (or would not) predict close elections at all. This is most likely due to a class imbalance (most states either are solidly democratic or solidly republican, so there aren't as many data points for the close states).

Comparing the accuracy of this hypothesis to the hypothesis of prior work (sentiment analysis) yields promising results. Previous papers, such as one published in Columbia Undergraduate Review Journal, had 66.7% accuracy in using sentiment analysis to predict the 2016 election. Other papers have yielded similar/worse results. However, a big caveat for this project is the lack of training on the multiclass classifier. The multiclass classifier was trained on merely 50 data points. In future works, more data points (potentially from different races, such as state races) can be used to obtain better classification boundaries, especially for close races.

The significance in the results of the project is that the hypothesis described here is another viable way of understanding the political inclinations of the US population. It can provide political parties with another tool to align/evaluate their messaging in different regions of the country, which is especially important in winning tight elections. Other than use by political parties, it provides another potential avenue for interested users and websites like 538 to evaluate/create models for election predictions. Limitations of this methods include predicting states that have close races. In cases like these, it may be more appropriate to gather tweets of a larger pool of politicians, in the hopes of more accurately capturing the political winds in the moment.

Eshil Patel
NLP Project Write-up
Professor Stone

REFERENCES:

1. Iyyer, M. *Political Ideology Detection Using Recursive Neural Nets*
2. Lee, C., Shiff, J. , Thatipamala, S. *Prediction US Political Party Affiliations on Twitter*
3. Hamling, T., Agrawal, A., *Sentiment Analysis of Tweets to Gain Insights into the 2016 Election*
4. Heredia, B., Prusa, J., Location Based Twitter Sentiment Analysis for Predicting the US 2016 Presidential Election
5. Oikonomou, L. , Tjortjis, C., *A Method for Predicting the Winner of the USA Election using Data Extracted from Twitter.*
6. Metaxas, P., Mustafaraj, E., *How (Not) to Predict Elections*
7. Kaggle Dataset, *https://www.kaggle.com/kapastor/democratvsrepublicantweets/version/3?*
8. Keras Tutorial, *https://adventuresinmachinelearning.com/keras-lstm-tutorial/*
9. Google word2Vec, *https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit*
10. CNN, *https://www.cnn.com/election/2016/results*
11. 538, *https://fivethirtyeight.com/features/the-2018-map-looked-a-lot-like-2012-and-that-got-me-thinking-about-2020/*