Zohaib Mannan
Nico Bykhovsky
Alisha Bhatia

Sentiment Analysis of Tweets to Predict Political Affiliation

Abstract

This paper delves into code written to find the political polarity of tweets through different methodologies. Specifically, it is able to find political polarity by pulling tweets directly from Twitter. To initially analyze these different methodologies, a training data set of tweets was found that contained tweets and the associated polarity of those writing the tweets. This data set was used to analyze the connection between the tweet and the associated political party of the source. After the tweets and associated party were extracted, different models were used to measure political polarity, including linear SVM classifiers, naive bayes, logistic regression, and LSTM. The results of all of these pipelines were analyzed and it was found that linear SVC had the highest accuracy of 0.85. This highest performing methodology was then used to create a pipeline to predict the polarity of tweets coming from Twitter itself. The political parties that were used for classification included Republican (R) Democrat (D) or Independent (I).

Introduction:

When reading a tweet online or glancing through a news article, the political polarity of the type of writing may be unclear. Knowing this information could optimize a user's experience by helping ensure that a person receives quality information and help make the person more aware of where the information they are consuming is coming from. Furthermore, knowing the political polarity of different texts read online can help create a more pleasurable reading experience for the user. Knowing this information can help them decide if they want to continue reading the tweet/article or not. TF-IDF was implemented in conjunction with linear SVM classifiers to eventually take in a tweet as input and output the political party that the user is affiliated with.

State of the art:

Currently, there are different studies that have been made to help reach a similar goal of finding the political polarity of different tweets. Although there is no mainstream feature that finds this, there are different studies that have done research on this topic. Many of these studies relate to finding political polarity of a specific subset of users or focus on a specific time period. This paper focuses on the methods explored to create a generalized political polarity of any tweet given a hashtag. It also explores this feature using different methodologies and analyzes some of the highlights of each one. Specifically, the methods used to find polarity were TF-IDF along with linear SVC classifiers, LSTM and naive bayes.

Solution:

Before the code for the prediction pipeline was created, the appropriate training data set was needed. Finding this was done through searching through various data sets of tweets, and ensuring that the one selected included the political party of the source. Searching for this attribute of the dataset was necessary to ensure that there was a way to verify the results of the pipeline. Once the data set of senators was found, which matched these requirements, the data set was visualized to get further information on the data set. Spedicially, the division of those in both parties was visualized into a bar graph, and it was found that the majority of those in the data set identified as Republican, and that the party with the lowest number of senators was Independent (Figure 1). The data set was further visualized to show the number of tweets from senators from each state (Figure 2). It was found that senators from Alabama tweeted the least, at around 2000 tweets, and those from more active states had over 6000 tweets.

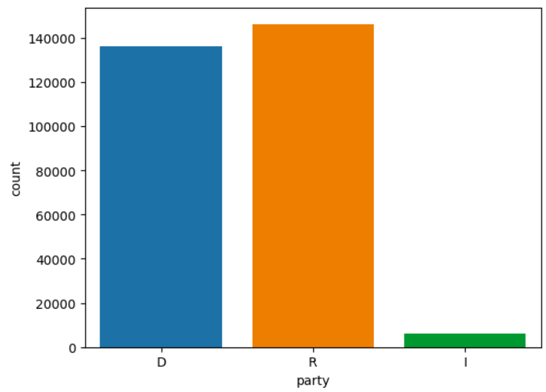


Figure 1

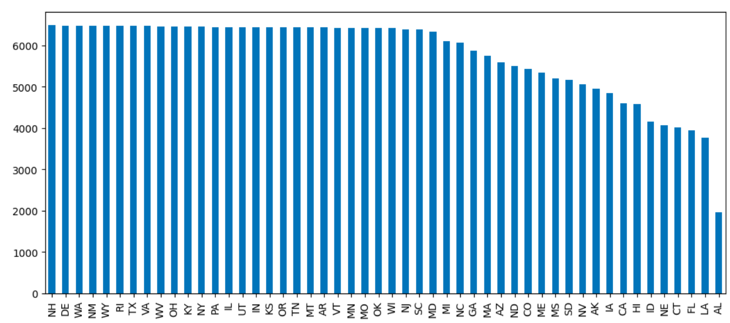


Figure 2

Once the data was analyzed, to analyze the different models, the raw data text was needed. Therefore, the data set was split so that the actual tweets were contained in the variable X and the party for corresponding tweets was contained in variable Y. Afterwards, the training data set was split into 4 parts so that each model had a different subset of the data to work on. Then, to be able to use the tweets and corresponding data in the models selected, the TfidfVectorizer function was used and stop words were removed. This function converted the strings from the data set into vectors that could be used within the models.

The first model that was used to find the political polarity of the data was naïveté bayes. This model was applied to the split data set and the results of the model were analyzed. Then a similar process was performed on the model called linear SVC, using part of the training data and the results were analyzed. This process was repeated again for logistic regression, and results were analyzed.

After all of these models were tested on the different portions of the data sets and the results were analyzed, the model that had the highest overall accuracy was used to implement a pipeline that took tweets from Twitter directly given a hashtag and outputted the found political polarity of the said tweet. The start of this implementation included adding a TF_IDF vectorization which would convert the found tweet into vector form. After this, a Linear SVC model was included which worked to find the political polarity from the tweet. The political polarity was reflected through explicitly stating the possible affiliated party of the source. Specifically, the results were either R (Republican), D (Democratic) or I (Independent).

Results:

With a test set of n = 115,446 tweets, *Figure 3* shows the results of the linear SVC model in a confusion matrix:

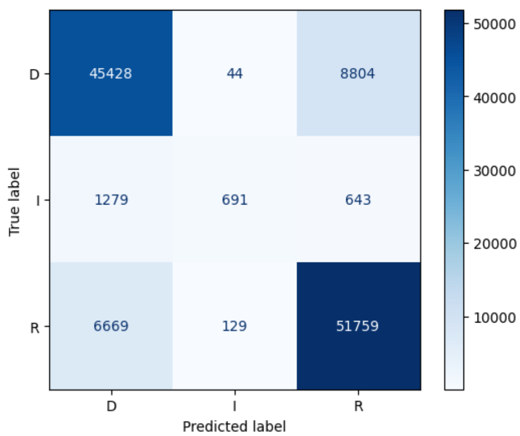


Figure 3

	precision	recall	f1-score	support
D	0.85	0.84	0.84	54276
I	0.80	0.26	0.40	2613
R	0.85	0.88	0.86	58557
accuracy			0.85	115446
macro avg	0.83	0.66	0.70	115446
weighted avg	0.85	0.85	0.84	115446

```
[[45428    44  8804]
 [ 1279   691   643]
 [ 6669   129 51759]]
```

Figure 4

In figure 3, the predicted label axis represents what the model predicted the results to be, versus, the true label axis represents the actual answers. The calculated precision, recall and f1-score for this model is then shown in figure 4. As mentioned beforehand, the results of the linear SVC model had the best performance, with both the other models tested (naive bayes and logistic regression) tying for an overall accuracy of 0.84. The results for the naive bayes model is shown in Figures 5 and 6, and the results for the logistic regression model are shown in Figures 8 and 7.

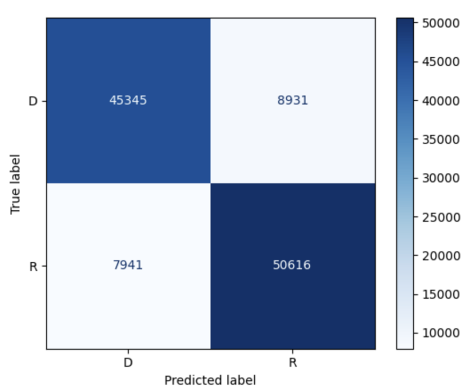


Figure 5

	precision	recall	f1-score	support
D	0.83	0.84	0.83	54276
R	0.83	0.86	0.85	58557
micro avg	0.83	0.85	0.84	112833
macro avg	0.83	0.85	0.84	112833
weighted avg	0.83	0.85	0.84	112833

```
[[45345  8931]
 [ 7941 50616]]
```

Figure 6

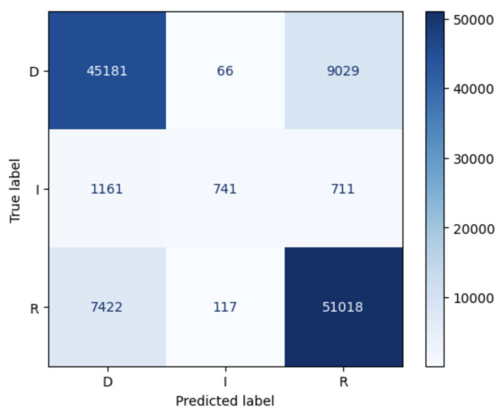


Figure 7

```
              precision    recall  f1-score   support

           D       0.84      0.83      0.84     54276
           I       0.80      0.28      0.42      2613
           R       0.84      0.87      0.86     58557

    accuracy                           0.84    115446
   macro avg       0.83      0.66      0.70    115446
weighted avg       0.84      0.84      0.84    115446

[[45181    66  9029]
 [ 1161   741   711]
 [ 7422   117 51018]]
```

Figure 8

After training the model for several hours, we were able to achieve this 85% test set accuracy , essentially surpassing our initial target goal of 80%. Below are some examples of tweets and their associated classifications:

Table 2: Examples of Correctly Classified Tweets

Tweet	Prediction
How could the Obama administration pay Iran $1.7 billion out of a fund that should already be depleted?	Republican
Obamacare is giving us higher premiums and canceled policies -- much bigger problem than the website. trainwreck	Republican
SCOTUS should strike down TX law that makes it much more difficult for women to make their own health care choices.	Democrat
Cyber is a national security imperative, essential to protecting Maine infrastructure—like our electric grid and healthcare system—from bad actors.	Independent

Table 3: Examples of Misclassified Tweets

Tweet	Prediction	Actual
Flynn's resignation is tip of iceberg. So many Qs unanswered about Russia's interference in election and its communication with Trump team	Democrat	Republican
Proud to join with Senator Shaheen at York Hospital to call for funding to fight opiate epidemic	Democrat	Independent
My bill to help recruit and retain federal workers would support law enforcement, permitting, agriculture and many	Republican	Democrat

other jobs and biz across ND		
Jim Marshall bravely served during WWII. It was an honor to deliver the medals he earned to his family yesterday.	Republican	Democrat

Once it was clear that linear SVC was the best performing model, the pipeline was created, and this included TF-IDF vectorization to pull tweets directly from Twitter. Thenthe linear SVC model was applied to these pulled tweets to classify the tweet into its proper political category (again with R indicating Republican, “D” indicating Democratic and “I” indicating independent. An analyzed tweet for this pipeline along with the associated prediction are shown in Figure 9.

```
**Tweet Text**
 Yup but there’s die hards who are Obama Crazy lol #Obamasucks #MichaelObama #obamacare https://t.co/Zuga59GcJb
['R' 'R' 'R' 'R' 'R']
```

Figure 9

Long-Short Term Memory:

The final classification methodology we tested in this paper is an implementation of Long Short Term Memory using pytorch. In building the model, the authors first preprocessed and vectorized the tweet text data to implement a Long Short Term Memory (LSTM) model. This was done using nltk's tokenization and padding methods. They then used an embedding layer to map the vectorized tweet text data to a lower-dimensional space. Next, they defined the LSTM layer, which takes in the embedded tweet text data and outputs a sequence of hidden states. Finally, they added a fully connected layer on top of the LSTM layer to perform the classification. They used an activation function to output probability scores for each class. They then defined a loss function and optimized the model using an optimizer (specifically Adam) and back propagated to update the model's weights and biases during training.

The advantage of using LSTM over other methods was hypothesized to be in its ability to capture long-term dependencies in the data, making it useful for classification tasks where the order of the input data matters. Additionally, the authors expected its ability to handle variable-length input sequences and noisy data to handle the text from twitter better than other methods. Below is the performance of the LSTM model with regards to changing batch sizes and hidden sizing:

Figure 10

```
Accuracy  : 0.7947
Precision : 0.8003
Recall    : 0.7853
F1 Score  : 0.7928
```

Figure 11

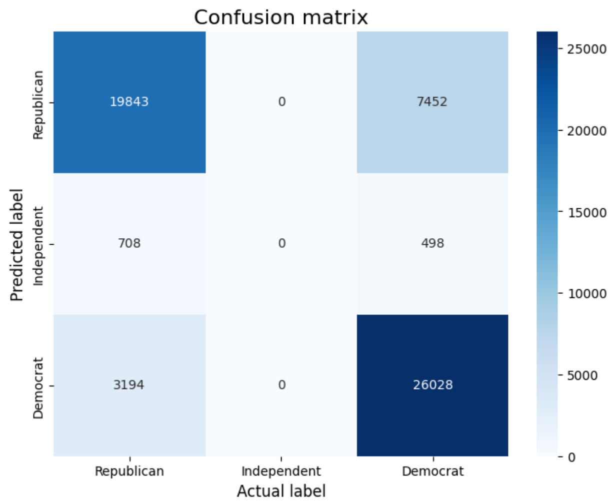
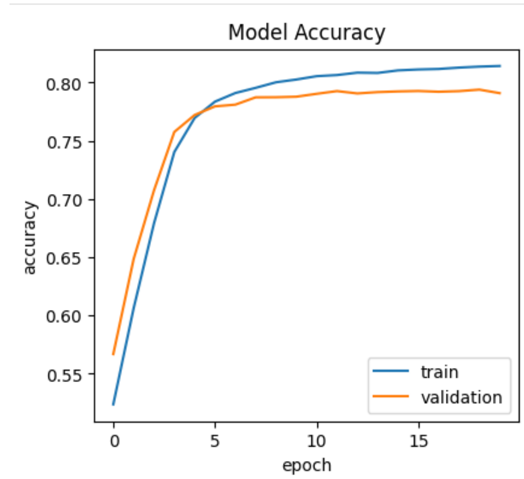


Figure 12



From the figures above, we see that LSTM performed exceptionally well, but with 5-6% less accuracy against previous linear SVM models.

Related work:

Certain papers have also found political polarity of tweets for a specific subset of users through related mechanisms. A specific study done found the political inclination of users during the 2016 election, and whether they were more likely to support Trump or Hillary. It also then went further to measure political homophily after classifying the Twitter users.

In this study, after classifying a tweet as political or not, the SentiStrength tool was used, which searches for words that it already “knows” to be positive or negative. After searching and finding these words in the sentence, values that indicate how positive or negative the word is are marked, and the difference between these two values indicates how positive or negative the final tweet is. This was used to help determine how the user felt towards either political candidate.

The approaches found in the study mentioned beforehand are similar to the approaches of sentiment analysis focused on in this paper. Specifically, the SentiStrength tool assigns a value to each word indicating positive or negative sentiment, and the code discussed in this paper acts similarly. Specifically, in the logistic regression model used i, the relationship between certain variables is used to find the overall sentiment, which is similar in certain ways to how SentiStrength works. One prominent difference between these two methods is that SentiStrength classifies words to be either positive or negative, but the models mentioned in this paper classify into political parties. Overall, both studies included sentiment analysis and worked with data sets that contained tweets, leading to certain similarities in finding these classifications.

A different study that is similar to the one mentioned in this paper is a study that focuses on tweets related to the Indian general election of 2019 and studies sentiments of Twitter users towards the major political parties in India. 3869 total tweets were analyzed and then classified into two parties, and any tweets that did not fall into these two parties were categorized into “other”. Before any sentiment analysis was performed on these tweets, the tweets were preprocessed, similar to what was done in the project for this paper, and stop words were also removed. Additionally, TF-IDF was used in this study to convert the raw documents into a matrix that represents term frequency vs inverse document frequency. Although the form of TF-IDF that was used in the project discussed in this paper was through vectorization, there was similarity in the models used to form this sentiment analysis, specifically in that logistical regression and LSTM were also used. In both studies the precision, recall and f1 score were also measured.

The results of the precision, recall and F1 scores for both studies showed some similarities and differences. The study discussed previously showed that the precision from using logistic regression was within the range of 0.70 - 0.72 versus in the study conducted in this paper, the precision for this model was higher at 0.80-0.84. For other factors such as f1-score and recall, there was a wider range of results in the study focused on in this paper, with the lowest recall being 0.28 and highest being 0.87. There was also a higher range in terms of the f1-score, as the highest value in this study was 0.84 and the lowest value in this category was 0.42. In the aforementioned study, there was more consistency within the ranges produced for both of these categories, although the values themselves were not as high.

Conclusion

Finding the political polarity of tweets is a study that has been done by multiple sources for different subsets of Twitter users, and this paper discusses the different methods that could be used to find this polarity and compares the use of each. This study goes even further to implement a prediction pipeline using the best performing model that can find the polarity of tweets from Twitter directly given a specific hashtag.

After carefully preprocessing, debugging, training, and fine-tuning the model, Linear SVM ended up with the highest testing accuracy of 85%, which has passed the target goal of 80% accuracy. After downloading and analyzing the dataset, there was a concern for its vastness and variability compared to datasets by related studies. As such, the results were not expected to be as comparable to previous work, whose top-performing model had 91.6% accuracy. It is believed that this accuracy, as well as the final accuracy, is far higher than most average humans' accuracy in predicting the political party of a random U.S senator, given one random tweet without context.

While it is a thrill that the target goal was met, if given more time, the results could have been even closer to the stretch goal of 90% accuracy. The main barrier was certainly the extensive amount of time that it took to run the model in its entirety, as it is apparent that further hyperparameter fine-tuning could have improved the accuracy. Another step that could have been taken, if there was more time, is to further analyze the dataset. With a bit more manual investigation, There would have been swift removal of tweets whose data seemed to be out of the ordinary (for instance, tweets that contained emoticons).

The main challenges throughout this project were determining how to best preprocess the data and constructing the model in Tensorflow, which was a framework new to all of us. The US senators' Tweets dataset was also very large and took many hours to train and output models from. There was an initial inquiry to include mentions in the vocabulary list, according to the hypothesis that including certain people would provide relevant information as to the political beliefs of the tweeter; with all of this data, the vocab size was around 286,000. No one in the group had much experience with Keras prior to implementing this model, so learning how to design, create, and train models in Keras and Tensorflow proved to be a steep learning curve for us. Familiarizing with the Keras framework was quite time consuming, and even after learning how to transfer the TensorFlow model into Keras, debugging in this unfamiliar framework was challenging. The first struggle came with Tensorflow, as the input data were improper types for many of the layers and functions. Simply getting accustomed to the differences between TensorFlow and Keras was a very consuming and challenging step in developing the model. Due to the sheer amount of data that the algorithm was attempting to train the model on, it took a very long time to fine-tune hyperparameters, even when training on SGD. However, there was still room to test various hyperparameters, including learning rate, batch size, more dense layers, momentum, and embedding size. After many iterations, the code was settled on a learning rate of 0.1, a batch size of 64, momentum of 0.8, and an embedding size of 32.

In addition, there was an original interest within the group in seeing how the model would be used to create a real-time tweet aggregator that can fetch tweets through an API and, targeting a specific user or keyword, identify what political party that tweet is likely affiliated with and filter those tweets in a live feed. There was also some dabling with the idea of doing news articles, not tweets, to create some real-time new aggregator for a specified news site. There was no anticipation that acquiring the data and running the model would take as many days/hours as they did, so there unfortunately was not enough time to explore this task. If the opportunity for a similar project came again, it would definitely be interesting to attempt a real-time tweet aggregator. Other than not reaching this last step, the program stuck pretty closely to the original plan for implementing this project, and while many parts took longer than expected, the group is overall pleased with the trajectory of the project. This project goes to show how much coded language is used on social media to further party agendas. The fact that it can become so easy for a deep learning network to determine posters' political party affiliation from a single tweet about nearly any topic demonstrates how much political content is consumed by users on a daily basis. It truly demonstrates the importance of being vigilant users of social media.

References

Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. Computing Research Repository.

Gustavo Laboreiro, Lu´ıs Sarmento, Jorge Teixeira, and Eug´enio Oliveira. 2010. Tokenizing microblogging messages using a text classification approach. In AND '10: Proceedings of The Fourth Workshop on Analytics for Noisy Unstructured Text Data, New York, NY, USA, October. ACM.

Wang H, Can D, Kazemzadeh A, Bar F, Narayanan S
A system for real-time Twitter sentiment analysis of 2012 US Presidential election cycle., ACL (System Demonstrations) (2012), pp. 115-120

Conover M, Gonçalves B, Ratkiewicz J, Flammini A, Menczer F. (2011a) "Predicting the political alignment of Twitter users." In Proceedings of SocialCom/PASSAT Conference, pp. 192–199

Conover M, Ratkiewicz, J, Francisco M, Goncalves B, Menczer F, Flammini A. (2011b) "Political Polarization on Twitter." In International AAAI Conference on Web and Social Media.

Choy M, Cheong MLF, Laik MN, Shung KP (2011) "A sentiment analysis of Singapore presidential election 2011 using twitter data with Census correction." CoRR abs/1108.5520

Mishra P, Rajnish R, Pankaj Kumar
Sentiment Analysis of Twitter Data: Case Study on Digital India. INCITE-2016, Amity University, India (2016)

Marchetti-Bowick M, Chambers N. (2012) "Learning for microblogs with distant supervision: political forecasting with Twitter." In Proceedings of the EACL Conference, pp. 603–612

K.Kohnson, D. Jin, D. Goldwasser. *Leveraging Behavioral and Social Information for Weakly Supervised Collective Classification of Political Discourse on Twitter.* ACL, 2017. Github repository, https://github.com/kmjohnson/twitter-framing

E. Summers, N. Ruest. *Twarc*. University of Maryland, 2017. GitHub repository, https://github.com/DocNow/twarc

C. Olah. *Understanding LSTM Networkds*. 2015. Github repository http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Author links open overlay panelMohd ZeeshanAnsariaM.B.AzizaEnvelopeM.O.SiddiquibH.MehraaK.P.Singha, et al. "Analysis of Political Sentiment Orientations on Twitter." *Procedia Computer Science*, Elsevier, 16 Apr. 2020, www.sciencedirect.com/science/article/pii/S1877050920306669.

Caetano, Josemar A., et al. "Using Sentiment Analysis to Define Twitter Political Users' Classes and Their Homophily during the 2016 American Presidential Election - Journal of Internet Services and Applications." *SpringerOpen*, Springer London, 3 Sept. 2018, jisajournal.springeropen.com/articles/10.1186/s13174-018-0089-0.