Modeling Politicization on Short Form Text Data

Noah T. Olsen, M.S.

The George Washington University

Table of Contents

[Abstract 3](#_Toc39498460)

[Modeling Politicization on Short Form Text Data 4](#_Toc39498461)

[Introduction 4](#_Toc39498462)

[Literature Review 5](#_Toc39498463)

[Data & Data Analysis 6](#_Toc39498464)

[Research Methodology 12](#_Toc39498465)

[Results 13](#_Toc39498466)

[Future Work 14](#_Toc39498467)

[Conclusion 15](#_Toc39498468)

[Biography 16](#_Toc39498469)

[References 17](#_Toc39498470)

Abstract

Social media platforms have become one of the most popular places for individuals to express their political opinions and beliefs. One of the topic areas where this trend is the most pronounced is politics, with supporters from all sides of the political arena actively engaging with each other and voicing their beliefs and opinions. This project attempts to use the tweets of members of congress to establish a model whereby text can be categorized into bins based on political ideology. Using pre-trained GloVe vectors as input to a recurrent neural network using Gated Recurrent Units (GRUs) tweets classified as either moderate, liberal and conservative.

Keywords: NLP, Twitter, Politics, Machine Learning, Neural Networks, GloVe, Gated Recurrent Units

Modeling Politicization on Short Form Text Data

# Introduction

In recent years social media platforms have become one of the most popular places for individuals to express their opinions and beliefs on a wide variety of topics. One of the topic areas where this trend is the most pronounced is politics, with supporters from all sides of the political arena actively engaging with each other and voicing their beliefs and opinions. While this kind of back and forth is normal in a healthy democracy, in recent years hostile foreign actors have exploited these conversations to influence public opinion (Badawy, Ferrara, Lerman, 2018). Even before the emergence of social media platforms like Twitter as an arena for cyber warfare and social engineering, much research was going in to developing machine learning methods to help identify bots (Wang, Foresti, & Jarodia, 2010).

These early attempts focused primarily on class of bot known as spam bots, which while not necessarily political in nature, were automated with the intention of spreading spam or otherwise fake information. The “Internet Research Agency” (IRA), the arm of Russia’s intelligence apparatus responsible for the vast majority of the meddling in the 2016 election, took these early strategies and further intensified and weaponized them.

These efforts intensified after Russia’s successful interference in the 2016 election, with government agencies like The Defense Advanced Research Projects Agency (DARPA) sponsoring competitions to help combat this sort of misinformation. Many of the machine learning models that have been developed up to this point have focused on graph theory in attempt to identify large networks of accounts working in concert with each other to sow discord. While many of these models have had success at identifying these bot networks, groups like the IRA have continued to adapt to make some of the more traditional network analysis methods less effective at identifying their interference attempts.

The IRA’s strategy had three main pillars. The first pillar was focused on creating campaigns and petitioning African American voters to boycott the election, or to follow incorrect voting procedures that would make them unable to vote. This strategy has also been applied to make Hispanic voters distrust government agencies, and thus be less likely to vote. The second pillar was focused on encouraging members of the far right to become more confrontational. The last pillar is broader and was simply to spread sensationalist, conspiratorial, or otherwise fake information. Interestingly, rather than focusing on voters of one political ideology they attempted to sow discord among both liberal and conservative voters.

While the model presented in this paper will not focus on the identification of bots like the ones described above, its ability to identify politically polarizing text has the potentially to be implemented in future bot detection algorithms that can identify politically charged information even before the network of bots itself is identified.

# Literature Review

Several prior models have looked at trying to classify text based off political ideology. That being said, for the most part these models have focused on longer pieces of text like news articles to try and detect political bias in the news media (Doumit & Minai, 2011), or documents that are almost always political in nature like congressional floor speeches (Iyers, Enns, Boyd-Graber, & Resnik, 2014) rather than short form content like tweets that will be used in this paper’s model.

Doumit & Minai 2011 collected articles from over 30 worldwide news organizations and used Latent Dirichlet Allocation to extract a set of topics for each news outlet and proceeded to apply a sentiment analysis model. While the study did not identify a clear answer for whether or not there is an overwhelming bias in the media, the authors note the importance of key terms and phrases in their model.

Iyers, Enns, Boyd-Graber, & Resnik (2014) used the Convote dataset from Thomas et al. (2006) containing congressional floor speeches from 2005, labeled according to the political party of the speech’s author. They then trained a recurrent neural network (RNN) after filtering the speeches for a set of explicitly political key phrases the authors took from Yano et al. (2010). This was due to the authors’ belief that that not all words in sentences in political speeches are inherently political, and that by filtering for a smaller set of key topics they could improve the accuracy of their neural network. Their approach was novel as similar attempts at these sorts of tasks had relied on logistic regression using the bag-of-words technique along with word2vec which the authors’ model outperformed significantly. Compared to their logistic regression benchmark which achieved 64.7% accuracy on their classification task, their best model which used a recurrent neural network with word2vec word embeddings to achieve over 70% accuracy. The authors of this paper note the importance of word embeddings in their success and will also be extremely important in the model we present below.

Not many models have been developed to identify the political ideology have been trained off of Twitter data, but those that have almost exclusively focus on a set of linguistic features to help make these determinations rather than relying on recurrent neural networks. In Djemili et al. (2014) the authors trained such a model whereby they applied a set of linguistic rules to a dataset of tweets they scraped containing tweets from French politicians resulting in a corpus of just over 34,000 tweets. It is important to note that while this corpus was in French, the methods and conclusions of the paper are applicable across languages. The model classified tweets as ideological or not ideological rather than ascribing them to a particular political ideology like is done in many of the models described up to this point.

# Data & Data Analysis

The initial dataset was obtained by scraping the most recent 3,200 tweets from 563 accounts belonging to members of the United States Senate and House of Representatives. The number of accounts is greater than the total number of Senators and Representatives because some members have more than one account. The initial scrape resulted in 1.44 million tweets authored between February 10th, 2009 and March 3rd, 2020. A graphical representation of how these tweets are distributed over time can be seen in the graph below.

A screenshot of a cell phone

Description automatically generated

*Figure 1*. Line graph showing the distribution of the data over time.

The process described in the methodology section will involve scoring each tweet according to the political ideology of its author. Looking at the distribution of tweets by political party we get the graph below showing slightly more tweets from democrats than republicans, but the data set is still fairly balanced.

A screenshot of a cell phone

Description automatically generated

*Figure 2*. Distribution of tweets by party.

The ideologies scores for each member of Congress were taken from Govtrack. Their scores are set on a scale of 0 to 1 with 0 being extremely liberal and 1 being extremely conservative. The average ideology score for each party can be seen in the graph below.

A screenshot of a cell phone

Description automatically generated

*Figure 3*. Average Ideology Score by Party.

More importantly for this project than the ideology distribution of the people writing the tweets, is the content of the tweets themselves. One of the peculiarities of using twitter as a dataset is the length of tweets, which for our purposes we will refer to as documents. The average number of characters per tweet was just under 170 characters, the left side of the distribution looks like you'd probably expect with a long tail. However, the right side of the distribution lacks this same tail due to Twitter limiting the number of characters per tweet to 280, resulting in hard stop on the right side of the distribution.

A screenshot of a cell phone

Description automatically generated

*Figure 4*. Histogram of Tweet Length in terms of characters.

One of the conditions we will need to hold in order for a model like the one we are trying to construct to work is the fact that Democrats and Republicans are talking about significantly different topics to differentiate between the two. As such, gaining an understanding of what each group is talking about is important step in establishing the viability of the model. When looking at the top 30 words overall regardless of party, the results are fairly standard for what you might expect members of Congress to be talking about. The vast majority of the words have to do with government or political issues.

A screenshot of a cell phone

Description automatically generated

*Figure 5*. Word cloud of most frequently used words in the data.

When looking at the same plot broken down by party does show small differences, at first glance it doesn't create the stark contrast that you might expect. This is likely because at the end of the day, both Republicans and Democrats are talking about the same issues just from very different perspectives. That being said, there are small differences that can be observed, for example it appears Republicans use the word "democrat".

A screenshot of a social media post

Description automatically generated

*Figure 5*. Word cloud of most frequently used words in the data by party.

Another way to look at this problem, is to look at the top words for each party in terms of the term frequency–inverse document frequency (TFIDF) which can be seen as a measure of the most unique words for a document, or in this case a group of documents.

A screenshot of a cell phone

Description automatically generated

*Figure 6*. Top TF–IDF terms by party.

Now we really start to see the difference between the two classes, with Democrats talking about issues like "#goptaxshutdown","#trumpshutdown", and "#gunviolence" while Republicans are discussing issues like "#taxreform", "#obamacare", and "unemployment". Is it also very interesting to see that the graph looking at the top words for each party doesn't have a single Twitter hashtag (#) but the top words in terms of TF–IDF for each party are almost hashtags, showing their potential value and importance in an eventual model. For our model though rather than simply labeling text to a certain party, we will split the data into five equal bins, which results in the TF–IDF plot below.

A screenshot of a cell phone

Description automatically generated

*Figure 7*. Top TF–IDF terms by model group.

This strategy has several advantages discussed in the methodology section of the site, but the groups have the average scores shown in the plot below.

A screenshot of a cell phone

Description automatically generated

*Figure 8*. Ideology score by model group.

# Research Methodology

For this project, the data scraping pre-processing and visualization were done in R, with the model itself be run in Python. Using the rtweet R package (Kearney, 2019) I was able to pull over 1.4 million tweets from members of the U.S. Congress. Each tweet was then assigned an ideology score based off the ideology of the tweet’s author. The congressional ideologies were taken from govtrak.us, whose annual ideology scores rank each member of congress on a scale from 0 to 1 with 0 being extremely liberal and 1 being extremely conservative. Next, using the popular Tidyverse suite of packages (Wickham et al., 2019), the data was preprocessed to remove unnecessary variables as well as to remove tweets with missing entries. Lastly the data was visualized using the ggplot2 package from the Tidyverse.

The goal of this model to accurately assess whether strings of text are partisan. As described in the data section, I am training the model on over a million tweets from members of Congress and assigning each tweet a partisan score based on a partisan score of the tweet's author. To solve this problem, we will be treating it as a text classification problem, with each tweet assigned to one of 5 bins according to its partisan score I described above. These bins were labeled "Very Liberal", "Moderately Liberal", "Moderately Conservative", and "Very Conservative". More information on these bins can be found in the data section. The model itself is a recurrent neural network using Python and Keras (Chollet et al., 2015) where the inputs are pretrained GloVe word vectors, specifically their Twitter model trained on 2B tweets resulting in 27B tokens. I decided to use these vectors instead of something like TF-IDF vectors because of the breadth of tokens as well as the fact the vectors were trained directly on Twitter data, the same source as the data for my model. To train the classifier, I decided to use Gated Recurrent Units (GRUs) which tend to perform better than Long Short-Term Memory (LSTM) units for shorter pieces of text like tweets. As will be discussed in the Results section, the version of the model that attempted to classify tweets into the 5 categories mentioned above performed quite poorly due primarily to the fact that while the model was very effective at categorizing very liberal and very conservative tweets, it really struggled with more moderate content. As a result, the final model consolidated the categories down to three with the labels "Liberal", "Moderate", and "Conservative". For more details on the performance of the model please see the results section.

# Results

As discussed in the methodology section, we ended up running two models, while the specifications for the model stayed the same, the big change was the number of categories used for categorization. The first model produced the confusion matrix by classification category below.

A picture containing bird

Description automatically generated

*Figure 9*. Model 1 Confusion Matrix.

The model performed very well at the extremes with the "Very Liberal" and "Very Conservative" groups, with an accuracy of almost 70% for the "Very Liberal" group. However, when trying to analyze more moderate content, the model really struggled accurately classifying only about 30% of tweets.

To fix this problem, I consolidated the "Very Liberal" and "Moderately Liberal" groups into a single "Liberal" bin, and the "Very Conservative" and "Moderately Conservative" categories into a single "Conservative" bin, and then kept the "Moderate" bin as is. This drastically improved the accuracy model, resulting in the confusion matrix below. While the model still struggled with accurately categorizing the "Moderate" category, it's improved performance with the "Liberal" and "Conservative" groups raised the overall accuracy of the model to about 70%, with the “Conservative” group categorized at approximately 90% accuracy. That being said, more work needs to be done trying to accurately categorize the more neutral content in the "Moderate Category", for which there are several strategies and tactics that could be applied.

A screenshot of a cell phone

Description automatically generated

*Figure 10*. Model 2 Confusion Matrix.

# Future Work

Models using Recurrent Neural Networks and word embeddings like the one we have presented in this paper are extremely effective at classifying political tweets at the margins achieving over 90% accuracy in some cases, however when it comes to classifying more moderate content to the model struggled mightily achieving no better than random accuracy. One reason for this is because of the size of the corpus used for the model, manually scoring the ideology of each tweet was impossible. The result of this was scores for tweets that accurately reflected the general political orientation of their authors but failed to detect the nuance of individual tweets. Given that members of Congress don’t only tweet about topics that are political in nature, some tweets were labeled partisan according to their author but in reality, were fairly neutral. In future work I would like to train the model on a dataset that was annotated at the individual tweet level or borrow methods from papers like Yano et al. (2010) where tweets were labeled for specific topics there were inherently political using a set of key words. This will allow on the model to pick up on more of the nuance of political discourse than it currently does.

Additionally, as I mentioned in the introduction one of this model’s potential applications is to be included in future bot detection and foreign interference initiatives where the manipulation of political decisions and beliefs is a core objective. Incorporating models like the one presented in this paper will allow for bot detection algorithms to more accurately differentiate complex bot networks with political social engineering objectives from smaller more harmless networks.

# Conclusion

Twitter, and other social networks like it, have become popular forums for political discourse. While these conversations are often legitimate and healthy, foreign actors like Russia’s Internet Research Agency have increasingly used them to influence elections and other political activity. Simply understanding what content is political in nature is an incredibly important starting in any attempt to find such networks. The model presented in this paper takes short form text like tweets and assigns a political label to it, achieving extremely high accuracy when identifying content at the margins (very liberal and very conservative) but more work is needed to accurately classify more moderate content. Borrowing methods from similar papers trained on political speech data will hopefully allow for increased accuracy in the future.

# Biography

**Noah Olsen** is a graduate student at The George Washington University studying Data Science. His interests lie at the intersection of natural language processing and network analysis, analyzing NLP and neural networks can better inform the analysis and discovery of complex networks. He has worked in the communications and advertising industries. In his free time, he enjoys running, skiing, and cooking.

References

Badawy, A., Ferrara, E., & Lerman, K. (2018, February 12). Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign. 258–265. <https://doi.org/10.1109/ASONAM.2018.8508646>

Chollet, F., & others. (2015). Keras. [https://keras.io](https://keras.io/).

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Transactions on Dependable and Secure Computing, 9(6), 811-824. doi:10.1109/TDSC.2012.75

Djemili, S., Longhi, J., Marinica, C., Kotzinos, D., & Sarfati, G. E. (2014, October). What does Twitter have to say about ideology?.

Doumit, S., & Minai, A. (2011). Online news media bias analysis using an LDA-NLP approach. In International Conference on Complex Systems.

Howard, P. N., Ganesh, B., Liotsiou, D., Kelly, J., & François, C. (2019). The IRA, social media and political polarization in the United States, 2012-2018.

Iyyer, M., Enns, P., Boyd-Graber, J., & Resnik, P. (2014, June). Political ideology detection using recursive neural networks. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1113-1122).

Johnson, K., & Goldwasser, D. (2016, November). Identifying stance by analyzing political discourse on twitter. In Proceedings of the First Workshop on NLP and Computational Social Science (pp. 66-75).

Kearney MW (2019). “rtweet: Collecting and analyzing Twitter data.” *Journal of Open Source Software*, 4(42), 1829. doi: [10.21105/joss.01829](https://doi.org/10.21105/joss.01829), R package version 0.7.0, <https://joss.theoj.org/papers/10.21105/joss.01829>.

Pennington J., Socher R., & Manning CD., (2014). GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

Thomas, M., Pang, B., & Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. arXiv preprint cs/0607062.

Wang, A., Foresti, S., & Jajodia, S. (2010). Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach. In Data and Applications Security and Privacy XXIV: 24th Annual IFIP WG 11.3 Working Conference, Rome, Italy, June 21-23, 2010. Proceedings (Vol. 6166, pp. 335–342). <https://doi.org/10.1007/978-3-642-13739-6_25>

Wang A., "Don't follow me: Spam detection in Twitter," 2010 International Conference on Security and Cryptography (SECRYPT), Athens, 2010, pp. 1-10.

Wickham et al., (2019). Welcome to the Tidyverse. Journal of Open Source Software, 4(43), 1686, <https://doi.org/10.21105/joss.01686>

Yano, T., Resnik, P., & Smith, (2010, June). Shedding (a thousand points of) light on biased language. In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (pp. 152-158). Association for Computational Linguistics.