Capturing Political Polarization of Reddit Submissions in the Trump Era

Virginia Morini[1][0000−0002−7692−8134], Laura Pollacci[2][0000−0001−9914−1943], and Giulio Rossetti[2][0000−0003−3373−1240]

[1] University of Pisa, Italy
virginiamorini95@gmail.com
[2] ISTI-CNR, Pisa, Italy
{laura.pollacci,giulio.rossetti}@isti.cnr.it

Discussion Paper

Abstract. The American political situation of the last years, combined with the incredible growth of Social Networks, led to the diffusion of political polarization's phenomenon online. Our work presents a model that attempts to measure the political polarization of Reddit submissions during the first half of Donald Trump's presidency. To do so, we design a text classification task: political polarization of submissions is assessed by quantifying those who align themselves with pro-Trump ideologies and vice versa. We build our ground truth by picking submissions from subreddits known to be strongly polarized. Then, for model selection, we use a Neural Network with word embeddings and Long Short Time Memory layer and, finally, we analyze how model performances change trying different hyper-parameters and types of embeddings.

Keywords: Political Polarization · Classification · Text Analysis.

1 Introduction

During the last decade, the rise of social networks has drastically changed how people interact and communicate. More than 45% of world population uses Social network for an average of 3 hours per day. As these platforms grow, the number of opinions shared publicly among users increases accordingly.

In this multifaceted panorama, particular attention turned on the diffusion of political discourse online and its implications. With the significant increase of user-Generated content, research in [3] underlines that sharing political beliefs and opinions online makes people feel more active, engaged, and interested. This attitude, combined with the contemporary political situation, leads to the spreading of political polarization. This phenomenon refers to the increasing gap between two different political ideologies.

During Donald Trump's presidency, online polarization has found its fertile ground: the debate between Trump supporters and anti-Trump citizens is becoming even more complex and uncivil [1].

In this paper, we propose a model that aims at measuring the political polarization of Reddit submissions in the Trump Era. To solve the issue, we model it as a text classification task. Given new submissions, we assess their political polarization by quantifying how those align themselves with pro-Trump ideologies and vice versa.

This task is a key step of a wider study, still ongoing, that attempts to identify and analyze Echo Chambers on Social Networks. Since an Echo Chamber is a strongly polarized environment, to assess its existence, it is necessary to dispose of a tool able to measure its possible components' polarization.

For our purposes, we choose Reddit as a subject of study. Reddit, as stated by its slogan *'The front page of the internet'*, is not a traditional social network but rather a website dedicated to the social forum and discussion threads. Founded in 2005, Reddit is now the nineteenth most visited website on the internet.[1] Also, Reddit is composed of thousands of communities, each dedicated to a specific topic, called *subreddits*. Its internal structure makes it easy to find politically polarized communities to analyze. Additionally, since users can write anonymously and posts aren't limited in length, this platform is particularly active in political discussions [11].

The rest of the paper is organized as follows. In Section 2 we discuss the literature involving text classification techniques in general as well as applied to the political domain. Section 3 describes the phases of data extraction and data preparation necessary to build our final dataset. In Section 4 we explore the different stages of model selection. Finally, Section 5 concludes the paper and set future research directions.

2 Related Works

With the growth of Social Networks, researchers have focused on the study of methods to extract information from a large quantity of unstructured text data.

The first key component of this process is text preprocessing, a combination of text mining techniques that allows cleaning data for further steps. Both works of Camacho-Collados et al. [4] and Uysal et al. [16], show how a weighted usage of these methods can remarkably improve final performances of a text classifier.

Another aspect to take into account when performing classification is the type of words' representation used. Social networks textual data, as posts, are sequences of words. Capturing their semantic valence is important to represent not only single words but also the context in which they are inserted. As stated in [8], traditional types of word representations, as a bag-of-word model, encode words as discrete symbols that cannot be directly compared to one another. Consequently, this approach is not suitable for the sentences' representation.

[1] https://www.alexa.com/topsites

Table 1. For each subreddit: number of submissions collected, average number of words in submissions and political ideology.

Subreddit	# submission	mean length	Political ideology
r/The_Donald	151,395	92.02	pro-Trump
r/Fuckthealtright	78,200	82.09	anti-Trump
r/EnoughTrumpSpam	73,168	79.34	anti-Trump

Word embedding (e.g., Word2vec [10] and Glove [12]), instead, maps words to a continuously-valued low dimensional space, capturing their semantic and syntactic features.

Recently has been shown how deep learning models can be fruitfully used to address Natural Language Processing related classification tasks. The survey in [9] synthesizes powerful models for sequence learning. The authors underline how Feed Forward Neural networks do not fit well with data sequences because after each word is processed, the entire state of the network is reset. Recurrent Neural Networks (RNNs), instead, assigning more weights to previous words of a sequence are suitable for this kind of data. In particular, Long Short Time Memory network (LSTM), a type of RNN, can maintain long term dependency through a complex gates mechanism, overcoming the vanishing gradient problem of RNN [15].

Social Network textual data are widely used for various text classification applications such as information filtering [7], personality traits recognition [13] or hate speech detection [17].

Concerning political leaning classification on such type of data, literature is not so exhaustive. In [5], Chang et al. propose a model to predict the political affiliation of Facebook posts in Taiwan. They build two models, one using the K-Nearest Neighbors algorithm and the other one AdaBoost combined with Naive Bayes classifier. Rao et al. [14] use word embeddings and LSTM to predict if a Twitter post belongs to Republican or Democratic beliefs. To the best of our knowledge, researches about political classification on Reddit is sparse or nonexistent.

3 Data Description and Preparation

In this section, we describe the phases of Data Extraction and Data Preparation necessary to build our final dataset.

Data was obtained through the Pushshift API [2] that offers aggregation endpoints to analyze Reddit activity from June 2005 to nowadays. We use the submission endpoint to pick submissions posted from January 2017 to May 2019, a period covering the first two years and a half of Donald Trump's presidency.

For this text classification problem, we select submissions belonging to subreddits known to be either pro-Trump or anti-Trump. Based on subreddits descriptions and considerations explained at the end of this section, we choose r/The_Donald for the first group and r/Fuckthealtright and

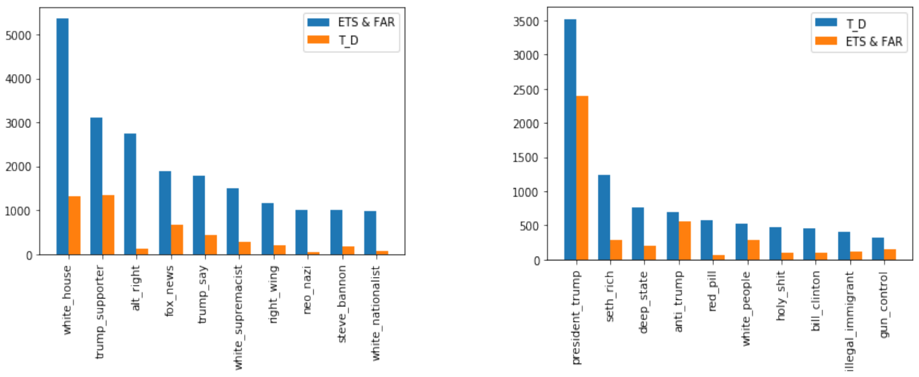

Fig. 1. Comparison between some of the 20 most frequent bigrams of each subreddit and their frequencies in the opposite one.

r/EnoughTrumpSpam for the second. To have a balanced dataset, for the anti-Trump data we merge two subreddits, strictly related both on the users and the keywords.[2] In Table 1 are illustrated some dataset statistics.

For each selected submission, we collect the fields *id*, *selftext*, and *title*, respectively, the identifier, the content, and the title of the submission. We merge the latter two fields in a unique one to give it as input for LSTM, because the *selftext* of a post may be empty or just a reference to the *title* itself. By doing so, we make sure to have a text capturing what the user is actually trying to convey. Then, we assign to each submission a label: 1 if it belongs to pro-Trump subreddit, 0 otherwise.

Due to the nature itself of the social network that promotes free speech and expression, textual data gathered from Reddit platform is dirty and noisy. For this reason, we apply a text preprocessing pipeline to give clean data to LSTM: in particular, we convert text to lowercase, then remove punctuation and number, as well as stop words.

Also, by looking at the submissions extracted, we observe that several of these are composed only by few words. This is probably because, during data extraction, we only select textual data, removing all multimedia contents related to each submission. To avoid affecting LSTM performance, we remove from our original dataset all submissions shorter than six words.

Lastly, to check the validity of our initial choice of polarized subreddits, we identify some of the most frequent bigrams of each subreddit to analyze their frequencies in the opposite one. As we can see in Fig. 1, these words are discriminant and semantically related to their belonging subreddit.

[2] https://subredditstats.com/r/EnoughTrumpSpam
https://subredditstats.com/r/Fuckthealtright

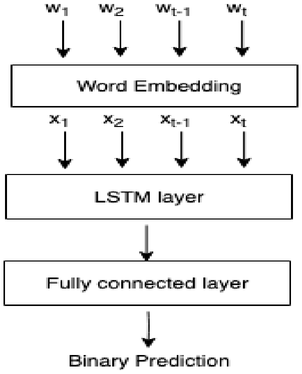

Fig. 2. High level architecture of our LSTM model.

4 Measuring Political Polarization

In this Section, we describe the model selected to measure the political polarization of Reddit submissions.

To assess the best model, we follow these subsequent steps: preprocessing of input text sequences, model selection, and testing on new instances.

Neural Network Preprocessing. To give our submissions as input for our model, we have to vectorize them, turning each one in a sequence of integers.

To do so, we create a lexicon index based on word frequency, where 0 represents padding: indexes are assigned in descending order of frequency. Each text is then transformed into a sequence of integers, replacing each word with the corresponding index of the lexicon (e.g., the sentence "president trump says" becomes [5,1,39]). We use the whole lexicon for training the model. Lastly, to optimize the matrix operations in the batch, we pad each sequence to the same length, based on the mean length of all submissions. Sequences longer than this length are truncated.

Neural Network Architecture. As stated in Section 2, since our training data consists of sequences, standard neural networks are not suitable for our task. For such a reason, we adopt a Long Short-Term Memory network [6], a type of Recurrent neural network able to model the meaning of a sentence by taking into account its paradigmatic structure. Fig. 2 shows the high level architecture of our model, composed of three layers:

1. *Embedding Layer:* This layer takes as input the sequences of integers w_1,..., w_t previously created and turns them into 100-dimensional dense vectors x_t. t is the length of the sequence. During model selection, we decide to see how performance changes based on the use of Glove pre-trained word embeddings or embeddings learned directly from the text.

Table 2. Hyper-parameters tuning for models with learned embeddings and Glove ones.

Emb. type	LSTM units	Training acc.	Validation acc.
Learned	32	0.832	0.816
	64	0.837	0.816
	128	0.828	0.816
Glove	32	0.831	0.814
	64	0.839	0.819
	128	**0.846**	**0.829**

2. *LSTM Layer:* This layer consists of multiple LSTM units. Each unit maintains a memory cell. Cells encode the information of the inputs observed up to that step through the gates mechanism. The first gate is called input gate and controls whether the memory cell is updated. The second gate is the forget gate that controls if the cell is zero; lastly, the output gate controls whether the information of the cell state is visible. To avoid over-fitting, we add a dropout regularization of 0.3.
3. *Output Layer:* This layer is a fully connected layer which outputs a single neuron to perform binary predictions. We use the Sigmoid activation function to have a probability output between 0 and 1. Submissions with a probability score greater or equal than 0.5 will be labeled as pro-Trump, the others as anti-Trump. As loss function, we use Binary cross-entropy and as optimizer Adam.

Model Selection. We use a training set of 302,763 instances, perfectly balanced among the two target classes. To find the model with the highest performance, we perform a 3-fold Cross-Validation trying different values of LSTM units [32,64,128] with a fixed embedding dimension of 100. We experience such configurations, both with learned embeddings and pre-trained ones. Results are shown in Table 2. The model with higher performance is the one obtained using 100 dimensions Glove pre-trained embeddings and 128 LSTM units that reaches an accuracy score of 84,63% on the training set and 82,96% on validation.

Model Evaluation. Model performances are firstly assessed on a polarized test set built by picking unseen submissions belonging to the previously selected subreddits.[3] In detail, we extract 38,906 submissions posted from 2nd May to 1st December 2019. As shown in the first line of Table 3, model reaches an accuracy of 74%.

We further assess model performances on less polarizing topics wrt Donald Trump's persona. To do so, we choose four main topics addressing sociopolitical issues, and for each of them, we select several related subreddits via Reddit lists.[4] Three out of four are fine-grained (i.e., gun control and legalization, minority discrimination, immigration policies) while the latter is more general in the topic (i.e., political discussion). Submissions extracted are posted from January 2017

[3] r/The_Donald, r/EnoughTrumpSpam, r/Fuckthealtright

[4] http://redditlist.com/
https://www.reddit.com/r/ListOfSubreddits/wiki/listofsubreddits

Table 3. Model performances on polarized topic and on less polarized ones.

Test set	# post	Accuracy	Precision	Recall	F1-score
Polarized test set	38,906	0.741	0.691	0.812	0.746
Gun control	2,247	0.712	0.751	0.692	0.720
Minority discrimination	955	0.723	0.780	0.731	0.754
Immigration policies	400	0.722	0.776	0.725	0.749
Political discussion	**80,961**	**0.825**	**0.943**	**0.851**	**0.894**

to December 2019.
In this scenario, we do not have a ground truth to evaluate model results among the different topics. Thus, we try a different approach: validating our model through polarized users. In detail, we first compute for each user in training set his *polarization score* by averaging his posts' prediction scores. Then we select the most polarized ones (i.e., score ≤ 0.2 for anti-Trump users and score ≥ 0.8 for pro-Trump ones), obtaining 1,150 final users. So, we look for their posts among the aforementioned four topics, labeling them according to the user *polarization score* previously obtained. Finally, we assess model performances on the four new test sets.

By comparing model performances across the five differently polarized topics, we can assess model's ability to generalize. As shown by Table 3, even though test sets are not directly comparable, due to their difference in size, our model reaches quite good results with accuracy always greater than 70%. Lastly, from the sample of predictions in Table 4 we denote similar trends across different topics: submissions with an higher prediction score convey a strongly Republican view while the lower ones a definitively Democratic leaning.

5 Conclusion and Future Works

In this study, we propose a model that aims at measuring the political polarization of Reddit submissions in the Trump Era. This task is a key step of a wider study focused on the identification of Echo Chambers on Reddit. Since an Echo Chamber is a strongly polarized environment, to assess its existence it is necessary to find a way to measure its possible components' polarization.

To do so, we use a Neural Network with word embeddings and Long Short Time Memory layer to quantify how submissions align themselves with Trump's ideologies and vice versa. The best model is the one built with Glove embeddings

Table 4. For each submission: text, topic and prediction score.

Sumission	**topic**	**score**
Tropical storm Barry: Obama has transformed his hatred for America into a new type of treason	Polarized topic	0.93
Trump's re-election crisis: His internal polls show it, national polls show it, and even a poll in reliably conservative Texas shows it.	Polarized topic	0.02
Never forget: Hillary worked hard to keep wages of Haitian garment workers under 31 cents per hour	Political discussion	0.93
American soldiers aren't dying for our freedom in Syria, Iraq and Afghanistan. They're dying for nothing	Gun control	0.15
Poor Immigrants Are The Least Likely Group To Use Welfare, Despite Trump's Claims missing	Immigration	0.03
Feminist deliberately acts like a condescending asshole to men. When they react like she's being an asshole, declares that men are an issue and Buzzfeed trends it.	Discrimination	0.90

that reaches 83% of accuracy on validation and 74% on the test set. Furthermore, we assess model performances also on less polarizing topics to evaluate model's ability to generalize. We achieve quite good results across four differently polarized topics with accuracy ranging from 72% to 82%.

As future research direction, we will leverage the proposed model to discover political Echo Chambers on Reddit: more specifically, analyzing their internal structure, their size and their persistence over time.

References

1. L. M. Bartels. Partisanship in the trump era. *The Journal of Politics*, 80(4):1483–1494, 2018.
2. J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn. The pushshift reddit dataset. *preprint arXiv:2001.08435*, 2020.
3. S. Boulianne. Social media use and participation: A meta-analysis of current research. *Information, communication & society*, 18(5):524–538, 2015.
4. J. Camacho-Collados and M. T. Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *preprint arXiv:1707.01780*, 2017.
5. C.-C. Chang, S.-I. Chiu, and K.-W. Hsu. Predicting political affiliation of posts on facebook. In *International Conference on Ubiquitous Information Management and Communication*, pages 1–8, 2017.
6. S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
7. M. S. Jones. The reddit self-post classification task (rspct): a highly multiclass dataset for text classification (preprint). 2018.
8. K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.
9. Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *preprint arXiv:1506.00019*, 2015.
10. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *preprint arXiv:1301.3781*, 2013.
11. R. Nithyanand, B. Schaffner, and P. Gill. Online political discourse in the trump era. *preprint arXiv:1711.05303*, 2017.
12. J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
13. B. Y. Pratama and R. Sarno. Personality classification based on twitter text using naive bayes, knn and svm. In *2015 International Conference on Data and Software Engineering (ICoDSE)*, pages 170–174. IEEE, 2015.
14. A. Rao and N. Spasojevic. Actionable and political text classification using word embeddings and lstm. *preprint arXiv:1607.02501*, 2016.
15. M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *International speech communication association*, 2012.
16. A. K. Uysal and S. Gunal. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112, 2014.
17. Z. Zhang, D. Robinson, and J. Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer, 2018.