Lakshmi S. Menon, Parthvi Sanjay Shah, Anusha R. Patil, and Muhammad Shujaat Mirza

Final Report for COVID19: War of Twitter Narratives

Introduction

We are interested in political discourse analysis on Twitter in the US with respect to the ongoing COVID-19 pandemic. The motivation is to build a model that can automatically distinguish between tweets expressing views on either side of the political spectrum, thus allowing the ability to assess narratives that harnessed more support among the users of the micro-blogging site. To build the training corpus, we utilize tweets made by the US Governors hailing from Democratic or Republican parties during the COVID-19 pandemic, allowing us an efficient way to identify and label tweets representing discourse pushed from either side of the political spectrum to deal with the same crisis. After experiments with multiple configurations such as SGD, LDA, biLSTM, etc., we achieved the best F1 score of 0.74 using logistic regression with TFIDF. We further perform data exploration using LDA and report themes that have dominated discussions during this pandemic.

Approach

Data Collection

We obtained the list of US governors' Twitter handles [5] and manually double-checked for its completeness by visiting each of the twitter handles and removing the outdated ones. We identified at least 6 accounts that were no longer active, mostly because the curated list did not include governors elected in the recent months. Our updated list of current governors contained 26 members of the Republican party and 24 members of the Democratic party. During the process of verification, we also identified that a majority of the governors reached out to their constituencies using two Twitter handles: a personal one and an official one. Given that most governors were either using both of these accounts regularly or preferring one over the other, we decided to include both of these for collection of relevant tweets. By the end of this process, we identified 94 active Twitter handles (46 Democratic and 48 Republican). For these Twitter handles, we collected tweets using the Twitter API [3] for the duration starting from Jan 21, 2020 to April 21, 2020. We decided it was appropriate to start collection of tweets from the day America's patient zero of COVID'19 was identified [1]. Overall, we managed to collect 29,210 tweets that were made in this duration. Table 1 details the breakdown of the collected tweet corpus based on party affiliation and account types.

	#governors	#handles	# of tweets collected official	personal
Democratic	24	46	11083	3042
Republican	26	48	8188	5720

Table 1. Detailed breakdown of the tweet corpus based on the party affiliation and twitter handle type.

Pre-processing and Vectorization

Raw tweets without pre-processing are highly unstructured and contain redundant information. To overcome these issues, we processed our tweets by taking multiple steps. We removed all digits, URL's, '@' mentions, punctuation, special symbols and stop words. We lower-cased tweets and performed tokenization. We further undertook lemmatization to better contextualize the words. Another important issue to deal with was the names and abbreviations of states and cities that could act as a leakage variable since, in the US, most states and cities are linked to a particular party. Given there are 25000+ cities in the US, we decided to remove names of all major cities with population of more than 50,000. Lastly, we added the corresponding label (Republican or Democrat) to the tweets based on their party affiliation.

Since hashtags are an important way of highlighting information on the platform, we extracted those and built a separate feature. Figure 1 shows a word cloud that provides insight into the 50 popular topics in the hashtags during this three month period. It is clear that pandemic related terms, such as 'covid19', 'flatten the curve', and 'social distancing' dominated the postings.

We have chosen TF-IDF as our vectorization technique with an n-gram range of (1,10) and maximum features of 3000. TF-IDF represents term frequency times inverse document frequency and it assigns weights to tokens in such a way that it scales down the impact of a word that occurs more frequently in a document, since it is likely to be less informative. n-grams is used for

Lakshmi S. Menon: lsm454; Responsible for submissions
Parthvi Sanjay Shah: pss434@nyu.edu
Anusha R. Patil: arp624@nyu.edu
Muhammad Shujaat Mirza: msm622@nyu.edu

Fig. 1. Word cloud on top 50 words used in the hashtags (excluding city/state names).

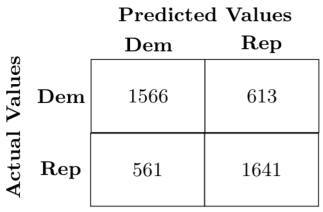

Fig. 2. Confusion Matrix of Test Set based on Logistic Regression with TFID.

developing features and hence helps with classification.

Selection of Baseline

Stochastic Gradient Descent (SGD) performs and generalises well on large datasets and it converges fast, reducing the training time. It is easily interpretable and highly efficient. Considering the size of the TF-IDF matrix, SGD seems like an appropriate model to experiment with. Hence, we fit our training data using SGD as the baseline model.

Models to Experiment

We decided to experiment with multiple models fed with different data exploration methods to understand which configuration works best for our task.

Logistic Regression is one of the most used algorithms when it comes to binary classification. It is a linear classifier with decision boundary of $<\Theta,x> = 0$. Logistic Regression works fairly well on text data and it offers great interpretability and transparency [7]. It also works really well on large documents and with balanced data, which seems like a perfect fit for our model.

Additionally, we used Latent Dirichlet Allocation (LDA) as a data exploration method to find latent topics within the tweets and to see whether there was a difference in the topics that Republicans and Democrats choose to tweet about. Although most tweets are COVID related, we wanted to find latent topics within this field such as business, healthcare, or preventive measures. We then used the topic distribution produced by LDA as features for Logistic Regression to improve the model's classification performance.

As an additional experiment, we also use GloVe embeddings with Recurrent Neural Network using the bidirectional Long Short Term Memory (BiLSTM) architecture to understand how it fares at our task of text classification.

Experiments

Dataset Splits

Data is split in the ratio 70-15-15 into train, validation and test sets respectively. Data is shuffled, making sure that all data points are present and that no indices overlap. The resulting set sizes were as follows: Training set (20448), Validation set (4381), Test set (4381).

Comparison between Models	Precision	Recall	F-score
Stochastic Gradient Descent	0.75	0.63	0.69
Logistic Regression	0.73	0.75	0.74
Logistic Regression using LDA	0.67	0.70	0.69

Table 2. Performance comparison between models on the test set (15% of the dataset). 'Stochastic Gradient Descent' acts as our baseline model.

Model Evaluation

`Stochastic Gradient Descent`: For our Baseline Model, we fit our training data using SGD (alpha = 10^{-3}) which gave us a fluctuating recall value each time we shuffled the data and trained it. Intuitively, alpha = 10^{-3} overshoots the minima most of the time, so we reduced our alpha value. After tuning the hyper-parameters, we achieved steady evaluation metrics with F1 Score of 0.69 at alpha equal to 10^{-6}. We later used this value for alpha to evaluate on the test set.

`Logistic Regression`: Logistic Regression with TF-IDF vectorizer and n-gram range (1,10) was performed on the data. Without any hyper-parameter tuning, F1-score was 0.7. After tuning the hyper-parameters, we observed, as we decrease the l1-ratio, the F1-score increases. Hence, finally we reduced l1-ratio to 0, penalty as l2-penalty, solver to lbgfs and C was increased to increase regularization to 0.96. With the hyper-tuned parameters our model performed fairly well, giving us an F1-score of 0.74 on the test set. Later, we calculated the confusion matrix as seen in Figure 2. Here, the false positives and false negatives have nearly the same numbers. This suggests that there are some tweets the models can't differentiate well, as they perhaps belong to the topic both the parties equally emphasize on. We later explored this using the output from our LDA model, incorporating it with the tweets dataset and fitting it using Logistic Regression.

`Logistic Regression with LDA`: Using our tweets along with LDA and fitting with Logistic Regression did not reduce the number of false positives and negatives as we had expected. However, we considered it would help to solve fluctuations caused in the evaluation metrics using SGD, which quite certainly worked. It combines the tweets along with the probability of that tweet belonging to a particular topic. After combining the text data with topics generated by LDA, we obtained a steady recall and precision rate. Logistic Regression could distinguish the topic emphasized by each party which increased the robustness of the model. By hyper-tuning the parameters, increasing the regularization parameter 'C' from 0.1 to 0.96, there was an increase in F1-score by 0.2.

`BiLSTM Classifier` As an additional experiment, we have implemented another common model for text classification: BiLSTM. A BiLSTM uses two LSTMs to learn each token of the sequence based on both the past and the future context of the token. [4] Using GloVe embeddings, we created 300-dimensional word vectors. After minimal data preprocessing, we proceeded to the next step of tokenization using Moses Tokenizer from Sacremoses. [2] We performed hyper-parameter tuning on *batch_size* and *sequence length* and trained for five epochs, the results of which are shown in Table 3. The training loss history can be seen in Fig. 7 in the Appendix. We get a maximum accuracy of 0.646 on our validation set.

This indicates that for our problem statement, the BiLSTM model is not able to efficiently classify the tweets into Democratic and Republican parties as all the other models outperform this one. This could be because LSTM generally performs better when there are many more features.

BiLSTM: Hyper-parameter tuning

batch_size	seq_length	accuracy
32	128	0.637
64	256	0.644
128	256	0.646

Table 3. Best accuracy results for each batch size

Topic Modelling with LDA

To better understand the distribution of topics in our dataset, we built an LDA model using the *gensim.models.lda* python package [11]. We trained the model on both a Bag-of-Words and a TF-IDF dictionary [13] and saw better coherence overall on the TF-IDF model, so we proceeded with this dictionary. In order to choose the optimal number of topics to be trained by the LDA model, we trained multiple models with different values for the parameter *num_topics* within the range 5 to 100 with a step size of 5 and plotted the coherence for each plot [12]. This plot can be seen in Fig. 3.

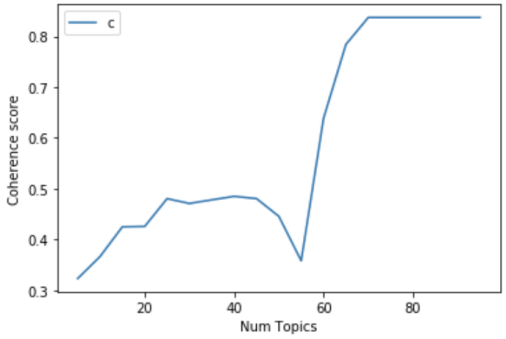

Fig. 3. Plot of coherence against number of topics.

The sharp increase in coherence after 65 topics is due to overlapping topics, which results in multiple topics having the same representative terms. To pick the value that balances high coherence with minimal overlap, we selected 25 topics. Our final LDA model was thus a TF-IDF model that determines 25 topics. After training the model, we created an interactive visualization of the topics using the *pyLDAvis* package [8] [14]. A snapshot of this visualization is shown in Fig. 4 below, which displays the intertopic distance along the top two principal components. Each circle represents a topic.

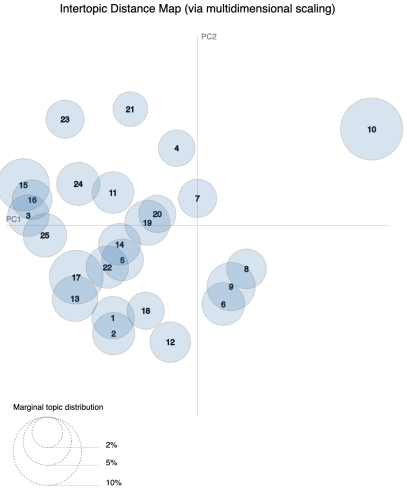

Fig. 4. Visualization of the intertopic distance

We then got the topic distribution for each of the 29,210 tweets, which gives the probability of each tweet belonging to each topic. We later incorporated this topic distribution into the logistic regression, where the probabilities were used as added features along with the text and hashtags of the tweet. Using these values, we took the highest probability for each tweet to be its 'Most

Likely Topic'. We then plotted the proportion of tweets, per class, that had each topic as its most likely topic. This can show whether any topics are seen significantly more or less in a particular class. The plot of select topics is shown in Fig. 5. The full distribution can be seen in Fig. 6 in the Appendix.

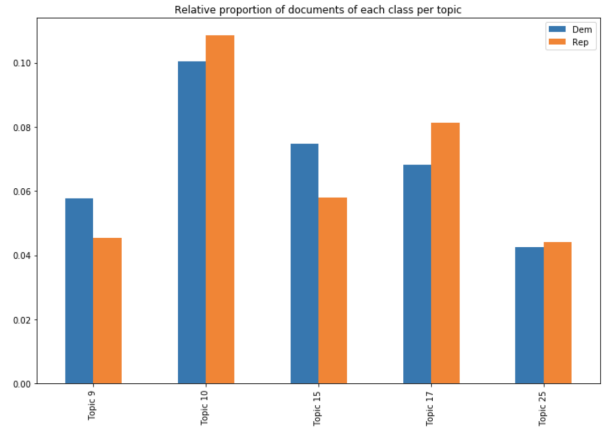

Fig. 5. Distribution of 'Most Likely Topic' per class

This distribution shows that while most topics are roughly equal between the two parties, there are certain topics that are more commonly used in one party over the other. For example, topics 10 and 17 are seen more in Republican tweets, while topics 9 and 15 are seen more in Democrat tweets. These differences are discussed in the next section.

Discussion

Explanation of Findings

The LDA topic model shows that there are some differences between the topics covered by Republicans and Democrats in their tweets. As mentioned before, topics 10 and 17 were seen as the most likely topic for a higher proportion of Republican tweets. These topics include representative terms such as 'live', 'watch', 'tune', 'executive', 'order', 'signed', and 'issued'. The topics which were more present in Democratic tweets, included terms such as 'stay', 'home', 'social', 'distancing', 'keep', 'safe', in topic 9, as well as 'case', 'positive', 'total', 'confirmed', and 'tested' in topic 15. This suggests that the Republican tweets focused more on encouraging citizens to watch press briefings for updates, as well as on highlighting measures taken by the governor, such as signing and issuing executive orders. On the other hand, the Democrat tweets focus more on the citizens' well being, using phrases such as 'keep safe', as well as by reminding them to stay home. Topic 15 also suggests that the Democratic Party governors provide updates about new confirmed cases and testing directly in their tweets, whereas the Republican governors encourage citizens to tune in for this information. Another interesting finding is that topic 25, which is nearly equal between the two parties, includes words such as 'thank', 'save', 'life', 'serve', 'asking', and 'increase', 'funding'. This could represent the universal gratitude towards healthcare workers, as well as a pervasive overburdening of the healthcare system, leading governors to ask the national government for increased support and financial assistance.

Error analysis

We performed human evaluation on the misclassification of parties in the test set. The classifier could not correctly predict tweets where they were closely related to the one of the main topics used by the other party, as discovered using LDA. For instance, 'live state briefing today pm press join say watch home' was tweeted by a Democratic Party Governor but was predicted as belonging to a Republican Party Governor, mainly because of the words 'briefing', 'watch', and 'live', which are the main topics of Republican class. Similarly, 'today state spread need thank update yesterday health home help' and 'health state business know stay make home care national update' were predicted as belonging to Democratic Party because of the words 'home', 'health', 'care' which belong to the main topics of Democratic class.

Comparison with Related Work

For our approaches, we drew inspiration from several previous publications that showed promising results in the field of text classification. Since TF-IDF is the most preferred weighting method, Prasetijo et al. [10] compare performance of different classification algorithms when coupled with TF-IDF. They find support for the use of SGD for this task, which further convinced us to use it as our baseline model. We decided to implement Logistic Regression after we found support for its use for text classification [6]. The empirical analysis of LDA-based topic modelling in text classification conducted in this paper [9] inspired us to build a logistic regression model that incorporates LDA to test its performance against simople logistic regression.

Possible Next Steps

For future work, other models, such as support vector machines for tweet classification can be experimented to check their impact on accuracy metrics. For application of the model, one could build an app that tracks, in real-time, the political sentiments on Twitter in response to real-world events such as the White House briefings.

References

[1] First patient with wuhan coronavirus is identified in the u.s. [Online]. Available: https://www.nytimes.com/2020/01/21/health/cdc-coronavirus.html

[2] Nltk documentation. [Online]. Available: https://www.nltk.org/_modules/nltk/tokenize/moses.html

[3] Twitter developer api. [Online]. Available: https://developer.twitter.com/en/docs

[4] Understanding lstm networks. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[5] CivilServiceUSA, "Contact information of US governors," Jan. 2020, https://civilserviceusa.github.io/us-governors/, as of May 19, 2020.

[6] B. Gebre, M. Zampieri, P. Wittenburg, and T. Heskes, "Improving native language identification with tf-idf weighting," 2013. [Online]. Available: https://pure.mpg.de/rest/items/item_1740046_12/component/file_1867917/content

[7] S. Liu, "Sentiment analysis of yelp reviews: A comparison of techniques and models," 2020.

[8] B. Mabey. pyldavis documentation. [Online]. Available: https://pyldavis.readthedocs.io/en/latest/readme.html

[9] A. Onan, S. Korukoglu, and H. Bulut, "Lda-based topic modelling in text sentiment classification: An empirical analysis," 2016. [Online]. Available: https://www.gelbukh.com/ijcla/2016-1/IJCLA-2016-1-pp-101-119-preprint.pdf

[10] A. B. Prasetijo, Semarang, R. Rizal, and D. Eridani, "Hoax detection system on indonesian news sites based on text classification using svm and sgd," 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8257673

[11] R. Rehurek. Gensim lda documentation. [Online]. Available: https://radimrehurek.com/gensim/models/ldamodel.html

[12] Selva Prabhakaran. Topic Modelling with Gensim (Python). [Online]. Available: https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

[13] Susan Li. Topic Modelling and Latent Dirichlet Allocation (LDA) in Python. [Online]. Available: https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

[14] Susan Li. Topic Modelling in Python with NLTK and Gensim. [Online]. Available: https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21

Appendix

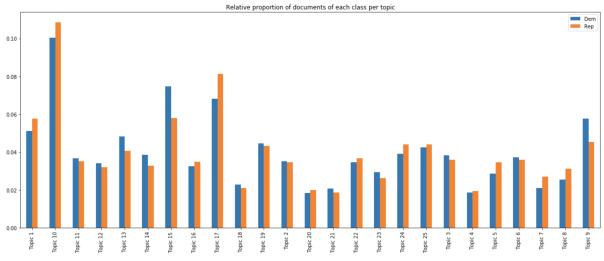

Fig. 6. Distribution of 'Most Likely Topic' per class

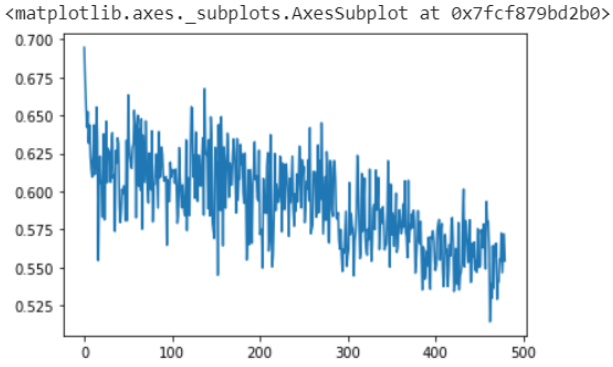

Fig. 7. Training loss history for BiLSTM classifier