Final Project Report - Future Spotify Engineers
CS5304 - Data Science in the Wild
05/06/2020

Tom Shanahan - tks46
Bailey Wei - bw489
Anirudh Shah - as3947
Howey Qiu - hhq3
Frans Fourie - fjf46

Abstract

In this report, we address the problem of 'fake news' on Twitter by training a model that can identify 'troll tweets'. The model is trained using a combination of known 'troll' tweets and control tweets surrounding political events from 2015 to 2018. The model presented combines Google's BERT semantic embedding tool that computes the sentiment of tweets from word embeddings, and binary classification models, which use the sentiment score from BERT with other features to indicate a 'troll' or 'not troll' tweet. The model was able to obtain a 99.81% classification accuracy on our test set, and BERT itself reached 95.4% accuracy, which indicates that there is a clear semantic difference between non 'troll' and 'troll' tweets. With these results we determined that it is possible to retrospectively identify 'troll' tweets that have been made using sentiment analysis, although we also note some potential pitfalls of applying this model on live tweets.

Background

Misinformation and disinformation related to elections are a huge challenge for democracies. We are attempting to tackle the problem of 'fake news' by identifying troll accounts on Twitter. Trolls are malicious users who deliberately bait other users through inflammatory messages for the purpose of eliciting an emotional response. Often, this includes spreading confusion and untrue information. The aims of trolls vary greatly from personal entertainment to geopolitical strategy. During the 2016 U.S. presidential election, entities affiliated with Russian intelligence agencies made extensive use of social media in attempting to influence the outcome of the election.

The aim of our project is to build a tool that can identify troll accounts by building a classifier trained on a combined dataset of Russian troll tweets, from accounts identified by the U.S. House intelligence Committee, and non-troll political tweets from the same time period.

Related Works

Badawy et Al. used natural language processing techniques on tweet text to attempt to identify the ideology of the tweets as well as bot detection techniques to determine if a tweet was sent by a real person or bot.[1] The model architecture used by Efthimion et al. was able to achieve 97.75% accuracy on this task taking inputs such as length of user names, temporal patterns and sentiment expression[2]. Chun et al. showed that using Google's BERT natural language processing technique provided the best results for this task[3] with higher results than Word2Vec, support vector machines and neural network models, achieving an accuracy of 99% on this task.

[1] A. Badawy, E. Ferrara and K. Lerman, "Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, 2018, pp. 258-265.

[2] Efthimion, Phillip George; Payne, Scott; and Proferes, Nicholas (2018) "Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots," *SMU Data Science Review*: Vol. 1 : No. 2 , Article 5. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss2/5

[3] Chun, Soon Ae et al. "Detecting Political Bias Trolls in Twitter Data." WEBIST (2019).

Data Sets

Russian Troll Tweets

We make use of a set of Russian troll tweets that were collected from Twitter handles that were connected to the "Internet Research Agency", a Russian troll factory. The database of tweets are from the period between February 2012 and May 2018 and were collected by Clemson University in 2018 using the Social Studio tool. The researchers used this data to write a paper on the methods employed by Russia's Internet Research Agency to influence politics in the United States. This dataset consists of 2,973,371 tweets from 2,848 Twitter accounts, along with the Twitter API data corresponding with the post.

Control Data

As a result of computation time issues, we decided to use a dataset of tweets collected by a team from Harvard University[4]. The dataset we identified contained over 170 million tweets related to the 2018 U.S. Congressional Election. These tweets were collected between January 22, 2018 and January 3, 2019 using a variety of election related hashtags.[5] Twitter's terms of service only allow the publication of datasets of tweet identification codes, not the tweets themselves. To reconstitute these tweets from their codes, we used the Hydrator tool developed by DocNow[6]. The use of an existing dataset of tweets allowed us to increase our timespan of study and to include a major national election. However, there is a disadvantage. Twitter does not allow a deleted tweet to be recovered. Additionally, if a tweet is retweeted and the original is later deleted, the entire chain from tweet to retweet is lost. Our preliminary analysis of a limited selection of tweets showed this loss rate to be around 10%. Our supposition is that much of this stems from a small number of controversial tweets that were retweeted a large number of times but later deleted.

Exploratory Analysis

Before beginning our model development, we did some exploratory analysis on the troll dataset to better inform our feature selection, and potential traps that could cause our model to train in an unrealistic way. From Figure 1, it is interesting to note that the majority of the accounts have few followers, followed accounts, and updates when compared to other influential accounts. From Figure 2, the interesting observation to note is that the majority of the tweets supposedly originated from the United States, but there were tweets from around the world so it is difficult to use location as a predictor. Figures 3 and 4 show different publish-date breakdowns of the tweets in the troll dataset. We found that there isn't any particular seasonality or regular grouping to the number of tweets per day, but that there tended to be large spikes of tweets on days surrounding hot-button political events, especially the 2016 election, and the week of August 12-18 2017. There also seems to be a general increase in day-to-day troll tweets after the 2016 election. From this we gathered that the troll

---

[4] Wrubel, Laura; Littman, Justin; Kerchner, Dan, 2019, "2018 U.S. Congressional Election Tweet Ids",https://doi.org/10.7910/DVN/AEZPLU, Harvard

[5] Hashtags followed by Harvard: #Nov2018, #Election2018, #Nov18, #Election18, #Midterms2018, #Midterms18, #Midterm2018, #Midterm18, #midtermelection, #election, #vote, 2018 election, election 2018, midterm election, #BeAVoter, #IVoted - Active

[6] DocNow and their Hydrator are a collaboration between Shift Design, Inc., the University of Maryland, and the University of Virginia.

bots tend to be reactive to hot-topic events, and oftentimes on specific days. Therefore, we could judge that publish date, followers, and updates, may be good variables to consider for our model.

Methodology

We combined two Machine Learning methods to maximize prediction power over the data available. First we used Google's pretrained BERT basic cased model, fine-tuned to our dataset, to analyze the tweets and create sentence embeddings and then passed these embeddings through a softmax layer to generate a probability of being troll for each tweet. This, along with other features pulled from the Twitter API data we trained a binary classification model to output our final prediction of whether a tweet was a troll tweet or not. The binary classification models we tested were logistic regression, logistic regression with normalized input features, and support vector machine with a radial-basis function kernel (SVM-RBF).

To prepare the data for the model, we first had to preprocess and tokenize the tweet for BERT, and also had to do some feature selection and manipulation for the binary classification. For BERT, at the start and end of each tweet we had to add a [CLS] and [SEP] token respectively, then add padding so all the tweets were the same length. We then used the BERT tokenization format to split the tweet into individual words and handle unknowns. Then finally we had to add attention masks to identify the padding of tweets. Due to mismatches in features between the control and troll datasets we could only use 4 additional features: publish date, retweet ({0,1}), account followers (integer>0), and total account posts (integer>0). To turn publish date into a ratio input variable, we decided to transform all dates to the form new_date = old_date - min(all_dates), which gives the number of days since the first tweet in both datasets.

We trained our BERT embeddings on a 200k tweet train set and a 30k tweet test set, checking validation set accuracy and test set accuracy. We then assigned scores to the 30k test set, and split into a 20k train and 10k holdout set for the binary classification model. The decision to feed the BERT test set into the binary classification set was to not double-train on the data points already used for the initial embedding training, and our sets were fairly small compared to the full dataset for computing issues, although after collecting results we re-trained the model on new sampled sets and achieved similar accuracy.

Results

After training the BERT model, we achieved an initial 96% accuracy on the validation set and a 95.4% accuracy on the test set, which is near the standard for other models using BERT for binary sentiment embedding. Then, after adding the additional features to the sentiment scores, we achieved a 97.04% test accuracy using logistic regression. On top of this, after standardizing the publish date, followers, and total updates features, we re-trained the logistic regression model and got a test accuracy of 99.74%. Finally, we wanted to try a non-linear classification model so we trained a support vector machine with RBF kernel, and got a final test accuracy of 99.81%.

While we were unable to get the significance of individual features in the models in sklearn, a cursory glance at the coefficients tells us that the sentiment score had the largest influence on the tweet being classified as troll or not, which is also backed up by the high accuracy achieved right after training BERT. Publish date seemed to have the second highest effect on prediction, although this may be attributable to the different range of dates the control and troll tweets were pulled from. Publish date also seemed to be the feature that

benefited most from normalization. With the normalized logistic regression model, the false positive and false negative rates were 0.15 and 0.11% respectively, and in the un-normalized version these rates increased by a factor of 10.

Discussion

We achieved high accuracy with an out-of-the-box BERT-based model. This indicates that troll and non-troll tweets have a very clear semantic difference. Reviewing the false positives and false negatives of our model for both of these tweet types revealed to us that certain political tweets were common mistakes our model made. For example, extremely leftist or rightist tweets were classified wrongly. From this, we can conclude that our model is detecting instances of political vocabulary well. Furthermore, our researchers have also reached a comparable accuracy score of 99%[7]. This could also further indicate our model is not out of the ordinary.

Our model had high computation times for training and evaluation. Just training on 200k tweets took approximately 1 hour per epoch, and in terms of inference, predicting on 20k tweets took 4 minutes. These statistics are using the Google Colab GPU, and indicate the difficulties of applying this model in the wild where the amounts of data are vast compared to our investigation.

For our dataset, we were not able to fully scrape current Twitter data, due to issues with the API and a Twitter development account. Thus, we were forced to utilize already curated datasets for tweets. Given a newer dataset from troll and non-troll accounts could potentially give us different results. However, we tested our model with a select few hand-pulled tweets and achieve similar results.

Lastly, BERT models have a tendency to be both unstable and prone to quick convergence. Our high accuracy could potentially have been a result of this. However, we ensured to have several different iterations of BERT and different GPUs to avoid this.

Conclusion

Given adequate training data and enough processing power it is possible to train a model that can identify ‘troll’ tweets with high accuracy. The current version of the model makes use of BERT sentiment embedding and an SVM binary classifier, which allows the model to achieve an accuracy of over 99%. The model accuracy and performance could be made even more robust or trained quicker by making use of a newer and more sophisticated version of BERT, such as AlBERT, RoBERTa, or DistillBERT. The problem with implementing a model such as this on actual tweets is the sheer volume of tweets generated every second. There are currently roughly 6000 tweets made per second and 500 million tweets per day. This massive volume of tweets that would all need to be checked and evaluated by the model would require an immense investment in processing power solely dedicated to running this algorithm attempting to identify ‘troll’ tweets, not to mention the expense of regularly re-training to learn new terminology surrounding current events. An additional weakness of a model like this is that if it becomes known that the model is based on an algorithm such as BERT, it could be exploited and bypassed. When Google announced BERT people were quickly able to create BERT optimization articles, thus if an advanced trolling agency such as the “Internet Research Agency” learned this they might be able to find a way to formulate tweets in order to not be detected by the model.

---

[7] Chun, Soon Ae et al. “Detecting Political Bias Trolls in Twitter Data.” WEBIST (2019)

Figure Appendix:

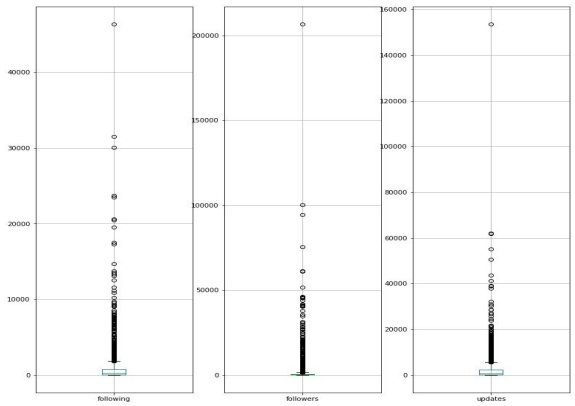

Figure 1 - Boxplot of number of followers, number following and number of updates for each Russian troll Twitter account

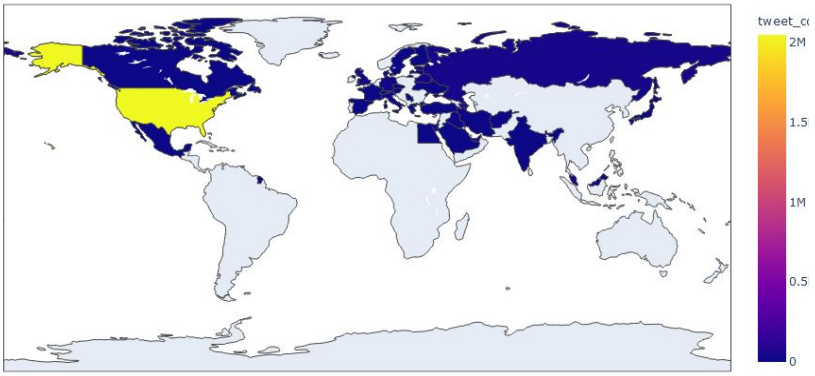

Figure 2 - Map showing where each tweet originated from

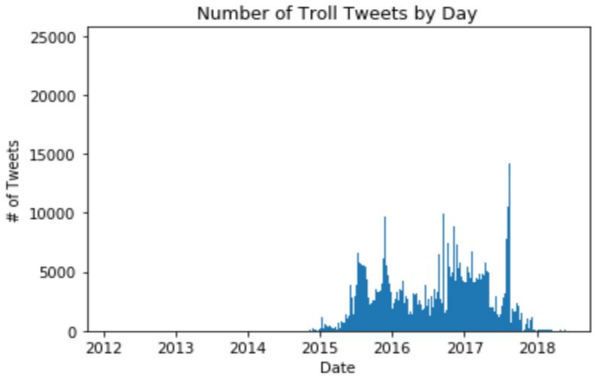

Figure 3 - Distribution of tweets over time

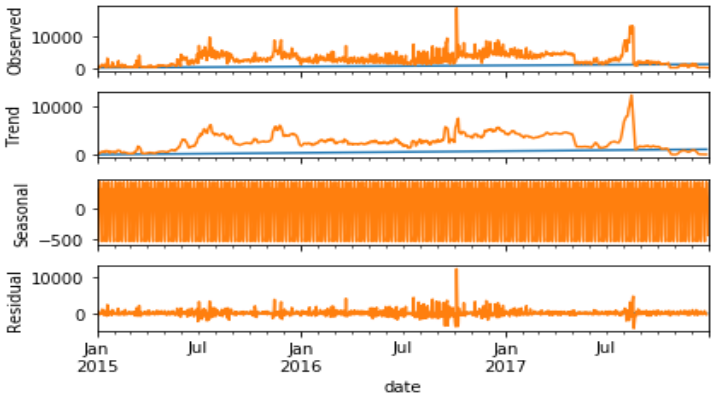

Figure 4 - Seasonal decomposition of the number of tweets per day