Detecting Fake Tweet Activity based on NLP and Sentiment Analysis

Final Project Report

TCSS-554 : Information Retrieval

J. McFadden, E. Han, K. Chandrasekaran,
S. Ananthapadmanabha, K. R. Kalyanam

1 Problem Description

Our proposal is to analyze Twitter data to detect fake/bot accounts and fake retweet activity by any account. Specifically, our first goal is to be able to determine if a given account is a fake/bot account. If time permits we would like to extend this by quantifying the probability of a given account being fake as a percentage. This will be followed, and complimented, by our second goal, detecting fake retweet activity. Should time permit, we will again seek to quantify the probability that a given retweet is fake, in other words automated, activity. This problem is important due to the large quantity of false and misleading information circulating on social media; and solving even part of it would help to improve the dialogue that all free and open societies must have.

2 Proposed Solution

Our solution aims to provide an efficient method of determining if a given post is generated by an actual human or an AI bot. Since even sophisticated AI bots have difficulty expressing emotions, we are using this weakness to help hunt for them. We hypothesize that the manner in which bots express emotions in text should be markedly different from the manner in which humans express emotions. Therefore our model seeks to quantify the emotions expressed in the text of a given tweet so that various machine-learning algorithms can be trained to detect the

difference between human and bot generated Tweets. If time permits, we would like to extend this model so that it provides a probability that a given tweet is real or fake.

We quantify the emotions of a given tweet using the NRC and Afinn natural language processing and sentiment analysis frameworks. The NRC framework consists of ten emotions with each word in the English language being associated with one of those ten emotions. This was used to quantify emotions in a given tweet by determining the frequency with which emotions occurred in each tweet. The other framework used, Afinn, provided an overall score for the emotion of each tweet based on the words in the tweet. Each word in a given tweet was scored according to the Afinn framework, then these scores were all added for that tweet to determine its Afinn score. A negative Afinn score would represent a negative tweet, while a positive Afinn score would represent a positive tweet.

This model is then used to produce what we are calling the *Tweet-Emotion Matrix*. This matrix has thirteen columns. The first column contains the ID numbers of the tweets. The next ten columns contain the scores for the ten emotions in the NRC framework. The twelfth column contains the score from the Afinn framework; and the thirteenth column contains a binary classifier which classifies tweets as their *real* (1) or *fake* (0). Finally, the rows of this matrix represent individual tweets, while the elements of the matrix represent the strength of a given NRC emotion, the Afinn score, or the real/fake classifier for the tweet represented by the row. Ultimately, this *Tweet-Emotion Matrix* will be run through various machine learning algorithms.

Running the data from the *Tweet-Emotion Matrix* through machine-learning algorithms serves two purposes. The first purpose is to determine if there is any statistically significant correlation between the emotional data in the matrix and the classification of a tweet as real or fake, as well as quantify any statistical significance based on both the t-test and a ten-fold cross validation for accuracy. The second reason for running the data from the *Tweet-Emotion Matrix* through various machine learning algorithms is to create a classifier to analyze tweets in real-time. This real-time classifier would only be created if the accuracy of the various machine learning methods was high enough.

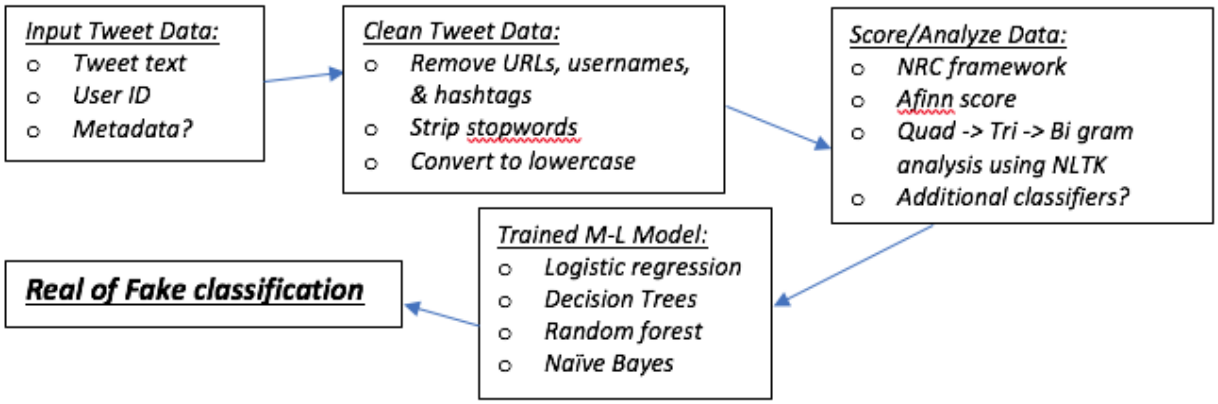
3 Dataset and Features

Our primary datasets come from the Fake Project, which used link analysis and feature selection to determine if a given Twitter user is real or fake. The data contains approximately two million fake tweets and two million real tweets, along with all metadata associated with those tweets. This metadata includes, but is not limited to, date of

creation, user ID of author, text of tweet, number of likes, geolocation data, and number of retweets. Additionally, the data provides a listing of all users whose tweets are included in the data. Again, this user data contains metadata about each user, including but not limited to, user geo-data, account creation date, number of followers, number of following, users followed, users following, and aggregate activity data.

Datasets from the Fake Project were divided into one of four categories: fake followers, genuine accounts, social spam bots, and traditional spam bots. The highly curated nature of this dataset is the reason we chose to exclusively use it to train our machine learning algorithms. The other dataset we have comes from *CarverLee, Cheng, and Lee*. This second dataset is not as well curated as the ones from the Fake Project, therefore we only used the second dataset for testing our train machine learning models.

4 Process Flow/Architecture



5 Result Analysis

Interestingly, a simple statistical analysis of the scores data contained in the *Tweet-Emotion Matrix* provided useful insights. The scores data for the fake Tweets is very sparse, while the score data for the real Tweets is considerably less sparse. The sparsity of the data provides some information about the emotions expressed, namely the number of different emotions being expressed simultaneously. That is to say, the greater the sparsity of the data, the fewer emotions being expressed. Given the previously described difference in the sparsity between our real and fake scored data, this appears to support our hypothesis. Additionally, the 2nd p-Norm and the mean row norm for the *Tweet-Emotion Matrix* of fake tweets were both significantly lower than for the *Tweet-Emotion Matrix* of real tweets. That said, the ultimate test is running our trained M-L models on test data randomly sampled from the

overall dataset.

We chose to run four machine learning algorithms on our data. These algorithms were

- Decision Tree
- Binomial Logistic Regression
- Random Forest
- Naive Bayes

From the Decision Tree algorithm, we determined that positive emotions were strong factors in determining if a tweet was real or fake. Interestingly, this separation applied to fake tweet generated by software bots as well as those generated by humans acting surreptitiously. This was an unexpected, but welcome outcome. Furthermore, this outcome (with respect to positive emotions) was supported by the results from both the binomial logistic regression and random forest models.

Running a binomial logistic regression on the data produced results similar to those of the decision tree. Moreover, the regression generated showed that all variables were statistically significant (using the t-Test), with positive emotions being considerably more significant than negative ones. Furthermore, the binomial logistic regression was used to create a predictor based on a randomly sampled subset of data with a 50% baseline. Training the binomial logistic regression with 10,000 data points provided a 54.5 % accuracy against fake humans and a 67.5 % accuracy against software bots. Interestingly, as more data was added to the train sets, the accuracy of the model began increase (50,000 and 500,000 data points) before decreasing (1 million data points), which likely indicates an over-fit. The max accuracy we obtained with a binomial logistic regression was 57.9 % against fake humans and 71.1 % against software bots.

Running a random forrest on the model and dataset provided similar results to the binomial logistic regression and decision tree methods. Interestingly, the random forrest produced better accuracy than the logistic regression, despite showing a similar relation between the size of the training set and accuracy. For a training set with 10,000 data points and a 50% baseline, the random forest provided a 54.92% accuracy against fake humans and a 73.41% accuracy against software bots. The increase in accuracy with increasing training set size behaved the same way as with the binomial logistic regression with accuracy topping out at 60.3% against fake humans and 83.5% against

software bots (based on a training set with 500,000 data points).

The final machine learning algorithm we ran was a Naive Bayes learner. This algorithm has the worst performance out of all the machine learning algorithms tested. With a training set of 50,000 data points and a 50% baseline, accuracy was 51.3% against fake humans and 53.7% against software bots. Surprisingly, adding more data to the training set did not affect accuracy like the other algorithms. This lack of increase accuracy suggests that a multinomial distribution is at work here and, by extension, that a multinomial naive bayes learner might be more appropriate and more accurate.

6 Conclusions

In addition to not running a Multinomial Naive Bayes learner, we also did not apply Neural Networks, Support Vector Machines, or Stochastic Gradient Descents. This was due to the limited timeframe of the project and limited computational resources available. We believe that running these additional machine learning algorithms against our datasets and models may provide a slight increase in accuracy over the algorithms we *did* apply. Since this is a simple binary classification, this expected increase in accuracy should be especially true in the case of a neural network. All that said, the unambiguous conclusion is that our model works, as desired, against software bots. Moreover, the model also works against fake humans, albeit with reduced accuracy.

Based on the fact that the model works against fake humans, our other conclusion is that, if the model could be expanded, its accuracy against fake humans might increase. Specifically, we suggests expanding the model in two ways. First, we could build on the NRC framework and add more emotions to obtain a more granular model which should provide additional bias for distinguishing fake humans from real ones, as well as humans from different countries and cultures. Second, we could also incorporate other NLP and sentiment analysis frameworks into our model, however, unless the provide more granularity than the frameworks our model is currently based on, we hypothesize that they are unlikely to significantly increase accuracy. Finally, based on the difference in emotional content between real humans and fake humans, we also hypothesize that humans express emotions differently between various countries and cultures. Unfortunately, we do not believe that this hypothesis can be properly tested with significantly increasing the granularity of our model as well as obtaining additional data. That said, testing this hypothesis and increasing the model's granularity could prove of interest for future work.