CMPS 287 - Final Report

Rita Tohme, 201807842
Ali Fayad, 201802049

May 12^{th} 2020

1 Abstract

The aim of this project is the detection of Social Media Trolls. A troll is a person who makes a deliberately offensive or provocative online post. In other words, a troll is a person whose main objective is to start quarrels, spread fake information, and provoke people by sending offensive messages. General wely, trolls tend to preserve an anonymous identity so they could feel at ease on social media networks.
In order to maintain the authenticity and quality of the information displayed online and create a friendlier social media experience, it has become necessary to monitor the content posted, and identify the trolls to minimize the level of damage against users (bullying, harassment...), and lessen the amount of fake information spread.
In our project, we will be looking forward to detect the probability of a tweet being posted by a troll, since Twitter is one of the most famous platforms where people interact and share their thoughts towards a certain subject.

2 Introduction

Social media, due to some factors like dissociative anonymity (i.e. online actions not linked to real identity), lack of consequences, and widespread of controversial social opinions, has provided trolls with an increasingly attractive platform where they can express themselves in ways that are unacceptable, otherwise, in society. In this project we aim to classify accurately whether a tweet was posted by a troll or not, using data of confirmed Russian trolls and normal tweets posted by genuine users.

3 Dataset

In order to obtain accurate results, our goal was to gather a large dataset for both the trolls related tweets and the authentic ones. Therefore, we relied on two main resources to get the needed data. The whole dataset is made up of

approximately 4.6 million tweets, 65% of them being 'trolls tweets' and 35% 'normal tweets'.
The dataset contains data on nearly 3 million tweets (2,973,371 tweets to be exact) connected to the "Internet Research Agency", a Russian "troll factory" and a defendant in an indictment filed by the Justice Department in February 2018, as part of special counsel Robert Mueller's Russia investigation, obtained from FiveThirtyEiAght's story "Why We're Sharing 3 Million Russian Troll Tweets". Additionally, it contains 1.6 million tweets from random users collected using Twitter api, obtained from kaggle.

4 Methodology and Related Work

1. **Logistic Regression:**
   Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. It maximizes the likelihood of labels given input data and tuned weights. In particular, logistic regression uses the sigmoid function, given by $g(x) = \frac{1}{1+e^{-z}}$, to express $p(y = 1|x)$. Logistic Regression uses the gradient ascent to maximize the log likelihood given by :

   $$l(\theta) = \sum_{i=1}^{n} y^{(10)} \, log \, g(\theta^T \, x^{(i)}) \; + \; (1 - y^{(i)}) \, log(1 - g(\theta^T x^{(i)}))$$

2. **LSTM Neural Network:**
   Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.
   A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.
   The forget gate weights different attributes of the context on a 0-1 scale using the equation $f_t = \sigma(W_f \, . \, [h_{t-1}, x_t] + b_f$, where f_t represents the new weightings for all of the old information in the context, W_f is all the previous weightings, h_{t-1} is the previous output (probability that the tweet is made by a bot or not), x_t is the new input (new feature of the tweet), and b_f is the bias for the weights. The sigmoid activation is chosen to put values between 0 and 1.
   The final context is simply an aggregation of the new weightings for the old context attributes and the weighting for the new context attributes and the new output h_t is a function of the output of the standard neural network equation: $o(t) = W_t \, . \, [h_{t-1}, x_t] + b_t$ and the context.

Social media bot detection is still emerging and thus there is minimal work done in this area in contrast with other fields. There was an attempt to classify bots in 2017 using Logistic Regression and Decision Trees, however accuracy wasn't too high and reach a value of 78.5%. Furthermore, there is also a lack of direct applicability from the current work in the field to bot detection on social media today.

5 Experiments and Results

5.1 Experiments

We conducted the experiments using the train and test datasets described above. Before proceeding to the training part, we did some prepossessing on the training data. For instance, we performed stemming and lemmatization, as well as removing punctuation, non-ascii characters, stopwords, etc. Additionally, we trained and tested Logistic Regression using TFIDF, and trained the LSTM model with custom word embeddings of length 50 (GloVe).
We made use of *sklearn* library for our implementation of Logistic Regression. The Logistic Regression algorithm assigns a weight for each words and then classifies them as troll and non-troll by forming a linear boundary.
As for the LSTM neural network, we utilized *keras* library. Before deciding on the final aspect of out model, we did multiple assessments and experimented with different LSTM layers, dropout rates, batch sizes, activation functions, and optimizer algorithms. Our final model consisted of seven layers, namely an LSTM layer followed by a conventional densely connected layer and an output layer. We initialized the weights of the model from the ones generated using GloVe. Moreover, we used the *ReLu* activation function and set the dropout rate to 0.1 as we found them best suitable for our LSTM layer. On the densely connected layer, we also used the *ReLu* activation function, utilized the sigmoid activation to put the output to be a probability between 0 and 1. We trained the model using binary crossentropy loss, $-(y\,log(p)\ +\ (1-y)\ log\,(1-p))$, and optimized it using the RMSProp algorithm.

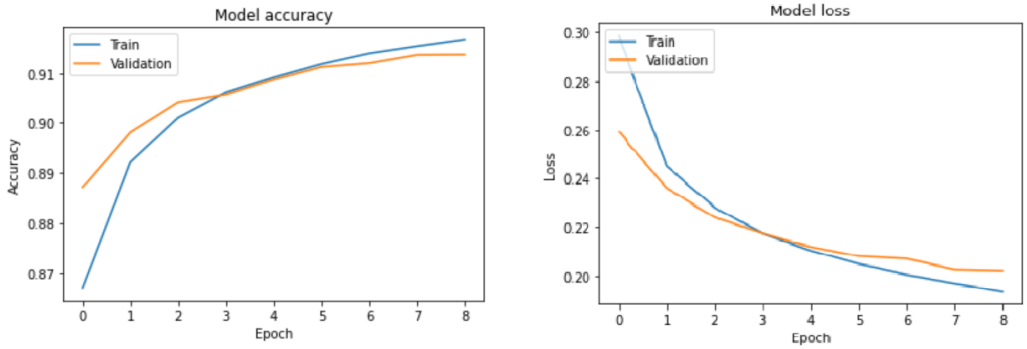
5.2 Results

After training both models, we obtained the following accuracies:

1. **Logistic Regression:** Logistic Regression with TFDIF gave an accuracy of 0.8403 on the test dataset.

2. **LSTM:** The LSTM neural network yielded a test accuracy of 0.905, outperforming Logistic Regression by 6% .

Below are the plots for the LSTM model accuracy and model loss:



Finally, we can conclude that the detection of tweets made by bots is not overly complex as we expected since we were able to achieve a 84% accuracy with a simple algorithm like Logistic Regression. Furthermore, we faced limitations in the scope of our dataset; the troll tweets were politics related ones, and we were hoping we could incorporate more general tweets posted by social media trolls.

6 References

1. **Datasets references:**
   (a) FiveThirtyEight (2018). 3 million Russian troll tweets: **here**
   (b) Kazanova (2017). Sentiment140 dataset with 1.6 million tweets: **here**

2. **Used academic resources references:**
   (a) Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning.
   (b) Cramer, J. S. The origins of Logistic Regression.
   (c) Wang-chun Woo (2015). "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting"
   (d) Gers, F. A.; Schmidhuber, J. "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages"
   (e) Gers, Felix. "Long Short-Term Memory in Recurrent Neural Networks". PhD Thesis.