Detecting Fake News Using a LSTM Neural Network

Brandon Lasher
Hood College
401 Rosemont Ave
Frederick, MD 21701

ABSTRACT

Fake news has become a major issue in society today. With the social media dominating the way news spreads and the lackluster verification on any news article that is shared, fake news affected all aspects of society. It effects what news stories are covered; to the way government policies are talked about; to the way public campaigns are run; to even discussions around the dinner table. The standard way of fact checking to rule out fake news, now takes too long in the incautiously changing new cycle. In this paper we discuss building a LSTM model to detect if a text if fake or true. We find that if the text is similar enough to the training set, we are able to predict at a very high percentage that if the news is indeed fake or true.

Approach

In this section, we present details of our approach and the framework which define it. The process we will take will be first to clean the data. We will remove any items from the text which might contribute noise to the model. Next, we will analyze the data. We will do this by looking at various features of the data as well as deriving new features based on publicly available natural language processing (NLP) models. Lastly, we will take the step necessary to build an LSTM to make our predictions. As an extra task, we will then use a second data set with information from around that same time period to test our model.

The Data

The dataset we will use can be found on Kaggle[1] This data consists of two csv files which denote a set of fake and true news article and social media posts. Fake.csv consists of 23481 entries and True.csv consists of 21417 entries. It is laid out as follows:

Feature Name	Data Type	Description
title	object	Title of the article
text	object	Text of article, may include publisher and author
subject	object	Type of News
date	date	Date of publication

Cleaning the Data

To optimize the learning and make the most of out the data provided we need to do our best to remove as much erroneous information as possible.
The first step to clean the data is to combine the title and the text fields to a new field called ‘news’. This way when cleaning and training we will be using all the text information available about the entry. We then create and apply a function the ‘news’ field to clean the data which does the following.

- Removes tags, urls, and html
- Remove any additional punctuation.
- Remove any numbers
- Make all text lowercase
- Remove any stopwords
- Combine Fake and True data sets into one, but addition an additional classification column called ‘type’

Prior to cleaning

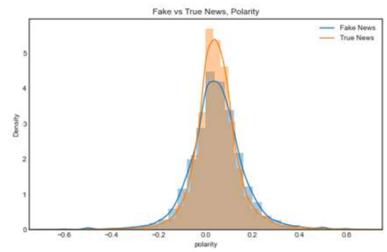
```
' Donald Trump Sends Out Embarrassing New Year’s Eve Message; This is Disturbing Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had to give a shout out to his enemies, haters and the very dishonest fake news media. The former reality show star had just one job to do and he couldn t do it. As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year, President Angry Pants tweeted.
```

After **cleaning**

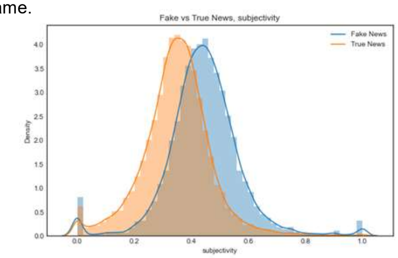
```
'donald trump sends embarrassing new year’s eve message disturbing donald trump wish americans happy new year leave instead give shout enemies haters dishonest fake news media former reality show star one job country rapidly grows stronger smarter want wish friends supporters enemies haters even dishonest fake news media happy healthy new year president angry pants tweeted
```

Inspecting the data

Checking the Sentiment and Subjectivity

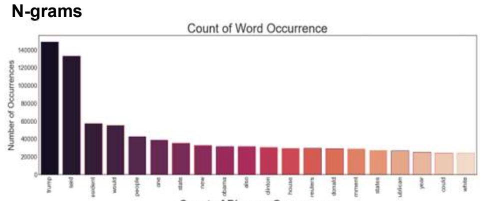


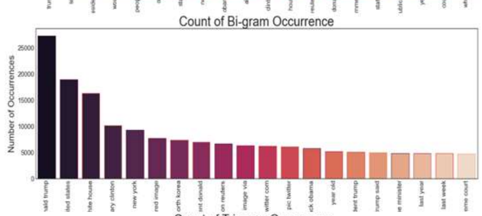
Polarity, Sentiment, is not very much between the two datasets. While true news had a smaller standard deviation than the fake news, the mean was approximately the same.

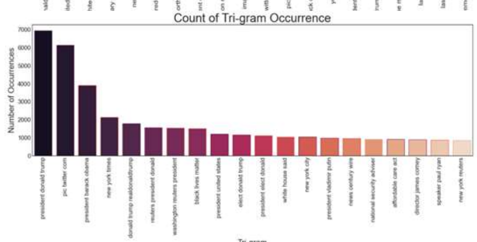


Subjectivity was significantly different between the two data sets. Fake news had a much higher mean in the subjectivity metric, which means it was much more subjective in its presentation.

N-grams





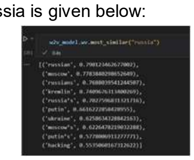


As expected, because this dataset was taken between and shortly after the 2016 election, the most common topics are all political related. With Donald Trump dominating the majority of the news cycle, followed closely by White house, Barack Obama and Hillary Clinton related topics. In the tri-gram chart we can start to see some the data sources, Twitter and Reuters.

Building and testing the Model

Long short-term memory (LSTM) Neural Network

An LSTM is a type of recurrent neural network (RNN) that is capable of learning and remembering over long periods of time. This makes it well-suited for NLP tasks that require the model to remember and use past information in order to make predictions or decisions. To build the embedding vector, which will help the model focus on relevant information, we used the Word2Vec library. Most of the articles have less than 1000 words, we will limit our text to 1000 words each. Then we tokenize the words and use them to build a vocabulary list which has the word associations given by Word2Vec. An example for the word russia is given below:



For this model we choose to make it 128 nodes and used 75% of the data for training and validation it. The model was trained using .7 of the training and validation set(52.5% of the overall data), it used .3 of training and validation set (22.5% of the overall data) to check to see how the model training did. After training and validation test was complete, we used the 25% we kept aside to see how accurate we made our model. Below is a summary of the results

	Loss Value	Accuracy
Training data	.0948	.9693
Validation data	.0284	.9926
Test Data	-	.9926

Building this model is highly dependent on the dataset, so we decided to try it on a different dataset[2] which was from the roughly same time period, With the new dataset we only ended up with a 73.9% accuracy score. The likely cause of this was that a large enough number of words fell into the bucket which we did not train on, thus it was not able to know the associated semantics of the word.

Conclusions

We saw that when using a subset of our training data to evaluate our model we ended with a very high rate of accuracy. However, if we used a dataset which we didn’t train on and likely had different words and semantics our accuracy rate dropped significantly. 73.9% is still much better than randomly guessing, but it still is not enough to be able to reliably use this model for practical purposes.
The data itself had several key properties. Frist, Donald Trump dominated the topics. Fake news had much more subjectivity.
Several avenues could be explored to improve this project. First, by training on a wider dataset to increase the vocabulary. It could also be improved by utilizing the polarity and subjectivity information. Lastly, is to change the model into a bidirectional LSTM.

Datasets/Source code/Notes:
[1] “Only one word 99.2%.” https://kaggle.com/code/josutk/only-one-word-99-2 (accessed Dec. 11, 2022).
[2] “Fake News.” https://kaggle.com/competitions/fake-news (accessed Dec. 11, 2022).
[3] Project Notebook https://github.com/blasher565/dm_cs522/tree/main/final%20project/Code
[4] See github.com link for Associated paper and list of complete references.