

# Fake News Detection (Political Dataset)

Amina Qazi<sup>1</sup>, Swapnil Anand<sup>2\*</sup>, Amar Singh<sup>3</sup>, Bhavya Shettigar<sup>4</sup>, Shivam Shukla<sup>5</sup>  
<sup>1,2,3,4,5</sup>Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India

**Abstract:** The increase in misleading information in everyday life through media outlets such as social feed, news blogs and online newspapers has made it very difficult to identify the trustworthy news sources, thus increasing the need for new technologies and computers to play the role by providing the better insights from data from online contents in online news. In this paper we have introduce the way BiLSTM algorithm can increase the chances of predicting news content accurately. In this paper we conduct a set of learning experiments to build accurate fake news detector. In addition, we have provided the analyses of the dataset.

**Keywords:** BiLSTM, Deep Learning, R Language, Keras.

## 1. Introduction

Fake news detection has attracted a lot of attention through platforms like online newspaper, social media and blogs, general public and researchers as the circulation of this misinformation on online platform has increased dramatically in the last two decades. For instance, Facebook will start testing one of the features that was launched recently by Twitter a couple of months ago. Just like Twitter, Facebook will soon start reminding users to read the article before sharing. It is yet another step by Facebook to curb the spread of fake news on its platform. The feature has already been tried and tested by Twitter before but unfortunately results were not impressive. Facebook will initially pick a few users across the world to test this feature. Since almost all the population across the globe gets on the internet through social media, online forum or online newspaper, being able to identify fake content in online source is necessary step for stopping the increase of fake news. To date, computational approaches for identifying the fake news detection have relied on an unconventional way such as “The Onion” and fact-checking websites such as “Snopes”. However, by using these sources it poses a challenge and potential drawbacks. In this paper we have develop computational model for the task of fake news detection. We also conduct several exploratory analyses to identify the patterns that are predominantly present in fake content, and we build the fake news detectors relying on this pattern that achieve accuracies of up to 84.5%.

## 2. Fake News Dataset

The dataset which we have used is the dataset from Kaggle competition which was specifically aimed at political data. In

this data there are three csv files namely test.csv, train.csv and submit.csv. This dataset contains 20800 training sample and with 5200 testing sample, which is marked as 0 and 1 where 0 represents true (reliable) and 1 represents false (not reliable). Further we have performed some exploratory analysis for getting all the insights from data, we have used R as our base language.

## 3. Exploratory Data Analysis

Performing EDA (Exploratory Data Analysis) on the dataset posed some challenge like which all words to keep, punctuations, stemming etc. before performing the EDA on the training dataset we have to clean the dataset such that the data only contains the relevant information for the goal of this project. We have used the libraries such as Tidyverse and qdap to perform the data preprocessing step in this project.

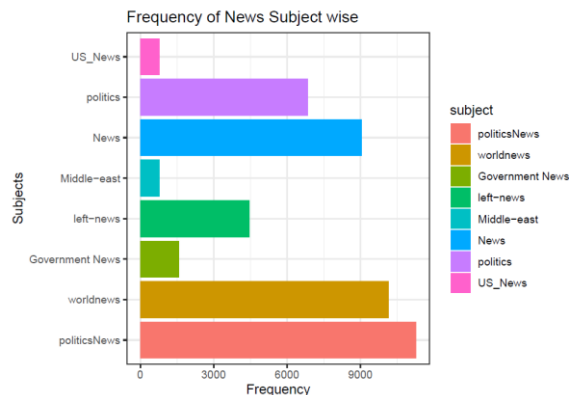


Fig. 1. News distribution subject wise

The pre-processing steps includes such as removing the most common words from the dataset, removing the word whose length is three or less because such words are not going to fetch us any meaningful results and most of the three letter words are stop words, since we are using the dataset which is collected in English language, we will keep it as default for removing the stop words. We have also converted all the words to lower case for ease of finding pattern in the dataset and similarly we have also stemmed the whole dataset for the analysis purpose.

Now that we have completed and preprocessing steps of our dataset, we can move towards the exploratory data analysis where we can plot all the necessary charts related to the data

\*Corresponding author: swapnilan@gmail.com

and find the insights from it so that we can make a decisive decision while building neural network using keras package.

After careful observation from fig. 1, We can see that in the dataset we have most of the news sample related to politics and majority of the news sample is related to world news and politics news.

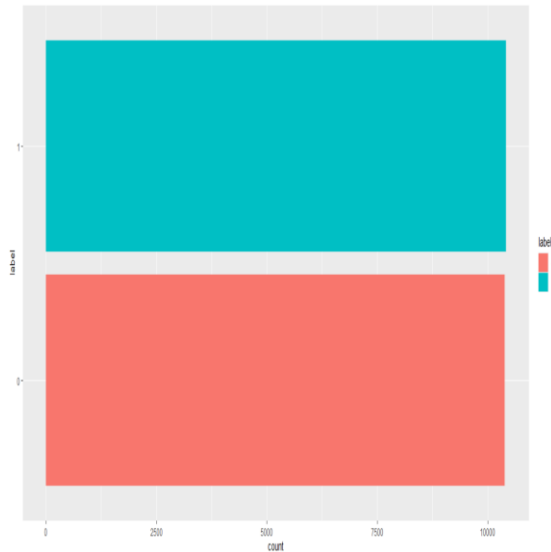


Fig. 2. News Sample Distribution true or false

As seeing from fig. 2, we can see that we have almost the same amount of true and false news sample. This means we have equilibrium between the true and false data in our dataset.

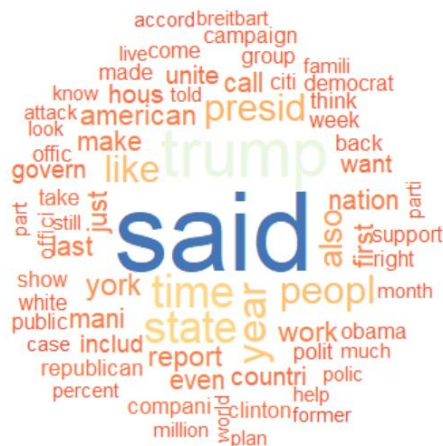


Fig. 3. Word cloud of true news in dataset

From fig. 3, we can see the top 70 words which are plotted according to the most repeating sequence in the true news in our dataset.

From Fig. 4, we can see the top 70 words which are plotted according to the most repeating sequence in the fake news in our dataset.

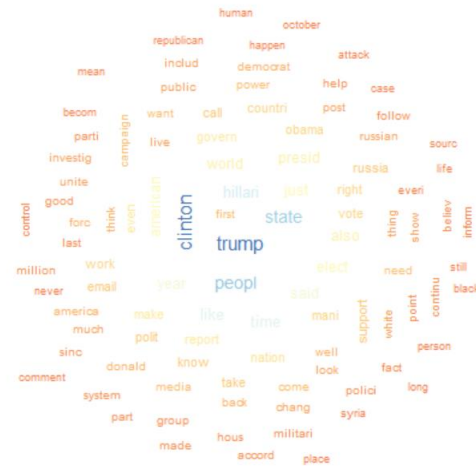


Fig. 4. Word cloud of fake news in dataset

### 4. Model Creation

Now that we have completed our data preprocessing and exploratory data analysis, we are now ready to create our neural network model.

#### A. Neural Network

Neural networks come under deep learning which also comes under machine learning and artificial intelligence, it is set of algorithms, designed on basis of how human brain works, that are designed specifically for find a pattern to a single task.

#### B. Keras

Keras is a library used when you have to create for deep learning model. In short keras is designed specifically for deep learning purpose in python but it is also supported on R language as well. Keras makes it very easy for us to write all the deep learning algorithms.

#### C. BiLSTM

A BiLSTM (Bidirectional Long Short Term Memory) is a type of LSTM gate which let's you memories the long term sequence in your data. Bidirectional means data can go forward and backward while our model is to memories the sequence for the corresponding value i.e., true or false.

```

Model
Model: "sequential_7"

Layer (type)                Output Shape                Param #
-----
embedding_7 (Embedding)     (None, 100, 32)            960000
bidirectional_8 (Bidirectional) (None, 64)                  16640
dense_13 (Dense)            (None, 1)                   65
-----
Total params: 976,705
Trainable params: 976,705
Non-trainable params: 0
    
```

Fig. 5. Sequential Model

Above is the model for which will detect the news sample for the prediction, since it is fairly a small model, we have minimized the sent length to 100.

In the first layer we have used embedding layer which will

internally convert the integer into dense vector of fixed size, this layer can only be used once in any neural network model and only as the first layer.

In the second layer we have BiLSTM which will take the input of 32 neurons and will train up to 64 memory capacity, we have also explained what BiLSTM means. In the final layer we have used dense layer with the sigmoid activation function since the whole project is a binary classification project, we will use sigmoid as the activation function. Since we have vocabulary size of 120000 unique words, to avoid any overheating and any other hardware issues we have cut down the vocabulary size to 30000 and sent length to 100.

### 5. Algorithm and Model Training

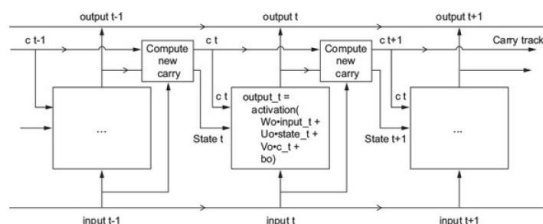


Fig. 6. Anatomy of LSTM

Above is the anatomy of lstm layer you can say that multiplying  $c_t$  and  $f_t$  is a way to deliberately forget irrelevant information in the carry dataflow. Meanwhile,  $i_t$  and  $k_t$  provides information about the present, updating the carry track with new information. But these operations means nothing because what these operations determines the content of weight parameterizing them and are learned in end-to-end fashion in each round making it difficult to remember long sequences.

```

output_t = activation(dot(state_t, Uo) + dot(input_t, Wo) + dot(C_t, Vo) + bo)
i_t = activation(dot(state_t, Ui) + dot(input_t, Wi) + bi)
f_t = activation(dot(state_t, Uf) + dot(input_t, Wf) + bf)
k_t = activation(dot(state_t, Uk) + dot(input_t, Wk) + bk)
c_{t+1} = i_t * k_t + c_t * f_t
    
```

Fig. 7. Pseudocode for LSTM

Using the above pseudocode, we can achieve the  $c_{t+1}$  state in the training.

Now once we have created our model skeleton, now comes the part for optimizer and training our dataset which we have pre-processed in earlier section.

For optimizer we have used adam as our optimizer and for loss we have used binary cross entropy since this project is closely related to binary classification problem we have to use adam as our optimizer and binary cross entropy as our loss optimization.

As from the above graph we can see the loss has been significantly reduced over time and accuracy has increased till 96%.

Now let's talk about how we achieved this result on our training set, we have trained our model for 20 epochs at first with batch size of 128, and we are facing the overfitting problem during our training set and then we have reduced the epoch to 5 and the issue was resolved.

Epochs is a hyper parameter which means how many times

your model will be trained on the train data.

Batch size refers to number of training sample utilized at a single iteration.

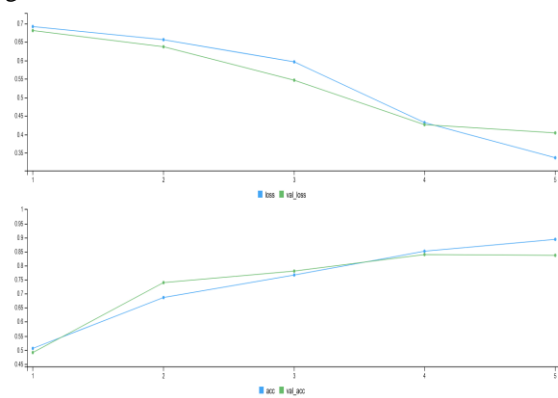


Fig. 8. Training Graph

### 6. Results

Now let's take a look at the result, which our neural network gives.

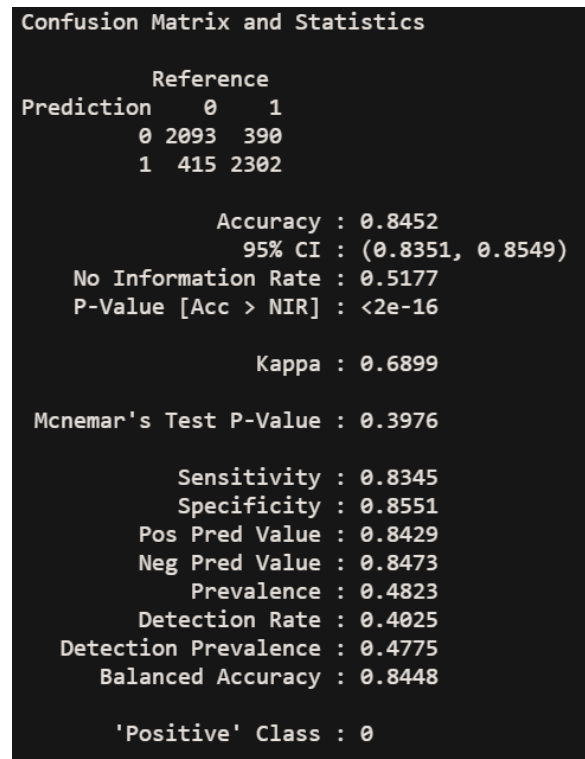


Fig. 9. Confusion matrix on test data

Above are some important parameters where you can see the accuracy, p-value and confidence-interval. We are mainly interested in these parameters and not on other parameters, this confusion matrix is created using "caret" package in R language

Now to understand what is confusion matrix, let's describe it as "A confusion matrix is a way that is often used to describe the performance to the model on two or more classes".

If you look closely in fig. 9, we have achieved the accuracy of almost 85% on the testing dataset, which is a good result for fairly small-scale network, also the confidence interval is also

between 0.8351 and 0.8549 which means it will accurately predict the sample in these ranges by using the parameters described in Model Creation section.

### 7. Conclusion and Future Scope

- Reducing the need for manual search of news whether it's reliable or not.
- It is easily accessible to the general public for fact check in political domain.
- Further work can be done is that one can include multiple domain news sample.
- Use of larger network or ensemble learning can drastically increase the scope of prediction.

### Acknowledgement

We would like to express our gratitude and extend our thanks to the entire team by giving us this opportunity and guiding us.

### References

- [1] Gary D Bond and Adrienne Y Lee. 2005. Language of lies in prison: Linguistic classification of prisoners' truthful and deceptive natural language. *Applied Cognitive Psychology* 19(3):313–329.
- [2] Molly E Ireland and James W Pennebaker. 2010. Language style matching in writing: synchrony in essays, correspondence, and poetry. *Journal of personality and social psychology* 99(3):549.
- [3] Allcott, H., & Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, 31(2), 211–236.
- [4] Bolsen, T., & Druckman, J. N. (2015). Counteracting the Politicization of Science. *Journal of Communication*, 65(5), 745–769.
- [5] Budak, C., Agrawal, D., & El Abbadi, A. (2011). Limiting the Spread of Misinformation in Social Networks. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 665–674). New York, NY, USA: ACM.
- [6] Flynn, D. J., Nyhan, B., & Reifler, J. (2017). The Nature and Origins of Misperceptions: Understanding False and Unsupported Beliefs About Politics. *Political Psychology*, 38, 127–150.
- [7] Flaxman, S., Goel, S., & Rao, J. M. (2016). Filter bubbles, echo chambers, and online news consumption. *Public Opinion Quarterly*, 80(1), 298–320.
- [8] Core Team. 2016. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- [9] Harriss, L., & Raymer, K. (2017). Online Information and Fake News (Post Notes). Post Note. Retrieved from <http://researchbriefings.parliament.uk/ResearchBriefing/Summary/POST-PN-0559#fullreport>
- [10] Mustafaraj, E., & Metaxas, P. T. (2017). The fake news spreading plague: Was it preventable? In *WebSci 2017 - Proceedings of the 2017 ACM Web Science Conference*, pp. 235-239.