TEXT CLASSIFICATION USING BERT AND LSTM:

FAKE NEWS DETECTION

By

Varsha Rajasekar

A Research Project Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Statistics

Specialization in Data Science At

Rutgers, the State University of New Jersey

August 2021

ABSTRACT

News media has become an important source to pass on the information about everything that is happening in the world to the people. However, the authenticity of this information has become a longstanding issue affecting businesses and society, both for printed and digital media. The media has often been notorious for sensationalizing inaccurate news to retain the attention of public. Often people perceive whatever conveyed in the news to be true. With the reach of the media far enough and effects of the information being spread at a fast pace, such distorted and false information has the potential to cause adverse impacts on the world and its peace. Some news has significant impact on the economy. Such news can bring the economy down and affect the livelihoods of millions of people. To overcome the problem of fake news, we can use one of the most common tasks of Natural Language Processing. Text Classification, also known as text tagging or text categorization is the process of categorizing text into organized groups. Text classifiers can automatically assign a set of pre-defined tags or categories after analyzing its content. The purpose of this project is to detect fake news from the Kaggle dataset using pre trained BERT model, one of the most popular transformer models and bidirectional LSTM which belongs to a larger category of neural networks called Recurrent Neural Network (RNN). Through the evaluation of the two models, we find that BERT model performs impressively well and better than Bi-LSTM.

ACKNOWLEDGEMENTS

This work would not have been possible without the support of many people. I would like to give special thanks to my project advisor Dr. Ajita John for guiding me. I enjoyed being in her Advanced Statistical Software class because of which I cultivated interest in Natural Language Processing. Dr. Ajita John’s teachings have inspired me to take interest in exploring different projects in the field of NLP. I am also grateful for the support Dr. Neville O’Reilly. And finally, nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. Also, thanks to my sister who has encouraged me in every step and continues to do so.

Table of Contents

List of Illustrations/Graphs

1. Introduction

Misinformation presents a huge challenge in online society. In today’s world, social media platforms are a major source for online news and very influential. According to internet live stats the estimated daily number of tweets is about 500 million. Hence, there have been many attempts to identify and classify misinformation. Especially in times like this, where a global pandemic is wrecking lives, any fake news regarding COVID-19 can affect the public health negatively. According to WHO, fake news spreads faster and more easily than the virus. Companies like Facebook, Twitter, TikTok, Google, Pinterest, Tencent, YouTube, and others are working with WHO to mitigate the spread of rumors. Their efforts aim at filtering out content that is a danger to public health.

BERT stands for Bidirectional Encoder Representations from Transformer. BERT is based on the Transformer model architecture, instead of LSTMs. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. Unlike the previous language models, it has a non-directional approach where it takes both the previous and next tokens into account at the same time.

The existing combined left-to-right and right-to-left LSTM based models were missing the same-time part. Bi-directional LSTM allows looking at sequences from front-to-back as well as from back-to-front. Each sentence is converted from a list of words to a sequence of indices where each index corresponds to a word. In order to feed the sentences into the model, they all need to be of the same length. So, all the sequences are padded to the same length. The embeddings are created by the embedding Layer which takes in the padded sequences of variable length and transforms each token into a vector of n-dimensions.

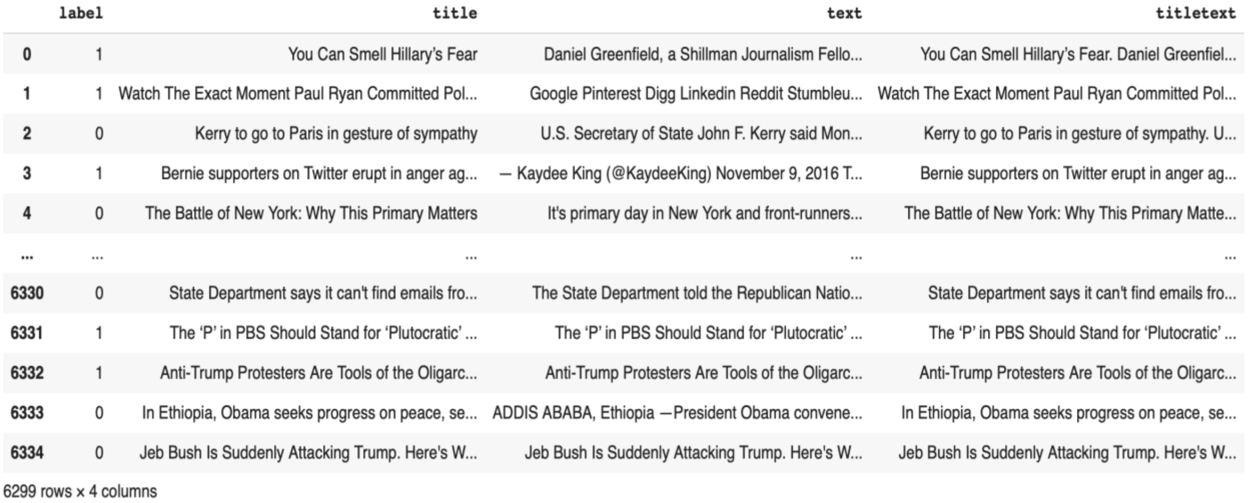
2. Dataset

I have used the REAL and FAKE news dataset from Kaggle which consists of 6299 rows out of which 50% are fake news and the remaining 50% are real news. The file has following columns:

- Unlabeled index column
- title - contains news headlines
- text - contains news content
- label – 0(real)/1(fake)

2.1 Pre-processing

First, the ‘title’ and ‘text’ column are concatenated to form a new column ‘titletext’. Records with empty text are dropped. A trim_string function is defined that takes in n_words as input which is the number of words specified. This function trims each sentence to the first n_words to enable faster training. Next, the dataset is divided into two data frames based on the label column. The fake news and real news data frames were split into training, test and valid sets separately using the train_test_split method of the scikit-learn library. The training, test and valid sets for fake news and real news data frames were finally concatenated to be passed to the model.



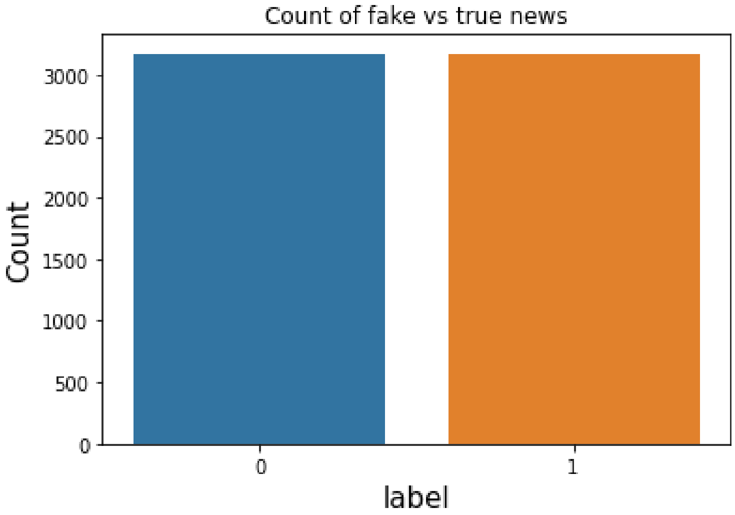
	label	title	text	titletext
0	1	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	You Can Smell Hillary’s Fear. Daniel Greenfiel...
1	1	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	Watch The Exact Moment Paul Ryan Committed Pol...
2	0	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	Kerry to go to Paris in gesture of sympathy. U...
3	1	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	Bernie supporters on Twitter erupt in anger ag...
4	0	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	The Battle of New York: Why This Primary Matte...
...	...	...	...	...
6330	0	State Department says it can't find emails fro...	The State Department told the Republican Natio...	State Department says it can't find emails fro...
6331	1	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...	The ‘P’ in PBS Should Stand for ‘Plutocratic’ ...
6332	1	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligarc...
6333	0	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	In Ethiopia, Obama seeks progress on peace, se...
6334	0	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...

6299 rows × 4 columns

To visualize our dataset better in order to understand it more clearly, all punctuations and English stop words are removed using regex and nltk respectively.

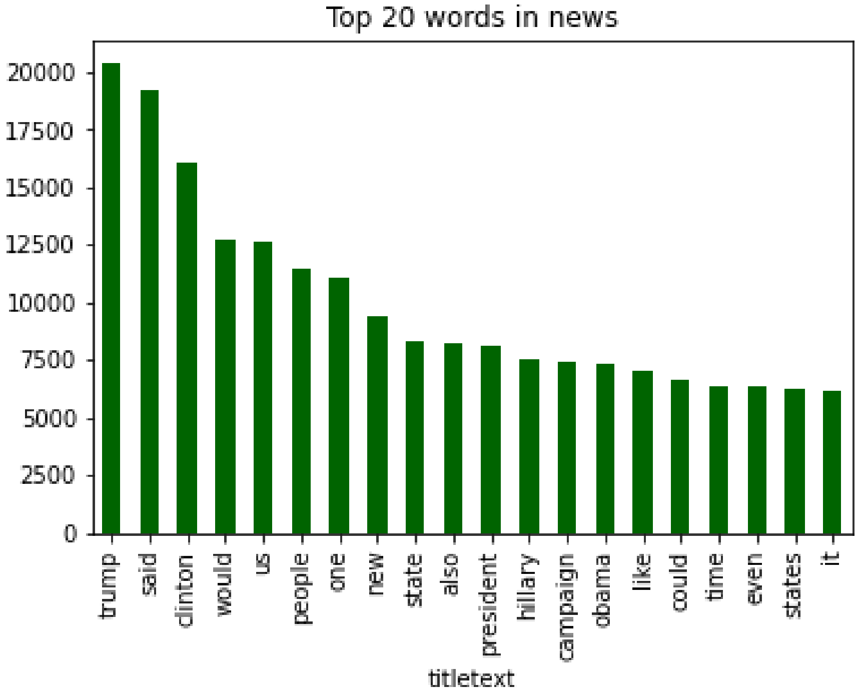
2.2 Exploration

1. Count of Fake news vs Count of True news



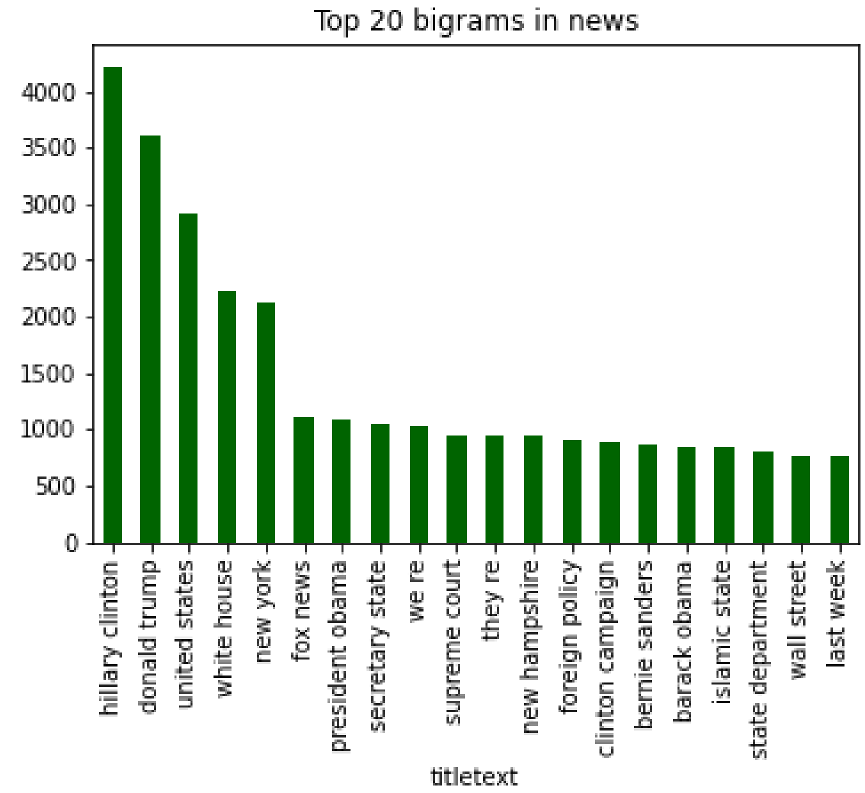
The graph gives us an idea about the number of fake news and real news in the dataset. We can see that the dataset equally distributed between fake and real news.

2. Top 20 words in News



Here, the graph shows top 20 words with most occurrences in the dataset. For better results, all the stop words are removed before finding the top 20 words. We can see from the graph that 'trump' is the most common word with 20344 appearances.

3. Top 20 bigram words



Here, the graph shows top 20 word pairs with most occurrences in the dataset. For this visualization too, stop words and punctuations are removed first. We can see that 'hilary clinton' is the most appeared word pair with 4205 appearances in the dataset.

3. Models

3.1 Libraries

I imported Pytorch for model construction, torchText for loading data, matplotlib for plotting, and scikit-learn for evaluation. Huggingface library is used in this project, which is the most well-known library for implementing state-of-the-art transformers in Python.

3.2 BERT

I used the bert-based-uncased version of BertTokenizer which is the smaller model trained on lower cased English text with 12 layer, 768-hidden, 12 heads, 110M parameters. Using TorchText, first the Text Field and the Label Field are created. The Text Field contains the news articles and the Label is used as the target. Each article is limited to the first 128 tokens for BERT input. Then, a TabularDataset is created from the dataset files based on the two Fields to produce the train, validation, and test sets. Then Iterators is created to prepare them in batches. By setting use_vocab=False and tokenize=tokenize.encode, we let the torch text know that we will be using pre-trained BERT and its word-to-index mapping.

The save and load functions are created for model checkpoints and training metrics, respectively. The training metric stores the training loss, validation loss, and global steps so that visualizations regarding the training process can be made later. I used Adam optimizer and a learning rate of 0.00002 to tune BERT for 5, 7 and 10 epochs. Since fake news detection has binary labels, BinaryCrossEntropy is used as the loss function. The output is passed through Sigmoid before calculating the loss between the target and itself.

During training, the model parameters are evaluated against the validation set. The model is saved each time the validation loss decreases so that we end up with the model with the lowest validation loss, which can be considered as the best model.

3.3 Bi-LSTM

Like the BERT model, torchText is used to create a label field for the label in the dataset and a text field for the title, text, and titletext columns. The TabularDataset is then built by accessing the train.csv, valid.csv, and test.csv dataset files. The train, valid, and test iterators are created to load the data, and finally, the vocabulary is built using the train iterator counting only the tokens with a minimum frequency of 3.

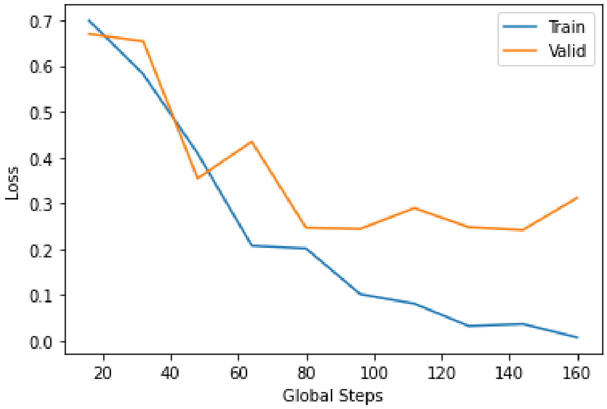
The LSTM class is constructed, inside which an Embedding layer is built, followed by a bi-LSTM layer, and ending with a fully connected linear layer. In the forward function, the text IDs are passed through the embedding layer to get the embeddings, which are then passed through the LSTM accommodating variable-length sequences. After learning from both directions, the output from the previous layer is passed through the fully connected linear layer, and finally through sigmoid to get the probability of the sequences belonging to FAKE.

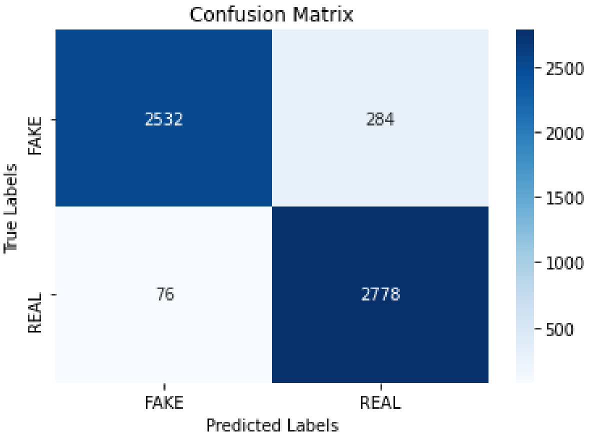
Before training, save and load functions are built for checkpoints and metrics. For checkpoints, the model parameters and optimizer are saved; for metrics, the train loss, valid loss, and global steps are saved so diagrams can be easily reconstructed later. After training the LSTM with 10 epochs, the checkpoints and metrics are saved whenever a hyperparameter setting achieves the lowest and the best validation loss.

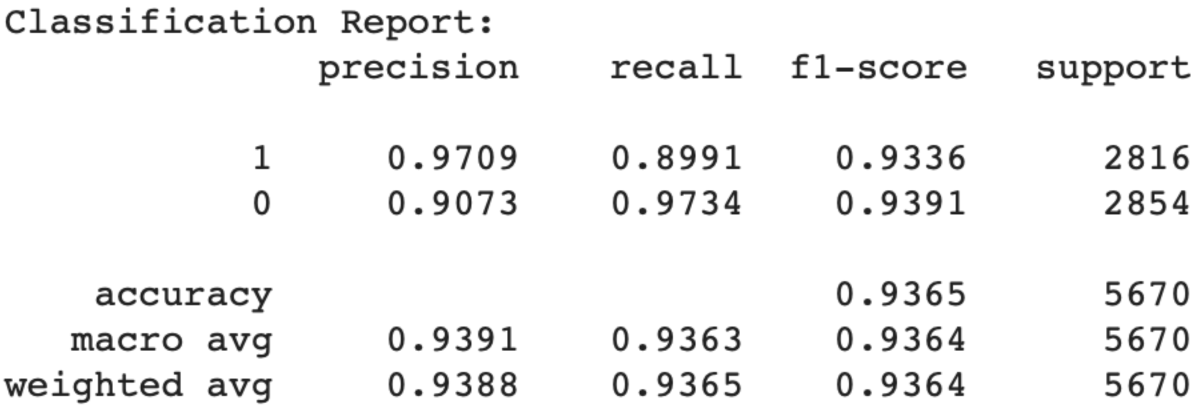
4. Evaluation

4.1 BERT

For evaluation, we predict the articles using our trained model and evaluate it against the true label. After training the model for 5, 7 and 10 epochs, the model trained for 5 epochs gave the highest accuracy. From the figure, we can see validation loss decreases till a point and then slightly increases and training loss keeps decreasing as the number of steps increase. As the number of steps increases, the gap between the training and validation loss, also known as the generalization gap, increases. This indicates overfitting which leads to increase in generalization error. After evaluation the model's accuracy is 0.9365.



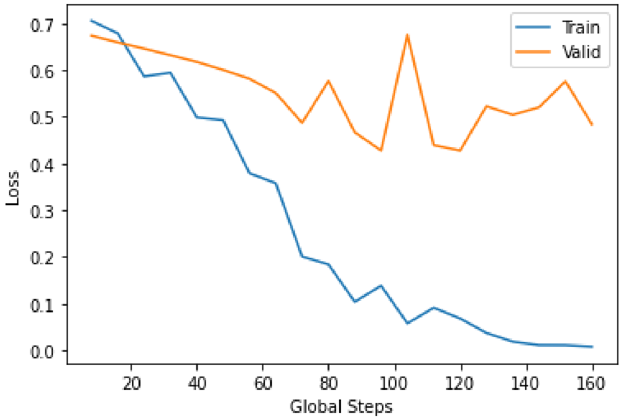


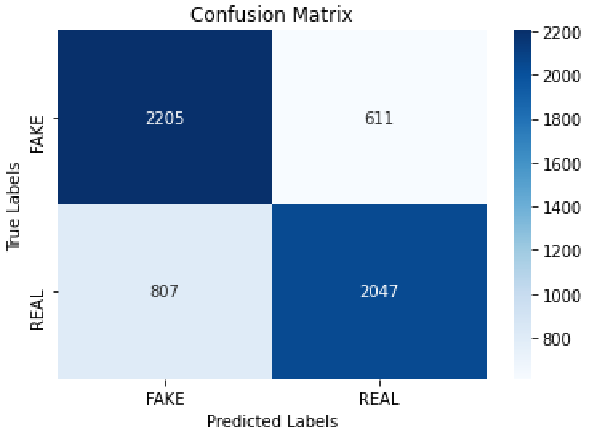


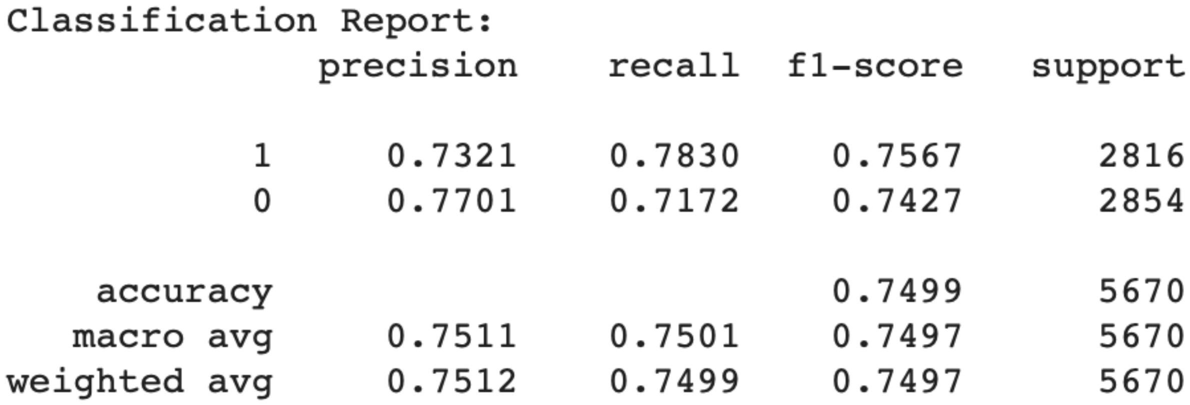
```
Classification Report:
              precision    recall  f1-score   support

           1     0.9709    0.8991    0.9336      2816
           0     0.9073    0.9734    0.9391      2854

    accuracy                         0.9365      5670
   macro avg     0.9391    0.9363    0.9364      5670
weighted avg     0.9388    0.9365    0.9364      5670
```

4.2 Bi-LSTM

A default threshold of 0.5 is used to decide when to classify a sample as FAKE. If the model output is greater than 0.5, we classify that news as FAKE; otherwise, REAL. From the graph we can see that the training loss keeps decreasing and the validation loss is. The very large gap between the training and validation loss shows that the model is overfitting and fails to generalize well. After evaluation the model's accuracy is 0.7499.







```
Classification Report:
              precision    recall  f1-score   support

           1     0.7321    0.7830    0.7567      2816
           0     0.7701    0.7172    0.7427      2854

    accuracy                         0.7499      5670
   macro avg     0.7511    0.7501    0.7497      5670
weighted avg     0.7512    0.7499    0.7497      5670
```

5. Conclusion

By evaluating both the models we can clearly see how well BERT performs. Despite overfitting occurring in both models, BERT can be evaluated as a better model based on its high accuracy. Bi-LSTM achieves an acceptable accuracy for fake news detection but still has room to improve. BERT model correctly predicted 2532 fake news and 2778 real news whereas Bi-LSTM correctly predicted 2205 fake news and 2047 real news. BERT has a better prediction ability with a macro avg score of 0.9364. Both the models can be further improved through advanced fine tuning. Next steps include reducing the overfitting problem by adding more dropout to BI-LSTM model, adding regularization, or adjusting the learning rate and batch size. The issue of overfitting could also be due to bias in the dataset, or the training set could unrepresentative.

6. References

i. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

ii. https://searchenterpriseai.techtarget.com/definition/BERT-language-model

iii. https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0

iv. https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/

v. https://towardsdatascience.com/fake-news-detection-with-machine-learning-using-python-3347d9899ad1

vi. https://towardsdatascience.com/bert-text-classification-using-pytorch-723dfb8b6b5b

vii. https://towardsdatascience.com/lstm-text-classification-using-pytorch-2c6c657f8fc0

viii. https://www.kaggle.com/benroshan/fake-news-classifier-lstm

ix. https://www.who.int/news-room/feature-stories/detail/fighting-misinformation-in-the-time-of-covid-19-one-click-at-a-time

x. https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

7. Bibliography

i. Sadrach Pierre, *Fake News Classification with BERT*, https://towardsdatascience.com/fake- news-classification-with-bert-afbeee601f41

ii. Kaliyar, R.K., Goswami, A. & Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed Tools Appl* **80,** 11765–11788 (2021). https://doi.org/10.1007/s11042-020-10183-2

iii. Ahmed, B., Ali, G., Hussain, A., Baseer, A., & Ahmed, J. (2021). Analysis of Text Feature Extractors using Deep Learning on Fake News. *Engineering, Technology &Amp; Applied Science Research*, *11*(2), 7001–7005. https://doi.org/10.48084/etasr.4069

iv. Ahmed H., Traore I., Saad S. (2017) Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham. https://doi.org/10.1007/978-3-319-69155-8_9

v. Allcott, Hunt, and Matthew Gentzkow. 2017. "Social Media and Fake News in the 2016 Election." *Journal of Economic Perspectives*, 31 (2): 211-36. DOI: 10.1257/jep.31.2.211