The Use of Deep Neural Networks to Determine the Authenticity of News

Tomer Atzili
Department of Computer Science
University of Maryland, College Park
tatzili@terpmail.umd.edu

Kyla Perkinson
Department of Computer Science
University of Maryland, College Park
kperkin1@terpmail.umd.edu

Srilasya Poruri
Department of Computer Science
University of Maryland, College Park
sporuri@terpmail.umd.edu

Riya Boyapati
Department of Computer Science
University of Maryland, College Park
rboyapat@terpmail.umd.edu

1 Introduction

Fake news is everywhere, especially in recent years as access to technology spreads and the world becomes further politicized. We wish to use a deep learning algorithm to create a model that evaluates a news article to determine its authenticity. Our decided approach is to use a Long Short-Term Memory (LSTM) network as it is well suited for sequential data, which we have with text articles. In terms of representing the written data numerically, we used a pre-trained BERT model to generate embeddings, which values word importance and context in a text. The combination of these methods, with data and text preprocessing, gave us a trainable model that can pick out key characteristics of a fake news article and use them as predictors for determining authenticity of test articles at 96% accuracy. Given more time, there would also be more improvements we could bring to the model, which will be detailed in section 7.

1.1 Motivation

Fake news is a topic that comes up constantly since it made headlines for allegedly contributing to President Donald Trump's victory in 2016. However, the concept has been in existence for all of humanity as false ideas have been shared, intentionally or unintentionally. These can include satire, parody, misleading, biased, false, or manipulated ideas. In the modern day, it is crucial to evaluate what news is real or fake. According to Vosoughi, et al. (2018), fake news today spreads six times faster than real news. This can be largely attributed to the desire of social media to entertain and spread. Much of fake news is crafted to appeal to this desire, and is therefore propagated around the internet.

Fake news is also extremely present on the internet. Watson (2022) reports that according to a recent survey, about 80% of adults have seen news they deemed to be fake on the internet. It is likely that the true figure is even higher. Due to the clear prevalence of this issue, we thought that a technical solution could and should be reached to combat misinformation. In an ideal world, human's would immediately discern fake from real news. However, as shown in figure 1, a majority adults are not very confident in their ability to make this distinction.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

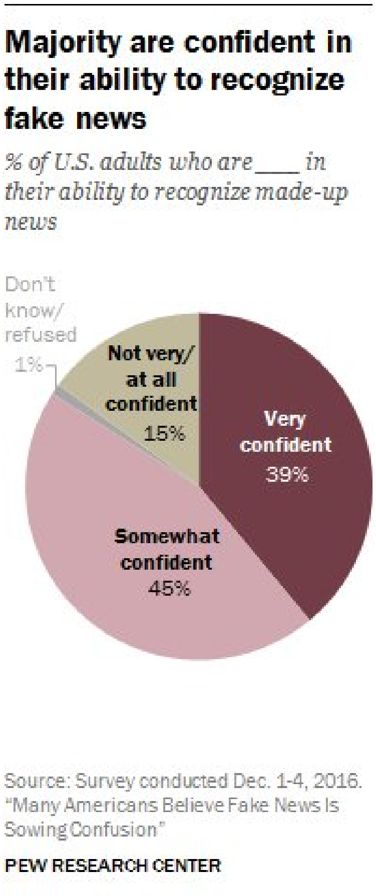

Figure 1. Pie Chart displaying how confident adults feel that they are in recognizing fake news.

This is where a deep learning model that can take a piece of made-up or biased news and determine it as such with high accuracy would become very valuable. If we consider half of the "Somewhat confident" section in the above figure as confident, we get a human confidence rate of 61.5%. A model that outperforms this figure on test data will be considered an improvement. Fake news is as dangerous as it has ever been, and we need to look beyond human training to identify and tag such misinformation. We believe our project can contribute to this effort.

2 Related Work

Previous attempts at using deep learning models to detect unreliable news have been made, with one such example being https://github.com/Abhradipta/Fake-News-Detection. In this project, the dataset is a collection of real and fake news articles from various sources. The text in these articles is converted to numbers and then used as input for recurrent neural networks and long short-term memory. The project provides insight on how to use long short-term memory networks (LSTM), which is helpful as we planned on utilizing LSTM networks as well.

To preprocess our data, we decided to utilize a bi-directional encoder representations from transformers (BERT) model, instead of using tf-idf vectors as we had originally planned. We referenced the code found at https://towardsai.net/p/l/fake-news-detection-using-bert-model-python while using BERT to preprocess our data. We were able to build off this existing model to implement this technique in our model as well.

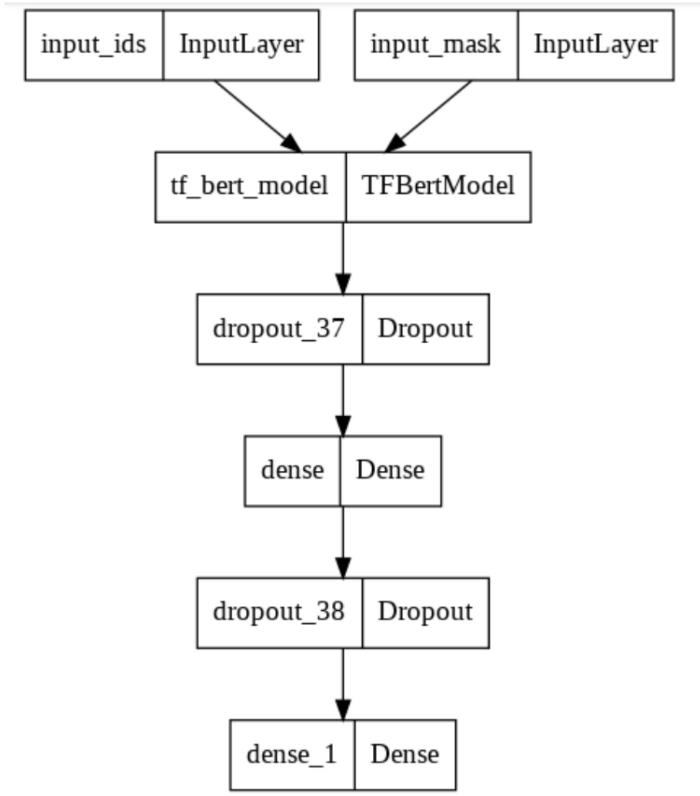

Figure 2. Diagram explaining the parts of the model and how they are related.

3 Data

3.1 Data source(s)

The data set we used for our project is available publicly at https://www.kaggle.com/c/fake-news/data. It consists of 20,800 news articles, which are labeled as unreliable (1) or reliable (0). Each news article in the data set has a unique id, title, author, text, and label. After preprocessing the data by removing articles that did not contain any text, the data set consisted of 20,671 total news articles, with 10,284 of those articles being unreliable news sources and 10,387 being reliable news sources. We chose this data set as it contains a large sample of articles, allowing us to train our model on a wide variety of news articles and capture a broad range of fake news articles. Additionally, this data set had been commonly used by others to make a deep learning model, giving us the option to compare our results with those who also trained on this data set.

We also looked at a public data set from https://github.com/KaiDMML/FakeNewsNet that contains tweets with real and fake news about politics and celebrities. Each news article in the data set has a unique id, URL of the website the article was taken from, title, and tweet ids of tweets sharing that news. However, the full data set is not allowed to be distributed due to copyright, therefore the first data set mentioned above was more useful for our project.

4 Approach

4.1 Data preprocessing

Our data contained articles as text, which we wanted to convert into word vectors to input into our LSTM network using BERT, a language model used to understand the context and meaning of words in text. We chose BERT to vectorize our articles since it is a pre-trained model for generating embedding vectors that has had significant success in various natural language processing tasks.

To preprocess our data, we first took all of the article data and dropped those with no text. This involved checking if the articles contained "na" or blank spaces as the text and removing those that did. Next, we loaded the pre-trained BERT tokenizer and BERT model. Then we went through each article in the data set and tokenized each article, returning a list of tokens for each article. We padded each list of tokens as necessary to ensure that each input list would be the same length of 512 tokens. Finally, we ran each article's token list through the BERT model, creating a vector embedding for each article and saving each embedding as an individual file.

4.2 Architecture

We used an LSTM network as our model to classify news as real or fake, since LSTMs are powerful models when used on sequential data as we have with our data set. The architecture for our network used PyTorch's LSTM function, a fully connected layer, and a sigmoid layer. The input word embeddings were passed through the LSTM function. The resulting output from the LSTM was then fed into a fully connected layer, and finally, this output was passed through a sigmoid layer, which generated a number ranging from 0 to 1 for each input. If the output for an embedding was greater than 0.5, its corresponding article was classified as unreliable, whereas if it was less than or equal to 0.5, it was classified as reliable.

5 Experiments and Results

5.1 Training and Validation

The hyperparameters we defined for our model were batch size, learning rate, and number of epochs. The data set was split into training, validation, and testing sets with a 70/15/15 ratio. After this split, there were 14,500 training embeddings, 3,100 validation embeddings, and 3,070 testing embeddings. The training consisted of a loop with forward propagation and backwards propagation. Binary cross entropy loss was calculated after the forward pass and used to update weights during the backwards pass. The validation consisted of forward propagation and calculating binary cross entropy loss to compare with the training loss. After each epoch, the accuracy for the training and validation sets were calculated.

5.2 Performance

During training and validation, accuracy and loss were used to evaluate how our model was performing and how well it generalizes to new data. A graph of our training and validation loss can be seen in figure 3. Accuracy, recall, and precision were used to evaluate the performance of our model with the testing set. In the context of this project, accuracy represents the probability that the model correctly identified whether an article was unreliable or reliable. When using our testing set, the model achieved an accuracy of about 96%. Precision measures the probability that an article classified as unreliable was correctly labeled unreliable. Our model achieved a precision of approximately 97%. Finally, recall represents the probability that an unreliable article was identified by the model. Our model achieved a recall of about 95%. The results of these metrics demonstrate that our model performed well in distinguishing reliable news articles from unreliable news articles.

Figure 3. Graph depicting training loss and validation loss over several iterations.

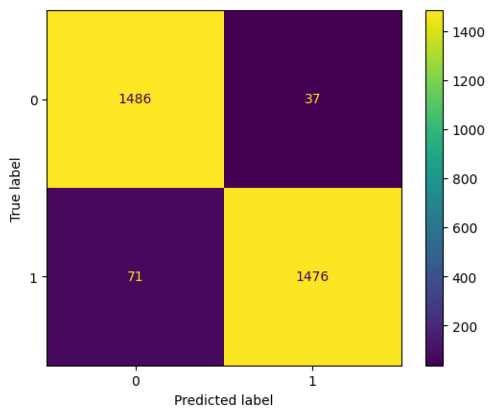

Figure 4. The Confusion Matrix depicting our model's performance on the test set.

6 Conclusion and Discussion

Overall we are very happy with the results of our model. We believe that having a deep network that can predict real or fake news with 96% accuracy can have massive implications in today's age. The 97% precision that accompanies the accuracy shows even more success in the exposure of fake news as fake. Our results place our model among 2020's top performances in terms of a good precision-recall balance as studied in Antoun et al.

6.1 Challenges

Working with a dataset of over 20,000 news articles, we ran into some challenges with how to handle the large data files. More specifically, we ran out of RAM when trying to create files with BERT embeddings output with 2,000 word embeddings each. To prevent this, we had to store each word embedding in its own file. We also were unable to download and store these files locally due to their size being 40 GBs collectively. This resulted in having to recalculate the word embeddings with every run, which is very computationally expensive compared to being able to load them in at runtime. We had similar issues while dealing with loading in the data during training. The RAM could not handle having all of the word embeddings loaded in at once, so the embeddings had to be loaded in through the dataset's 'getitem()' instead of 'init()' as usual. This loaded in one embedding at a time, which resulted in a very long training time, however in return the model performed very well within a small number of epochs because of the large amount of data it was able to train on.

7 Future Work

7.1 Short Term

With more time to work on the fake news model, we would likely make some logistical changes or tweaks to the model. As discussed in section 6.1 above, one of our major struggles was working with the size of the 20,000+ article dataset. Part of this struggle was that we could not store the BERT embeddings due to their size (40 GBs), which meant they had to be recalculated with every run. In the short term, we would try to find a way to improve the run time by storing the data somewhere capable of storing large items and then simply loading them in at runtime. In terms of tweaking the model, we could further alter hyperparameters to see if we could get even better performance. This could involve a grid search on learning rate and batch size. Finally, although our dataset is large, it is all from one source. Due to the high bias present surrounding the topic, there are two possible improvements we can make with data: to manually go through a subsample of our dataset and analyze it for biases (1), or to find more fake news data from other sources and run tests on it (2). For (2), if the tests perform poorly on the new data, we could consider training on a mix of multiple datasets.

7.2 Long Term

LSTM's introduce the idea of *attention*, but they are not the most cutting-edge models in the deep-learning community. Currently, transformers, using the idea of *self-attention*, are extremely popular for both image and text processing. These models have been producing great results, and it would be interesting to see how a transformer would perform on our BERT embeddings. BERT itself calculates these embeddings with a transformer architecture. Going beyond our architecture, there are some additional tasks that our model could be well suited for. Other binary text classification tasks, such as checking whether a review is positive or negative, would likely work very well with our model given there exists a good dataset to train on. Another extrapolation would be on spoken news. Fake news is not always written, so it would add to our goal to be able to take in audio files as well as text files for testing. This would likely require implementing a third network suited for speech to text tasks and then processing the text as we have.

8 Presentation Link

https://drive.google.com/file/d/1H9d0_UQI2005ohe5g_E3UFCmT79HeUm8/view?usp=sharing

References

[1] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. Science, 359(6380), 1146–1151. https://doi.org/10.1126/science.aap9559

[2] Watson, A. (2022). Topic: Fake news in the U.S. Statista. Retrieved March 29, 2023, from https://www.statista.com/topics/3251/fakenews/#dossier-chapter1

[3] Nasir, J. A., Khan, O. S., and Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. International Journal of Information Management Data Insights, 1(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

[4] Thota, A., Tilak, P., Ahluwalia, S., and Lohia, N.(2018) "Fake News Detection: A Deep Learning Approach," SMU Data Science Review: Vol. 1: No. 3, Article 10. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss3/10

[5] Fake news. Kaggle. (2018). Retrieved March 29, 2023, from https://www.kaggle.com/competitions/fake-news/data?select=train.csv

[6] KaiDMML. (2019). KaiDMML/FakeNewsNet: This is a dataset for fake news detection research. GitHub. Retrieved March 29, 2023, from https://github.com/KaiDMML/FakeNewsNet

[7] Abhradipta. (2020). ABHRADIPTA/fake-news-detection: Fake news detection using recurrent neural networks (rnns) amp; long short term memory (LSTM). GitHub. Retrieved March 29, 2023, from https://github.com/Abhradipta/Fake-News-Detection

[8] Rohith, M. S. (2022). Fake news detection using Bert Model Python. Retrieved from https://towardsai.net/p/l/fake-news-detection-using-bert-model-python

[9] W. Antoun, F. Baly, R. Achour, A. Hussein and H. Hajj, "State of the Art Models for Fake News Detection Tasks," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 519-524, doi: 10.1109/ICIoT48696.2020.9089487.