# Fake News Detection Using Machine Learning - Technical Report

By: Collin Kerker

Date: 12/1/2025

## Introduction

The goal of this project is, as the title describes, to detect fake news using machine learning, specifically to use a BERT transformer to differentiate between the two. This matters for many reasons, but specifically because of how important and relevant it is today. There is a large amount of false information spread across the internet because of its nature, that being that anyone can post anything to it at anytime and anywhere. And since millions of people can see any single piece of information at any time online, false information can be critically harmful with how much attention it is able to receive without fact checking.

To achieve this goal, I will be using the Kaggle dataset named "Fake and Real News Dataset." This dataset gives 4 columns with around 40,000 rows of data, each row gives the news data a title, the actual content of the news, what type of news it is, whether it be general political or something else, and it also gives the date that the news was published. The model I will be using is a BERT transformer model, specifically the bert-base-uncased model, as it is pre-trained on a large amount of textual data that gives it the right qualities to efficiently learn from the dataset that I will train it on. I will also be comparing this main model to a baseline model that will be used to determine whether BERT is a good way of detecting fake news or not. The baseline model uses TF-IDF and Logistic Regression to train off the data from Kaggle.

This project is also significantly ahead of my midterm project, not only in the complexity of the model, but now using a deep learning transformer instead of a shallow learning model. It also comes with a more difficult problem in general. More on the model, BERT alone, with it being able to capture bidirectional context and semantic relationships already makes it 100 times more capable and complex than the shallow learning model that I used back in the midterm. This along with the understanding of the information that I have now, I will be able to make this project more in depth and structured than the midterm overall.

Overall, this report will document the methods used, the results obtained, and the conclusions that I drew about the performance of the BERT transformer model.

## Related Work

Fake news detection is a widely researched field, taking only a quick search into google scholar to give me a vast number of papers of people doing similar research as me. And their projects, albeit more difficult and complex than mine, show me that this is a very good topic to go into with a lot of resources. In the first of them, Kaliyar et al. [1] explains in the FakeBERT paper how, instead of just using BERT on its own for the news detection, they also combined it with a 1D-CNN. Their setup was three parallel blocks of 1D-CNN's each with different kernel sizes so that each would capture their own unique patterns in the text. This allows for the BERT model to capture deep context while the CNN's capture the local context, and it worked amazingly well as it got about a 98.90% accuracy. Compared to mine it did get an overall less accuracy, but from what I can tell is that it is likely that they had much more tuned and "better" hyperparameters that made it so that their model did not overfit off their data. Overall, this is a good showing of how more advanced research often involves more complex and in-depth

models, like this one that combines two models, and for a future continuation of this project I can maybe do something similar if I want to be able to try and get more accurate results.

Secondly, I found a paper written by Shishah [2] that uses a "Joint Learning" approach, and I think is also called as such too. The idea of this person's design is that they would focus in on texts specific properties instead of just giving the model the entire text as a block and say to read it. The properties that they would home in on would be properties specific to certain texts, such as the names of politicians and how they relate to the text around them. This model not only performed very well but it also outperformed the previous example that I showed as it was able to handle long texts much better because of how the model was able to focus on the important parts of the news instead of just getting a jumble of all the words on the page because there was no specification in its training. This is almost similar to the BERT model that I used as it was trained on large English texts, and because of the HuggingFace trainer and tokenization used on the texts beforehand, with it giving specific and important words more weight than unimportant words. This means that my trainer would most likely be able to handle long-form text and it being able to do that might be a result of people realizing this concept of lengthiness and dealing with it from Shishah's report.

Lastly, I found a third interesting report from Szczepanski et al. [3] in the paper named "Explainability." Which just gave a very interesting deep dive into the model itself, as the papers main idea is that it is not enough to even have a model with 99% accuracy if we do not actually know why it is able to get that score in the first place, and how it gets to the conclusions it does. The author does this by using a tool called LIME to highlight exactly which words in the model were looked as mainly by the model. For example, the model correctly flagged a fake news article as fake by latching onto the phrase "just gave a wake up call," which is emotion language often found in fake news articles. This is extra interesting as I also had the same sort of questions, as I know that the model that I trained is VERY VERY good at determining if a news article is fake or not in theory sure, but as to why the news is fake that it did flag that way, I really do not know, so seeing this shows me that there is good reason as to why it was so good and that I can trust the model beyond thinking that the model may just be cheating by just looking at other tells that might fail when looking at variations of news articles, and just learning off of that.

**Methods**

For building all the models I used a variety of different techniques, although none too complicated. For the data preparation, I merged the fake and true CSV data files for ease of processing for both the baseline and main BERT model. I also did general cleaning on the files, such as removing duplicates, removing null values, merging title and text columns into one field, removing unnecessary textual clutter, and making the train/test split be 80/20.

For the Baseline model I used TF-IDF vectorization with a Logistic Regression classifier. I used TF-IDF because it effectively captures the important and meaningful words while also downplaying the meaningless and common words in each article. And it works well on text problems such as where there is a big jumble of words that some of the times do not really mean anything. I used Logistic Regression because of how fast and well it works on large datasets, and it also can handle the output given from the TF-IDF vectorization, as it produces thousands of features. Overall, both work very well with text problems and they can both be reliably compared with more advanced and complicated models, such as the main BERT model.

Moving on to the tokenization, I used tokenization so that the BERT model would be able to process the data given to it though the Kaggle dataset. I used a length of 256 tokens to balance training speed and the constraints of memory that BERT has, that being 512 as the very max limit. I also chose the amount of 256 tokens as it still captured the main and important contexts and meanings from the text to properly train off them. Bert-base-uncased was used because of how it is already pre-trained on a vast amount of English textual data that makes it perfect for capturing context and word meanings.

As for BERT tokenization, specifically converting the dataset to a HuggingFace dataset, I did this because I need to convert it to be able to use the HuggingFace trainer. I also did this because it gives many benefits like faster preprocessing, easy tokenization for large datasets, and it has compatibility with BERT models. I used batch tokenization because it is far more efficient for larger datasets. The PyTorch tensors are for use by the BERT transformer model, as the model is built in PyTorch so it needs the PyTorch tensors to work.

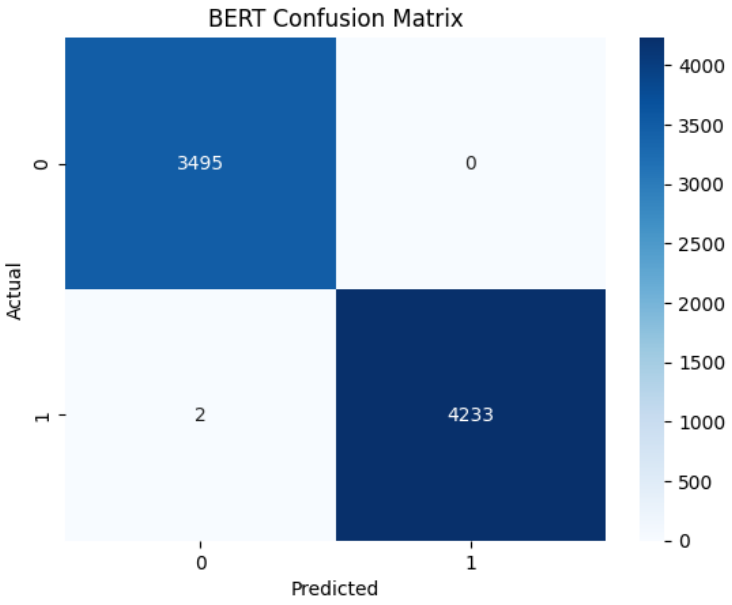
For the BERT model setup, I am, again, using bert-base-uncased because it is pretrained on vast amounts of English textual data and therefore is very good for the task at hand. It is specifically good at interpreting contextual and word meaning for text, giving another reason as to why this is the perfect model and that is the entire goal of what I am trying to give it. I gave it two labels as there is only two needed, 0 for if the news is predicted fake and 1 if the news is predicted true. The loss is already handled by the HuggingFace trainer as it automatically applies the appropriate loss function based on the number of labels.

Finally for the training setup, I used three epochs, the batch size is 8, and the learning rate is 2e-5. I chose three epochs because more than 3 would become way too sluggish and slow while on a CPU, so three were chosen as the sweet spot. I chose 8 as the batch size as BERT is very memory intensive and so anything bigger would have made it crash or run out of memory on the CPU. The learning rate was chosen because this is the recommended learning rate for fine-tuned BERT, and because BERT is very sensitive it was important to use the right learning rate, meaning that the recommended was very advisable.

**Results**

The results for both models were extremely strong, with the BERT model coming out on top and the baseline model not falling too far behind. The baseline model achieved an accuracy of 0.986 and an F1 score of 0.987. These scores are very good, especially for a simple TF-IDF and Logistic Regression approach. The results show that the baseline is highly effective and gives a good reference point for evaluating the BERT model.

The BERT transformer was able to achieve even better scores, with an accuracy score of 0.9997 and an F1 score of 0.9997. This means that the BERT transformer model was able to almost perfectly classify both real and fake news articles in the test set.




Finally, the confusion reflects these almost perfect scores in picture form. Out of all of the fake news articles 3495 were correctly predicted to be false, and 0 were falsely predicted to be true. This means that for the fake articles the BERT transformer got 100% percent accuracy. Out of the true news articles, the dataset correctly predicted 4233 to be true, while it only predicted 2 to be false when they were actually true. This is also almost perfect marks.

**Figure 1: Confusion Matrix for the BERT Model**

These results show that the BERT model was very effective at classifying the news articles and was most certainly the right choice for the picked model on this problem.

**Discussion**

There are honestly many reasons as to why both performed so well almost to the point of perfection. Getting right into it the baseline performed so well for two main reasons in my eyes, the first being that TF-IDF works very well for news style texts. It does because news articles are not only long enough for TF-IDF to see patterns in the articles, but it is also able to latch heavily onto the vocabulary used, as fake news articles usually use similar vocabulary like emotional words, trigger words, or political trigger words. These two facts make the word-based features very informative. This is coupled with the fact that Logistic Regression also handles the high-dimensional vectors from the TF-IDF very well, allowing the TF-IDF to fully work.

As for BERT, it performed even better and there are again good reasons for this performance. The biggest reason that I can see is how the BERT model that I used, bert-base-uncased, is pretrained on a very vastly on the English language, using sources like Wikipedia. This means that it can understand context in the words and that it understands relationships between words. These two factors alone are huge as they both contribute greatly to how well BERT would be in classifying fake or real news articles when all it must go off are the context clues and the words relationships.

Now looking at the datasets, both were nearly perfect, which does honestly make me very skeptical about whether I made a mistake or not, but I think that there are good reasons as to why the accuracy for both were almost perfect. Firstly, the dataset is very good, and by that, I mean that it has very clean and well-labeled data, and because of the light cleaning that I did as well, there are no null values or any major irregularities in the data, leading for the data to be very easy to use and interpretable. Secondly, because for reasons that I will mention in the limitations paragraph, the task was certainly too easy for a transformer, especially the BERT transformer. And Lastly because of how easy it was, BERT was able to get deep and meaningful patterns from the texts, and that just means that it would be able to perform very well, which it did.

| Step | Training Loss |
|---|---|
| 500 | 0.013000 |
| 1000 | 0.002300 |
| 1500 | 0.008700 |
| 2000 | 0.006900 |
| 2500 | 0.005000 |
| 3000 | 0.004500 |
| 3500 | 0.002800 |
| 4000 | 0.009300 |
| 4500 | 0.005600 |
| 5000 | 0.004400 |
| 5500 | 0.007100 |
| 6000 | 0.000000 |
| 6500 | 0.000000 |
| 7000 | 0.000000 |
| 7500 | 0.000000 |
| 8000 | 0.000000 |
| 8500 | 0.000000 |
| 9000 | 0.000000 |
| 9500 | 0.000000 |
| 10000 | 0.000000 |
| 10500 | 0.000800 |
| 11000 | 0.000000 |
| 11500 | 0.000000 |

As for the limitations, firstly, the training on the CPU is very slow and that caused a few big problems. The problem being that I could not run the trainer again because of the time constraints that I had in doing this project along with other projects for other classes. So being held back by not being able to rerun the trainer meant that I had to pull my results from the HuggingFace checkpoints instead of the trainer after I closed and reopened the main Jupyter file where I trained the model. And secondly, this problem leads right into the main biggest problem that I ran into, and what I mentioned in the last paragraph, that BERT was far too advanced for this problem, I know this because about halfway through training it entirely overfitted the data and had zero loss, and that meant that for my results I had to pull from the checkpoints halfway through the outputs from the trainer rather than actually just getting the last result from the trainer.

**Figure 2: HuggingFace trainer checkpoints overfitting mid model training**

Overall, the results were much better than I expected as I really did not think that it would get pretty much perfect accuracy. This showed me that the BERT transformer is a lot more powerful than I thought it was when I went into this.

## Conclusion

In conclusion, I was able to make and use a baseline and BERT model to compare against each other in detecting fake vs real news. Out of everything the most important takeaways were that the baseline and BERT performed exceptionally well. These results mean a few things, but the main thing that these results show me is that this was almost perfectly suited for the setup baseline model, but that this problem was far too easy for the BERT model. Something that I can do in the future to improve this is to get a more complex and difficult set of new articles that would make BERT struggle more, that and to have vastly more data that just 40,000 rows.

## References


1. [1] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimedia Tools and Applications*, vol. 80, pp. 11765–11788, 2021.
2. [2] W. Shishah, "Fake News Detection Using BERT Model with Joint Learning," *Arabian Journal for Science and Engineering*, vol. 46, pp. 9115–9127, 2021.
3. [3] M. Szczepański, M. Pawlicki, R. Kozik, and M. Choraś, "New explainability method for BERT-based model in fake news detection," *Scientific Reports*, vol. 11, no. 23705, 2021.