Issues in Information Systems
Volume 24, Issue 2, pp. 164-177, 2023

DOI: https://doi.org/10.48009/2_iis_2023_114

The efficacy of detecting AI-generated fake news using transfer learning

Jake Stewart, *Purdue University Northwest, stewa277@purdue.edu*
Nikita Lyubashenko, *Purdue University Northwest, nlyubash@purdue.edu*
George Stefanek, *Purdue University Northwest, stefanek@purdue.edu*

Abstract

This research investigates the detectability of AI-generated fake news by a fine-tuned BERT transformer that does text classification. The performance of the model is evaluated by testing it on GPT-generated (Generative Pre-trained Transformer) news articles as well as real-world news articles from a Kaggle dataset. The study focuses on how fake news articles can be created by AI tools and how effective a fine-tuned, open-source large language model is at detecting the GPT-generated fake news. The model was fine-tuned with Kaggle real-world articles and GPT-generated articles from OpenAI ChatGPT and Davinci. The use of state-of-the-art tools like the BERT pre-trained, large language model transformer were found to effectively classify GPT-generated articles but were less effective at classifying fake news articles that were not GPT generated.

Keywords: machine learning, GPT, transfer learning, BERT, LLM, fake news, AI

Introduction

The goal of this research was to determine the effectiveness of a fine-tuned, large language model transformer to classify GPT-generated (Generative Pre-trained Transformer) news articles as either fake or true. A pre-trained BERT transformer (Bidirectional Encoder Representations from Transformers) large language model (LLM) was fine-tuned and used for classifying the news articles. The model was fine-tuned (i.e., additionally trained) using news articles that were labeled as either real or fake from three data sets containing news articles from real sources downloaded from Kaggle and with articles generated by OpenAI's ChatGPT and Davinci. The fine-tuned BERT model was tested using articles from the same datasets.

Transfer learning is the practice of leveraging pre-trained machine learning models to fit a problem and dataset specific to something else. The main benefit in doing so is that the weights of the pretrained model are oftentimes very precise, having been trained on extremely large datasets. Transformers like the BERT transformer are state-of-the-art tools which are the newest deep learning models that are extremely powerful, built by big-tech companies such as Google, Meta, and OpenAI using billions of gigabytes of training data. For example, GPT-3 has 175 billion trainable parameters and was trained on about 45 TB of text data from multiple sources that included Wikipedia and books (Brown, T. et. Al., 2020). These models have been made available for public use, which is a huge benefit for the machine learning community because these models greatly improve natural language processing (NLP) in question answering (e.g., Chat scenarios), reading comprehension, and textual entailment over previous models and techniques primarily due to the use of massive training sets and many trainable parameters.

This study focused on the topic of fake news detection because of the ability of online and social media to potentially spread misinformation to its users. This study investigates how the latest models can be used to help detect AI-generated real and fake news articles. Described in this paper are the model used, the model's metrics, fine-tuning of the model, the methods to build the pipeline, results from classifying against several different data sets, analysis of the results, and conclusions, insights, useful tips and recommendations.

Literature Review

The study conducted by Gao, et al. (2020) looked at how well OpenAI's ChatGPT model was at generating scientific research abstracts based on the title and journal. This paper was relevant to our study since it looked at the ability of a ChatGPT-like model to generate articles and test them using an AI trained model. Ten research abstracts were gathered from five high-impact medical journals and used as a reference to evaluate the generated abstracts. The results that were gathered showed that the articles were clearly written but only 8% of them followed the formatting requirements. Most of the generated abstracts were easily detected using an AI output detector but were found to have a high score of originality using a plagiarism detector. However, when evaluated by human reviewers, 68% of the generated abstracts were correctly identified, but 14% of the original abstracts were mistakenly identified as being AI-generated. The study concluded that ChatGPT can write believable scientific abstracts with completely fake data.

A study by Zellers, et al. (2019) focused on building a model specific to identifying online disinformation or fake news. The authors built an AI model called Grover to address the problem of growing fake news. It was designed to detect fake news generated by other AI models. The authors found that the best way to detect fake news is to use an AI model that can also generate news articles. This is relevant to our study in that our study also fine-tuned a model to detect AI-generated fake news. Grover was trained on publicly available news and was able to spot its own generated fake news articles, as well as those generated by other AI models. Grover achieved over 92% accuracy in detecting human-written and machine-written articles. Overall, the authors suggest that Grover can be an effective tool in detecting fake news.

Ozbay and Alatas (2020) performed a study where they used twenty-three supervised artificial intelligence algorithms to look for fake news on social media. In this study, text is mined from online social media news articles. The twenty-three supervised learning models that were used were BayesNet, JRip, OneR, Decision Stump, ZeroR, Stochastic Gradient Descent (SGD), CV Parameter Selection (CVPS), Randomizable Filtered Classifier (RFC), Logistic Model Tree (LMT), Locally Weighted Learning (LWL), Classification Via Clustering (CvC), Weighted Instances Handler Wrapper (WIHW), Ridor, Multi-Layer Perceptron (MLP), Ordinal Learning Model (OLM), Simple Cart, Attribute Selected Classifier (ASC), J48, Sequential Minimal Optimization (SMO), Bagging, Decision Tree, IBK, and Kernel Logistic Regression (KLR)).

Data preprocessing of the unstructured new articles consisted of tokenization which divided the text into smaller parts (tokens) and removed all the punctuation from the text. Numbers, punctuation, and words less than N number of characters were removed. Case was converted, stop words were removed (i.e., words frequently used to complete sentence structure and connect expressions such "are", "a", "that", "but", "when", "where", "what", etc.). Finally, stemming consisted of transforming words into a root form (such as "connection", "connective", "connected" converted to "connect").

The next step consisted of feature extraction from the pre-processed data after which the terms in the data set for each document were weighted and each document converted into a vector of term weights. Next the various supervised artificial intelligence algorithms were trained from the structured training set where the labels of the instances in the data sets are already known. The performance of the various supervised algorithms was evaluated according to accuracy, recall, precision, and F-measure values and displayed in a confusion matrix. There were several datasets from different news sources used. The Decision Tree algorithm, ZeroR, CVPS, and WIHW algorithms performed the best.

Aldwairi and Alwahedi (2018) performed a study to detect fake news in social media networks by carefully selecting features from the news post that accurately identifies fake posts. A logistic classifier was used to perform the detection. Their experimental results show a 99.4% accuracy of detecting fake and real news. The software system they designed starts by analyzing web pages returned by a browser, after a user inputs search terms, to identify sites whose links contain words that may have a misleading effect on the reader. Next the system examines the title of a news article for the number of words in the title. Those titles with many words (e.g., over some threshold such as eight) are often used as clickbait and thereby are considered as potentially fake news. The tool also monitors how punctuation marks are used in web pages. The model flags sites whose headlines contain extensive usage of exclamation marks and question marks and designates those sites as potential clickbait.

Additionally, bounce rates to other sites are used as an indicator of the quality of information. For instance, links to sites with very different content. The methodology consisted of first creating a clickbait database by crawling the web to collect URLS for clickbaits and social media sites that are likely to have more fake news or clickbait ads. The second step was to create a script that computed the attributes from the title and content of the web pages, extract the features from the web pages such as keywords in English, titles that start with numbers, capitalized words, and question marks and exclamations marks. The BayesNet, Logistic, Random Tree, and Naïve Bayes classifiers were used from the WEKA machine learning library to perform classification. The Logistic classifier had the highest precision at 99.4% and therefore the best classification quality. Logistic and Random Tree had the best recall at 99.3%.

Gereme and Zu (Gereme & Zhu, 2019) performed a study to detect fake news using a Naive Bayes, Deep Convolutional Neural Network and Long Short-Term Memory (LSTM) neural network. The data sets were created from the Kaggle dataset that contained real-world articles and the George McIntire dataset. These two data sets were concatenated and articles shuffled randomly to create a combined dataset. The typical evaluation metrics of accuracy, precision, recall and F1 were used. The code was written in Python using the TensorFlow, NumPy and Keras libraries. The dataset was preprocessed by removing unlabeled data, very short articles were ignored, stop words were removed, and the Keras Tokenizer class was used to split the text data with space, filter out punctuation marks, and change all text to lowercase. The convolutional neural network model had the best performance with above 94% accuracy on the combined Kaggle-George McIntire dataset. This was followed by LSTM with 90% accuracy and Naive Baye's with over 89% accuracy.

Girgis er al. (2018) performed a study where they used deep learning for detection of fake news. The classifiers used were the Recurrent Neural Network (RNN), GRU, and LSTM models. The dataset used was called LIAR, a publicly available dataset for fake news detection that consists of 12.8K manually labeled short statements from POLITIFACT.COM. Data was preprocessed to split each sentence, remove stop words, and perform stemming. A Vanilla single layer recurrent neural network, a Gated Recurrent Unit (GRU) neural network and a LSTM (Long Short-Term Memories) network were trained and tested against an extracted test set of articles from the LAIR dataset. The GRU classifier performed the best of the three. However, compared to other results from Wang (2017), a CNN classifier did better than the GRU.

Mahir et al. (Mahir, et al., 2019) performed a study to detect fake news using five machine learning algorithms. The machine learning algorithms used for classification were Support Vector Machine, Naive Bayes, Logistic Regression and Recurrent Neural Network models. This study used a dataset of Twitter threads that included information about the 2010 Chile earthquake that had 20,360 data items that were labeled as either fake or real. Preprocessing included creating feature vectors of a count vector which represented each document by each row, each column represented a corpus term and each cell represented the frequency count of a particular term in the document. Next, the pre-processing created TF-IDF information used in text mining which represented how frequently a term is found in the entire document with a metric value assigned to represent the presence of a term. Additional feature vectors included Word Level TF-IDF, N-gram Level TF-IDF, and Character Level TF-IDF. The results found that SVM and Naïve Bayes classifiers outperformed the other algorithms.

Manzoor, et al. (2019) performed a systematic review of machine learning algorithms for the detection of fake news. They looked at Naïve Bayes, Decision trees, Support Vector Machines (SVM), Neural Networks, Random Forest, and XG Boost. They also looked at various types of fake news generation strategies such as visual-based, user-based, knowledge-based, style-based, and stance-based.

Conroy, et al. (2015) also did a survey paper on methods for finding fake news. The paper looks at veracity assessment methods including linguistic cue approaches with machine learning and network analysis. Also, they looked at a hybrid approach that combines linguistic cues and machine learning with network-based behavioral data. They also proposed operational guidelines for creating a feasible fake news detection system including how to represent data, the analysis of word-use along with tools for doing this, semantic analysis and rhetorical structure and discourse analysis. Finally, they concluded that linguistic and network-based approaches have shown high accuracy within limited domains. They found that linguistic processing should be built on multiple layers from word analysis, network behavior should be combined to incorporate a trust dimension by identifying credible sources, tools should be designed to augment human judgement, and that publicly available gold standard datasets should be in linked data format to assist in up-to-date fact checking.

Statement of the Problem

The goal in this study was to determine the efficacy of a fine-tuned transfer learning model such as BERT in detecting fake news articles generated by GPT models. Additionally, the efficacy of the model would also be tested on real-world articles from a Kaggle dataset. During our preliminary research, we found multiple studies online which used AI models for detecting fake news.

Our study was different in two ways:

1) this study used model-generated news articles using Davinci GPT-3 and ChatGPT along with real news articles from the Kaggle dataset that were scraped from real news sites

2) this study used a pre-trained, large language model BERT transformer for the detection of fake and real news articles

This study does not contribute to theory but shows the results of using one of the latest models (i.e., BERT which uses bi-directional encoder representations from transformers architecture) to determine how effective it is at classifying AI-generated fake news. The BERT classifier is fine-tuned with AI-generated fake and real news from OpenAI's Davinci and ChatGPT and from a news dataset from Kaggle. Previous work primarily used models such as logistic regression, Naïve Bayes, Convolutional neural networks, Recurrent neural networks, and Long Short-Term Memory neural networks for classification.

Methodology

The methodology for the study was as follows:

1) identify sources of pre-compiled datasets that contain fake and real news from on-line data repositories

2) search for and down-select at least one dataset from the identified on-line repositories that contain news stories from a variety of mainstream news sources and has data from at least thousands of news articles

3) select a widely used large language model that uses a transformer encoder architecture that will be fine-tuned for fake and real news article classification

4) determine the number of data items (i.e., news stories) to use for fine-tuning the model from each pre-compiled dataset

5) identify and select large language models to be used for generating fake news

6) determine input prompts for generating fake news from the large language models

7) create AI-generated datasets from the large language models

8) create training sets with equal input from all data sources for fine-tuning the transformer architecture LLM

9) create test sets for testing the fine-tuned LLM

10) test the LLM classifier for its effectiveness at classifying AI-generated fake news and real news

11) show results using confusion matrices

12) perform analysis by comparing percent effectiveness from the various test sets from various sources.

The large language model used to classify the news articles as fake or real was a pre-trained BERT model, specifically the ‘bert-based-uncased’ model from the Huggingface ‘transformers’ library. BERT was trained using 3.3 billion words with 2.5 billion coming from Wikipedia and 800 million from BooksCorpus (Muller, 2022). We selected BERT because it has proven to be highly effective for various NLP tasks including text and news classification.

Trying additional classifiers such as ChatGPT, logistic regression and naïve bayes was considered, but since previous work had already looked at logistic regression and naïve bayes the focus was on using only BERT. We fine-tuned the model which has already been trained on massive amounts of text data, so that it can learn representations of words and phrases often seen in news articles and articles with misinformation. By fine-tuning on a labeled dataset of both real and fake news, the model can learn to make distinctions between the two classes.

After initial testing of the BERT model and validation of it for functionality, we gathered data for fine-tuning. Real news article data came from a dataset from Kaggle. The specific Kaggle news article dataset contained news articles from Wikipedia, Reuters, and POLITIFACT. Two additional datasets were created of AI-generated news articles from OpenAI's Davinci and ChatGPT models.

Real news was defined as news that is truthful regardless of its author. The prompt for generating real news is: "Generate a (one of three topics) news article, keep it below 510 words." The Kaggle news datasets use various methods to determine fake and real news either by manual fact-checking by experts or websites, looking on linguistic cues or network analysis of the news content, and some articles are based on machine learning models that learn from labeled data.

Article generation by OpenAI's Davinci and ChatGPT went through many iterations to tweak the quality and consistency of the articles to our criteria. The first criterion was whether to include the headlines of the articles. Headlines were deemed valuable since they are the first thing readers see and, in many cases, determine whether the reader will click on the article.

Another factor in the decision to keep the headlines was that the language models automatically generated headlines. The OpenAI ChatGPT model was used to generate datasets of fake news. Initially, it was not possible to get ChatGPT to generate "fake news" as it violated OpenAI's guidelines for the model. However, it was possible to "bypass" the model restrictions by using alternative terms such as "imaginary" and "pretend," which generated some promising model-generated articles. Another OpenAI model called "test-davinci-003" did not have the same restrictions as ChatGPT, so it was used to generate articles with "fake news" into a dataset.

Finally, OpenAI released ChatGPT as a query-able model in their API, which did not have the restrictions that its web-app counterpart did and thereby allowed for fake news generation into another dataset. Table 1 shows the number of articles that were model-generated and used in training (fine-tuning) and testing the BERT model. The dataset sizes were partially arbitrary and to create a balanced dataset between the three sources with regard to the origin of a given article. The total number of selected articles from all of the three sources was 8634 with a training/test split of approximately 80/20 which was close to the 2,400 articles for training and 478 articles for testing for each dataset." It was decided to use equal contributions of data from the three sources for this study to eliminate the influence of one dataset over another.

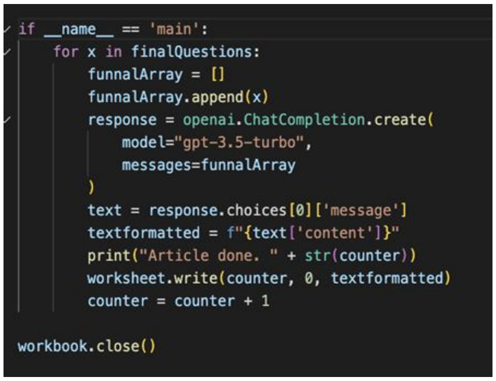
Table 1: Number of Articles for training and testing.

Article Sources	Training Size	Test Size
Kaggle (Reuters, Politifact, Wikipedia)	2,400	478
OpenAI Davinci (model-generated)	2,400	478
OpenAI ChatGPT (model-generated)	2,400	478

The next step was to do "prompt" engineering and formulate questions that would yield articles that were not only believable, but also have a balanced distribution of topics. Questions were split into three different topics - argumentative, persuasive, and entertainment. BERT also had a 512 token limit per text block, so we made sure to specify within our prompts that articles should not be longer than the token limit. The final fake news prompts combined all these factors into one sentence such as “Generate a (one of three topics) fake news article, keep it below 510 words.” No significant differences were identified between false negatives and false positives between the 3 different topics. Next, a Python script was written to take these questions, format them in a way that the API would accept and capture the model’s response. To do this, the exact number of articles that we needed for training and testing had to be determined. Initially our Kaggle training set consisted of 21,000 real and 18,000 fake articles compiled from real world sources.

Initially, there were 300 real and fake model-generated articles for testing. This did not generate the results that we wanted so we decided to insert model-generated articles into the training set. Our final training set included 2400 articles from real world sources in the Kaggle dataset, 2400 generated by text-davinci-003, and 2400 articles from ChatGPT with 1200 of the articles being fake and 1200 being real for each model. Our test set consisted of 478 articles from each of the three sources with half fake and half real.

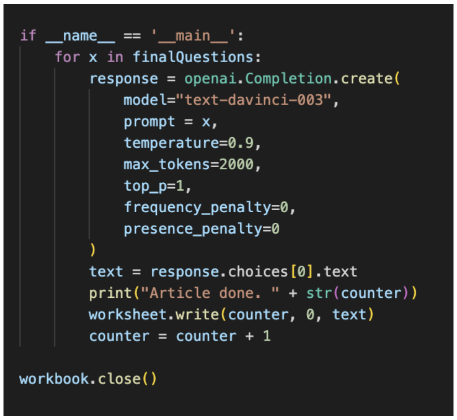
A python script was created to query the API with our prompts. After the results came back, we ran the article corpus through a formatting script to remove any unnecessary characters or lines and insert an article into a “tab-separated values” file for ease of use with a Pandas DataFrames python library. Articles would then be converted from .xlsx to .tsv files through a cloud-based service. The code for standardizing the format for the model-generated articles is shown in Figure 1 and Figure 2 and Figure 3 illustrates the pipeline for processing.



```python
if __name__ == 'main':
    for x in finalQuestions:
        funnalArray = []
        funnalArray.append(x)
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=funnalArray
        )
        text = response.choices[0]['message']
        textformatted = f"{text['content']}"
        print("Article done. " + str(counter))
        worksheet.write(counter, 0, textformatted)
        counter = counter + 1

workbook.close()
```

Figure 1: ChatGPT Formatting Script



```python
if __name__ == '__main__':
    for x in finalQuestions:
        response = openai.Completion.create(
            model="text-davinci-003",
            prompt = x,
            temperature=0.9,
            max_tokens=2000,
            top_p=1,
            frequency_penalty=0,
            presence_penalty=0
        )
        text = response.choices[0].text
        print("Article done. " + str(counter))
        worksheet.write(counter, 0, text)
        counter = counter + 1

workbook.close()
```

Figure 2: Davinci Formatting Script

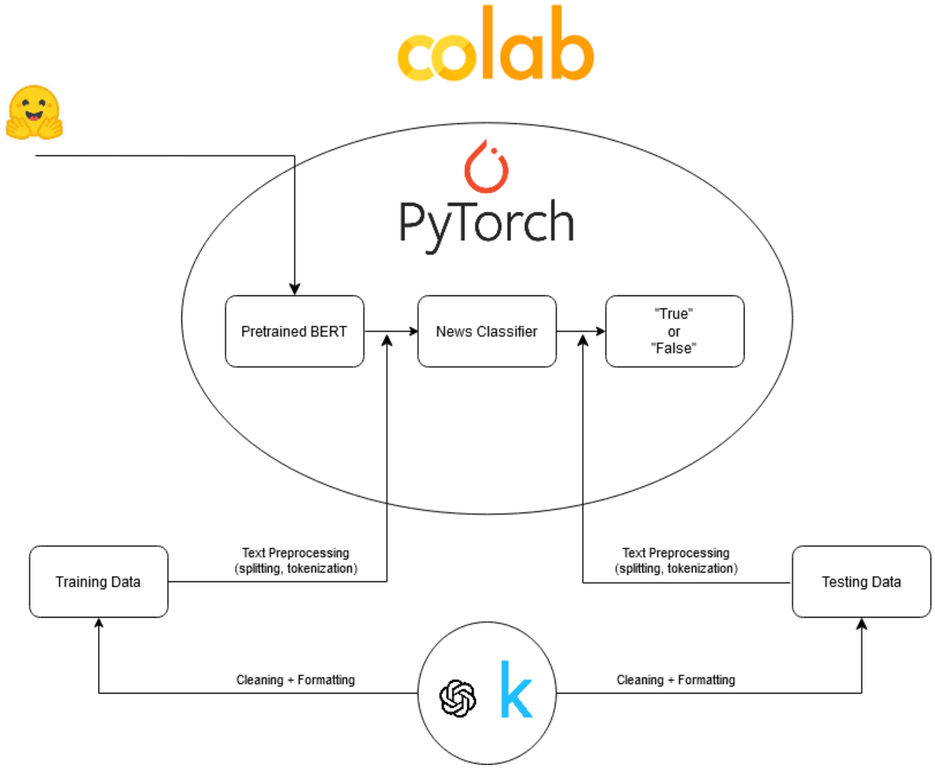


Figure 3: Processing Pipeline.

Results

The results are shown using confusion matrices. Confusion matrices show the results from model classification by the matrix comparison between the True Positive, True Negative, False Positive and False Negative results. An example is shown in Table 2.

Table 2: Example Confusion Matrix Layout

		Predicted	
		Real Articles	**Fake Articles**
Actual	**Real Articles**	True Positive	False Negative
	Fake Articles	False Positive	True Negative

Figure 4 shows the results of classifying Kaggle fake and real articles. There was a total of 239 fake articles and the model was able to correctly classify all of them as truly fake. This resulted in an accuracy score of 100%.

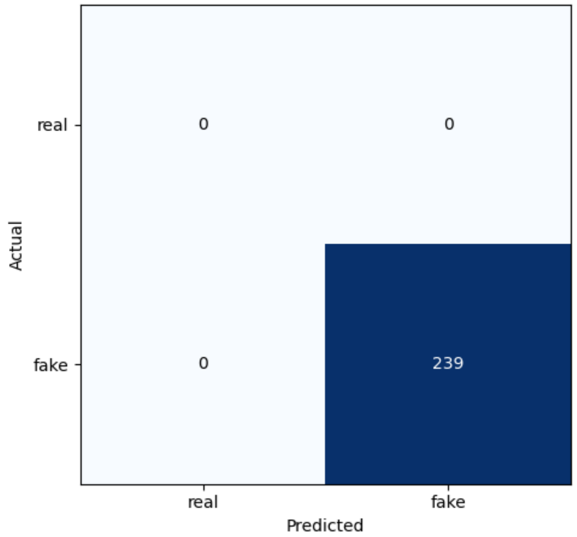


Figure 4: Kaggle Fake Article Results

In Figure 5 there were a total of 239 Kaggle true articles and the model was able to correctly classify 118 of them being truly true. The model incorrectly predicted 121 of them as being fake. This results in an accuracy score of 49%.

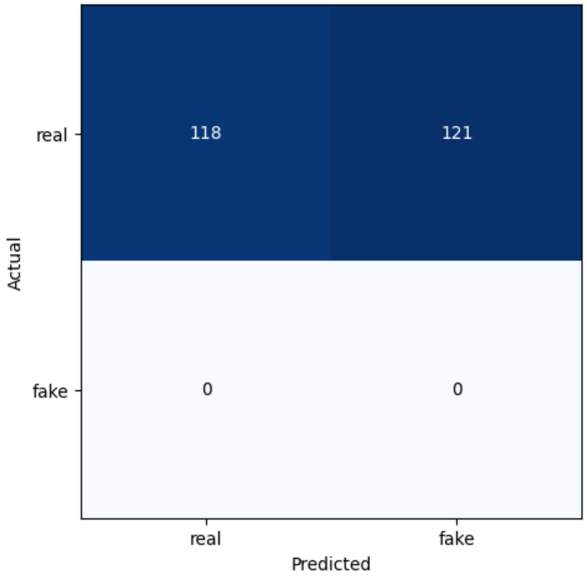


Figure 5: Kaggle True Article Results

Figure 6 shows the results for classifying AI-generated fake articles using ChatGPT. There was a total of 239 fake articles and the model correctly classified 201 of them as fake. The model incorrectly classified 38 of them as real. This results in an accuracy score of 84%.

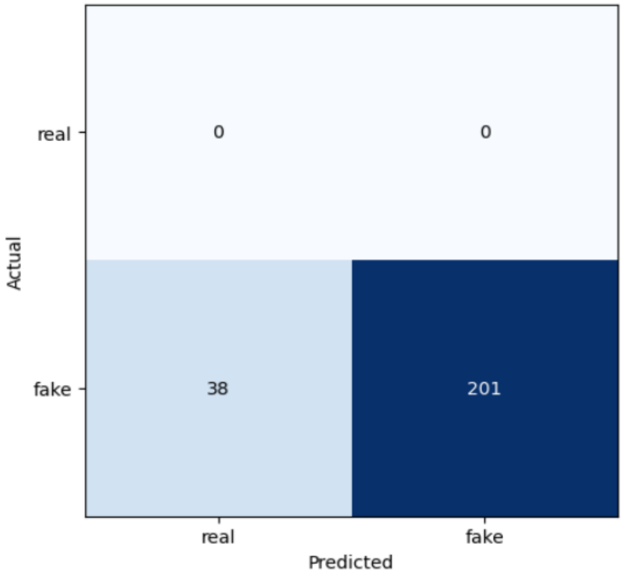


Figure 6: ChatGPT-generated Fake Article Classification Results

Figure 7 shows the results for ChatGPT-generated true articles. There was a total of 239 true articles and the model was able to correctly classify 51 of them as true. The model incorrectly classified 188 of them as fake. This results in an accuracy score of 21%.

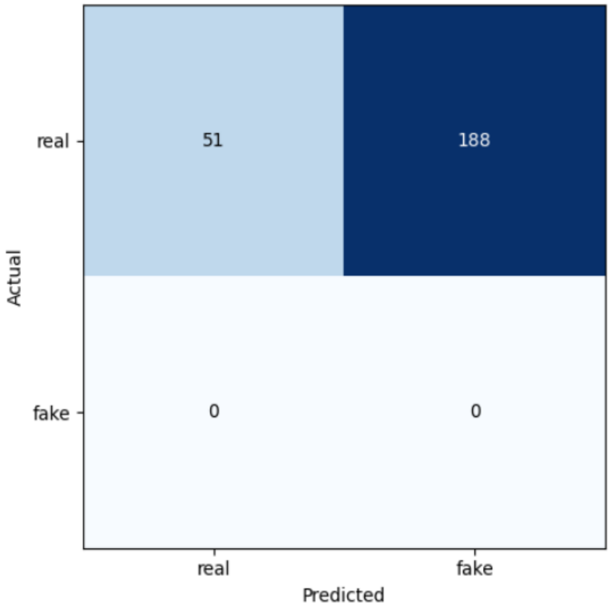


Figure 7: ChatGPT-generated True Article Classification Results

Figure 8 shows the results from Davinci-generated fake article classification. There were a total of 239 articles and the model was able to correctly classify 215 of them as fake. The model incorrectly classified 24 of them as real. This results in an accuracy score of 90%.

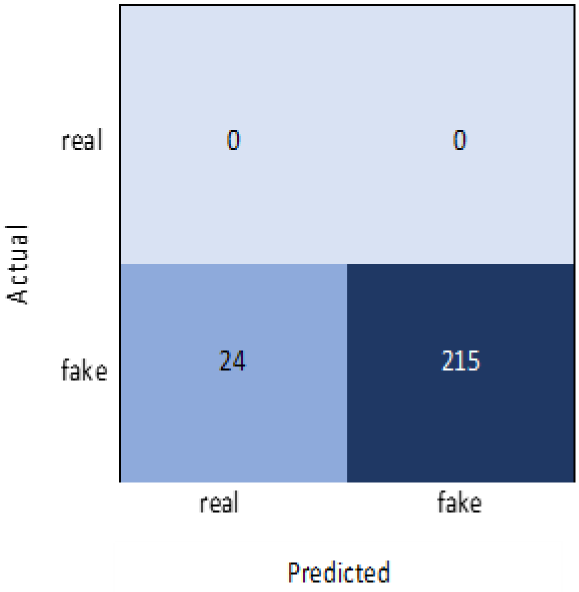


Figure 8: Davinci-generated Fake Article Classification Results

The final set of results was for classification of Davinci-generated true articles shown in Figure 9. There was a total of 239 articles and the model was able to correctly classify 51 of them as true. The model incorrectly classified 188 of them as fake. This results in an accuracy score of 21%. The overall results are shown below in Figure 10 in an aggregate confusion matrix and in tabular form in Table 3.

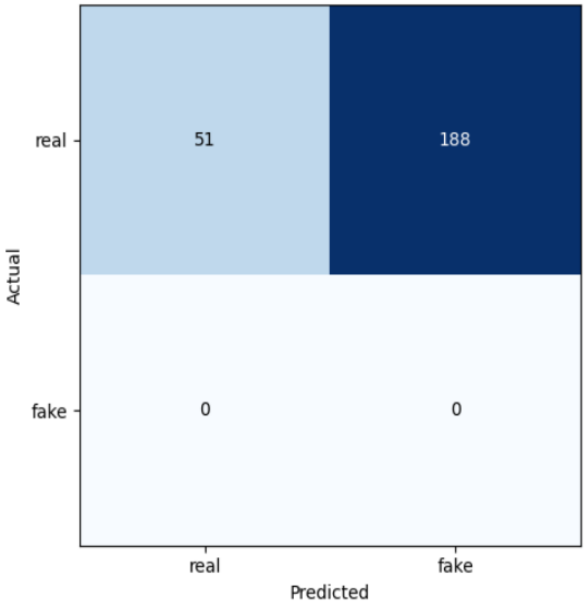


Figure 9: Davinci Real Article Results

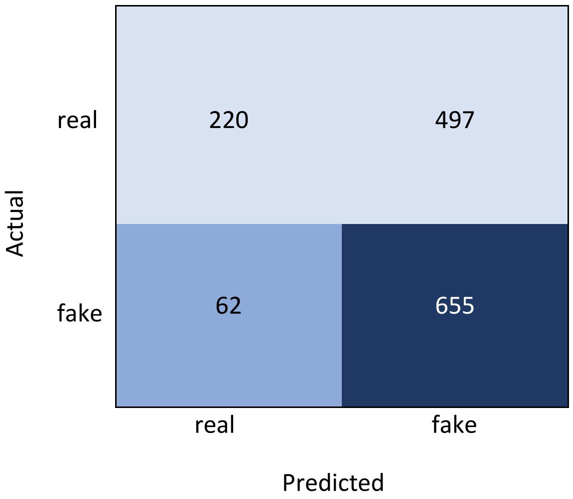


Figure 10: Aggregate Classification Results in Confusion Matrix

Table 3: Classification Results

Dataset Sources	Fake Data Accuracy	True Data Accuracy
Kaggle	100%	49.37%
Davinci	89.96%	21.34%
ChatGPT	84.1%	21.34%

Discussion

Reviewing the results reinforced that the model's metrics were better than we originally anticipated. A major cause for this effect was the tweaks that were made to the text preprocessing function. The delta value that was modified was a parameter that dictates how the text should be divided in preprocessing, before feeding into the classifier. Our texts were about 500 words each, so the delta value denotes making 5 splits of 100 words. There was a 25% increase in accuracy across the board when the function was modified to fit our data. Specifically, we changed the way that the input text gets split and tokenized by modifying a delta value and the associated subtext length.

However, while the fine-tuned model has proven to be effective in the detection of fake news, both human-generated (from the Kaggle dataset) and AI-generated (from Davinci and ChatGPT), it is not as accurate in classifying real news articles. We have seen in the results a 100% correct detection rate for fake Kaggle articles which are entirely written by humans, an 84% correct detection rate for fake ChatGPT-generated articles, and a 90% accuracy for fake Davinci-generated articles. On the other hand, for the true news, we have had accuracy scores of 49% for Kaggle, 21% for ChatGPT, and 21% for Davinci.

One factor that may have affected this is an imbalanced dataset. We had knowledge of this beforehand, so we made sure that our datasets are extremely balanced, which leads us to believe that the cause is likely to be either that we have a bias (potentially mislabeled data) or we are missing a valuable component in feature selection. When we decided to add the ChatGPT articles we needed to substantially decrease the number of total articles generated due to restrictions on time and cost. Due to this, we had high average loss numbers on the training which would have contributed to a bias. To fix this we would have to increase the total number of articles back to the original numbers we had before using ChatGPT (e.g., the Kaggle dataset had an original set of human-generated 39,000 true and fake articles). Also, implementing a "fairness metric" to make sure that the model does not add bias to either direction would help with these scores. It is recommended that additional fine-tuning be performed using more training data to improve true article classification.

Conclusion

The findings are relevant to the current challenges of identifying misinformation and generally support using a fine-tuned BERT transformer for doing such classification. As large language models progress and improve it is likely that it will become easier to detect model-generated misinformation.

The limitations of the research were using only one type of GPT model. It would help to compare the various GPT models fine-tuned with the same data to see how these compare at classification. Additionally, it would be beneficial to create larger datasets of AI-generated fake and real news to see how much more effective classification can get with incrementally larger AI-generated datasets used for fine-tuning training of the model. The limitation to doing this is cost since some of the more popular GPTs request you to pay for their use above some limited free amount of queries.

The research study using the BERT model has shown that with very small training sets (total of 7200 stories from three sources with equal fake and real news stories) that were used to fine-tune the BERT model, the results were very good at identifying AI-generated fake news ranging in accuracy from 84.1% – 100% depending on which data source the test set came from. Providing some preliminary evidence that the use of large training sets to fine tune the model may not be necessary. The limitation of the model was identifying real news where there could have been a bias of mislabeled data and the possibility of having better feature selection to improve this percentage. Future work would need to focus on improving the identification of real news using the BERT model.

There are still elements of this research that can be changed and improved. To improve the quality of this research, using our model-generated fake news articles together with more real articles may improve classification. Also, it would be valuable to gather natural reactions and opinions via human feedback. Not only should this improve the credibility of the generated content but would also help in understanding how people consume and react to news articles. The feedback that we would receive could be used to identify which subset of topics are most engaging, how headlines affect the reader's desire to continue reading the articles, and how humans evaluate the truthfulness of what they are reading.

In addition to human feedback, another way to improve the quality of the model could be to incorporate a wider range of news sources and topics. Rather than using our three blanket topics we could implement many more topics and genres of news. For example, real news could take articles from a wider variety of sources. This would not only add more variety to our dataset but also improve the efficacy of the model at detecting fake news across multiple sources and contexts. Additionally, incorporating more nuanced features into the model, such as sentiment analysis, linguistic cues, and reinforcement learning could further enhance the model's accuracy in detecting fake news.

Considering recent studies from OpenAI, we think it would be beneficial to explore the ethical implications of using AI-generated text detectors to combat fake news. For example, how can we ensure that such detectors are not biased or used to further self-interest and political agendas? How can we balance the need for accurate news reporting with the right to free speech and expression? These are complex questions that require careful consideration and discussion. Overall, this research has the potential to make a positive impact in the ever-growing fight against misinformation. By continuing to refine and improve news classifiers, we can help to ensure that accurate and reliable news remains accessible to all.

References

Aldwairi, M., & Alwahedi, A. (2018). Detecting fake news in social media networks. *Procedia Computer Science*, *141*, 215-222.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Conroy, N. K., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the association for information science and technology*, *52*(1), 1-4.

Gao, C. A., Howard, F. M., Markov, N. S., Dyer, E. C., Ramesh, S., Luo, Y., & Pearson, A. T. (2022). Comparing scientific abstracts generated by CHATGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. https://doi.org/10.1101/2022.12.23.521610

Gereme, F. B., & Zhu, W. (2019, August). Early detection of fake news" before it flies high". In *Proceedings of the 2nd International Conference on Big Data Technologies* (pp. 142-148).

Girgis, S., Amer, E., & Gadallah, M. (2018, December). Deep learning algorithms for detecting fake news in online text. In *2018 13th international conference on computer engineering and systems (ICCES)* (pp. 93-97). IEEE.

Manzoor, S. I., & Singla, J. (2019, April). Fake news detection using machine learning approaches: A systematic review. In *2019 3rd international conference on trends in electronics and informatics (ICOEI)* (pp. 230-234). IEEE.

Mahir, E. M., Akhter, S., & Huq, M. R. (2019, June). Detecting fake news using machine learning and deep learning algorithms. In *2019 7th international conference on smart computing & communications (ICSCC)* (pp. 1-5). IEEE.

Muller, B. (2022, March 2). BERT 101 State Of The Art NLP Model Explained. https://huggingface.co/blog/bert-101

Ozbay, F. A., & Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: statistical mechanics and its applications*, *540*, 123174.

Wang, W. Y. (2017). " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. *Advances in neural information processing systems*, *32*.