APS360 PROJECT FINAL REPORT: AI GENERATED TEXT VS HUMAN WRITTEN TEXT

Vivian Huynh
Student# 1006978521
vivian.huynh@mail.utoronto.ca

Amelie Smithson
Student# 1006684651
amelie.smithson@mail.utoronto.ca

Ji sung Han
Student# 1006581815
jishan.han@mail.utoronto.ca

Hana Truchla
Student# 1006988422
hana.truchla@mail.utoronto.ca

ABSTRACT

This project final report continues to build on our project proposal and our project progress report on creating a model that can decipher between AI generated text and text written by people. On top of providing background information on the project, this report describes all aspects of our final model including how we processed our data, the architecture of our final model, and the results of the model on new data.

—-Total Pages: 10

1 INTRODUCTION

In today's digital age, distinguishing between human-generated and AI-generated text is becoming increasingly important. As AI technology advances, there are increasing instances of indiscriminate use of AI, which raises questions about academic integrity, fairness, and the reliability of online communication (Kannan (2024), Woodall (2024)). This situation poses a significant challenge in many areas related to content verification. Therefore, our project aims to develop a machine-learning model that accurately identifies AI-generated essays. The motivation for this project stems from the growing use of AI in content creation. AI technology offers many benefits, but at the same time, concerns about authenticity and potential abuse are being raised. We aim to create a reliable tool that can identify AI-generated texts, thereby contributing to maintaining trust in written communication and ensuring the responsible use of AI technology. Machine learning is a reasonable approach to addressing this issue because it can analyze complex patterns in text data (Mahapatra (2018)). In particular, by leveraging advanced models such as RNNs and LSTMs that are well-suited for sequence prediction tasks.

2 ILLUSTRATION/FIGURE

The model consists of several main portions as depicted in Figure 1. First, the data is processed and embedded before training. During training, the data is passed through an LSTM model (more details in the Architecture section), and then layer normalization is applied. Layer normalization reduces learning biases within the model. Then it is passed through a linear layer to consolidate the output, and max pooling is applied. Max pooling was chosen over average pooling since it allows the model to effectively use relevant information. Finally, the output is used to calculate loss and complete the backward step necessary in training a neural network.

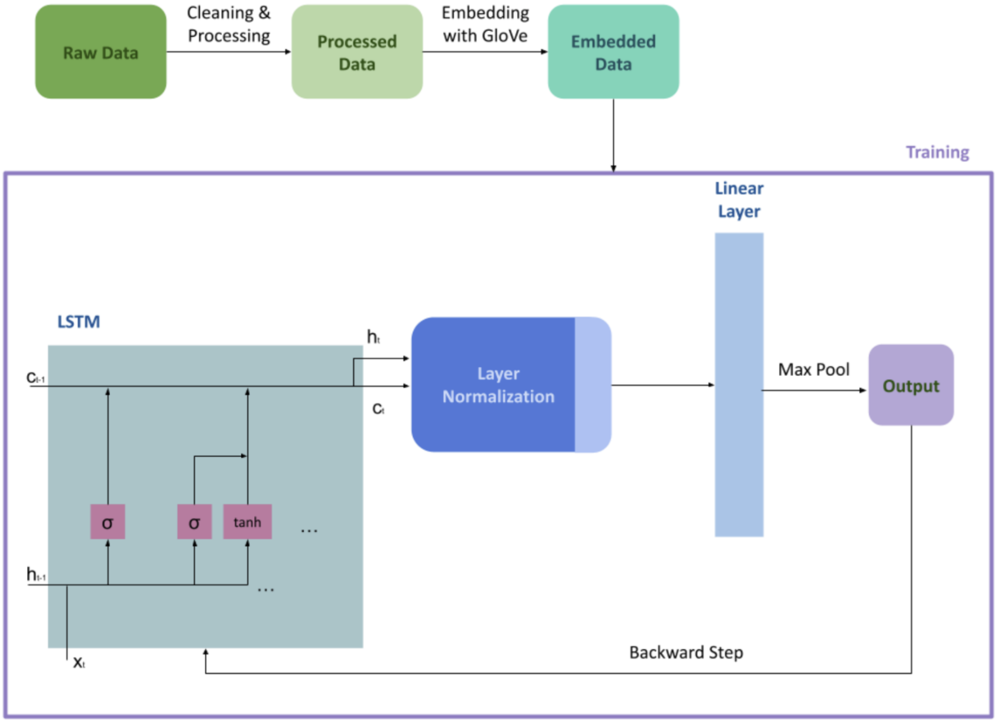


Figure 1: Simplified flow chart of the overall model including data processing, and training.

3 Background and Related Work

QuillBot (QuillBot (2024)) and Scribbr (Scribbr (2024)) are free online AI detectors where the user can submit English text and detect if it is AI-generated content. It can identify text written by ChatGPT, GPT4 or Google Gemini.

“How to Detect AI-Generated Texts?” (Nguyen et al. (2023)) was written to discern synthetically generated text (SGT) from hand-written text (HWT) by academics from Winona State University, AI Center of Excellence Fidelity Investments and the University of Southern Mississippi. Some methods used in this study were dataset creation, feature engineering, dataset comparison and result analysis.

“Detecting AI Generated Text Based on NLP and Machine Learning Approaches” (Prova (2024)) addresses the ethical, legal, and social repercussions of using AI models to create writing that is capable of being identical to that of a human in the future. It uses several learning methods including XGB Classifier, SVM, and BERT architecture deep learning models to differentiate text created by AI and finds that the BERT is the most effective.

“Evaluation of machine-generated text detectors’ (Šigut (2023)) compares how the AI detection tools Compilatio, Turnitin, and GPT-2 Output Detector perform when AI-generated text in English, Slovak, and Czech are tested. Only Compilatio was capable of detecting some AI-generated text in the Czech and Slovak languages however, it was still only capable of performing at an accuracy of 56% and 67% in the Czech language which is slightly above random choice. When the texts were translated to English, Turnitin was capable of receiving an accuracy of 9% with Czech and 92% with Slovak. It concludes that texts generated by ChatGPT-4 are not as detectable as those generated by ChatGPT-3.5 with all languages.

4 Data Processing

The data processing and splitting portion consisted of 4 sections: receiving the input data, data formatting, text preprocessing, and data splitting.

4.1 Receiving the Input Data

There are 2 datasets that are used for this project.

The first dataset is the AI vs Human generated essay data from the daigt-v3-traindataset from Kaggle (Kleczek (2023)). The CSV file contains 5 columns, "Essay", "Category label", "Prompt name", "Source", and "RDizzle3-seven competitor". Where in "Category label" 0 is human-written and 1 is AI-generated.

The second dataset is also obtained from Kaggle and titled "AI Vs Human Text" (Gerami (2024)), which will only be used to do the final testing on the model to gain an accurate evaluation. This dataset is also in the form of a CSV file with the columns "Essay" and "Category". Simiarly, the "Category" uses 0 for human-written and 1 for AI-generated.

4.2 Data formatting

There are two main data preprocessing steps that are required; extracting the important information and removing duplicate entries from the CSV files.

Firstly, the important information in each dataset needs to be extracted. For the daigt-v3-traindataset dataset the important columns were deemed to be the essay and the corresponding category only. The prompt column is unnecessary since the model should be able to determine if it is AI or human generated regardless of the topic being written about. Since the label of whether the essay is AI or human is predetermined, the source is also unnecessary. Lastly, the model does not require knowing if the data is present in a competition or not, and thus it was removed. The "AI Vs Human Text" dataset did not contain the 3 columns that were removed from the daigt-v3-traindataset dataset and thus, this processing step did not need to be completed as all information present is important.

Furthermore, it is important that the labels of the columns which contain the essay and the category for both the daigt-v3-traindataset and "AI Vs Human Text" datasets remain consistent. The column labels were changed using the create_consistent_labels function. This resulted in columns "Text" and "Category".

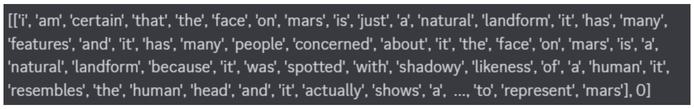
4.3 Text Preprocessing

There are a couple steps that were conducted in the text preprocessing outlined.

The first step was to remove duplicate entries in the CSV files. This was conducted with the remove_duplicates function that loops through both datasets and removes duplicates. Next, the remove_uppercase function was used change all text to lowercase. Unwanted characters were also removed using the csv_remove_nonwanted_characters. The only retained characters were within the string, "abcdefghijklmnopqrstuvwxyz".

4.4 Data Splitting

The daigt-v3-traindataset dataset was split up into 60% training, 20% validation and 20% testing data. The split_data function was created that goes through the daigt-v3-traindataset dataset and splits the required amount of data into a separate CSV file, one for each training, validation, and testing. The testing data is stored in a completely separate CSV file it can be assured that the model will not see the test data before the testing phase. Furthermore, the dataset contains 27,371 human-generated essays and 37,962 AI-generated entries. Thus steps must be taken to ensure that there is a balanced dataset. The splitting of the data will only include 27,371 AI-generated essays to ensure a 50/50 split of AI vs human text. Once the data splitting was run, it was found that there was a total number of 32,846 training samples, 10,498 validation samples and 10,948 testing samples. To allow for a more consistent format of data entering the model, the function csv_to_list function took each essay and split into a list of the corresponding words and then entered into a list of the form: [[[word1], [word2], . . .], category]], [[word1], [word2],...], [category]],...] An example of a processed essay can be seen in Figure 2.



```
[['i', 'am', 'certain', 'that', 'the', 'face', 'on', 'mars', 'is', 'just', 'a', 'natural', 'landform', 'it', 'has', 'many',
'features', 'and', 'it', 'has', 'many', 'people', 'concerned', 'about', 'it', 'the', 'face', 'on', 'mars', 'is', 'a',
'natural', 'landform', 'because', 'it', 'was', 'spotted', 'with', 'shadowy', 'likeness', 'of', 'a', 'human', 'it',
'resembles', 'the', 'human', 'head', 'and', 'it', 'actually', 'shows', 'a',  ..., 'to', 'represent', 'mars'], 0]
```

Figure 2: An example of how the text is split.

The reasoning for not solely using the “AI Vs Human Text” dataset for testing is to determine if the model behaves differently when exposed to similar but new data. Ideally, the model should behave similarly when testing on both datasets. The same csv_to_list function was used to split each essay into a list of the same form shown in Figure 2. There are 10,949 essays in the “AI Vs Human Text” dataset with 1036 being AI generated with the average length of each essay being 410 words.

5 Architecture

The final neural network model is a Long Short-Term Memory (LSTM) network. This model is designed to process sequence data, making it particularly effective for handling sequential data like text (GeeksForGeeks). The architecture of the model is composed of the following key components:

- **Embedding Layer**: Utilizes pre-trained GloVe vectors to convert input text into dense vectors of a fixed size. This layer captures semantic relationships between words in the text data, enabling the model to better understand and process the text.
- **LSTM Layer**: Consists of a single LSTM layer with 100 hidden units and is unidirectional (i.e., bidirectional=False). This LSTM layer processes the input sequence data, maintaining contextual information throughout the sequence.
- **Layer Normalization**: Applied to the output of the LSTM layer to enhance the convergence speed and improve the stability of the model during training. The normalization process is adjusted based on the hidden size and bidirectional settings.
- **Fully Connected Layer**: A dense layer responsible for the final classification, producing outputs for the two classes (human-written text and AI-generated text). The input to this layer is the max-pooled output from the LSTM layer.

The model was trained with a learning rate of 0.001 and a dropout rate of 0.2 to prevent overfitting. Training was conducted over 2 epochs, with loss and accuracy tracked for each epoch.

Although theoretically, model combination 6, derived through hyperparameter testing, was expected to have the highest accuracy, additional testing of various combinations revealed that model combination 1 (which was ultimately selected as the final model) achieved even higher accuracy. This outcome underscores the importance of testing various configurations to achieve the best-performing model.

6 Baseline Model

A Random Forest Model can be used to compare the final model. To process the text data, the Bag of Words model will also be used to process the text for the baseline model. This converts text data into numerical data by using a word frequency count. Each data point has a list of words and the number of appearances. Then the numerical data is passed through the baseline model. Within the baseline model, there are many decision trees where sample sets of data are passed through. These sample sets are extracted using bootstrapping, meaning it has the same number of data points as the original data set, and potentially repeat data points (due to sampling with replacement). The decision trees are created using the training dataset, and the validation and test set only produce final accuracy results. Each decision tree will produce an output and the final output is determined with the majority result. Both the Bag of Words and Random Forest models (Chidananda (2018)) were created using the Python module sklearn (see the full code in the Google Colab link at the end of the report).

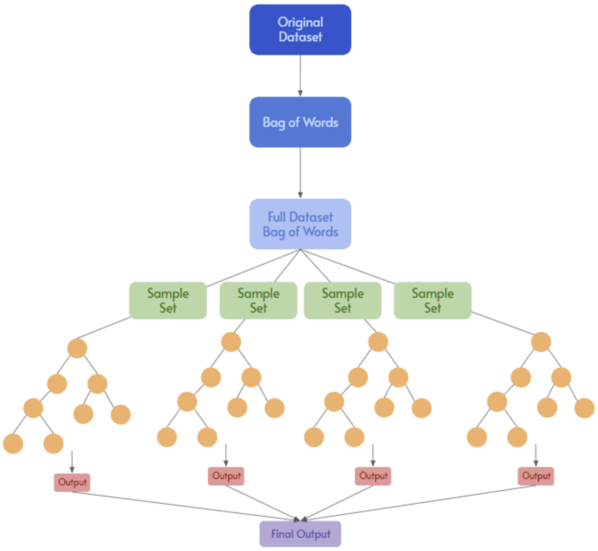


Figure 3: A simple diagram of the Random Forest Model with Bag of Words.

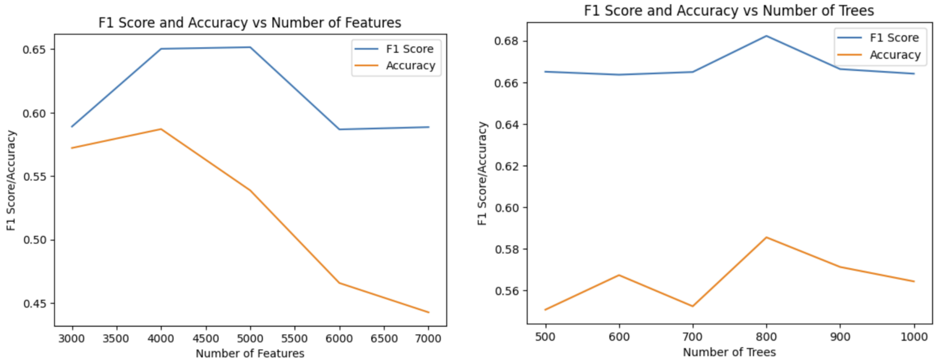


Figure 4: Tuning the hyperparameters, number of features, and number of trees. For the final model, it was decided that 4000 features and 800 trees would be used.

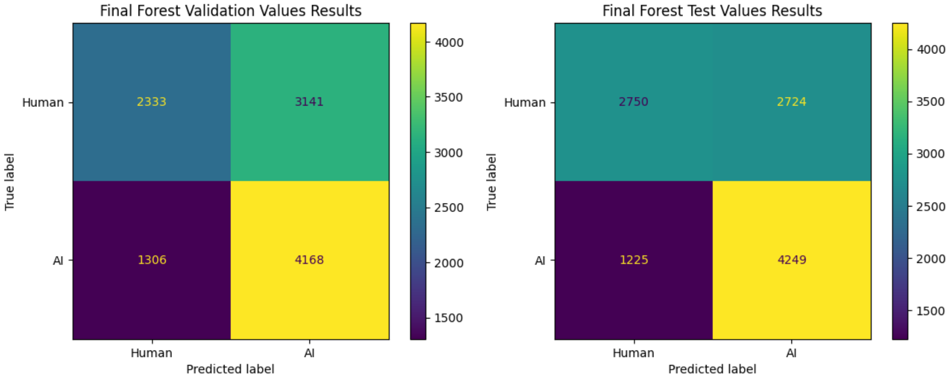


Figure 5: Confusion matrices for the final validation and test results.

The first iteration of the baseline model used 5000 features, 500 trees and the rest of the parameters are set to the common values as determined through research (Probst et al. (2019)). These parameters are as follows, Gini impurity rule for splitting, min leaf node samples = 1, number of features at a split = sqrt (number of samples). This default Random Forest model resulted in an accuracy of 55% and an F1 score of 0.671.

The largest challenge was the training time. Since the dataset is large, creating the Bag of Words followed by training took several minutes. As a result, the parameters had to be tuned slightly to allow the model to train within 10 minutes. Additionally, the larger parameters could not be too large as it would max out the Google Colab resources. After some trial and error, setting the number of features to 5000 was chosen since it did not max out the resources, allowing for proper hyperparameter tuning later.

The chosen hyperparameters to tune were the number of features and the number of trees. For the number of features, values in increments of 1000 were chosen from 3000 to 70000. This maximized the use of computational resources while keeping runtime for each iteration at a reasonable length (<10mins). As demonstrated in Figure 4, 4000 features maximized the accuracy and F1 score of the model. Similarly, with the number of trees, values in increments of 100 were chosen from 500 to 10000. This also maximizes the computational resources while exploring within a reasonable range as determined by research (Probst et al. (2019)). In Figure 4 it is found that 800 features maximizes the accuracy and F1 score of the model.

Finally, after tuning the hyperparameters to 800 trees and 4000 features, the final test accuracy was 64% and F1 score of 0.693. The validation and testing confusion matrices for the final model are presented in Figure 5.

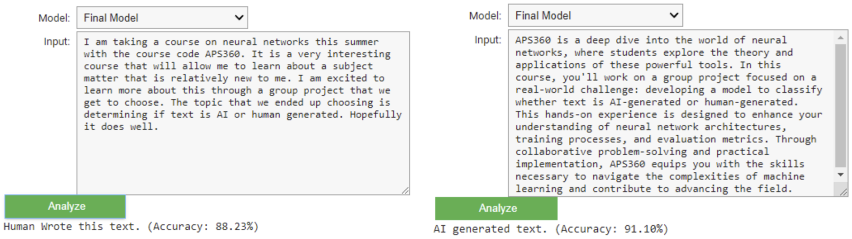
7 Quantitative Results

To evaluate the model performance, the training accuracy, validation accuracy, training time, test accuracy, and testing time will be analyzed. Higher accuracy and lower runtime results in a better model. To train the model, default values for the parameters were selected. This included a learning rate of 0.01, a hidden size of 100, 1 layer, and dropout rate of 0.2. After the data was processed and the training and validation data were trained on the model, the training accuracy was about 97.89% and it took over 9 minutes for training. While the initial training accuracy was relatively high, the training time was relatively long. The validation accuracy was also low at about 74.59%. After tuning the hyperparameters, the highest accuracy was found with the following hyperparameter values: a learning rate of 0.001, a hidden size of 100, 1 layer, and a drop out of size 0.2. The training accuracy was very similar to the previous training accuracy at about 97.87% averaged across two epochs. The validation accuracy increased to an average value of about 84.94% over two epochs. The time also decreased to approximately 6.5 minutes.

Once the model had been trained and the hyperparameters tuned, it was time to test the model on a completely new dataset not seen in training. Running the testing data through the model resulted in a test accuracy of 86.49%. This is about 30% greater than the base model accuracy and over 15% the initial goal. While it was lower than the training accuracy by about 10%, it was slightly higher than the validation accuracy. Therefore the model performed relatively well on new data and remained accurate when presented with unseen data.

8 Qualitative Results

The model was able to sort AI vs Human generated text well as seen by the quantitative data section analysis. An example of both AI and Human generated input can be seen in Figure 6 and Figure 6 below.



Model: Final Model

Input: I am taking a course on neural networks this summer with the course code APS360. It is a very interesting course that will allow me to learn about a subject matter that is relatively new to me. I am excited to learn more about this through a group project that we get to choose. The topic that we ended up choosing is determining if text is AI or human generated. Hopefully it does well.

Human Wrote this text. (Accuracy: 88.23%)

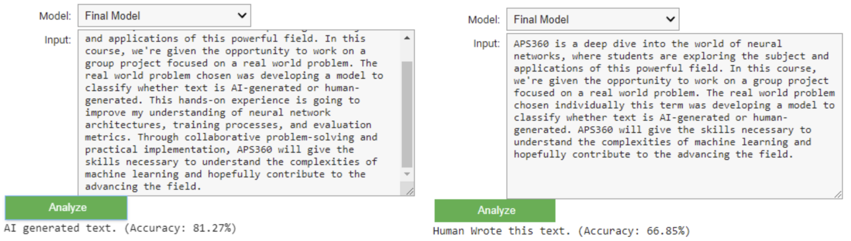
Model: Final Model

Input: APS360 is a deep dive into the world of neural networks, where students explore the theory and applications of these powerful tools. In this course, you'll work on a group project focused on a real-world challenge: developing a model to classify whether text is AI-generated or human-generated. This hands-on experience is designed to enhance your understanding of neural network architectures, training processes, and evaluation metrics. Through collaborative problem-solving and practical implementation, APS360 equips you with the skills necessary to navigate the complexities of machine learning and contribute to advancing the field.

AI generated text. (Accuracy: 91.10%)

Figure 6: Accurate classification of human generated text (left) and AI generated text (right).

One strength of the model occurs when a portion within an AI generated text is edited by a human. The entire text should still be classified as AI generated since changing a couple words does not make it an authentically human generated piece of work. The prompt from Figure 6 above was directly copied; however, the results were now slightly modified by a human. This was then input into the model again and as seen in Figure 7, the model still classifies the text as AI generated which is correct. There is however a balance between how much of the AI generated text is actually changed by the human which impacts the classification. As seen in Figure 7, even more of the text was edited and when input into the model after these changes, the model misclassified it as human generated.



Model: Final Model

Input: and applications of this powerful field. In this course, we're given the opportunity to work on a group project focused on a real world problem. The real world problem chosen was developing a model to classify whether text is AI-generated or human-generated. This hands-on experience is going to improve my understanding of neural network architectures, training processes, and evaluation metrics. Through collaborative problem-solving and practical implementation, APS360 will give the skills necessary to understand the complexities of machine learning and hopefully contribute to the advancing the field.

AI generated text. (Accuracy: 81.27%)

Model: Final Model

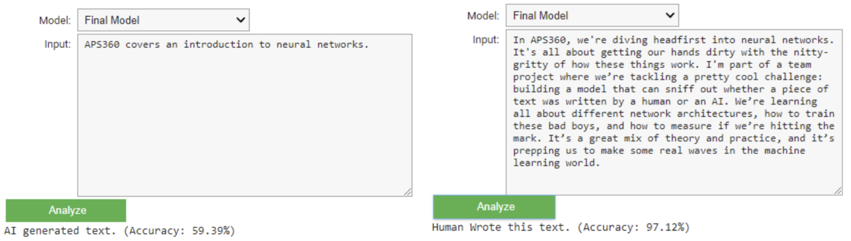
Input: APS360 is a deep dive into the world of neural networks, where students are exploring the subject and applications of this powerful field. In this course, we're given the opportunity to work on a group project focused on a real world problem. The real world problem chosen individually this term was developing a model to classify whether text is AI-generated or human-generated. APS360 will give the skills necessary to understand the complexities of machine learning and hopefully contribute to the advancing the field.

Human Wrote this text. (Accuracy: 66.85%)

Figure 7: (Left) Text correctly classified as AI generated despite human tampering of exact ChatGPT result. (Right) Text incorrectly classified as human generated after making further edits to AI generated text.

Although the model was able to predict the classifications with a relatively high degree of accuracy, there were still more misclassifications that occurred.

A common misclassification error that was observed was when a wide variety of text lengths were input into the model. The data set which was used for training and validation had an average word count of 427 words. Thus, since the information originally fed into the model were of similar lengths, the model became very accurate and performed well at determining the classifications for these lengths. The testing data further confirmed that the model was accurate, however, it is important to note that the testing data set used contained texts that were a similar length of 410 words as the training and validation data. As seen in Figure 8. below, an input of 7 total words was fed into the model that was written by a human, however the model classifies it as AI generated. Since the model was never fed a text with a similar length to this input, it leads to misclassifications.



Model: Final Model

Input: APS360 covers an introduction to neural networks.

AI generated text. (Accuracy: 59.39%)

Model: Final Model

Input: In APS360, we're diving headfirst into neural networks. It's all about getting our hands dirty with the nitty-gritty of how these things work. I'm part of a team project where we’re tackling a pretty cool challenge: building a model that can sniff out whether a piece of text was written by a human or an AI. We’re learning all about different network architectures, how to train these bad boys, and how to measure if we’re hitting the mark. It’s a great mix of theory and practice, and it’s prepping us to make some real waves in the machine learning world.

Human Wrote this text. (Accuracy: 97.12%)

Figure 8: (Left) Human generated text incorrectly classified as AI generated. (Right) Misclassified text as human generated when copying a result directly from ChatGPT with explicit instructions to use personal pronouns.

Lastly, the model tends to classify texts that use a significant amount of personal pronouns as human generated regardless of the true origin of the text. Figure 8 shows a text that was a result of a prompt into ChatGPT that specified that a large amount of personal pronouns should be used. As seen, the model incorrectly classified it as human generated. This is largely due to the fact that many GPTs do not use personal pronouns unless explicitly specified. All the AI generated texts that the model was trained and validated with do not include personal pronouns which opens up the model to errors.

9 Evaluate Model on New Data

To ensure the results are a good representation of the model's performance on new data, the dataset was split into 3 sections. 60% of the data was put aside for training the model, 20% was separated for validating the model and tuning the hyperparameters, and the final 20% was designated to testing. Each dataset had an even split of AI-generated and real essays. In an unbalanced dataset a quantitative bias could occur; the model outputting one label more frequently due to the higher likelihood of appearance. By splitting the training dataset and validation dataset to have 50% AI-generated and 50% human-written essays, this was avoided.

The 80% of the data that made up the training and validation dataset was then used to train and validate the model and the 20% of testing data was set aside. Furthermore, the final decision was to use a separate dataset to create the testing dataset, ensuring complete isolation of the test data (the "AI Vs Human Text" dataset discussed in the Data Processing section). The model was then completely finalized and optimized before testing. The training data was used while developing and modifying the model to ensure adequate training accuracy. Then the validation data was used to determine the best set of hyperparameters that resulted in the highest accuracy. The training and validation datasets were used to ensure that the model worked as intended. Finally, the isolated testing dataset was used. This test data was run through the model once and the results were recorded and analyzed.

One issue that may result in the model's performance on new data appearing better than expected is that testing, validation, and training data were formatted similarly. All of the data was in the form of short essays. If the new data was comprised of shorter messages (i.e. emails or text messages) the performance of the model would likely decrease as it was not trained on that form of data.

Due to the many precautions to ensure the model was unbiased and not memorizing data, the model can be evaluated on new data. As the datasets used to train the model were made up of essays, it will perform best on new data of a similar format and length. To see some examples of new samples being tested against the model, please refer to the Qualitative Results sections.

10 Discussion

The model has a 86.49% test accuracy; above the desired 70% test accuracy. Therefore the model meets our requirements previously outlined. However, if this model were to be deployed for consumer usage, it would require further training/testing to cover more styles of text/writing. While the accuracy appears suitable, there is still a non-negligible chance genuine texts may be falsely identified as AI-generated. In that case, the writer may unfairly face negative consequences. This concern is further discussed in the Ethical Considerations section. Thus, the model accuracy achieved is suitable in the context of this report. However, greater accuracy should be strived for if this model were to be implemented for public usage.

An unexpected quality of the model is its ability to achieve a higher test accuracy than the validation accuracy. Generally, the opposite is expected because hyperparameter tuning is done to maximize the validation accuracy. Although, the testing dataset is taken from an entirely different text dataset which may have positively influenced the accuracy.

Another unexpected feature of the model is that it functions well after one epoch of training. It is also notable that the single epoch can take a few minutes to several minutes to run depending on the hyperparameters. However, in each case, the model has a training accuracy greater than 60%. Although, the training data set is relatively large (in terms of the course) at the size of tens of thousands. So one epoch covers a large amount of data, explaining the high accuracy.

Overall, the development of this model brought a new realization of how text can be processed numerically in various methods (embedding) and how a neural network can process that numerical data. Neural networks operate using numerical values, even when processing data that is not inherently numerical (images, audio, text etc.). As it follows, any non-numerical data then must be converted or interpreted numerically. However, different methods can affect the efficiency/effectiveness of the overall model. In this project, the GloVe embedding protocol was used as it is a proven reliable and effective method used in text-processing neural networks. As a result, this aided in creating an effective model for identifying AI-generated text.

11 Ethical Considerations

A major ethical consideration for our model is that it may be used as the sole identifier of authenticity. If it is incorrect, there is a risk that a person could face negative consequences even if the text is original and authentic. In the past, people have failed assignments and even courses due to false positives from an AI detector (Coffey (2024)). These major consequences can greatly and negatively impact someone's life. Ensuring that our detector is as accurate as possible and reiterating that it should only be used as a resource and is not definitive will help combat this risk.

Another ethical issue is that the used database has been collected from only one person. Depending on the sourcing of writing samples and geographical location, this may result in a biased dataset that is more accurate for texts written by people from a specific area. Many English-speaking countries have different styles of writing and commonly used words (i.e. American English versus British English). If all of the writers are from one area of the world or write using similar English, the model will perform better in identifying text that uses that variation of English. As a result, the test accuracy will decrease when English text from different geographical locations is used. By being aware of this possible bias, we can better understand the test accuracy of future uses of our model.

Lastly, the data this model will be trained on is mostly medium-length text as it focuses on short essays. As a result, the model may perform worse on shorter texts or texts in different contexts like messages and emails. This limitation of the training data could result in an increased inaccuracy with results when inputting a shorter or different style of text. Identifying this limitation is important to best inform the user of the model's intended usage.

12 Link to Github or Colab Notebook

APS360 Final Project Code:
https://colab.research.google.com/drive/1oHYC-AJUoxM88Yx6Ji08Lv3mMriI9FMf?usp=sharing

Data Processing:
https://colab.research.google.com/drive/1sULFK7nwkcg_HvZY831L0iaxVmfOHXPE?usp=sharing

Random Forest Model:
https://colab.research.google.com/drive/1tgLEdtP1XONEFNAL67NICj9AJ6v7VU5y?usp=sharing

The link to the Github: https://github.com/ji24077/Cerberus-Detection

The link to the processed data:
https://drive.google.com/drive/folders/10BG9Az2990SuMbWicfOIrbFvGWeZxSw8?usp=drive_link

The link to the demo and model paths:
https://drive.google.com/drive/folders/1_Eb6H6vdEdKwK-WjCAOBQwLWNRT0Y3UA?usp=sharing

REFERENCES

Rajath Chidananda. Bag of words meets bags of popcorn. https://www.kaggle.com/datasets/rajathmc/bag-of-words-meets-bags-of-popcorn-, 2018. [Accessed 05-07-2024].

Lauren Coffey. Professors cautious of tools to detect ai-generated writing. https://www.insidehighered.com/news/tech-innovation/artificial-intelligence/2024/02/09/professors-proceed-caution-using-ai, 2024. [Accessed 01-08-2024].

GeeksForGeeks. Rnn for text classifications in nlp. https://www.geeksforgeeks.org/rnn-for-text-classifications-in-nlp/. [Accessed 05-07-2024].

Shayan Gerami. Ai vs human text, Jan 2024. URL https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text.

Prabha Kannan. How much research is being written by large language models? https://hai.stanford.edu/news/how-much-research-being-written-large-language-models, 2024. [Accessed 07-06-2024].

Darek Kleczek. Daigt-v3-train-dataset, Dec 2023. URL https://www.kaggle.com/datasets/thedrcat/daigt-v3-train-dataset?resource=download.

Sambit Mahapatra. Why deep learning over traditional machine learning? — towardsdatascience.com. https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063, 2018. [Accessed 05-07-2024].

Trung Nguyen, Amartya Hatua, and Andrew Sung. How to detect ai-generated texts? pp. 0464–0471, 10 2023. doi: 10.1109/UEMCON59035.2023.10316132.

Philipp Probst, Marvin N Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9 (3):e1301, 2019.

Nuzhat Prova. Detecting ai generated text based on nlp and machine learning approaches. *arXiv preprint arXiv:2404.10032*, 2024.

QuillBot. Free ai detector. https://quillbot.com/ai-content-detector, 2024. [Accessed 07-06-2024].

Scribbr. Free ai content detector. https://www.scribbr.com/ai-detector/, 2024. [Accessed 07-06-2024].

Petr Šigut. Evaluation of machine-generated text detectors. 2023.

Nate Woodall. Why it's getting harder to tell ai-generated images from the real deal online. https://www.abc.net.au/news/2024-04-27/artificial-intelligence-ai-faces-fake-images-social-media/103627436, 2024. [Accessed 07-06-2024].