APS360 PROJECT PROPOSAL: 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 proposal introduces our project for the semester on creating a model that can decipher between AI generated text and text written by people. This report includes an introduction to the project, a figure that demonstrates the possible model, background and related work to the project, a description of the data source and how the data will be cleaned, architecture and baseline model that may be used, the ethical considerations, project plan and possible risks, and a link to the GitHub with the project.

—-Total Pages: 7

1 INTRODUCTION

1.1 GOAL AND MOTIVATION

For our project, we will be developing a model to discern between text generated by an AI application and text written by humans. After training the model using essays written by people and essays written by AI, testing text data can be inputted into the model. It will report back on which party the text was written by with a goal accuracy of 80%. The outlook is that this model will help improve people's confidence in confirming if online texts are written by people. Furthermore, it can provide universities with an extra resource to check if assignments, lab reports, research papers, and projects are written by students.

1.2 IMPORTANCE

AI is being used increasingly by students of all ages to submit reports, essays, research papers and more. With AI improving, it is getting harder for people to decipher between the two (Woodall (2024)). There is also an increase in the number of published resources that are not being written by people alongside an increase of researchers utilising AI to assist them in writing reports. James Zhou, an associate professor at Stanford University found that 16.9% of peer review text had at least some content drafted by AI (Kannan (2024)). This model would act as an extra resource to help people determine if pieces are written by AI and prevent potential issues. This would ensure that people are being fairly evaluated (pieces written by AI receive deductions) and only acceptable papers are published.

1.3 WHY DEEP LEARNING

As AI text generators continue to learn from publicly available text, it is getting harder for a person to identify if AI wrote a block of text. Many people are not experienced at detecting AI in papers. Even if people are experienced, this model can act as a confirmation and may even be faster than the average person in classification. Moreover it is also redundant and time consuming for simple code to search and compare predefined features of text. The features may also be biased/inaccurate due to

the predefinition. The classification task can also be complex for a simple model due to intricacies of language and text. Conversely, deep learning can better handle complex problems and classification due to its utilization of layers. Due to its advantages in speed, accuracy, and classification abilities, a deep learning model is an appropriate tool for this task.

2 ILLUSTRATION

Figure 1 demonstrates the objective of this project and outlines the data processing and architecture routes the program will follow.

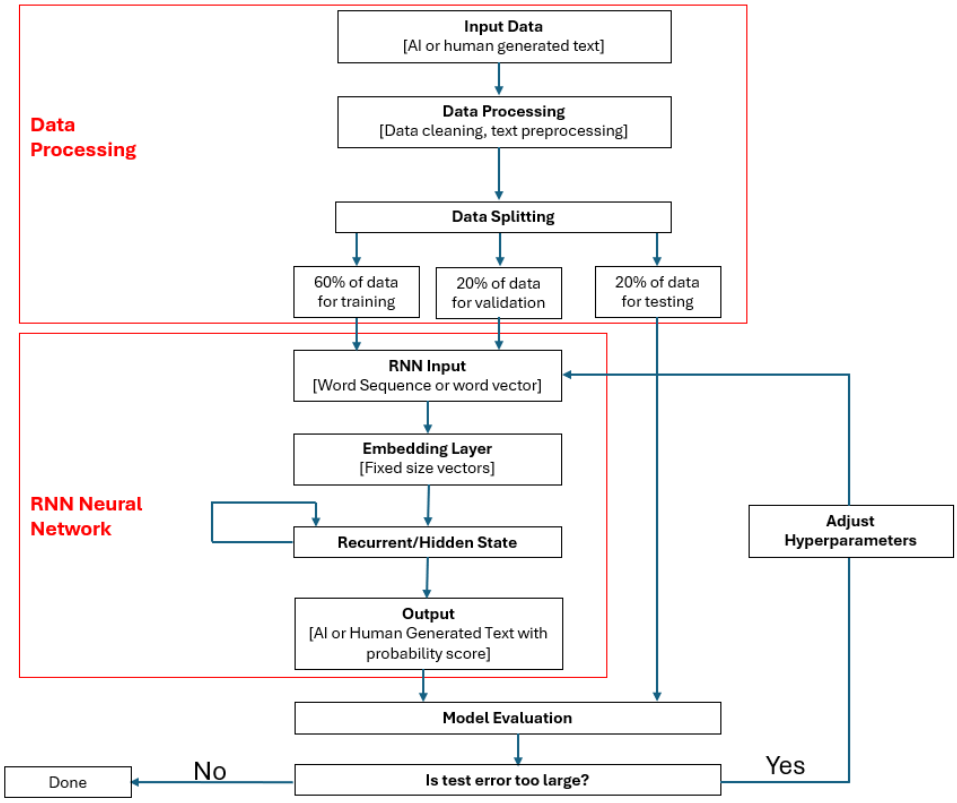

Figure 1: Objective, Data Processing Outline, and Architecture Routes

3 BACKGROUND AND RELATED WORK

QuillBot (QuillBot (2024)) and Scribbr (Scribbr (2024)) are free online AI detectors where the user can paste english text or upload a document 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

Data processing is one of the most important steps in any machine/deep learning project. It ensures that the data is clean, consistent, and ready for analysis, which ultimately affects the accuracy and reliability of the model.

Our primary dataset is sourced from Kaggle's "AI Generated Text Detection" dataset (Kleczek (2024)).This dataset contains both AI-generated and human-written texts, labeled accordingly for binary classification.

4.1 Data Review

Initially, we will review the data to understand its structure and contents. This includes examining the labels, prompt names, sources, and the RDizzl3_seven column. The label distribution will indicate the ratio of AI-generated text to human-written text. The prompt name distribution will show the various prompt names and their proportions. The source distribution will display the sources of the text and their respective proportions. Lastly, the RDizzl3_sevendistribution will represent the ratio of in-domain and out-of-domain text.

4.2 Data Cleaning

The first step in data cleaning is handling missing values. We will check for any missing values and handle them appropriately by either removing or replacing rows or columns with missing values. Following this, we will remove any duplicate data to ensure the dataset's integrity. Additionally, we will detect and address any outliers, which are unusually large or small values in the text data(we may use data visualization to detect outliers), to prevent them from skewing the analysis.

4.3 Text Preprocessing

During text preprocessing, we will remove stopwords, which are common words that do not contribute significant meaning to the text, such as 'the', 'is', and 'in'. We will also remove punctuation from the text to focus on the pure words for analysis. Unnecessary spaces in the text data will be eliminated to ensure consistency.

4.4 Data Splitting

Finally, we will split the data into training, validation, and testing sets to evaluate the model's performance accurately. Our data will be divided into 60

5 Architecture

Recurrent Neural Networks (RNNs) are a type of neural network designed to process sequence data, making them very effective in dealing with sequential data such as text. RNNs have several key components that work together to process and analyze sequence data. These are explained in the following section.

5.1 Input Layer

The input layer is where sequence data is inputted into the network. For text data, the input is typically expressed as a word sequence or word vector. Text data is inputted by vectorizing it, which converts the text into a numerical format that can be processed by the neural network.

5.2 Embedding Layer

The embedding layer converts words into dense vectors of fixed size to capture the semantic relationships between words. This layer is essential for transforming each word into a vector, allowing the neural network to process and understand the text data more effectively.

5.3 Recurrent Layer

As the core component of RNNs, the recurrent layer processes sequence data sequentially and remembers information from previous sequences. At each timestep in each sequence, a new hidden state is created by combining the current input and the previous hidden state. This mechanism allows the network to maintain a memory of previous inputs, making it well-suited for tasks that require an understanding of context over time.

5.4 Output Layer

The output layer is the final layer that outputs the prediction result. For binary classification problems, the output consists of a single node, which uses a sigmoid activation function to output a probability score. This probability score indicates the likelihood of the input belonging to a particular class.

Among RNNs, it is essential to compare the performance of different models, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) etc.., These models have different architectures and capabilities, and their performance can vary depending on the specific task and dataset. Most likely we will mainly use LSTM networks, a popular variant of RNNs that are commonly used for text learning tasks, such as Natural Language Processing (NLP). LSTMs are designed to handle the vanishing gradient problem, allowing them to capture long-term dependencies in sequence data more effectively than standard RNNs.

5.5 Hyperparameter Choices

As of now, we are unsure of our specific choices for hyperparameters. Hyperparameters such as the learning rate, batch size, number of epochs, and the number of units in each layer can significantly impact the model's performance. Given the variety of available models and the importance of hyperparameter tuning, it would be prudent to test and train multiple models with different configurations. By comparing the performance of each model, we can determine which one is best suited for our specific task.

6 Baseline Model

A Random Forest model will be used to compare results, speed, and accuracy with the neural network. This will be created using the same features extracted from the data in the processing step. A series of decision trees will be created, using a randomized sample selection (with replacement) from the data set. The sample size can be the same as the data size (using bootstrapping sampling) or be tuned to a different value. For classification, each of the decision trees should use approximately p random features, where p is the total number of features (Probst et al. (2019)). The number of trees will be tuned when the model is created but it most commonly is between 500 and 1000 (Probst et al. (2019)). Finally, the overall model will have testing data passed through it. Each piece of data will pass through all the decision trees to receive a result from each. Then the majority of the results (0 or 1) will be taken as the final result.

The previously mentioned values to be tuned will be done to reduce overfitting.

7 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 been expelled from universities or failed courses due false positives from an AI detector. 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 database to be used has been collected from only one person. Depending on who they asked for writing samples and their 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. 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 our future model.

Lastly, the data this model will be trained on is mostly long lengths of text as it focuses on essays. As a result, the model may perform worse on shorter lengths of 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. Being aware of this limitation is important so that we can best explain what this model should be used for.

8 Project Plan and Risk Analysis

We will be using a Discord group chat to communicate and will meet every Sunday evening for two hours to make incremental progress on the project. In order to ensure we do not overwrite other people's codes, it will be split up into sections and assigned to each person. This will be decided as a group with all group members present. For complicated strings of code, they may be done all together with everyone collaborating on the section and providing input. We will also split each part of the project up into smaller internal deadlines. For now, we have outlined the internal deadlines and distribution of work for the project proposal and the major internal deadlines for the Project Progress Report, Project Presentation, and Final Project Report in Table 1 below. As the summer continues, we will continue to evenly split the work for the Project Progress Report, Project Presentation, and Final Presentation so that everyone is contributing fairly. We also will continuously communicate with each other so that everyone is aware of where each person is in the project. Furthermore, we will continuously clone the repository when working on the code individually and communicate clearly when we push a new change.

While it is unlikely that anyone will drop the course, it is still a possibility. All of us are taking this course outside of our degree requirements, so there is less obligation to complete the course. If this is the case, we will re-distribute the remaining work evenly. Additionally, we plan to keep organized notes and comments on the work we do so that others can pick up on the work seamlessly.

One of the other likely risks is the difficulty of implementing the model since it will be covered much later in the course. That would mean we will not have any in class experience with it for a large portion of the project. To address this, we will find resources online to learn the content earlier than the course schedule. We can also utilize office hours and email TAs for additional help or advice.

Another likely risk is the training time for the model. Depending on the data we choose to use, we may have to process and train the model on large pieces of text. Additionally, we may have to train it on a large set of data to get acceptable results. If the training time exceeds our resources we can re-evaluate what size of data (length of text) we want to use. Moreover, the neural network could be adjusted to allow for shorter training times.

In addition, scheduling is another time risk that needs to be considered. The work may take longer than expected or some issues may arise during the timeline. However we have a deadline to meet so the solution is to diligently schedule earlier internal deadlines. This provides some buffer days to account for any issues that may occur. Also important, the schedule allows everyone to plan accordingly and spread their time out properly to avoid crunch time.

Below is a general outline of the internal deadlines for this project.

TASK	PERSON	DEADLINE
Create Group	All	May 22
Project Proposal: Intro	Amelie	June 5
Project Proposal: Illustration	Hana	June 5
Project Proposal: Background	Hana	June 5
Project Proposal: Data Processing	Ji	June 5
Project Proposal: Architecture	Ji	June 5
Project Proposal: Baseline Model	Viv	June 5
Project Proposal: Ethical Considerations	Amelie	June 5
Project Proposal: Project Plan	Amelie	June 5
Project Proposal: Risk Register	Viv	June 5
Project Proposal Review and Submit	All	June 6
Project Code	All	Ongoing
Project Progress Report	All	June 10
Project Presentation	All	August 10
Final Project Report	All	August 10

Table 1: Internal Deadlines of Project Proposal and Final Project

9 Link to Github or Colab Notebook

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

REFERENCES

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. https://www.kaggle.com/datasets/thedrcat/daigt-v3-train-dataset, 2024. [Accessed 07-06-2024].

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. Can you spot which of these faces is generated by ai? probably not — here's why — abc.net.au. https://www.abc.net.au/news/2024-04-27/artificial-intelligence-ai-faces-fake-images-social-media/103627436, 2024. [Accessed 07-06-2024].