AI WRITING DETECTOR: A DEEP LEARNING APPROACH TO DISTINGUISH HUMAN AND AI-GENERATED TEXT

Zahra Mohamed Suhail
Student# 1008997016
zahra.suhail@mail.utoronto.ca

Tajrian Islam
Student# 1007939251
tajriansyeda.islam@mail.utoronto.ca

Xin Ling (Grace) Li
Student# 1010143113
xlgrace.li@mail.utoronto.ca

Sophia Hill
Student# 1007865883
sophia.hill@mail.utoronto.ca

1 INTRODUCTION

With the introduction of ChatGPT in 2022, generative Artificial Intelligence (AI) has become widely accessible, particularly to generate written content at a level similar to that created by humans. However, this quickly raised ethical concerns, as people have begun to pass off AI-written content as their own, which has created issues of plagiarism in educational, professional, and creative spaces. Thus, a need has arisen for a deep learning algorithm that can efficiently and accurately detect AI-written content to mitigate the implications to academic, professional, and artistic integrity, and to prevent inaccurate accusations of plagiarism. Therefore, we propose to create an AI Writing Detector that can accurately detect the complexities of human-written, AI-written, and mixed content. AI algorithms used to generate written content have detectable structures that differ from that of humans. Through a Multi-Layer Perceptron (MLP) model, our team will create an AI Writing Detector that will help resolve the issues of integrity and of the authorship in the age of AI.

2 ILLUSTRATION

Our AI Writing Detector's overall pipeline consists of text preprocessing, feature extractions, and a MLP neural network to classify between AI-generated and human-written text. An illustration of our proposed AI Writing Detector Pipeline is shown in Figure 1.

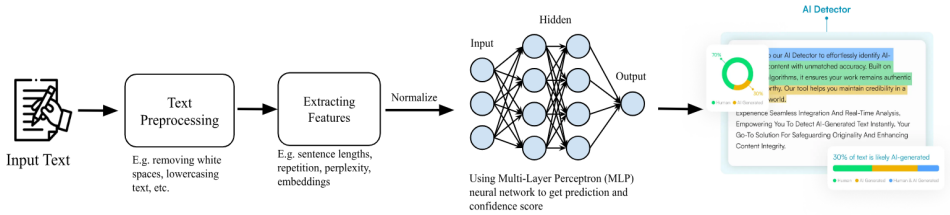

Figure 1: An illustration of our proposed AI Writing Detector Pipeline is shown (Freepik)

3 BACKGROUND & RELATED WORK

3.1 RELATED PAPERS

"Detecting Generative Artificial Intelligence Essays using Large Language Models: Machine and Deep Learning Approaches" (Tariq et al., 2024)

This study analyses the efficacy of different deep learning models to identify human written and AI written essays. The algorithms it discusses are logistic regression, Support Vector Machine (SVM), decision trees, random forests, K-Nearest Neighbours (KNN), and Long Short Term Memory (LSTM). The research used Term Frequency-Inverse Document Frequency (TF-IDF) to prepare their data. It found that SVM and LSTM were the most accurate algorithms, though LSTM is more demanding computationally.

"Deep learning detection method for large language models-generated scientific content" (Alhijawi et al., 2024)

This paper outlines a model called AI-Catcher developed to identify AI generated writing, citations, and data in scientific papers, treated as a binary classification problem. AI-Catcher uses two models to identify human versus AI generated text: the MLP and a Convolutional Neural Network (CNN). For the MLP, it prepares the data through text cleaning, text encoding, and padding. The MLP model then has five hidden layers, each with 256 neurons, where each layer applies a linear transformation and then ReLU activation. For the CNN model, it prepares the data through extraction of thirteen linguistic and statistical features from the text. Then, it uses the embedding layer to put the encoded integers into vectors, followed by a dropout layer, and then a convolutional layer connected to a global max pooling layer. The results of the two are then concatenated into a single feature vector,

which goes through two additional hidden layers. The accuracy, precision, recall, and F1 score of the model were assessed based on confusion matrices. This research was only conducted using scientific papers written by humans or generated by ChatGPT (as opposed to other LLMs).

"Detecting AI-generated essays: the ChatGPT challenge"

This research paper from early 2023 investigates effective algorithms to identify human and ChatGPT generated essays. They used a n-gram bag-of-words (BOW) language model for the classifier input with n=5. They tested the performance of Support Vector Machine, Naïve Bayes, Logistic Regression, Random Forest, and Neural Network classification algorithms and then tested the performance of an Ensemble Learning (EL) classifier with the five algorithms fed to it. The findings stated that their SVM, which was modified to eliminate False Negatives entirely, was the most efficient algorithm. The study emphasized eliminating False Negatives (human written essays labelled as AI) due to the ethical implications of incorrectly blaming students for plagiarism with ChatGPT. SVM alone had a slightly lower accuracy than EL, but had better recall and F2 scores. Their SVM model also had fewer False Negatives than OpenAI Detector, GPTZero, and Copyleaks on test data and a 100% accuracy in detecting human-written essays, despite a lower overall accuracy. This research did also have a relatively small sample size, with 230 total essays for training and 150 for testing (Cingillioglu, 2023).

3.2 Current Software Solutions

GPTZero

GPTZero is an AI detector that scans for AI generated writing on a sentence, paragraph, and document level. It can detect AI written content generated by ChatGPT, GPT-4, GPT-3, GPT-2, LLaMA, and derivatives of those models. It claims to have a 99% accuracy and has an extension that can be added to Google Classroom and Google Docs. It can also identify if a text is entirely human written, entirely AI written, or mixed. However some of its features are locked behind a paid subscription, such as its ability to indicate the most human or AI written parts of a sample of writing and the number of characters, words, or files that can be submitted to it at a time, and it is also primarily for the analysis of English text.

Copyleaks

Copyleaks is a plagiarism detector and AI detector. It works with 30+ languages and claims to have a 99.8% accuracy and a 0.2% false positive rate. It can detect text written by LLMs like ChatGPT, Gemini, DeepSeek, Claude, Jasper 3, LLaMA, and T5. It also shows the percentage of AI in a piece of text and can account for different detection sensitivity levels. It claims to have the capability to identify plagiarism and paraphrasing done with AI and that it can identify human written, AI written, and mixed text. It has a minimum requirement for the number of characters (350) to accurately determine the presence of AI. Copyleaks also has Google Docs extension and mostly requires a paid subscription, with only a limited number of free uses.

Grammarly

Grammarly, an AI based grammar checker, also has its own AI detector. It can detect writing generated by Grammarly, ChatGPT, Google Gemini, and Claude, and it can show the percentage of text that is AI generated. However, it is only available with certain paid subscriptions and they have not stated the accuracy rate of their checker.

4 Data Processing

Figure 2 illustrates the data process pipeline. It demonstrates the effort to source, repurpose, clean, and structure data rather than just using a pre-existing dataset.

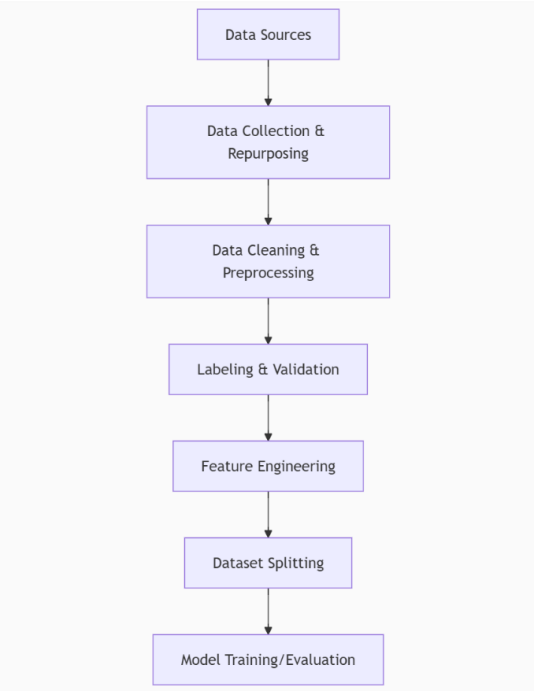

Figure 2: Schematic overview of the proposed data processing methodology

4.1 Data Sources

To ensure a robust and diverse dataset, we will undertake an extensive data collection and processing effort, prioritizing originality and rigor over the use of pre-existing datasets. Our data is sourced from multiple origins to enhance generalization:

- **Human-Written Text:**
  - **Academic Papers & Essays:**
    - Sources: arXiv, PubMed, OpenAlex (for older literature)
    - Why: High-quality, structured, and varied writing styles
  - **News Articles:**
    - Sources: CBC News, BBC News, Reuters
    - Why: Factual, professionally edited, diverse topics
  - **Creative Writing:**
    - Sources: Wattpad, Fanfiction.net, literary blogs
    - Why: Informal, narrative-driven, varied author styles
  - **Social Media & Forums:**
    - Sources: Reddit (long-form posts), Quora, Facebook
    - Why: Conversational, personal, and opinionated writing
- **AI-Generated Text:**
  - GPT Models (OpenAI, Gemini, Deepseek):
    - Using the API calls, one can generate synthetic text with varying prompts (e.g., essays, stories, technical writing)

        - Use different model versions (GPT-3.5, GPT-4, etc.) to capture evolution
    - Kaggle:
        - Provides various styles of texts that one can compare, for example one can fetch a dataset that contains generated texts from LLM.
- **Hybrid (Human + AI Edited):**
    - Wikipedia Edits:
        - By using the edit history one can compare pre- and post-GPT-4 Wikipedia edits
    - GitHub Docs:
        - Some repositories use AI for README generation, this can be identified via commit logs

4.2 Data Collection & Repurposing Efforts

We gathered data in several careful ways to make sure it was both useful and ethical:

- **Web Scraping (Respectful & Legal):**
    - By using open source frameworks that can help with extracting data in an efficient manner such that delays of overloading servers are avoided, use: *scrapy* or *BeautifulSoup*
    - Web scraping is legal so long as the data is publicly available
- **Using APIs to Generate Synthetic Data:**
    - OpenAI API with varied prompts (i.e "Write an essay about AI").
- **Data Augmentation:**
    - Paraphrase human text using AI to create "AI-like" samples
- **Partnerships:**
    - Collaborate with universities for student-written vs. AI-assisted essays

4.3 Data Cleaning & Preprocessing

Before using the data, we must clean and organize it to ensure quality and fairness. First, we removed duplicate or nearly identical content to avoid repetition:

- **Deduplication & Noise Removal:**
    - Near-Deduplication:
        - Use MinHash/LSH (Locality-Sensitive Hashing) to remove near-identical samples
    - Language Filtering:
        - Keep only English text (fastText language detection, developed by Facebook AI Researchers).
    - Quality Filtering:
        - Remove low-quality text (e.g., incomplete sentences, excessive repetition)
- **Text Normalization:**
    - Lowercasing (optional, case-sensitive models may retain it)
    - Remove boilerplate (HTML tags, ads, headers/footers)
    - Expand contractions ("can't" $\rightarrow$ "cannot") for consistency
- **Balancing & Stratification:**
    - Ensure balanced classes (human vs. AI) across various domains of writing (academic, creative, etc.)

4.4 Labeling & Validation

There needs to be a way in which human verification is used for labeling and validating the data.

- **Human Verification:**
  - Crowdsource (Amazon Mechanical Turk) to label ambiguous cases
  - Expert review for disputed samples (e.g., linguistics PhDs)
- **Confidence Scoring:**
  - Use models (RoBERTa, GPT-detectors) to flag uncertain samples for rechecking
- **Kaggle:**
  - By using a dataset of LLM generated texts, provided from this source one can compare and contrast when validating data.

4.5 Feature Engineering

- **Stylometric Features:**
  - Sentence length variance, word rarity, POS tag ratios
- **Perplexity Scores:**
  - Compare human vs. AI text using small GPT-2
- **Watermarking Detection:**
  - If watermarks are presented in the AI text(e.g., OpenAI's), incorporate detection.

4.6 Dataset Splitting

- Split data into: Train (70%) / Validation (15%) / Test (15%)
- Ensure each split has no overlapping sources/authors.
- If testing newer AI Models. One can split based on time (e.g., train on GPT-3, test on GPT-4).

5 Architecture

The MLP stands out as the most effective neural network architecture for AI text detection. Studies comparing various machine learning models have shown that MLPs strike the best balance between accuracy, efficiency, and flexibility. While SVMs require manual feature engineering and LSTM networks are computationally intensive, MLPs deliver high accuracy without these drawbacks. As highlighted in the AI-Catcher study, MLPs outperform CNNs in capturing the overall coherence of text, which is a key factor in detecting AI-generated content. Unlike complex ensemble methods, which added little value in challenges like ChatGPT detection, MLPs offer a simpler, more interpretable solution that scales well. MLPs further enhance their practicality by working with both engineered features and raw text embeddings. Altogether, MLPs combine performance, simplicity, and real-world validation, making them an ideal choice for reliable, production-ready AI text detection system.

6 Baseline Model

Most of the research that investigates effective AI writing detection algorithms found that SVM was the best model for accuracy. Sklearn has a built-in SVM model package that can be built given specific parameters and trained on the data. This would require hyperparameter tuning of the model, which would initially be based off of what was found to be effective in the literature. A 2024 study focused on the detection of human-written and AI-written texts found the default settings of Sklearn's SVM model were sufficient in training (Tariq et al., 2024). With the default settings, the kernel will be "rbf" (Radial Basis Function) for non-linear classification, the regularization parameter will be set to 1.0, and the gamma parameter set to "scale" (Tariq et al., 2024).

7 Ethical Considerations

When creating an AI detection tool, there are many ethical concerns such as accuracy, bias, and privacy. While we aim to make a model as accurate as possible, there is still a chance for human writing to be detected as AI. Bias in the model is another issue, which can occur from the training data. A group particularly affected by AI detection systems is that of non-native English speakers, whose writing is flagged as AI-generated more often than that of native speakers due to differences in writing styles and levels of sophistication (Myers, 2023). Also, the risk of a biased model is that individuals begin to fear being flagged, even without AI usage. Such fears can stop people from openly expressing themselves and limit creativity. Another ethical consideration is that many individuals may not be comfortable in having their work used for model training, and the use of their data without consent would be a breach of privacy. We need to be careful in training our AI model as there are a lot of consequences of false detection of AI generated writing since plagiarism is tied into a person's credibility. With a larger dataset and the ability to learn from a variety of different texts, the team will aim to train the model to be as accurate as possible and reduce its bias.

8 Project Plan

We will hold weekly meetings on Mondays to discuss current progress and future steps. All forms of communications will take place online via Discord, unless specified otherwise. To ensure we do not overwrite each other's code, each member will create and work on their own branches through the use of version control. Each member will open a pull request when they wish to merge with the main branch after code reviews have been completed.

A summary of the Project Plan timeline with assignees and deadlines is shown in Figure 3.

Task	Priority	Owner	Status	Start Date	End Date	Completed	Notes
Project Brainstorming	High	All	Completed	5/26/2025	6/4/2025	100%	
Project Proposal Draft	High	All	Completed	6/4/2025	6/10/2025	100%	
Project Proposal Final	High	All	Completed	6/11/2025	6/13/2025	100%	
GitHub Setup	High	Sophia	Completed	6/12/2025	6/12/2025	100%	
Collecting AI-generated data	Medium	Tajrian	Not started	6/14/2025	6/16/2025		
Collecting Human-written data	Medium	Zahra	Not started	6/14/2025	6/16/2025		
Preprocess and clean the data	Medium	Sophia, Grace	Not started	6/16/2025	6/18/2025		
Investigate Other Architectures (e.g. LSTM, transformers, CNNs, etc.)	Medium	All	Not started	6/15/2025	6/30/2025		
Feature extraction	Low	All	Not started	6/16/2025	6/23/2025		
Baseline experiment	Low	Sophia, Tajrian	Not started	6/23/2025	6/30/2025		
MLP Implementation	Low	Zahra, Grace	Not started	6/26/2025	7/5/2025		
Performance Evaluation	Low	Sophia, Tajrian	Not started	7/1/2025	7/6/2025		
Progress Report Draft	Low	All	Not started	7/5/2025	7/9/2025		
Progress Report Final	Low	All	Not started	7/9/2025	7/11/2025		
Hyperparameter Tuning	Low	Zahra, Grace	Not started	7/12/2025	7/17/2025		
Prepare Demo	Low	All	Not started	7/17/2025	7/31/2025		
Project Presentation Prepation	Low	All	Not started	8/1/2025	8/15/2025		
Final Deliverable	Low	All	Not started	7/31/2025	8/15/2025		

Figure 3: Project Gantt Chart

9 Risk Register

In the creation of an AI model to detect if written work is made by AI, there are many problems that the team may encounter. Despite the chance of these challenges occurring, we have planned ahead on how to handle them.

9.1 High Risk — Long Training Time Due to Large Dataset

Risk: Since our dataset has a large dataset, each with at least 350 words entered at a time, training the model on the full set is likely to take a significant amount of time.
Solution: To mitigate the risk of not completing the training on time, we intend to reduce our sample

size accordingly, potentially by half, ensuring that our set is still diverse. If time allows later on, we can expand the dataset for further training.

9.2 Medium Risk — Breach of User Privacy

Risk: Since our model will be trained on written content, there is a chance that users may not want their own work to be used, making the likelihood of invading user privacy medium risk.
Solution: To mitigate any breach of privacy, we will warn users of how their data is being used. That way, whoever uses the model is giving their consent for their work to be used for training purposes and they are aware of how the data they are giving is being handled.

9.3 Medium Risk — Detecting Human Writing as AI in Plagiarism Detection

Risk: As the model is limited to the training of our dataset, the likelihood that it becomes biased and can mistake human writing as AI is a medium risk.
Solution: To mitigate the chance of the model mistaking human writing as AI, we have a large dataset with a variety of writing, teaching the model as best as possible in detecting if the written work was done by AI. This way, the model can learn from a more complex dataset and be better prepared in differentiating AI-written work from human-written work.

9.4 Medium Risk — Missing Deadlines

Risk: Our members have busy schedules with many of us working or having other activities to take care of, making the likelihood of missing deadlines a medium risk.
Solution: To mitigate the chance of missing major deadlines for the project or procrastination, the team will break tasks up and create internal deadlines using gantt charts. That way if any team member is unable to complete their part on time, there is still time for other members to help them and meet the major deadline.

9.5 Low Risk — Team Members Dropping the Course

Risk: As all of our team members need this course for our degrees, the likelihood of anyone dropping out is low risk.
Solution: To mitigate the damage to the project timeline if a team member were to drop, the team member that is leaving must notify the group as soon as possible. The team member leaving should ensure others understand the next steps of any incomplete work. Also having gantt charts and weekly meetings will help the other members redistribute responsibilities.

10 Link to GitHub/Colab

GitHub Repository

References

Bushra Alhijawi, Rawan Jarrar, Aseel AbuAlRub, and Arwa Bader. Deep learning detection method for large language models-generated scientific content. *Neural Computing and Applications*, 37: 91–104, 2024.

Freepik. *Writing free icon*. URL `https://www.flaticon.com/free-icon/writing_12054824?related_id=12054818&origin=search`.

Rasikh Tariq, Casillas-Muñoz F., Waqar Muhammad Ashraf, and Maria Soledad Ramírez-Montoya. Detecting generative artificial intelligence essays using large language models: Machine and deep learning approaches. In *2024 International Conference on Engineering Computing Technologies (ICECT)*, pp. 1–6, 2024. doi: 10.1109/ICECT61618.2024.10581394.