Group 25 Progress Report:
AI Essay Detector

Fei Xie, Ghena Hatoum, Haniye Hamidizadeh
{xief17,hatoumg,hamidizf}@mcmaster.ca

1 Introduction

The widespread and accessible nature of Generative AI (GenAI) tools, such as ChatGPT and Gemini, has created a significant shift in the educational landscape. While these tools are valuable for quick information retrieval, their rapid adoption precedes the necessary adaptations in academic practices.
The core issue is that unchecked reliance on GenAI for academic work can have long-term developmental consequences for students. It prevents them from engaging in the process of formulating original thoughts and arguments, which is critical for developing their unique intellectual voice. A reliable AI Text Detector, integrated within the education system, would allow educators to foster responsible use of GenAI as an educational aid rather than a replacement for student effort.
Developing an effective detector is challenging for several reasons:

- Low Accuracy of Existing Models: Current models are not accurate enough for real-world academic use.
- Evolving Realism: GenAI systems are constantly improving, generating increasingly human-like text, requiring the detector to be dynamically updated.
- Critical False Positive Rate: The model must achieve a near-zero false positive rate to ensure no student is wrongfully accused of plagiarism while maintaining high accuracy in identifying genuine misuse.

2 Related Work

Detecting AI-generated text is closely connected to research in text classification. An overview of how methods such as TF-IDF have been used in traditional machine-learning models before the development of deep-learning techniques as seen in the survey “A Survey on Text Classification: From Traditional to Deep Learning” (Li et al., 2022).
In practical applications, AI text detection has been explored through competitions, such as the kaggle AI Text Detection Competition (Al-Ahmadi). Public notebooks from the competition show examples of baseline systems that use TF-IDF combined with classifiers such as Logistic Regression or SVM. Two examples are the notebooks by Faris Al-Ahmadi (Al-Ahmadi) and Jasmine Mohamed (Mohammad), where TF-IDF is used to represent essays numerically before training classification models.
More recent work focuses on neural network approaches. LSTM models are often used for text classification tasks because they can process text as a sequence and keep track of earlier context in the input. An example of this is shown in an LSTM text-classification notebook (Khotijah). Transformer-based models such as DistilBERT have also been applied to similar classification problems. The KerasNLP starter notebook from the same Kaggle competition shows how DistilBERT can be fine-tuned for detecting AI-generated text (Audevart).
There is also research specifically focused on identifying patterns unique to AI-generated writing. The “DetectGPT” paper (Mitchell et al., 2023) shows that AI-generated text often has different patterns in how words are used compared to human writing and discusses ways to detect those differences.

3 Dataset

We will be using the DAIGT V2 Train Dataset from Kaggle (Kleczek). This comprehensive dataset is a collection of other datasets. One of the sub datasets contain argumentative essays written by grade 6 to 12 students. The remaining sub datasets contain AI-generated essays from numerous models as seen in Table 1.

The dataset is prelabelled and additional columns include prompt_name (original persuade prompt), source (original dataset), text (actual text content), and RDizzl3_seven (Classifier built for a previous Kaggle competition, indicating whether the essays were written in response to one of the seven essay prompts for the competition).
Our team has dropped all the features other than text and the label.

3.1 Preprocessing

Using the text feature, we have extracted additional features. These include:
Word Statistics
The features include the total number of words in the text/document, the sum of words that repeat frequently (greater or equal to 5 times), and the sum of words that appear infrequently (less than 5 times). All these features are integers.
Punctuations Statistics
The features include the number and ratios of the following punctuations: [`’`, `-`, `;`, `:`, `.`, `!`, `?`, `(`, `)`, `[`, `]`, `{`, `}`, `/`, `"`, `'`, `_`]. The model incorporates the percentage of characters that are one of the 17 different punctuation marks shown above. These features are integers. It also includes the total punctuation percentage, which is the sum of these individual percentages. These values are between 0 and 1.
TF-IDF Vectorizer
The model uses the TF-IDF vectorizer provided by ski-kit as another feature set. Term frequency (TF) measures how often a word appears in a specific document as shown in Equation 1. A higher frequency suggests greater importance within that document. Inverse document frequency (IDF) captures the term popularity as an inverse of the overall corpus as shown in Equation 2. Our TF-IDF matrix has a max of 10000 features.

4 Features

Describe any features you used for your model, or how your data was input to your model. Are you doing feature engineering or feature selection? Are you learning embeddings? Is it all part of one neural network? Refer to item 2. This may range from 0.25 pages to 0.5 pages.

5 Implementation

Describe your model and implementation here. Refer to item 4. This may take around a page.

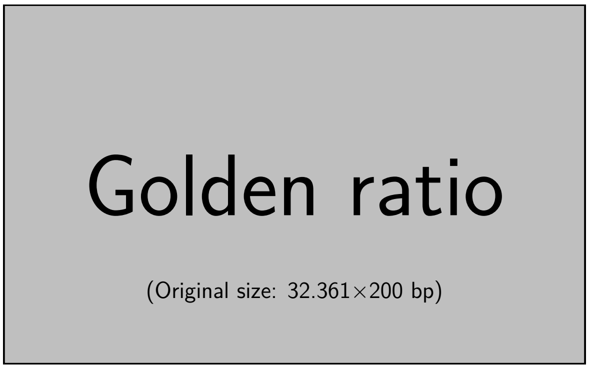

Figure 1: A figure with a caption that runs for more than one line. Example image is usually available through the `mwe` package without even mentioning it in the preamble.

6 Results and Evaluation

How are you evaluating your model? What results do you have so far? What are your baselines? Refer to item 5. This may take around 0.5 pages.

7 Feedback and Plans

Write about your plans for the remainder of the project. This should include a discussion of the feedback you received from your TA, and how you plan to improve your approach. Reflect on your implementation and areas for improvement. Refer to item 6. This may be around 0.5 pages.

7.1 Tables and figures

7.2 Citations

7.3 References

Many websites where you can find academic papers also allow you to export a bib file for citation or bib formatted entry. Copy this into the `custom.bib` and you will be able to cite the paper in the LaTeX. You can remove the example entries.

7.4 Equations

$$TF(w, d) = \frac{\text{count}(w, d)}{\text{total words in } d} \tag{1}$$

$$IDF(w, C) = \log\left(\frac{|C|}{|\{\, d \in C : w \in d \,\}|}\right) \tag{2}$$

Labels for equation numbers, sections, subsections, figures and tables are all defined with the `\label{label}` command and cross references to them are made with the `\ref{label}` command. This an example cross-reference to Equation **??**. You can also write equations inline, like this: $A = \pi r^2$.

Dataset Name	Description
Persuade Corpus 2.0	Provides argumentative essays produced by 6 to 12 grade students. It was created by The Learning Agency and Vanderbilt University, originally pulled from the following GitHub repository.
ChatGPT	Contains 2.5k student written texts sourced from the FeedBack Prize 3 Competition, and 2.5k AI-generated texts using ChatGPT. The compiled dataset includes only AI-generated texts and prompts.
Llama 70b + GPT-4	Contains 9k essays generated by Llama 70b and Falcon 180b. Prompts come from the Persuade Corpus and GPT4, using a total of 35 prompts for generation.
LLM Generated Essays	Contains 700 essays generated by LLMs 500 from GPT3.5Turbo and 200 from GPT4.
Claude Essays	Contains 1000 essays generated by Claude-Instant-1 using 15 prompts from the Persuade Corpus. Prompts were sourced from the competition discussion.
PaLM Essays	Contains 1384 essays generated by PaLM. Prompts were sourced from a Kaggle competition notebook.

Table 1: Summary of Datasets Used in the DAIGT V2 Train Dataset

Team Contributions

Write in this section a few sentences describing the contributions of each team member. What did each member work on? Refer to item 7.

References

Faris Al-Ahmadi. Ai text detection competition. https://www.kaggle.com/code/farisalahmdi/ai-text-detection-competition. Accessed: 2025-08-03.

Alexia Audevart. Kerasnlp starter notebook llm - detect ai generate. https://www.kaggle.com/code/alexia/kerasnlp-starter-notebook-llm-detect-ai-generate. Accessed: 2025-08-03.

Siti Khotijah. Using lstm for nlp: Text classification. https://www.kaggle.com/code/khotijahs1/using-lstm-for-nlp-text-classification. Accessed: 2025-08-03.

Darek Kleczek. Daigt v2 train dataset. https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-dataset/data. Accessed: 2025-08-03.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2).

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *Preprint*, arXiv:2301.11305.

Jasmine Mohammad. Detecting ai-text(svm, nn) + full ml guide. https://www.kaggle.com/code/jasminemohamed2545/detecting-ai-text-svm-nn-full-ml-guide. Accessed: 2025-08-03.

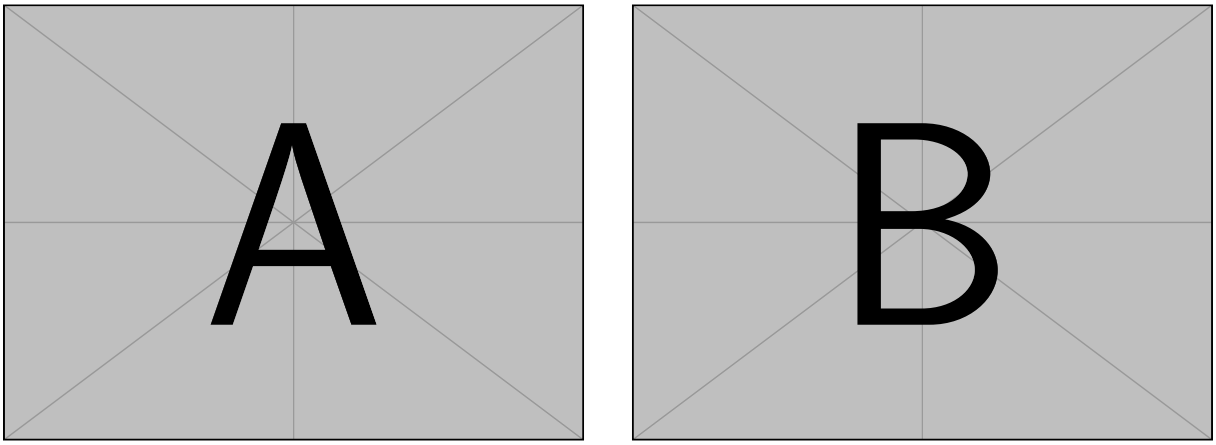

Figure 2: A minimal working example to demonstrate how to place two images side-by-side.