UCCD3074 Deep Learning for Data Science
Group Assignment
Fine-Tuning a Transformer for Detecting AI-Generated Text

Research-based [ ] Application-based [√]

Name	Charmaine Hooi Wai Yee (Leader)	Khow Kai Yong	Michelle Koh Mei Xian	Seow Yi Xuan
Programme	CS	CS	CS	CS
ID	2104533	2105725	2103784	2105524
Contribution	1/4	1/4	1/4	1/4

1. INTRODUCTION

Done by: Khow Kai Yong (2105725)

Problem Statement

With the rapid evolution and wide adoption of large language models (LLMs) such as GPT, Claude, DeepSeek and other generative AI systems, differentiating between AI-generated and human-written text becomes increasingly difficult [1]. This causes the authenticity of the work and assignment to be hard to determine [2]. This issue is extremely important in the education sector as assignments are the key metric to assessing students' performance. Besides, the correctness of information provided by AI is still questionable [3]. This may mislead the user. It is vital to create a reliable system to classify AI-generated text and human-authored text.

Motivation

i. Misuse of AI in Academic Study

The drastic growth of AI tools increases the challenges in academic fields as the students may use AI to complete their homework and assignments which should be completed by themselves. The education sector requires a reliable system to identify and check academic integrity so that the assessment evaluation can be done in a fair way [1][2].

ii. Content Accuracy and Misinformation

The reliability of AI-generated content is sometimes doubtful as the information or response provided can be misleading or incorrect. Some may post information with ambiguity from AI to social media which may mislead the public. Hence, it is important to innovate the current detection system to ensure the authenticity of information [3][4].

Background of Project

In the early days, AI text detection depended on the statistical analysis of linguistic features such as perplexity scores and language model likelihood, N-gram frequency analysis, Stylometric features and readability metrics and syntactic complexity measures [5][6].

Recent approaches have leveraged neural networks for AI-text detection including RoBERTa-based Detectors, BERT-based Classification and Ensemble Methods. Among the methods, RoBERTa models are used in OpenAI's GPT-2 and the following enhancements to fine-tune human and AI text detector [7][8]; BERT is used in various implementations for the binary classification of text authenticity [9][10]; Ensemble methods combine multiple detection ways together such as integrating transformer-based models with traditional linguistic features [11]. There are few commercial solutions existing; some have been discontinued while some are still in use. For example, OpenAI's AI Classifier, GPTZero, Originality.AI and Turnitin AI Writing Detection [12]. However, the current system has some limitations. The limitations include

vulnerable to paraphrasing and engineering attacks, having poor performance when facing different writing styles and topics, challenging in detecting text from the latest AI models and high rate in misclassification for the non-native writers and distinct writing styles [13][14].

Challenges of Project

i. Dataset Quality and Size

The amount of AI-generated and human-written text from the sample should be balanced to avoid bias in the training. The dataset should also be large enough for the model to learn [15].

ii. Rapid Improvement of AI Capabilities

When the AI model improves, the differences between the text written by humans and AI become fewer. This increases the difficulty of detectors in differentiating between them [16].

Project Scope

This project will fine-tune a transformer for the binary classification of text into AI-generated and human-authored categories using PyTorch and the dataset from Hugging Face. The transformer model used is DistilRoBERTa. The scope of the project includes data collection and preprocessing, model selection and fine-tuning, training and optimization, as well as evaluation and analysis. The project will only focus on the English text detection so that the project is feasible within the timeframe.

f. Objective

i. To develop a fine-tuned transformer model that can reliably detect AI-generated text across a wide range of contexts

This involves the training of a model on the dataset which consists of sample text written by humans and generated by AI to differentiate the AI content from human-written content.

ii. To assess the generalization of models across diverse text styles and domains

To test the robustness of system models in the accuracy of classifying the different types of writing into the correct categories (human-written or AI-generated).

2. RELATED WORK (IF ANY)

Done by: Michelle Koh Mei Xian (2103784)

The detection of AI-generated text has grown in importance field in cybersecurity, content production, and education in recent years. Many tools and studies have been created to tackle this problem. This section evaluates the advantages and disadvantages of current AI-generated text detectors.

I. GPTZero

Edward Tian launched GPTZero in early 2023 in response to increasing concerns about student abusing AI-generated material for academic assignments [19]. It differences between text produced by AI and text authored by humans using statistical measures like burstiness and perplexity [20].

- Burstiness: It assumes that human writing displays greater irregularity than AI-generated writing and assesses the variation of confusion across phrases within a text.
- Perplexity: It is an amount of uncertainty in the value of a sample taken from a discrete probability distribution. An observer's ability to guess the value that will be extracted from the distribution decreases with increasing confusion.

Strengths	Weaknesses
• Easy to use web-based tools • Quick analysis that is appropriate for journalists and educators. • Installation of sophisticated hardware or software is not necessary.	• Lower accuracy on short texts • Subject to false positives, particularly when it comes to writers who are not native English speakers. • Insufficient visibility about model upgrades and training

Table 2.1 Strengths and Weaknesses of GPTZero [21]

II. Turnitin AI Detection

In April 2023, the famous plagiarism detection website Turnitin released AI-generated writing detection capabilities. The system combines proprietary machine learning algorithms with the recognition of linguistic patterns. To determine if a text is artificial intelligence (AI) created, it examines syntactic patterns, word choice, and sentence structure [22]. Many academic institutions prefer Turnitin's technology to GPTZero because it is integrated into its broader plagiarism-checking ecosystem.

Strengths	Weaknesses
• Easily integrated with services that identify plagiarism. • Offer a probability score for AI authoring along with reports. • Reliable results on academic texts that are deeper and more structured.	• Proprietary algorithm: The detection process is not entirely revealed. • It requires institutional authorization and is not a stand-alone tool. • It might not work well for creative or informal writing styles.

Table 2.2 Strengths and Weaknesses of Turnitin AI Detection [23]

Academic conditions are the primary focus of both Turnitin and GPTZero. Turnitin provides a more comprehensive but institution-dependent solution, whereas GPTZero places a higher priority on ease of use and accessibility. In order to achieve performance on the same level with reliable systems like Turnitin while preserving the flexibility and model transparency desired in more approachable solutions like GPTZero, our project focuses on creating a refined transformer model that strikes a balance between detection accuracy and generalization across a variety of text styles.

3. SYSTEM DESIGN OF TRANSFORMER-BASED AI TEXT DECTION SYSTEM

Done by: Charmaine Hooi Wai Yee (2104533)

This chapter discusses in detail the design and development of a transformer-based AI text detection system that aims to classify input text as either human-written or AI-generated, using

a fine-tuned transformer-based binary classifier. The top-down system design diagram shown in Figure 3.1 is used to illustrate the overall system architecture of the system which will be further discussed in the subsections below.

The system is designed as an application-based project. As shown in Figure 3.1, the pipeline consists of four core components:

1. Data Preprocessing
2. Fine-Tuned Transformer
3. Binary Classification Layer
4. Evaluation Framework

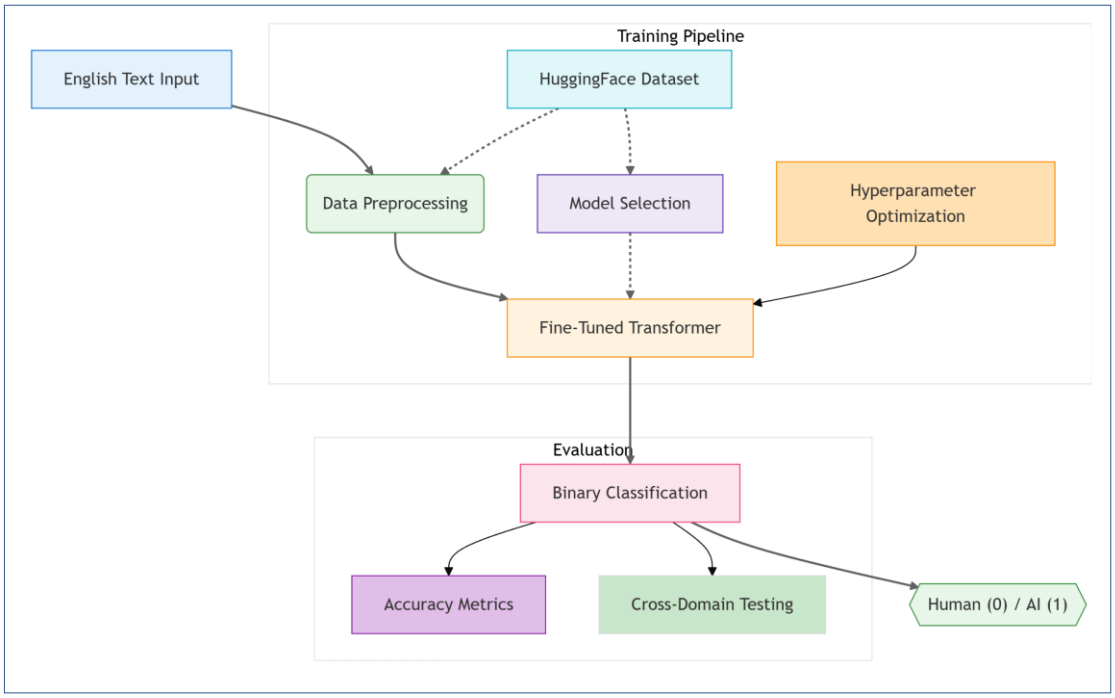


Figure 3.1: Use a top-down design diagram to show your overall system

The core system processes English text input before classifying it as either human-generated (0) or AI-generated (1). The English text input is then fed into the Data Preprocessing step. After data preprocessing, the data is inputted into a Fine-Tuned Transformer model that passes its output to a Binary Classification layer. This final output is a determination of whether the text was generated by a Human (0) or AI (1).

Data Collection

In the first step, the system accepts text passages ranging from 100 to 300 words. Data was obtained from Hugging Face's public repository [24]. This dataset contains roughly 400,000 rows of processed data with labeled text passages that are either human-written (0) or AI-generated (1).

Data Preprocessing

The Data Preprocessing step transforms raw text into model-digestible structured tensors, preserving discriminative features essential for accurate classification. **Tokenization** is the primary operation, mapping words/sub-words to token IDs using pre-trained tokenizers like RobertaTokenizer. To standardize input length and optimize resources, all sequences are

limited to a fixed maximum length, with shorter sequences padded and longer ones truncated. **Attention masks** are generated to distinguish content from padding. Preprocessed samples are saved in a format compatible with Hugging Face's datasets library.

Fine-Tuned Transformer

Preprocessed data is fed into a **Fine-Tuned Transformer**, leveraging pre-trained models such as DistilRoBERTa. These models are adapted for text classification, extracting complex patterns and contextual relationships crucial for distinguishing human from AI-generated content. A linear classification head is added atop the transformer's pooled output. During fine-tuning, the pre-trained weights and the new head are adjusted, specializing the model in AI text detection while benefiting from its prior language knowledge.

Binary Classification

Following the transformer, a Binary Classification layer takes the learned representations and produces a probability score, which is then thresholded to yield the final binary classification, namely Human (0) or AI (1).

Training Pipeline

The training pipeline will be built with Hugging Face Transformers, where all text samples were tokenized with the RoBerta tokenizer which are then padded to 512 tokens to balance efficiency and context preservation. RoBERTa-based model with a binary classification head is fine-tuned using the AdamW optimizer (learning rate 2e-5, weight decay 0.01) for 2 epochs with batch sizes of 32 (train) and 64 (eval). Training was managed by the Trainer API, which handled optimization, validation, and checkpointing. On the other hand, a custom compute_metrics function will report accuracy, precision, recall, and F1. At the end of training, the best model checkpoint (selected by F1-score) and tokenizer will be saved in a timestamped directory for reproducibility and later evaluation.

Evaluation Framework

Lastly, to ensure the robustness and accuracy of the system, a comprehensive Evaluation phase is conducted. The outputs from the Binary Classification are rigorously assessed using Accuracy Metrics to quantify performance.

- **Accuracy:** Proportion of correctly classified samples.
- **Precision:** Correctly identified AI texts among predicted AI texts.
- **Recall:** Correctly identified AI texts among actual AI texts.
- **F1-score:** Harmonic mean of precision and recall.
- **Confusion Matrix:** Visualizes true positives, true negatives, false positives, and false negatives.

Furthermore, Cross-Domain Testing is performed to evaluate the model's generalization capabilities across different types and styles of text.

4. EXPERIMENT & EVALUATION

Done by: Seow Yi Xuan (2105524)

In-Domain Testing

```
 Train Set RESULTS
Accuracy:  0.9861
Precision: 0.9742
Recall:    0.9985
F1-Score:  0.9862
ROC-AUC:   0.9997
PR-AUC:    0.9997

Classification Report (0=Human, 1=AI):
              precision    recall  f1-score   support

       Human     0.9985    0.9737    0.9859      6017
          AI     0.9742    0.9985    0.9862      5973

    accuracy                         0.9861     11990
   macro avg     0.9863    0.9861    0.9861     11990
weighted avg     0.9864    0.9861    0.9861     11990

Confusion Matrix [[TN, FP], [FN, TP]]:
[[5859  158]
 [   9 5964]]
```

```
 Test Set RESULTS
Accuracy:  0.9848
Precision: 0.9631
Recall:    0.9982
F1-Score:  0.9803
ROC-AUC:   0.9996
PR-AUC:    0.9995

Classification Report (0=Human, 1=AI):
              precision    recall  f1-score   support

       Human     0.9989    0.9767    0.9876     15125
          AI     0.9631    0.9982    0.9803      9231

    accuracy                         0.9848     24356
   macro avg     0.9810    0.9874    0.9840     24356
weighted avg     0.9853    0.9848    0.9849     24356

Confusion Matrix [[TN, FP], [FN, TP]]:
[[14772   353]
 [   17  9214]]
```

The performance metrics above show the system's ability to distinguish human-written and AI-generated text. On the training set, the system achieved an accuracy of 98.6%, while on the test set, it reached 98.5%, showing that the method works consistently well on unseen data. The recall values are extremely high (99.8% for both sets), meaning the system almost never misses identifying the correct category. Precision is also strong (97.4% on training and 96.3% on testing), indicating that the model is usually correct when predicting a text as AI-generated. The F1-scores, which balance both precision and recall, are above 98%, confirming the overall reliability of the system. In the test set of 24,356 samples, the model made only 370 mistakes in total. Out of these, 353 human texts were misclassified as AI-generated, and 17 AI texts were misclassified as human-written. Similarly, in the training set of 11,990 samples, the model made only 167 mistakes, with most errors coming from human texts being labeled as AI. These low error rates highlight that the system is more cautious with human text, but overall, the accuracy remains extremely high.

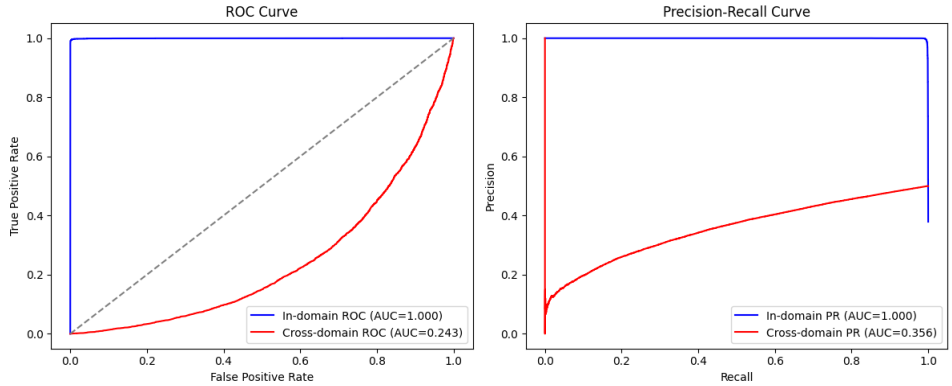
Cross-Domain Testing

The cross-domain test was conducted using the AH&AITD dataset which is larger, more diverse, and includes formal texts from older sources than the original dataset.

```
 Cross-Domain Test RESULTS
Accuracy:  0.4927
Precision: 0.4963
Recall:    0.9829
F1-Score:  0.6596
ROC-AUC:   0.2426
PR-AUC:    0.3557

Classification Report (0=Human, 1=AI):
              precision    recall  f1-score   support

       Human     0.1316    0.0026    0.0051      5790
          AI     0.4963    0.9829    0.6596      5790

    accuracy                         0.4927     11580
   macro avg     0.3140    0.4927    0.3323     11580
weighted avg     0.3140    0.4927    0.3323     11580

Confusion Matrix [[TN, FP], [FN, TP]]:
[[  15 5775]
 [  99 5691]]
```

The cross-domain test result's overall accuracy is only 49.27%, which is close to random guessing. While recall for AI-generated text remains very high (98.29%), the recall for human-written text collapses to near zero (0.26%). This means the model can correctly identify most AI texts but misclassifies almost all human texts as AI. The precision for both classes is also

imbalanced, with AI precision at 49.63% and human precision at only 13.16%, reflecting a bias towards predicting the AI class. The confusion matrix further illustrates this imbalance. Out of 5,790 human samples, only 15 were classified correctly, while nearly all were incorrectly labeled as AI. In contrast, most AI samples were correctly identified (5,691 out of 5,790). This extreme skew shows that the model has effectively lost the ability to recognize human-written text in the cross-domain dataset.



The results highlight a sharp contrast between in-domain and cross-domain performance. For the in-domain dataset, the model achieves nearly perfect separation, with ROC and Precision-Recall AUC scores close to 1.0, indicating excellent ability to distinguish between human-written and AI-generated text when evaluated on data similar to its training set. However, performance drops drastically on the cross-domain dataset, where the ROC AUC falls to 0.243 and the Precision-Recall AUC to 0.356. These values suggest that the model struggles to generalize outside its training domain and fails to reliably capture the distinguishing patterns in unseen data sources.

Several factors may explain this poor cross-domain performance. First, the training and cross-domain datasets may differ significantly in writing style, vocabulary, or topics, leading to a distribution mismatch. Second, the model may have overfitted to domain-specific patterns in the training data such as stylistic cues or formatting rather than learning more generalizable linguistic features. Third, it is possible that the cross-domain dataset introduces noise or label inconsistencies that confuse the model. Collectively, these issues highlight that while the system is highly effective on familiar data, its predictive power does not transfer well to new domains, emphasizing the importance of robust training strategies, domain adaptation, or more diverse training data to improve real-world applicability.

5. CONCLUSION

Done by: Khow Kai Yong (2105725)

In this project, a DistilRoBERTa Transformer model was fine-tuned to detect AI-generated text. The model achieved strong performance on the in-domain dataset, demonstrating its effectiveness in distinguishing between human-written and AI-generated content within the same distribution. However, cross-domain testing revealed a significant performance drop, highlighting the model's sensitivity to dataset differences and limited generalizability. These findings suggest that while distilled models can be efficient and competitive in resource-limited settings, future work should focus on improving robustness across domains. This may involve training on more diverse datasets, exploring domain adaptation techniques, and extending to multilingual contexts. Such improvements are necessary before considering deployment in practical applications, such as education, where input data may vary widely.

REFERENCES

[1] Cotton, D. R., Cotton, P. A., & Shipway, J. R. (2023). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International*, 61(2), 228-235.

[2] Tlili, A., Shehata, B., Adarkwah, M. A., et al. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments*, 10(1), 15.

[3] Kreps, S., McCain, R. M., & Brundage, M. (2022). All the news that's fit to fabricate: AI-generated text as a tool of media misinformation. *Journal of Experimental Political Science*, 9(1), 104-117.

[4] Goldstein, J. A., Sastry, G., Musser, M., et al. (2023). Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint arXiv:2301.04246*.

[5] Gehrmann, S., Strobelt, H., & Rush, A. M. (2019). GLTR: Statistical detection and visualization of generated text. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 111-116.

[6] Ippolito, D., Duckworth, D., Callison-Burch, C., & Eck, D. (2020). Automatic detection of generated text is easiest when humans are fooled. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1808-1822.

[7] Solaiman, I., Brundage, M., Clark, J., et al. (2019). Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

[8] Radford, A., Wu, J., Child, R., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.

[9] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171-4186.

[10] Liu, Y., Ott, M., Goyal, N., et al. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

[11] Rodriguez, J., Hay, T., Gros, D., et al. (2022). Cross-domain detection of GPT-2-generated technical text. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, 1213-1233.

[12] OpenAI. (2023). New AI classifier for indicating AI-written text. *OpenAI Blog*. Retrieved from https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

[13] Sadasivan, V. S., Kumar, A., Balasubramanian, S., et al. (2023). Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

[14] Liang, W., Yuksekgonul, M., Mao, Y., et al. (2023). GPT detectors are biased against non-native English writers. *Patterns*, 4(7), 100779.

[15] Kaplan, J., McCandlish, S., Henighan, T., et al. (2020). Scaling laws for neural language models. *arXiv preprint arXiv*:2001.08361.

[16] Clark, E., August, T., Serrano, S., et al. (2021). All that's 'human' is not gold: Evaluating human evaluation of generated text. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 7282-7296.

[17] Uchendu, A., Le, T., Shu, K., & Lee, D. (2020). Authorship attribution for neural text generation. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 8384-8395.

[18] Krishna, K., Song, Y., Karpinska, M., et al. (2023). Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Proceedings of NeurIPS*, 36, 15820-15837.

[19] M. Renbarger, "How a 23-year-old college student built one of the leading AI detection tools," *Business Insider ,* 2023.

[20] J. Hartman-Sigall, "Edward Tian ’23 creates GPTZero, software to detect plagiarism from AI bot ChatGPT," *The Daily Princetonian,* 2023.

[21] E. Tian, "GPTZero: Detect AI-generated text," 2023. [Online]. Available: https://gptzero.me/.

[22] M. O. IMAM, "AN EVALUATION OF THE LINGUISTIC ASPECTS OF TURNITIN," *ReasearchGate,* 2019.

[23] Turnitin, "Turnitin's AI writing detection capabilities FAQs," June 2025. [Online]. Available: https://guides.turnitin.com/hc/en-us/articles/28477544839821-Turnitin-s-AI-writing-detection-capabilities-FAQs.

[24] Hugging Face, “andythetechnerd03/AI-human-text · Datasets at Hugging Face,” February 27, 2024. Available: andythetechnerd03/AI-human-text · Datasets at Hugging Face.