nits_teja_srikar at GenAI Detection Task 2: Distinguishing Human and AI-Generated Essays Using Machine Learning and Transformer Models

L D M S Sai Teja, Annepaka Yadagiri, M Srikar Vardhan and Partha Pakray
Department of Computer Science & Engineering
National Institute of Technology Silchar, Assam, India, 788010
{lekkalad_ug, annepaka22_rs, mangadoddis_ug, partha}@cse.nits.ac.in

Abstract

This paper presents models to differentiate between human-written and AI-generated essays, addressing challenges posed by advanced AI models like ChatGPT and Claude. Using a structured dataset, we fine-tune multiple machine learning models, including XGBoost and Logistic Regression, along with ensemble learning and k-fold cross-validation. The dataset is processed through text cleaning, lemmatization, stemming, and part-of-speech tagging, followed by TF-IDF vectorization before training. Our team nits_teja_srikar achieves high accuracy, with DistilBERT performing at 77.3% accuracy, standing at 20th position for English, and XLM-RoBERTa excelling in Arabic at 92.2%, standing at 14th position in the official leaderboard, demonstrating the model's potential for real-world applications.

1 Introduction

Generative Artificial Intelligence (AI) tools (Tkachov, 2024) are revolutionizing the creation of text, images, and videos, reshaping how society consumes and produces online content. As these technologies advance, distinguishing between AI-generated and human-generated content has become increasingly challenging. AI-Generated Content (*AIGC*) (Staff, 2024) spans a broad range of media, including text, code, images, and music, with rapidly expanding applications in areas such as news reports, blog posts, scriptwriting, and marketing copy.

A study published in *The Public Library of Science (PLOS)* found that readers were more likely to agree with arguments in AI-generated essays than those in human-written ones (Bal and Veltkamp, 2013). These findings underscore the increasing need for models reliably differentiating between AI-generated and human-authored content. In this paper, we employ a fine-tuning approach utilizing multiple Machine Learning (*ML*) models, including XGBoost and Logistic Regression (Google-Research), along with their k-fold cross-validation variants (Ismail et al., 2023), and an ensemble learning method. A hybrid model integrating all models and their k-fold variants was developed using ensemble learning (Xiong et al., 2024) and transformer models using DistilBERT (Sanh et al., 2019) and XLM-RoBERTa (Conneau et al., 2019) to further improve detection accuracy in distinguishing between human and AI-generated essays.

2 Related Work

Liao et al. (Liao et al., 2023) proposed an ethical framework for Artificial Intelligence Generated Content (*AIGC*) within the healthcare sector. This study investigated the differences between medical texts authored by ChatGPT and those written by healthcare professionals. Additionally, the authors developed machine learning workflows aimed at identifying and distinguishing medical texts generated by ChatGPT. To achieve this, they curated datasets, with one dataset consisting of ChatGPT-generated medical texts and the other containing texts authored by human experts. Subsequently, they implemented ML methods to determine the source of the medical text content.

Alamleh et al. (Alamleh et al., 2023) assessed the effectiveness of various ML methods in differentiating AI-generated text from text authored by humans. To carry out this analysis, they collected responses from computer science students to both essay and programming assignments. Using this dataset, they trained and evaluated multiple machine learning models, including Support Vector Machines (*SVM*), Logistic Regression (*LR*), Neural Networks (*NN*), Random Forest (*RF*), and Decision Trees (*DT*).

Chen et al. (Chen et al., 2023) proposed a novel approach to distinguish between texts written

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 278–283
January 19, 2025.

by humans and those generated by ChatGPT using language-based techniques. The researchers collected and released a curated dataset named OpenGPTText, comprising rephrased content created through ChatGPT.

2.1 Problem Statement

The primary objective of this task is to classify the provided essays as either **Human-generated** or **AI-generated** in two languages: **English** and **Arabic**. This classification problem aims to assess the *authenticity* of the text, ensuring a reliable distinction between human-authored and AI-generated content. The ability to detect AI-generated text effectively is crucial for applications such as content moderation, academic integrity, and automated scoring systems (Genai-content-detection / Genai-content-detection-coling-2025 · GitLab, n.d.).

2.2 Dataset Description

We used data from the shared task dataset provided by the Linguistic Data Consortium (LDC) (ETS Corpus of Non-Native Written English, n.d.) by COLING 2025 (Chowdhury et al., 2025). The ETS Corpus of Non-Native Written English includes 12,100 TOEFL essays from speakers of 11 non-English languages. Additionally, AI-generated essays from models like GPT-3.5-Turbo (Mrbullwinkle, 2024), GPT-4o (Stryker, 2024), Gemini-1.5 (Google AI for Developers, n.d.), and others were incorporated. This combination of human and AI-generated content facilitates a thorough comparative analysis, which significantly contributes to research in authorship detection, automatic scoring, and understanding the distinctions between human and AI writing.

2.3 Dataset Visualization

This visualization highlights the key differences between AI-generated and human-authored essays [1] [2], focusing on several critical aspects. It presents the variations in essay length distribution (1b), (2b), and a comparative analysis of essay lengths by source. Additionally, it offers insights into key textual metrics, including word count, sentence count, and unique word count (1a), (2a), shedding light on the structural and stylistic differences between the two types of content. This analysis highlights the distinct features of human and AI writing, contributing to a deeper understanding of their unique characteristics.

3 Methodology

We leverage both traditional machine-learning models and advanced transformer-based models. Below, we provide a detailed technical description of our methodology for both datasets.

3.1 Preprocessing Techniques

We employed preprocessing techniques like Cleaning, lowercasing, lemmatization and stemming, POS tagging, removing stop words to enhance data quality: 1) Text Cleaning: Removal of punctuation, special characters, and redundant spaces. 2) Lowercasing: Uniform formatting of all text entries. 3) Lemmatization and Stemming: Standardization of word forms for better vectorization. 4) POS Tagging: Advanced feature extraction by identifying grammatical roles using spacy library. 5) Stopword Removal: Elimination of common words with minimal semantic contribution.

```
input_text = "The quick brown foxes were jumping
     joyfully, over the lazy dogs!"

After preprocessing:

pos = "quick/ADJ brown/ADJ foxes/NOUN jumping/
    VERB joyfully/ADV lazy/ADJ dogs/NOUN"

result = "quick brown foxes jumping joyfully
    lazy dogs"
```

3.2 TF-IDF Vectorization

To effectively analyze text data, we convert documents into numerical representations using **Term Frequency-Inverse Document Frequency (*TF-IDF*)** vectorization. This method assigns importance to words based on their occurrence within individual documents and across the entire corpus. TF-IDF helps emphasize significant terms while downplaying those that are common but carry little meaning.

Term Frequency-Inverse Document Frequency (*TF-IDF*):

$$\mathrm{TF}(t, d_i) = \frac{f_{t,d_i}}{\sum_{t' \in d_i} f_{t',d_i}} \tag{1}$$

$$\mathrm{IDF}(t, D) = \log\left(\frac{N}{|\{d_j \in D : t \in d_j\}| + 1}\right) \tag{2}$$

TF-IDF Calculation and Matrix Construction: The TF-IDF score for a term t in document d_i is computed as the product of its Term Frequency (TF) and Inverse Document Frequency (IDF):

$$\text{TF-IDF}(t, d_i, D) = \text{TF}(t, d_i) \times \text{IDF}(t, D) \quad (3)$$

Each document is represented as a vector of TF-IDF values, with $x_i = (\text{TF-IDF}(t_1, d_i, D), \ldots, \text{TF-IDF}(t_m, d_i, D))$, where m is the vocabulary size. Stacking these vectors for all N documents forms a TF-IDF matrix $X \in \mathbb{R}^{N \times m}$, where each row corresponds to a document, and each column corresponds to a term.

To ensure uniformity, we apply Euclidean (L2) normalization:

$$x_i^{\text{norm}} = \frac{x_i}{\|x_i\|_2} = \frac{x_i}{\sqrt{\sum_{j=1}^{m} x_{ij}^2}} \quad (4)$$

where x_{ij} is the TF-IDF value of the j^{th} term in the i^{th} document.

3.3 HyperParameters

In this training, SparseCategoricalCrossentropy was used as the loss function for multi-class classification (Mao et al., 2023). Optimizers Adam and AdamW (Jakartamitul, 2024) were applied with learning rates of 5e-4 and 2e-5, respectively. Key metrics included SparseCategoricalAccuracy, ROC AUC (Bowers and Zhou, 2019), Accuracy, Precision, Recall, F1 Score, and MCC (Chicco and Jurman, 2020). The training was performed over 3 epochs with batch size 16 and 5-fold cross-validation, preserving class distribution. TF-IDF vectorization used 5000 max_features to optimize efficiency, and soft voting was applied in the ensemble model for enhanced accuracy. Overall hyperparameters can be seen in Table 1

Parameter	Value
Loss Function	SSCE
Optimizer	Adam, AdamW
Learning Rate	5e-4, 2e-5
Metrics	Accu, Prec, Recall, F1
Batch Size	16
Number of Epochs	3 (Transformer models)
Early Stopping	No
n_splits	5
max_features	5000
Voting	soft

Table 1: Model Hyperparameters

3.4 Models for English Dataset

For the English dataset D_{eng}, we employ the following models:

Logistic Regression with K-fold Cross-Validation We split the dataset into k folds and train the logistic regression model iteratively. The logistic regression function is defined as:

$$\hat{y}_i = \sigma(w^\top x_i + b) = \frac{1}{1 + e^{-(w^\top x_i + b)}}$$

where $\sigma(\cdot)$ is the sigmoid function, and w, b are learned parameters.

XGBoost with K-fold Cross-Validation XGBoost minimizes the following objective function:

$$\mathcal{L} = \sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where $\ell(\hat{y}_i, y_i)$ is the loss function, and $\Omega(f_k)$ is the regularization term.

Ensemble Learning We combine predictions from logistic regression, XGBoost, and CatBoost using a weighted majority voting scheme:

$$\hat{y} = \arg\max_c \sum_{m=1}^{M} w_m \cdot \mathbb{I}(\hat{y}_m = c)$$

where M is the number of models, w_m is the weight of the m^{th} model, and $\mathbb{I}(\cdot)$ is the indicator function.

DistilBERT Transformer We use the DistilBERT pretrained transformer as a binary classifier to distinguish between human-written and AI-generated essays. DistilBERT, a lighter version of BERT, retains its ability to capture subtle linguistic and contextual patterns while being computationally efficient. The classifier processes input text to generate hidden representations h_i, and the final prediction is made with:

$$\hat{y}_i = \text{softmax}(W h_i + b)$$

where W and b are trainable parameters, and $\hat{y}_i$ is the probability distribution over the two classes. The output class is determined by the highest probability score.

3.5 Model for Arabic Dataset

Given that our dataset is in Arabic, XLM-RoBERTa ensures better performance by leveraging its pre-trained embeddings specific to the language. The model produces rich contextual embeddings for each input sequence, which are fed into a classification layer. For Arabic, the pre-trained embeddings are particularly significant, as they capture complex morphological and syntactic patterns that are challenging to model through linguistic features alone. The final prediction is computed as:

$$\hat{y}_i = \text{softmax}(Wh_i + b)$$

where h_i is the contextual representation of the input, and W and b are trainable parameters. The softmax layer outputs a probability distribution over the target classes, allowing the model to classify the text accurately within the given context.

Model	ROC	Acc	Pre	Rec	F1
LR	1.00	0.99	0.99	1.00	0.99
XGB	0.99	0.98	0.98	0.99	0.99
CatB	0.92	0.89	0.92	0.92	0.92
En	0.99	0.99	0.99	0.99	0.99

Table 2: Performance Metrics of Machine Learning Models on English Dataset

Model	Acc	Pre	Rec	F1
DistilBERT-En	0.99	0.95	1.00	0.97
XLM-R-Ar	0.94	1.00	0.91	0.95
DistilBERT-Ar	0.91	0.93	0.91	0.91

Table 3: Performance Metrics of Transformer Models

4 Results

The DistilBERT model excels in the English dataset, while XLM-RoBERTa performs well in Arabic. The confusion matrix highlights their accuracy and offers insights into their classification performance.

For English, DistilBERT-English achieves a perfect recall of 1.0 and an F1-score of 0.97. The ensemble model, combining Logistic Regression, XGBoost, and CatBoost, reaches an accuracy of 0.996 and an F1-score of 0.997. Overall results can be seen in Table 2. In contrast, Arabic models show lower performance, with XLM-RoBERTa Arabic leading in precision at 1.0 and an F1-score of 0.95, while DistilBERT-Arabic has an F1-score of 0.91. This highlights the challenges faced by Arabic models. The transformer-based models results can be seen in the Table 3

4.1 Ablation Studies

To evaluate the impact of preprocessing, feature extraction, and ensemble components, we conducted systematic experiments:

- **Preprocessing:** Removing *stemming and lemmatization* improved transformer models' performance by 1.43 on English datasets. Removing *POS tagging* reduced traditional model accuracy by 1.8, indicating its importance for feature engineering.
- **Feature Extraction:** Traditional models saw a 6% accuracy improvement using TF-IDF, while transformers performed better with raw text due to their pre-trained embeddings.

Preprocessing steps such as lemmatization and part-of-speech tagging play a vital role in feature extraction for traditional models. However, minimal preprocessing often yields better performance for transformer-based models by effectively utilizing raw textual features.

Model	Acc	Pre	Rec	F1
English	0.773	0.875	0.649	0.658
Arabic	0.922	0.943	0.882	0.904

Table 4: Final Test Results in Leaderboard

We got 77% accuracy on English essays, and performance on Arabic essays was significantly higher. Several linguistic and dataset-specific factors contribute to this gap. Firstly, Arabic's morphological complexity, including intricate word forms and inflections, presents challenges for tokenization and embedding generation. Unlike English, where words have simpler inflections, Arabic requires handling more complex word transformations, which can be harder for DistilBERT to capture. Additionally, Arabic's right-to-left writing and unique orthographic conventions further complicate processing.

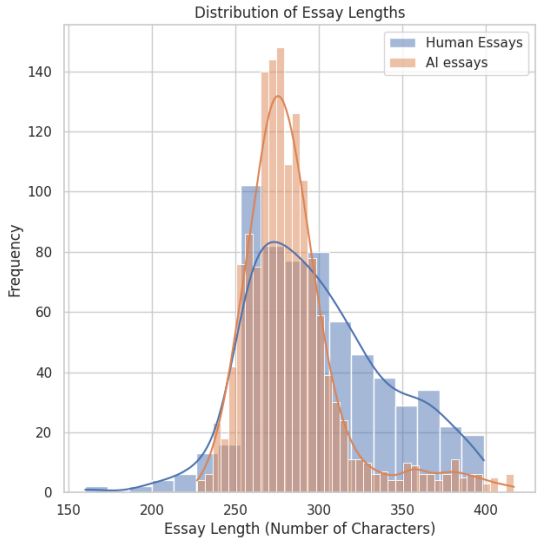
Moreover, Arabic requires more sophisticated preprocessing steps, such as handling diacritics and lemmatization, which might not have been as effectively implemented as for English. Finally, cultural and stylistic differences between English and Arabic writing may also contribute to the difficulty in detecting AI-generated content in Arabic.

5 Conclusion

Our study demonstrates the efficacy of various ML and transformer-based models in distinguishing between human-generated and AI-generated essays in both English and Arabic. Utilizing models like DistilBERT and XLM-RoBERTa, we achieved superior detection precisions of 0.875 for English and a strong performance of 0.943 for Arabic, as shown in Table 4, highlighting their adaptability to diverse linguistic contexts. The ensemble methods further enhanced classification accuracy, emphasizing the importance of robust detection systems as AI-generated content becomes increasingly prevalent. Future work could explore additional linguistic features and cross-domain applications to improve detection capabilities and address challenges in Arabic model performance.

Label	Count	Word	Sentence	Unique
Human	629	300.24	15.57	173.48
AI	1467	284.07	14.51	155.24

(a) Comparison of Metrics in Number Counts English

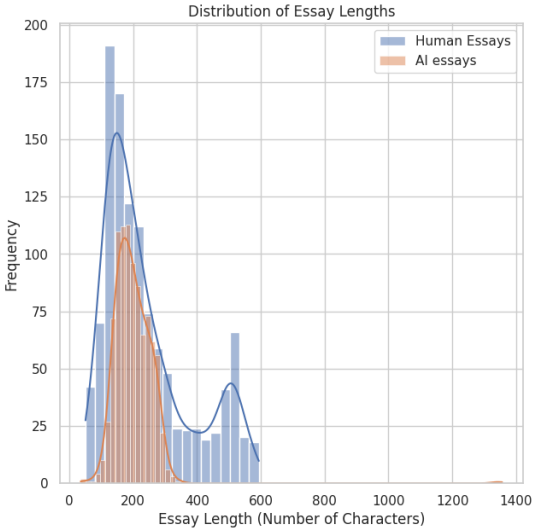


(b) Train Data Essay Length Distribution

Figure 1: Train Data Visualization for Language English

Label	Count	Word	Sentence	Unique
Human	1145	241.51	8.71	169.54
AI	925	200.96	14.33	137.57

(a) Comparison of Textual Metrics in Number Counts



(b) Train Data Essay Length Distribution

Figure 2: Train Data Visualization for Arabic

References

Hosam Alamleh, Ali Abdullah S AlQahtani, and AbdElRahman ElSaid. 2023. Distinguishing human-written and chatgpt-generated text using machine learning. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*, pages 154–158. IEEE.

P. M. Bal and M. Veltkamp. 2013. How does fiction reading influence empathy? an experimental investigation on the role of emotional transportation. *PLoS ONE*, 8(1):e55341.

A. J. Bowers and X. Zhou. 2019. Receiver operating characteristic (roc) area under the curve (auc): A diagnostic measure for evaluating the accuracy of predictors of education outcomes. *Journal of Education for Students Placed at Risk (JESPAR)*, 24(1):20–46.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.

D. Chicco and G. Jurman. 2020. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1).

Shammur Absar Chowdhury, Hind Al-Merekhi, Mucahid Kutlu, Kaan Efe Keleş, Fatema Ahmad, Tasnim Mohiuddin, Georgios Mikros, and Firoj Alam. 2025. GenAI content detection task 2: AI vs. human – academic essay authenticity challenge. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE. International Conference on Computational Linguistics.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv.org*.

ETS Corpus of Non-Native Written English. n.d. Ets corpus of non-native written english - linguistic data

consortium. https://catalog.ldc.upenn.edu/LDC2014T06. Accessed: 2024-10-28.

Genai-content-detection / Genai-content-detection-coling-2025 · GitLab. n.d. Genai content detection / genai content detection coling-2025. GitLab. Accessed: October 29, 2024.

Google AI for Developers. n.d. Gemini models.

Google-Research. Github - google-research/tuning_playbook: A playbook for systematically maximizing the performance of deep learning models. https://github.com/google-research/tuning_playbook. N.d., Accessed: 2024-10-27.

W. N. Ismail, H. A. Alsalamah, and E. A. Mohamed. 2023. Genetic-efficient fine-tuning with layer pruning on multimodal covid-19 medical imaging. *Neural Computing and Applications*, 36(6):3215–3237.

Jakartamitul. 2024. Exploring the adamw pytorch optimizer - scriptoverflow. https://scriptoverflow.com/exploring-the-adamw-pytorch-optimizer/. Accessed: October 30, 2024.

Wenxiong Liao, Zhengliang Liu, Haixing Dai, Shaochen Xu, Zihao Wu, Yiyang Zhang, Xiaoke Huang, Dajiang Zhu, Hongmin Cai, Quanzheng Li, et al. 2023. Differentiating chatgpt-generated and human-written medical texts: quantitative study. *JMIR Medical Education*, 9(1):e48904.

A. Mao, M. Mohri, and Y. Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. https://arxiv.org/abs/2304.07288.

Mrbullwinkle. 2024. Work with the gpt-35-turbo and gpt-4 models - azure openai service.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv.org*.

C. Staff. 2024. What is generative ai? definition, applications, and impact. https://www.coursera.org/articles/what-is-generative-ai.

C. Stryker. 2024. Gpt-4o. what is gpt-4o?

N. Tkachov. 2024. Chatgpt and other ai assistants: An ultimate comparison | beetroot. Accessed: 2024-10-27.

F. Xiong, T. Markchom, Z. Zheng, S. Jung, V. Ojha, and H. Liang. 2024. Fine-tuning large language models for multigenerator, multidomain, and multilingual machine-generated text detection. Ensemble Methods: Foundations and Algorithms.