Tesla at GenAI Detection Task 2: Fast and Scalable Method for Detection of Academic Essay Authenticity

Vijayasaradhi Indurthi
IIIT Hyderabad
vijayasaradhi.i@research.iiit.ac.in

Vasudeva Varma
IIIT Hyderabad
vv@iiit.ac.in

Abstract

This paper describes a simple yet effective method to identify if academic essays have been written by students or generated through the language models in English language. We extract a set of style, language complexity, bias and subjectivity, and emotion-based features that can be used to distinguish human-written essays from machine-generated essays. Our methods rank 6th on the leaderboard, achieving an impressive F1-score of 0.986.

1 Introduction

The emergence of large language models (LLMs) such as ChatGPT, GPT-4, Claude, and other similar applications has revolutionized text generation, creating highly coherent and human-like essays. While these advancements hold promise for educational tools and creative writing, they also pose significant challenges to academic integrity. Specifically, misusing machine-generated essays in educational settings threatens to undermine genuine learning and assessment. To address this issue, the task of distinguishing between human-authored and machine-generated essays has gained critical importance. This binary classification task focuses on identifying English-language essays as machine-generated or human-written. The task's details are outlined in the task overview paper (Chowdhury et al., 2025).

An ideal text detector should be accurate, easy to train and should be robust to any adversarial attacks. It should be easy to train, inference should be fast and probably is agnostic to the machine learning model that is used to generate the text.

Our approach involves identifying key textual features based on style and statistical counts of specific words, words involving complexity, bias, affect, and moral features. By transforming the text into various feature spaces and using the transformed vectors to train simple classical ML algorithms.

Participation in this task revealed several key insights. Our system achieved 6th position on the leaderboard. However, challenges persist in distinguishing highly sophisticated machine-generated texts from essays written by proficient human authors, particularly in cases where LLMs simulate idiosyncratic human writing styles. This highlights the need for further exploration of subtle linguistic markers and the inclusion of diverse training data. Quantitatively, our system achieved an F1 score of 0.986 on the test set, achieving 6th position on the leaderboard. We have made our codebase publicly available [1] to facilitate future research in this area. We hope this encourages further innovation and collaboration in safeguarding academic integrity through advancements in machine-generated text detection.

2 Related Work

Ever since Large Language Models have started to generate coherent, human like text, the task of identifying machine generated task has gained significant attention. As the complete list of works for detecting machine generated text is exhaustive, we list some of the key works that attempt to identify machine generated text.

Primarily, the methods can be broadly categorized into 3 categories: Methods involving language specific features with simple ML models. These models usually use a simple Bag of Words or extract specific linguistic features and train traditional classification models like Logistic Regression, SVM, Random Forests or a simple neural network on the extracted feature values. The advantages with these kind of models is that they are quick to train and evaluate.

Methods involving fine-tuning a transformer (encoder or decoder only) models. These models usually use a transformer encoder architecture like

[1] https://github.com/saradhix/COLING25_DAIGen_Task2

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 317–322
January 19, 2025.

BERT, RoBERTa or a decoder only architecture like GPT to learn a classifier. This involves fine-tuning the model on both the classes of data. As the models are originally pretrained on large quantities of text, these models can understand the structure of the text.

Methods that have the text generation model available. These methods use a language model and compute per-token probability and per-token ranks in the predicted next token distribution. These methods use these probabilities to train a simple classifier that are used to train classifier models. The primary disadvantage of these methods is that they require the text generation model to compute the token probabilities. In the real-world, access to the model that is used to generate the text may not be possible.

The creators of GPT model Solaiman et al. (2019) develop a simple model that uses tf-idf features on unigrams and bigrams fed to a logistic regression model for identifying text generated from GPT-2 models with an accuracy of 88%.

Uchendu et al. (2020) train simple models on psychological features with simple neural network architectures to determine if an article is written by a human or a language model.

Zellers et al. (2019) develop a model on top of GROVER model that can identify fake news articles written by GPT-2. Solaiman et al. (2019) fine-tune RoBERTa model for the task to identify the texts generated by the largest variant of the GPT-2 model with an accuracy of 95%.

Fagni et al. (2021) show that a fine-tuned RoBERTa model can spot machine generated tweets from human tweets with over 90% accuracy.

Gehrmann et al. (2019) develop a tool GLTR [2] that uses the per-token probability and per-token rank in the predicted next token distribution and the entropy of the predicted next token to determine if the text is human written or machine-generated.

Mitchell et al. (2023) introduced DetectGPT, a zero-shot detection method that leverages the curvature of a language model's probability function to identify machine-generated text without requiring any training data. This approach demonstrated effectiveness across various LLMs and datasets.

Yang et al. (2023) introduced DNA-GPT, a training-free detection technique that utilizes divergent n-gram analysis. By comparing regenerated text segments with original ones, this method effectively identified discrepancies indicative of machine generation, offering a promising direction for explainable detection.

Wang et al. (2023) developed M4, a benchmark dataset with texts generated from various generators, domains, and languages. Their empirical study revealed that existing detectors often misclassify machine-generated text as human-written, particularly when encountering unseen generators or domains, indicating significant room for improvement.

Bao et al. (2023) presented Fast-DetectGPT, an efficient zero-shot detection method that reduces computational costs associated with previous approaches like DetectGPT. By introducing conditional probability curvature, this method offers a scalable solution for real-time detection applications.

Li et al. (2024) presented MAGE, a comprehensive testbed evaluating detection methods across diverse domains and LLMs. They fine-tune a Longformer Beltagy et al. (2020) that can detect machine generated content with 86.61% on unseen models.

Dugan et al. (2024) propose RAID, a benchmark designed to evaluate the robustness of machine-generated text detectors against adversarial attacks and unseen models. Their study demonstrated that many detectors can easily be circumvented, setting a new standard for evaluating and improving detection methodologies.

3 System Overview

We formulate the problem of identifying the given essay in English as human-written vs. machine-generated as a binary text classification problem. We identify a comprehensive set of linguistically motivated and statistical features for text analysis. We transform the essay into this feature space and use the feature matrix to train classical machine learning algorithms.

We use a set of style, language complexity, bias and subjectivity, and emotion-based features of the text to train machine learning models on these features. These features capture the style, syntax, fluency and other psycholinguistic characteristics of the text. The features have been used to identify fake news (Horne et al., 2019). The code to extract these features has been packaged into a Python package that is easy and fast. [3]

[2] http://gltr.io/

[3] https://github.com/BenjaminDHorne/NELAFeatures

These features fall broadly into these categories:

1. **Style**: These include the fraction of quote characters, exclamation, number of words that are all capitalized, number of stop words, and counts of various parts of speech tags and counts of special characters. 50 features are identified in this category.

2. **Complexity**: Type token Ratio, average word length, word count, Flesch Kincaid Grade level, smog index, Coleman Liau Index and Lix scores. There are 7 features in this category.

3. **Bias**: Fraction of bias words, assertatives, factives, hedges, report verbs, positive opinion words, negative opinion words. There are 8 features in this category.

4. **Affect**: These include a fraction of positive opinion words, neutral opinion words, negative opinion words, Valence Arousal Dominance (VAD) scores of positive, negative, and neutral words, word level and sentence level sentiment scores. There are 9 features in this category.

5. **Moral**: These features include counts of words that indicate Harm Virtue and Vice, Fairness Virtue and Vice, Authority Virtue and Vice, Purity Virtue and Vice, and General Morality. There are 11 features in this category.

We deliberately avoided methods involving finetuning transformer-based models because we wanted to develop a lightweight, fast, and scalable model for detecting machine-generated texts.

4 Experimental Setup

The task organizers have shared a dataset that can be used to train the various machine-learning models. The dataset contained a predefined split to be used for training and evaluation.

The organizers shared the unlabeled test set for making predictions with the trained models. As the test set labels are not publicly released, we do not know the exact number of essays that were machine-generated vs written by Human in the test set.

Table 1 mentions the number of samples present in each of the splits provided by the organizers.

	Human	**Machine**	**Total**
Train	629	1467	2096
Dev	1235	391	1626
Test	Unknown	Unknown	1130

Table 1: Samples in various splits of the dataset

We used the same split as provided by the organizers. We trained multiple ML algorithms after transforming the text through feature extraction. We used the NELA-features python package to extract features from the text.

We experimented with 4 different machine learning algorithms: Logistic Regression, Random Forest, Randomized Decision Trees(Extra Trees), and XGBoost. We used scikit-learn for training our models. We have used the default hyperparameters provided by the scikit-learn while training our models.

The official metric for this task is the macro F1 score.

5 Results

The official *test* set results scored on CodaLab have been presented below in Table 3.

Features	**Model**	**F1 (macro)**
All	Logistic Regression	*0.9949*
All	Random Forest	0.9729
All	Extra Trees	**0.9859**
All	XGBoost	0.9355

Table 3: Results on the official test set

Since only 1 submission is considered for the final evaluation, we used the model that gave the best F1-score on the development set.

Table 2 shows the results of different models with different feature combinations on the development set.

From the results of the development set, we observe that the **complexity** features were most helpful for this task. Using just the **complexity** features alone gave very good results, with an F1 score of 0.9916 using Logistic Regression model. Style features gave an F1-score of 0.9442 using XGBoost. **Complexity** features again gave high performance of F1-score of 0.9916 with Logistic Regression model. **Moral** and **Affect** features did not perform as much as the other feature group.

We can observe that models trained with all concatenated features gave higher accuracy and F1

Model →	**LR**		**RF**		**ExtraTrees**		**XGB**	
Feature Group ↓	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Style	0.8204	0.7970	0.9699	**0.9603**	0.9668	0.9565	0.9569	0.9442
Complexity	0.9938	**0.9916**	0.9477	0.9247	0.9520	0.9315	0.9428	0.9184
Bias	0.2405	0.1939	0.6470	0.6250	0.6648	0.6486	0.7165	**0.6846**
Affect	0.2970	0.2724	0.7841	0.7533	0.7608	0.7322	0.8272	**0.7883**
Moral	0.2405	0.1939	0.5480	0.4967	0.5467	0.4944	0.5806	**0.5294**
All	0.9932	0.9908	0.9938	0.9916	**0.9969**	**0.9958**	0.9711	0.9614

Table 2: *Dev* Set Accuracy and Macro-F1 scores

metric than using any one of the feature groups.

Motivated by the high performance of 'All features' with Extra Trees model, we trained a model using all the training and development data. We used this model to make inferences on the test data.

However, after the shared task deadline has passed, the organizers have allowed for submitting the predictions of other models on the official test set for comparison. Table 3 show that Logistic Regression model with all the features has performed the best on the test set with an F1-score of 0.9949 that might have placed us in 3rd position.

The official test set results place us in the 6th position of the leaderboard with an accuracy of 0.9876 and a macro F1 score of 0.9859.

6 Discussion

From the trained models, we found the most important features that are useful for discriminating between machine-generated essays and human-written essays. Table 4 lists the top discriminatory features. We visualize the distribution of the feature values in the human-written and ai-generated classes as histograms to understand them in greater detail. Here are some of the observations:

- Human essays use more stop words than machine-generated essays.
- Human essays have less average word length compared to the essays generated by AI.
- Machine-generated essays have lower readability scores like the Coleman Liau Index, Lix Readability Index, Smog Index, and the Flesch Kincaid Grade level. This is because humans tend to write shorter sentences and use fewer words per sentence.
- Compared to machine-generated essays, the usage of coordinating conjunctions is slightly more in human-written essays.

Rank	**Feature**
1	Fraction of stop words
2	Average Word Length
3	Fraction of punctuations
4	Coleman Liau Index
5	Lix Readability Index
6	Fraction of Existential there
7	Smog Index
8	Fraction of Coordinating Conjunctions
9	Flesch Kincaid Grade Level
10	Type Token Ratio

Table 4: Top 10 most important features

- Human essays have more occurrences of existential 'there' usage than AI-generated essays.

Section A, shows the histogram of the top 10 features among both the classes. We can observe the differences in the distribution of the top 10 features.

7 Conclusion

We conclude that the NELA features can be used to identify machine-generated text with high accuracy, as shown through the evaluation of the test set.

We feel that our model is robust to adversarial inputs through perturbations. As a future work, we plan to evaluate our model by adversely perturbing the input. We plan to explore the robustness of our methods to domain shifts.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020.

Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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.

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. *arXiv preprint arXiv:2405.07940*.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Benjamin D Horne, Jeppe Nørregaard, and Sibel Adali. 2019. Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–23.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. Mage: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, et al. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

A Appendix

Figures 1 to 10 show the histograms of the top features across the two classes of the essays.

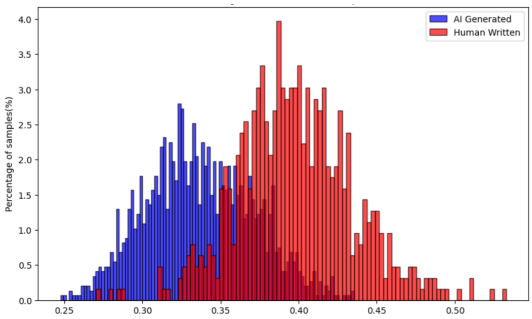

Figure 1: Histogram of the fraction of stopwords

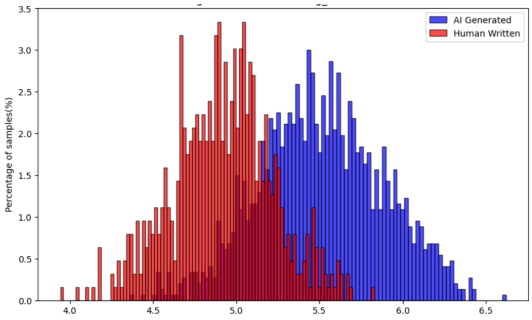

Figure 2: Histogram of Average Word Length

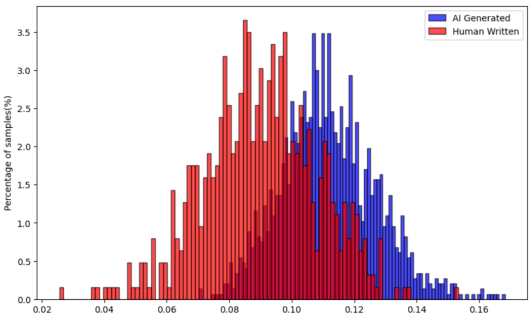

Figure 3: Histogram of the fraction of all punctuation words

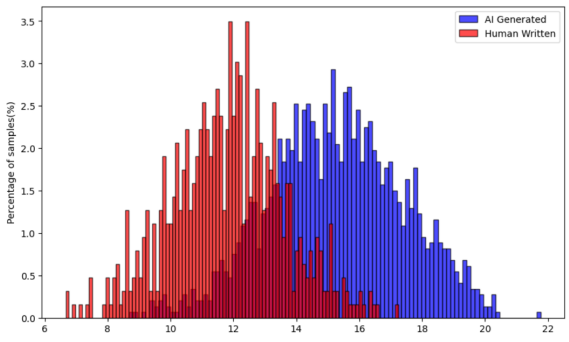

Figure 4: Histogram of Coleman Liau Index

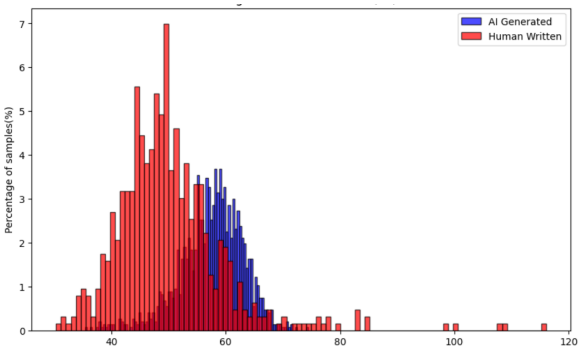

Figure 5: Histogram of Lix Readability Index

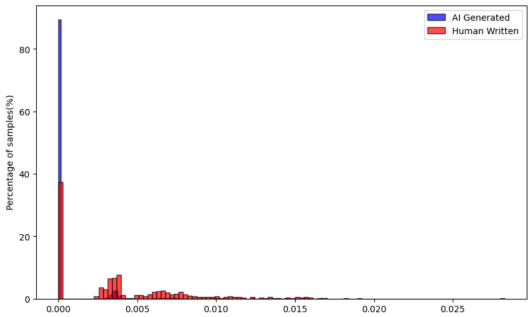

Figure 6: Histogram of Fraction of Existential 'there'

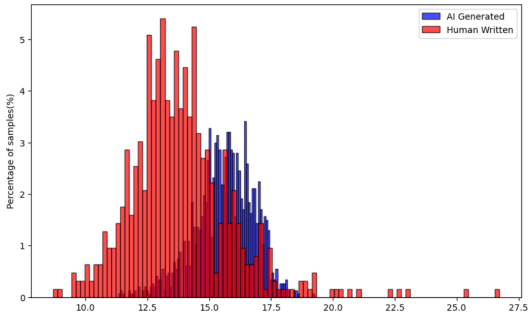

Figure 7: Histogram of Smog Index

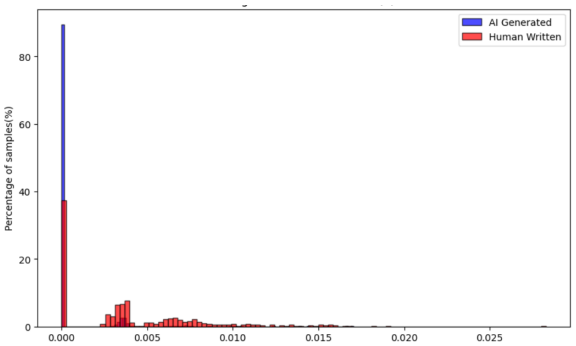

Figure 8: Histogram of Fraction of Coordinating Conjunctions

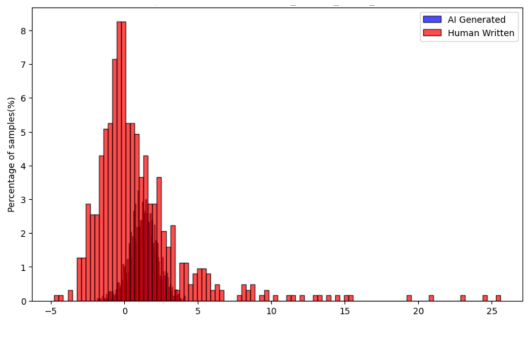

Figure 9: Histogram of Flesch Kincaid Grade Level

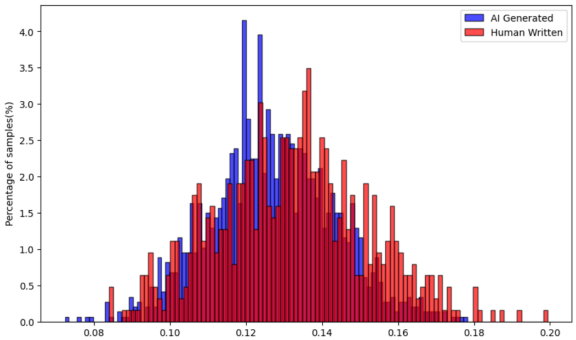

Figure 10: Histogram of Type Token Ratio