Identifying AI-Generated Text

Sai Rohini Godavarthi
Montclair State University
Montclair, New Jersey
godavarthis1@montclair.edu

Monira Enam Heya
Montclair State University
Montclair, New Jersey
heyam1@montclair.edu

Abstract

Identifying Artificial Intelligence(AI) generated text important in situations where authenticity is of utmost importance. This work utilize a dataset consisting of essays classified as either 'Human' or 'Generated', with a specific emphasis on several subset from each category. After carefully removing irrelevant data and ensuring consistency, this work utilizes statistical Machine Learning(ML) methods to differentiate between language written by humans and text generated by Large Language Models(LLM). The results of our tests provide valuable insights, on the performance of these models. We emphasize the model's effectiveness in precisely determining the source of text, while analyzing several features of the essays.

1 Introduction

The widespread adoption of AI technologies has brought about revolutionary changes in a variety of fields, including the production of written content (Gruetzemacher and Whittlestone, 2022). As the use of language created by AI becomes more widespread, the necessity of distinguishing it from content written by humans becomes more essential.

Chatbots are doing numerous tasks in education, law, advertising, scientific/creative writing, entertainment, and many other fields. LLMs also provide cybercriminals with more extensive and dangerous capabilities (Drolet, 2023). Detecting the abuses of academic dishonesty, fake news/reviews, spam/phishing, etc., is more complex (Pu et al., 2023). All educational institutes consider ChatGPT as a serious obstacle, because all the existing plagiarism-checking software are not trained to identify AI-generated content. Sometimes it is really difficult to tell the difference between AI-generated writing and human-written material (Crothers et al., 2023). In order to address this difficulty, the purpose of this research is to find a reliable model for identifying text that has been generated by AI. This will improve our capacity to evaluate the authenticity of textual information.

The study that we conduct makes use of a wide-ranging dataset (Kłeczek, 2023) that is derived from a number of different models, such as Cohere Command, Google Palm, and GPT4, which was included in the Radek contribution. These improvements increase our understanding of language produced by AI through introducing new patterns and qualities to the dataset and making it richer. In addition, the dataset comprises fresh prompts, such as source texts derived from original essays. Important contextual signals from these source texts can be utilized to interpret the created content.

Considering these developments in dataset construction, our analysis focuses on various subset of articles from both AI-generated and human sources. We make an effort to establish a solid basis for our study by performing thorough preprocessing in order to get rid of noise and ensure standardization. We intend to find a model that is capable of efficiently distinguishing between language generated by humans and text generated by AI by utilizing statistical machine learning techniques. Our work not only makes a contribution to the continuing discussion on the authenticity of textual information, but it also provides practical consequences for fields that are dependent on correct text analysis.

The following sections of this paper will provide a detailed explanation of our approach, an overview of the results of our experiments, and a discussion of the implications of our research in the area of AI-generated text identification.

2 Related Works

To create solutions that can differentiate writing samples of AI-generated text from human-written texts, a variety of modeling ways have been explored. These range from fundamental statistical techniques to state-of-the-art Transformer-based ar-

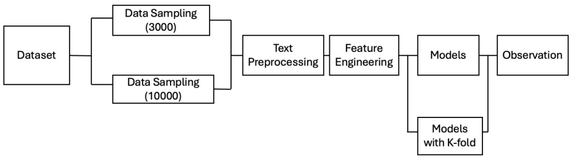

Figure 1: Proposed Workflow

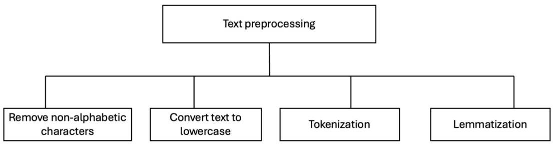

Figure 2: Text Preprocessing

chitectures. (Gehrmann et al., 2019) suggested simple statistical techniques to recognize text created by a model that might be used in a visual interface to help with the detection process. The authors made a prediction that AI systems generate text based on a small subset of highly confident linguistic patterns. In order to differentiate texts created by AI from those written by humans, (Wu et al., 2023) and (Yang et al., 2023) investigated entropy, n-gram frequency, and perplexity. To increase the accuracy and dependability of AI-generated text identification methods, more sophisticated deep learning frameworks, including Transformer-based models, have been investigated.

Several Natural Language Processing (NLP) methodologies are utilized in the feature-based identification strategy to extract relevant elements from the text. This strategy is inspired by the finding that generated text produced by Natural Language Generation (NLG) models exhibits unique artifacts (Crothers et al., 2023). Frequencies, linguistics, fluency, and fact checking are a few of the most significant features that have been suggested. The authors of (Tang et al., 2023) suggests statistical disparities, linguistic patterns, and fact verification can be used as features for detection. They also suggests traditional classification algorithms like Support Vector Machine (SVM) and Naive Bayes, as well as deep learning approaches, can be used for LM-generated text detection.

3 Methodology

Figure 1 shows the proposed workflow of this work.

3.1 Data collection and pre-processing

We acquired our dataset from the Kaggle competition “LLM – Detect AI-Generated Text.” This dataset, version 2 of DAIGT (Detect AI-Generated Text), consists of contributors from multiple individuals, comprising both human and AI-generated essay responses. The earlier version of the dataset showed an imbalance with more human-generated text. Our version 2 of DAIGT comprises 44,868 samples, with 27371 human-generated and 17.496 AI-generated essays.

3.1.1 Sampling

We employed various sampling techniques to better understand our dataset:

- Initially, we sampled 113 essays, then 3000 essays of each type.
- Next, we utilized k-fold cross-validation with 3000 samples of each type.
- Lastly, we sampled 10,000 essays of each human and AI-generated text, labeling each as human or AI-generated.

3.1.2 Text Pre-processing

During Sampling, we performed following text preprocessing techniques:

- Removing all non-alphabetic characters
- Converting text to lowercase
- Tokenization
- Lemmatization

3.2 Feature engineering

3.2.1 Flesch Reading Ease

One of the earliest and most popular readability assessments is the Flesch Reading Ease. Two concepts underpin the score:

- The material becomes more difficult to read the longer the sentences are on average.
- The text gets more difficult to read the more syllables a word typically contains.

For the concept, consider the following two sentences: "I live in my home" and "I reside in my domicile." Because the first one has fewer syllables per word, it is easier to read. The readability increases with the Flesch Reading Ease score. A higher score, therefore, denotes an easier-to-understand content.

3.2.2 Lexical Diversity

Lexical diversity (or lexical density) estimates the linguistic complexity in a written or spoken composition from the functional words (grammatical units) and content words (lexical units, lexemes). Use cases:

- You can investigate the correlation between these features and the target.
- These scores might be used as additional features for each text. And you don't have to choose the only one.

3.2.3 Text Length

Text length refers to the number of words or characters in a piece of text. We have considered number of words for our evaluation.

3.3 K-fold cross-validation

We applied the K-fold cross-validation procedure to analyze the performance of our model. By separating the dataset into numerous folds, we ensured that each data point was used for both training and validation. This strategy assists in determining how effectively our model generalizes to unknown data. We plotted the sum of squared distances (SSE) against several values of k (number of clusters) and identified an "elbow" form in the curve. This demonstrated that increasing the number of clusters beyond a certain point did not considerably enhance the SSE. We determined the ideal value of k as 3, which achieved an acceptable balance between capturing the underlying patterns in the data and avoiding unnecessary complexity. After the K-fold cross-validation was finished, a final split was conducted using the train-test-split function. This split retained 20% of the data for testing, while the remaining 80% was used for training. This last test set enabled an unbiased evaluation of the model's performance on unseen data. This made it possible for an in-depth assessment of the model's performance on unseen data (Fushiki, 2011).

3.4 ML Models

3.4.1 Naive Bayes

Naive Bayes is a probabilistic classification technique based on Bayes' theorem. It assumes that every attribute is independent of one other, which is why it is known as "naive." Despite its foundation, Naive Bayes works well in different text classification and document categorization problems. It is especially useful when dealing with high-dimensional data, such as natural language processing work (Yang, 2018).

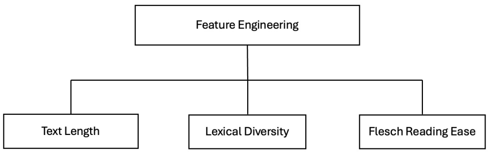

Figure 3: Feature Engineering

3.4.2 Logistic Regression

Logistic Regression is a classification system used to estimate the likelihood of a binary result (e.g., yes/no, true/false). It represents the connection between the independent factors and a binary dependent variable by fitting a logistic curve. Logistic Regression is frequently utilized because of its simplicity, interpretability, and efficacy in numerous applications (Zou et al., 2019).

3.4.3 Random Forest

Random Forest is an ensemble learning approach that mixes many decision trees to create more robust and accurate predictions. It provides a diversified collection of decision trees by injecting randomization throughout the tree-building process. Random Forest is less prone to overfitting and provides superior generalization compared to individual decision trees (Belgiu and Drăguţ, 2016).

3.4.4 DistilBERT: a distilled version of BERT

DistilBERT is a simplified version of the BERT (Bidirectional Encoder Representations from Transformers). This architecture is designed to provide a more efficient use of resources and a more streamlined approach to NLP. The DistilBERT model was proposed in the blog post Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT (Sanh, 2020), and the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter (Sanh et al., 2019). It is created using a procedure called distillation, in which the knowledge gained by the larger, more intricate BERT model is converted to a smaller, more straightforward equivalent. In order to minimize differences

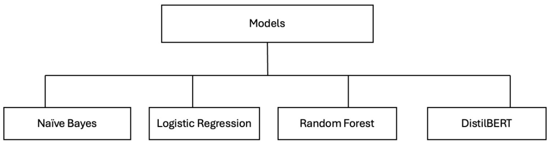

Figure 4: ML Models

between their outputs, DistilBERT is trained using a combination of original training data and predictions made by BERT on the same data. This information transfer is made possible by techniques such as knowledge distillation. DistilBERT's architecture keeps the core elements of BERT but significantly reduces its complexity: instead of BERT's 12 levels, 768 hidden units, and 12 or 24 attention heads, DistilBERT has just 6 layers, 768 hidden units, and 12 attention heads. DistilBERT further reduces its memory footprint without appreciable performance reduction by using parameter sharing and compression methods like quantization and trimming. The end product is a more lightweight, quicker, and smaller model that is very successful for a wide range of NLP applications. This makes it particularly appropriate for deployment in situations that prioritize computing efficiency or in environments with limited resources.

3.5 Performance Evaluation Metrics

3.5.1 Accuracy

The percentage of accurate predictions is expressed using the metric of accuracy in classification problems. It can be calculated by dividing the number of correct predictions by the total number of predictions.

$$\text{Accuracy} = \frac{Number\,of\,Correct\,Predictions}{Total\,Number\,Of\,Predictions}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Where, TP = True Positive TN = True Negative FP = False Positive FN = False Negative. Using this formulation, a general measure of model performance across the entire dataset is provided by the accuracy metric in its default form. The minority class must be correctly predicted since overall accuracy might be deceptive when the distribution of classes is unbalanced.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 1: Sample Confusion Matrix

3.5.2 Precision

The ratio of true positives to all expected positives is known as precision.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3.5.3 Recall

Recall is the proportion of real positives to all the positives in the ground truth.

$$\text{Recall} = \frac{TP}{TP + FN}$$

3.5.4 F1-Score

The F1 Score represents the Harmonic Mean between Recall and Precision.

$$\text{F1-Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

3.5.5 Confusion Matrix

A confusion matrix provides a comprehensive view of the performance of a classification model by displaying the counts of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions made by the model.

True Positive (TP): The number of instances correctly predicted as positive by the model.

False Positive (FP): The number of instances incorrectly predicted as positive by the model.

True Negative (TN): The number of instances correctly predicted as negative by the model.

False Negative (FN): The number of instances incorrectly predicted as negative by the model.

Table 1 provides a sample of the confusion matrix.

4 Results

4.1 Feature Engineering

Before training our models, we performed following feature engineering methods:

Sample Size	AI-generated Text	Human-generated Text
113	Less than 500: 0	Less than 500: 0
	Less than 1000: 8	Less than 1000: 5
	More than 1000: 105	More than 1000: 108
3000	Less than 500: 3	Less than 500: 0
	Less than 1000: 144	Less than 1000: 169
	More than 1000: 3856	More than 1000: 2830
10000	Less than 500: 10	Less than 500: 0
	Less than 1000: 457	Less than 1000: 572
	More than 1000: 9543	More than 1000: 9423

Table 2: Text Length

Sample Size	Human Lexical Diversity	AI Lexical Diversity
113	Min: 0.20	Min: 0.25
	Max: 0.58	Max: 0.68
	Mode: 0.20	Mode: 0.25
3000	Min: 0.10	Min: 0.21
	Max: 0.71	Max: 1.0
	Mode: 0.5	Mode: 0.5
10000	Min: 0.05	Min: 0.1
	Max: 0.71	Max: 1.0
	Mode: 0.5	Mode: 0.5

Table 3: Lexical Diversity

4.1.1 Text Length

We analyzed the number of words or characters in each piece of text: Table 2

4.1.2 Lexical Diversity

We measured the variety of unique words used in each piece of text: Table 3

4.1.3 Flesch Reading Ease

We assessed how easy or difficult the text is to read: Table 4

4.1.4 Observations

Text Length: AI-generated essays tend to be longer on average compared to human-generated essays, particularly in the 10,000-sample dataset where the majority of AI-generated essays exceed 10,000 words.

Sample Size	Flesch Reading Ease for AI Essays	Flesch Reading Ease for Human Essays
113	Moderate: 53	Moderate: 53
	Difficult: 43	Easy: 51
3000	Moderate: 1312	Moderate: 1443
	Difficult: 1048	Easy: 1349
10000	Moderate: 4394	Moderate: 4920
	Difficult: 3460	Easy: 4433

Table 4: Flesch Reading Ease

Sample Size	Metric	AI	Human
113	Precision	1.00	0.83
	Recall	0.77	1.00
	F1 Score	0.87	0.91
	Accuracy	0.89	-
3000	Precision	0.94	0.88
	Recall	0.87	0.94
	F1 Score	0.90	0.91
	Accuracy	0.91	-
10000 (k-fold)	Mean CV Accuracy	0.9216	-
	Train Set Accuracy	0.9226	-
	Test Set Accuracy	0.9213	-

Table 5: Naive Bayes

Sample Size	Metric	AI	Human
113	Precision	0.89	0.81
	Recall	0.77	0.92
	F1 Score	0.83	0.86
	Accuracy	0.85	-
3000	Precision	0.98	0.96
	Recall	0.96	0.98
	F1 Score	0.97	0.97
	Accuracy	0.97	-
10000 (k-fold)	Mean CV Accuracy	0.9846	-
	Train Set Accuracy	0.9876	-
	Test Set Accuracy	0.9878	-

Table 6: Logistic Regression

Lexical Diversity: Human-generated essays exhibit higher lexical diversity compared to AI-generated essays across all sample sizes.

Flesch Reading Ease: AI-generated essays generally have lower Flesch Reading Ease scores, indicating higher complexity, while human-generated essays tend to be easier to read.

4.2 Model Performance evaluation

4.2.1 Naive Bayes

Table 5 shows the results of the model performance assessment.

4.2.2 Logistic Regression

The evaluation of model performance is shown in the Table 6.

4.2.3 Random Forest

Table 7 displays the evaluation of the model's performance.

Sample Size	Metric	AI	Human
113	Precision	0.91	0.96
	Recall	0.95	0.92
	F1 Score	0.93	0.94
	Accuracy	0.93	-
3000	Precision	0.99	0.96
	Recall	0.86	0.99
	F1 Score	0.97	0.97
	Accuracy	0.97	-
10000 (k-fold)	Mean CV Accuracy	0.9699	-
	Train Set Accuracy	0.9840	-
	Test Set Accuracy	0.9720	-

Table 7: Random Forest

Sample Size	Metric	F1 Score
113	Confusion Matrix	0.76
3000	Confusion Matrix	0.84

Table 8: Distil BERT

4.2.4 Distil BERT

Model Performance evaluation is presented in Table 8

4.2.5 Observations

Performance Improvement with Sample Size: Across all models, there is a noticeable trend of performance improvement as the sample size increases. This is evident in the increase in accuracy, precision, recall, and F1 scores as the number of samples grows from 113 to 3000 and further to 10000 with k-fold cross-validation.

Robustness of Logistic Regression: Logistic Regression consistently shows robust performance across different sample sizes, with high precision, recall, and accuracy. This indicates its suitability for classifying AI-generated and human-written text.

Potential Overfitting with Naïve Bayes: While Naive Bayes achieves high accuracy and F1 Scores, especially with larger sample sizes, there is a warning of potential overfitting, indicated by its higher accuracy on the training set compared to cross-validation.

Random Forest Performance: Random Forest shows competitive performance, particularly in F1 scores, across different sample sizes. However, it also shows signs of overfitting, especially noticeable with the larger sample size of 10000.

Distil BERT for complex patterns: Distil BERT demonstrates its capability in capturing complex patterns within text, as evidenced by its F1 scores and confusion matrix analysis. However, its resource-intensive nature limits its scalability for larger datasets.

5 Conclusion

In this study, we investigated the efficacy of various machine-learning models in identifying AI-generated text. Our findings demonstrate that logistic regression consistently performs well across different sample sizes, exhibiting high accuracy and balanced precision and recall. Naive Bayes and Random Forest also show competitive performance but may suffer from overfitting, particularly with larger datasets.

Additionally, the use of Distil BERT shows promising result in capturing nuanced patterns within text, but its computational demands restrict its applicability to smaller datasets.

Limitations

While our study provides valuable insights into the performance of different models in identifying AI-generated text, it is not without limitations. These include:

Dataset Bias: Our analysis relies on the LLM dataset, which may not fully represent the diversity of AI-generated text. Future studies should consider using more diverse datasets.

Resource Constraints: The computational requirements of models like Distil BERT limit their scalability, particularly for large datasets. This restricts their practical utility in real-time applications.

Model Evaluation Metrics: While we have evaluated our models using standard metrics like accuracy, precision, recall, and F1 score, these metrics may not fully capture the real-world implications of misclassification, particularly in sensitive applications like content moderation.

Generalization: Our study focuses on a specific domain of text classification. Generalizing the findings to other domains may require further investigation.

References

Mariana Belgiu and Lucian Drăguţ. 2016. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

Michelle Drolet. 2023. Council post: 10 ways cybercriminals can abuse large language models.

Tadayoshi Fushiki. 2011. Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21:137–146.

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

Ross Gruetzemacher and Jess Whittlestone. 2022. The transformative potential of artificial intelligence. *Futures*, 135:102884.

Darek Kłeczek. 2023. Daigt V2 Train Dataset. Retrieved May 3, 2024, from `https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-dataset`.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1613–1630. IEEE.

Victor Sanh. 2020. Smaller, faster, cheaper, lighter: Introducing dilbert, a distilled version of bert.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. Llmdet: A third party large language models generated text detection tool.

Feng-Jen Yang. 2018. An implementation of naive bayes classifier. In *2018 International conference on computational science and computational intelligence (CSCI)*, pages 301–306. IEEE.

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.

Xiaonan Zou, Yong Hu, Zhewen Tian, and Kaiyuan Shen. 2019. Logistic regression model optimization and case analysis. In *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*, pages 135–139. IEEE.