Distinguishing Human-Written and ChatGPT-Generated Text Using Machine Learning

Hosam Alamleh
Computer Science
University of North Carolina Wilmington
Wilmington, North Carolina, USA
hosam.amleh@gmail.com

Ali Abdullah S. AlQahtani
Computer Systems Technology
North Carolina A&T State University
Greensboro, North Carolina, USA
alqahtani.aasa@gmail.com

AbdElRahman ElSaid
Computer Science
University of North Carolina Wilmington
Wilmington, North Carolina, USA
aelsaied@gmail.com

***Abstract*—The use of sophisticated Artificial Intelligence (AI) language models, including ChatGPT, has led to growing concerns regarding the ability to distinguish between human-written and AI-generated text in academic and scholarly settings. This study seeks to evaluate the effectiveness of machine learning algorithms in differentiating between human-written and AI-generated text. To accomplish this, we collected responses from Computer Science students for both essay and programming assignments. We then trained and evaluated several machine learning models, including Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), Neural Networks (NN), and Random Forests (RF), based on accuracy, computational efficiency, and confusion matrices. By comparing the performance of these models, we identified the most suitable one for the task at hand. The use of machine learning algorithms for detecting text generated by AI has significant potential for applications in content moderation, plagiarism detection, and quality control for text generation systems, thereby contributing to the preservation of academic integrity in the face of rapidly advancing AI-driven content generation.**

***Index Terms*—TextOriginClassifier, ChatGPT, human-written text, AI-generated text, machine learning, academic integrity, content detection, AI, NLP, TF-IDF**

I. Introduction

The rapid proliferation of AI has led to significant advancements in natural language processing (NLP) capabilities. Language models, such as OpenAI's ChatGPT, have transformed various sectors, including communication, customer service, and content creation, by generating human-like text with remarkable proficiency. These powerful models offer immense benefits in terms of efficiency and accessibility; however, they also present challenges that warrant careful consideration.

One of the foremost concerns arising from the widespread use of AI language models is the potential erosion of academic and scholarly integrity. As AI-generated content becomes increasingly indistinguishable from human-authored works, the ability to differentiate between human-written and AI-generated text assumes critical importance. The proliferation of AI-authored content may inadvertently undermine the credibility and authenticity of intellectual works, making it essential to develop methods for detecting and distinguishing between human and AI-generated text.

Machine learning has been suggested as a potential method for detecting text generators. In this study, we evaluate the accuracy and efficiency of 11 machine learning algorithms in distinguishing between human-written text and text generated by ChatGPT. The classification problem is presented to the machine learning algorithms, and a comprehensive comparison of their performance is conducted.

Through the development and evaluation of our ML framework, this paper not only addresses the immediate concern of distinguishing between human-written and ChatGPT-generated text, but also contributes to a broader understanding of how AI-generated content can be detected and mitigated in the future. Ultimately, our findings have significant implications for maintaining the trustworthiness of digital communication and the ethical use of AI technologies in the era of rapidly advancing natural language generation models.

The rest of the paper is organized as follows: In Section II, we provide a comprehensive review of related work in the field of machine learning-based detection of text generated by language models. Section III, describes our proposed approach in detail. In Section V, we evaluate the performance of our approach and present results. Finally, in Section VI, we conclude our paper.

II. Related work

Classifying GPT output vs human output has been a growing challenge in academia and education [1]–[3]. Despite the blurry boundaries between the work authenticity and plagiarism - considering the authors of its training data, its developers, and its prompts of the users - there is pressing calls to draw a line between human work and machine work [4]. That been said, efforts presented in literature have been introducing efforts to distinguish bot generated text from human output. Narasimhan *et al.* [5] found that

out of 80 programs generated by GPT-3, 38 were of good quality, concluding that the model can partially contribute to software development. Dou *et al.* [6] investigated GPT-3 ability to generate indistinguishable text to human writing and found that, among different error types noticed in the generations, the model beats humans at using technical jargon and knowledge, however, human writers were less prone to commit other eight error types. The study also found that model's overall writing ability grows with model's size. Wilson *et al.* [7] used unsupervised NLP models to categorize student reasoning in Physics Measurement Questionnaire. The researchers found that their logistic regression model was able to achieved the same level of agreement with the human rater. Solaiman *et al.* [8] (OpenAI in partnership with academic and research institutions) fine-tuned a RoBERTa [9] base model to detect GPT-2 output. The RoBERTa based model was able to classify 1.5 billion parameter GPT-2-generated text with about 95% accuracy [1].

This scientific paper investigates the growing popularity and usage of ChatGPT, a language model developed by OpenAI, capable of generating grammatically flawless and seemingly-human text [10]. With increased usage, concerns arise about the potential for abuse, leading to the research question: can a machine learning model accurately distinguish between original human-generated text and ChatGPT-generated text? To address this, the authors conduct two experiments, fine-tune a Transformer-based model, and utilize an explainable AI framework (SHAP) to understand the model's reasoning. The study's findings reveal that while disambiguation between human-generated and ChatGPT-generated text is challenging, especially when using rephrased text, the proposed approach achieves an accuracy of 79%. The authors identify specific patterns in ChatGPT-generated text, such as politeness, lack of detail, fancy vocabulary, impersonality, and reduced expressiveness. The paper emphasizes the importance of using explainability in AI models and offers valuable insight into the limitations and challenges of distinguishing between human-generated and ChatGPT-generated text.

The paper introduces a classification model for detecting text generated by ChatGPT [11], emphasizing the challenge of distinguishing between machine-generated and human-written text. The authors employ a dataset of human-written and ChatGPT-generated essays to train and evaluate the model, which is based on XGBoost. The model's performance is assessed using two feature extraction schemas, achieving a detection accuracy of 96%. These results demonstrate the feasibility of machine learning for detecting ChatGPT-generated text and provide valuable insights for researchers and policymakers concerned about malicious ChatGPT usage.

In [12], the authors report the outcomes of two experiments examining people's behavior towards text generated by GPT-2, a state-of-the-art Natural Language Generation algorithm, using poems as test material. The experiments aimed to evaluate whether participants could differentiate between human-written and GPT-2-generated poems. The first experiment provided participants with two poems, one human-written and one GPT-2-generated, either randomly selected or chosen as the best by a human. Results indicated that participants could not consistently detect algorithm-generated poems in the latter case, but succeeded in the former. The second experiment assessed participants' preference for human-written or algorithm-generated poems, revealing a slight aversion to the algorithm-generated poetry, regardless of awareness of its origin. The authors discuss the implications of these findings for Natural Language Generation algorithms and propose methods for future human-agent experimental research.

In [13], three experiments were conducted to study the impact of AI-generated text on online misinformation and its effect on foreign policy opinions. The first experiment examined people's perception of AI-generated text compared to an original story. The second explored the role of partisanship in perceived credibility of AI-generated news, and the third investigated how perceived credibility varied based on AI model size. The findings showed that people could not consistently distinguish between AI-generated and human-created text, and partisanship influenced perceived credibility. However, exposure to the text did not significantly change individuals' policy views. The authors conclude that these findings have important implications for understanding AI's role in online misinformation campaigns, as AI-generated text can be used to spread misinformation in a way that is difficult for people to identify, and partisanship can influence the perceived credibility of the presented information.

III. Methodology

This paper focuses on evaluating the performance of machine learning algorithms in distinguishing between human-written text and text generated by ChatGPT using various classifiers. In this section we will present the dataset used in this study, followed by a discussion of the feature extraction process.

A. Data Collocation Phase

A dataset comprising 500 data points was gathered by collecting answers to 250 computer science problems assigned in classes and quizzes from students. To generate this dataset, a response was selected from a random student for each question. The same questions were then asked

[1]The model is deployed publicly at: https://openai-openai-detector.hf.space

to ChatGPT 3.0, and the answers were recorded. Based on the source of the response (either student or GPT), the dataset was labeled accordingly. The resulting labeled dataset includes the list of assignment and quiz questions, along with the corresponding answers from students and ChatGPT.

Half of the assignment questions were essay prompts that required students to compose an essay-style answer. GPT was also tasked with answering these questions, with a constraint to limit the length of the responses to be similar to that of the student answers. This was done to prevent the classifier from relying solely on the length of the response to distinguish between student and GPT-generated answers. The other half of the assignment consisted of programming prompts that required students and GPT to write programs in both C and Python. The difficulty of these programming prompts ranged from simple programs to more complex ones that involved multiple functions and classes. Dataset is made publicly available here [14].

B. Feature Extraction

Feature extraction is an essential step in the machine learning pipeline, as it converts the raw text data into a numerical representation that can be utilized by various machine learning models. In this study, we employ the Term Frequency-Inverse Document Frequency (TF-IDF) technique for feature extraction. In this section, we will delve deeper into the TF-IDF technique and explain the process of applying it to our data-set.

1) TF-IDF

TF-IDF is a widely used method in natural language processing and information retrieval to quantify the importance of words within a set of documents. The TF-IDF technique consists of two components: term frequency (TF) and inverse document frequency (IDF).

- Term Frequency (TF): This component measures the frequency of a term (word) within a specific document (text sample). The idea behind TF is that if a word appears frequently in a document, it is likely to be more important and relevant to the topic of the document.
- Inverse Document Frequency (IDF): This component measures the importance of a term across the entire set of documents in the dataset. The rationale behind IDF is that words that appear in many documents are likely to be less informative, as they do not help differentiate between different documents.

By combining both components, the TF-IDF value of a term within a document captures its importance within that document relative to the entire dataset. High TF-IDF values indicate that a term is essential and unique to a specific document, whereas low values suggest that the term is either common across all documents or not relevant to the document's topic.

2) TfidfVectorizer

In this study, we use the TfidfVectorizer class from the scikit-learn library to perform the feature extraction. This class automatically calculates the TF-IDF values for each term in the dataset and converts the text data into a matrix of TF-IDF features. The process involves the following steps:

- Tokenization: The TfidfVectorizer first tokenizes the text data, splitting it into individual words or terms. This process can include removing stop words (common words such as "the," "and," "is," etc.) and applying stemming or lemmatization techniques to reduce words to their root forms.
- Calculating term frequency (TF): For each term in the document, the vectorizer computes its term frequency, which is typically represented as the raw count of the term in the document or a normalized value (e.g., dividing the raw count by the total number of words in the document).
- Calculating inverse document frequency (IDF): The vectorizer computes the inverse document frequency for each term in the dataset by taking the logarithm of the total number of documents divided by the number of documents containing the term. This value is then added to a constant (usually 1) to avoid division by zero and to ensure non-zero values.
- Computing TF-IDF values: The vectorizer calculates the TF-IDF values by multiplying the term frequency and inverse document frequency values for each term in each document.
- Creating the feature matrix: The vectorizer creates a sparse matrix of the TF-IDF values, with each row representing a document and each column representing a term. This matrix serves as the input for the machine learning models in the subsequent steps of the methodology.

By applying the TfidfVectorizer to our dataset, we obtain a matrix of TF-IDF features that capture the importance and relevance of each term within the text samples. This numerical representation allows us to leverage various machine learning models to discern between human-written and ChatGPT-generated text by exploiting subtle linguistic patterns and contextual discrepancies that are unique to each source.

IV. Experiment and Results

In this section, we evaluate and analyze the performance of various machine learning algorithms. The experiments were conducted on a MacBook Pro 2021 with an Apple M1 Max Chip and 32 GB of memory, except for the

BERT model. For the BERT model, the training and testing were performed on a higher performance computing device specifically designed for machine learning tasks.

A. Model Training

After feature extraction, the next step is to train various machine learning models on the dataset to discern between human-written and ChatGPT-generated text. The training process involves fitting the models to the training data, which consists of the matrix of TF-IDF features and the corresponding labels. In this study, we consider a diverse set of models to compare their performance and identify the most suitable one for the task at hand. This study evaluated a total of 11 machine learning models, consisting of nine classical models and two deep learning models. The models assessed are as follows:

1) Logistic Regression (LR)
2) Decision Tree Classifier (DT)
3) K-Nearest Neighbors Classifier (KNN)
4) Multinomial Naive Bayes (MNB)
5) Random Forest Classifier (RF)
6) Gradient Boosting Classifier(GB)
7) Support Vector Machine (SVC)
8) Fast Large Margin (FLM)
9) Generalized Linear Model (GLM)
10) Feedforward Neural Network (FNN)
11) Bidirectional Encoder Representations from Transformers (BERT)

For each model, we instantiate it using the default parameters from the scikit-learn library and fit it to the training data. The fitting process adjusts the model's parameters to minimize the differences between the predicted labels and the actual labels in the training data. This process effectively "teaches" the model to recognize patterns in the TF-IDF features that distinguish between human-written and ChatGPT-generated text.

B. Model Evaluation

After training the models, we evaluate their performance on the testing data, which is a separate portion of the dataset not used during the training process. This evaluation enables us to assess how well the models generalize to unseen data, providing a reliable estimate of their real-world performance.

We perform the following tasks to evaluate the performance of each trained model:

1) Making predictions: We use each trained model to predict the labels for the testing data, which consists of the matrix of TF-IDF features for the test samples. These predictions represent the model's best guess at whether the text is human-written or generated by ChatGPT, based on the patterns it learned during the training process.
2) Calculating accuracy: We compute the accuracy of each model by comparing its predictions to the actual labels in the testing data. The accuracy is defined as the proportion of correct predictions over the total number of predictions, expressed as a percentage. Higher accuracy values indicate better performance and a greater ability to discern between human-written and ChatGPT-generated text.
3) Measuring training and prediction time: We measure the time taken to train each model and make predictions on the testing data. This metric provides an estimate of the computational efficiency of each model, which is an important consideration when selecting a model for deployment in real-world applications.

By evaluating the models based on their accuracy; as can be seen from Table I, and computational efficiency; as can be seen from Table II, we can compare their performance and select the most suitable model for discerning between human-written and ChatGPT-generated text. This model can then be fine-tuned, if necessary, and deployed in a variety of applications, such as content moderation, plagiarism detection, or quality control for text generation systems.

As can be seen from Table I, the essay prompt experiment, the RF classifier achieved the highest accuracy of 93%, while the SVM classifier achieved an accuracy of 91.50%. In the programming prompt experiment, the RF classifier achieved 93.50% and the SVM classifier achieved 91%. In the combined prompt experiment, the RF classifier achieved an accuracy of 92.50%, while the SVM classifier achieved an accuracy of 91%.

TABLE I: Accuracy

Model	Essay Prompts	Programming Prompts	Combination
RF	93%	93.50%	92.50%
SVM	91.50%	91%	91%
LR	92.50%	85.50%	88%
DT	87%	87.50%	90.25%
KNN	65.50%	70%	61.75%
NB	89.50%	82.50%	88%
GB	92%	91%	88.75%
FLM	91%	89.50%	91.25%
GLM	91.50%	88.50%	91%
FNN	90.50%	86.50%	91.75 %
BERT	73.46%	62.00%	69.69%

V. Discussion

As can be seen from results, we can observe that the Random Forest (RF) model performs the best overall, achieving an accuracy of 93%, 93.5%, and 92.5% for essay prompts, programming prompts, and the combination, respectively. The Support Vector Machine (SVM) model is the second-best performer, with an accuracy of 91.5% for essay prompts and 91% for both programming prompts and the combination. The K-Nearest Neighbors (KNN) model has the lowest

TABLE II: Time Consumption

Model	Essay Prompts	Programming Prompts	Combination
RF	0.05 seconds	0.06 seconds	0.10 seconds
SVM	0.00 seconds	0.00 seconds	0.00 seconds
LR	0.00 seconds	0.00 seconds	0.01 seconds
DT	0.01 seconds	0.00 seconds	0.02 seconds
KNN	0.00 seconds	0.00 seconds	0.01 seconds
NB	0.00 seconds	0.00 seconds	0.00 seconds
GB	0.16 seconds	0.07 seconds	0.25 seconds
FLM	0.00 seconds	0.00 seconds	0.00 seconds
GLM	0.00 seconds	0.00 seconds	0.00 seconds
FNN	0.62 seconds	0.37 seconds	1.23 seconds
BERT	> 1000 seconds	> 1000 seconds	> 1000 seconds

accuracy among all models, achieving only 65.5%, 70%, and 61.75% for essay prompts, programming prompts, and the combination, respectively.

As evident from the results, machine learning algorithms were generally more successful in identifying text generated by GPT for essay prompts than for programming prompts. This is likely due to the fact that GPT's writing style is more distinct and recognizable in essays compared to programming code, which may be more complex and varied.

The results show that classical machine learning algorithms have better performance than deep learning algorithms. There are several possible reasons for this. For instance, classical machine learning algorithms may be more suitable for small datasets, whereas deep learning algorithms require larger datasets to perform well. In our case, the dataset was small, consisting of only 500 points, which may explain why the deep learning methods performed worse than classical machine learning algorithms.

The training time for the models was affected by the size of the dataset, with smaller datasets resulting in faster training times. Classical models generally had very fast training times, with the exception of GLM which took 0.16 seconds due to its use of iterative algorithms, such as maximum likelihood estimation. The two deep learning methods took longer to train, with BERT taking the longest time. This can be attributed to its use of a complex attention mechanism that requires processing each input token in relation to every other token in the input sequence, which adds to the computational overhead. Additionally, BERT often requires fine-tuning on specific tasks, which can further increase training time and complexity.

VI. Conclusion

In this study, we assessed the efficacy of machine learning algorithms in distinguishing between human-written and ChatGPT-generated text. Through the use of cutting-edge natural language processing techniques and the TF-IDF feature extraction method. Our comprehensive analysis involved training and evaluating a diverse array of machine learning models, which yielded valuable insights into their respective performances. By comparing accuracy and computational efficiency, we were able to identify the most appropriate model for effectively distinguishing between the two text types. Implementing such algorithms is critical, as they have considerable potential for real-world applications, including content moderation, plagiarism detection, and quality control for text generation systems. Furthermore, their effectiveness in tackling the challenges posed by the proliferation of advanced AI language models, such as ChatGPT, contributes to upholding academic integrity and mitigating the impact of AI-generated text on scholarly works.

References

[1] G. G. P. Transformer, A. O. Thunström, and S. Steingrimsson, "Can gpt-3 write an academic paper on itself, with minimal human input?" 2022.

[2] I. Islam and M. N. Islam, "Opportunities and challenges of chatgpt in academia: A conceptual analysis," *Authorea Preprints*, 2023.

[3] M. Zong and B. Krishnamachari, "a survey on gpt-3," *arXiv preprint arXiv:2212.00857*, 2022.

[4] N. Dehouche, "Plagiarism in the age of massive generative pre-trained transformers (gpt-3)," *Ethics in Science and Environmental Politics*, vol. 21, pp. 17–23, 2021.

[5] A. Narasimhan, K. P. A. V. Rao *et al.*, "Cgems: A metric model for automatic code generation using gpt-3," *arXiv preprint arXiv:2108.10168*, 2021.

[6] Y. Dou, M. Forbes, R. Koncel-Kedziorski, N. A. Smith, and Y. Choi, "Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text," *arXiv preprint arXiv:2107.01294*, 2021.

[7] J. Wilson, B. Pollard, J. M. Aiken, M. D. Caballero, and H. Lewandowski, "Classification of open-ended responses to a research-based assessment using natural language processing," *Physical Review Physics Education Research*, vol. 18, no. 1, p. 010141, 2022.

[8] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps *et al.*, "Release strategies and the social impacts of language models," *arXiv preprint arXiv:1908.09203*, 2019.

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[10] S. Mitrović, D. Andreoletti, and O. Ayoub, "Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text," *arXiv preprint arXiv:2301.13852*, 2023.

[11] R. Shijaku and E. Canhasi, "Chatgpt generated text detection."

[12] N. Köbis and L. D. Mossink, "Artificial intelligence versus maya angelou: Experimental evidence that people cannot differentiate ai-generated from human-written poetry," *Computers in human behavior*, vol. 114, p. 106553, 2021.

[13] S. Kreps, R. M. McCain, and M. Brundage, "All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation," *Journal of Experimental Political Science*, vol. 9, no. 1, p. 104–117, 2022.

[14] H. Alamleh, A. A. S. AlQahtani, and A. Elsaid, ""chatgpt vs. student: A dataset for source classification of computer science answers," 2023. [Online]. Available: https://dx.doi.org/10.21227/8g44-k803