Large Language Models (LLMs) Generated Text Detection using Machine Learning Algorithms and BERT Model

Muhammad Zahraan
2021-UOK-04805
Department of Artificial Intelligence
Faculty of Computing and Engineering
University of Kotli Azad Jammu and Kashmir

***Abstract*—The rapid growth in the field of Artificial Intelligence (AI) has led to the development of Large Language Models (LLMs), such as ChatGPT and Bard, which are capable of generating human-like text across various domains. However, this capability has raised concerns about the potential misuse of LLM-generated content, particularly in academic settings where it can lead to plagiarism and compromise academic integrity. This project aims to address the challenge of detecting text generated by LLMs using machine learning techniques. Specifically, we employ Logistic Regression, Random Forest and BERT (Bidirectional Encoder Representations from Transformers) model to differentiate between human-written and LLM-generated text based on language patterns, stylistic characteristics, and semantic nuances. By training the models on diverse datasets containing essays and utilizing techniques like text vectorization, we aim to develop a robust detector capable of accurately classifying text instances. Through this research, we seek to contribute to the development of effective methods for identifying LLM-generated content.**

Keywords—Artificial Intelligence (AI), Large Language Models (LLMs), Machine Learning, Logistic Regression, Random Forest , BERT, Classification, Text Vectorization

I. INTRODUCTION

Artificial Intelligence (AI) has progressed rapidly in recent years, contributing to the creation of large language models (LLMs), such as GPT (ChatGPT) and LaMDA (BARD). Large Language Models are a category of foundation models trained on large amounts of data making them capable of understanding and generating natural language and other types of content to perform a wide range of tasks.

As of now, there is no 100% reliable method for effectively detecting AI-generated text. Many detectors perform poorly when the text is paraphrased, the language is changed, or some words are replaced. The issue is increasingly relevant because future LLMs will require more text in their training data. If the models are trained on their own generated data, the results could be problematic. It's essential to independently verify facts and evaluate the credibility of sources and citations, check dates, and evaluate the believability of the content.

Choosing the right model is crucial in machine learning projects, especially for the goal of detecting text generated by Large Language Models (LLMs). As for binary classification, there are varieties of machine learning algorithm available but we utilize Logistic Regression, Random Forest and BERT model.

A. Logistic Regression

Logistic Regression stands out as the optimal choice for several reasons. Logistic regression is part of the Regression family as it involves predicting outcomes based on quantitative relationships between variables. However, unlike linear regression, it accepts both continuous and discrete variables as input and its output is qualitative. In addition, it predicts a discrete class such as “Yes/No” or “Customer/Non-customer”. It assigns probabilities to discrete outcomes using the Sigmoid function, which converts numerical results into an expression of probability between 0 and 1.0. Probability is either 0 or 1, depending on whether the event happens or not. For binary predictions, we can divide the population into two groups with a cut-off of 0.5. Everything above 0.5 is considered to belong to group A, and everything below is considered to belong to group B.

Additionally, Logistic Regression excels in handling noisy data, a common challenge in real-world datasets, including those associated with LLM-generated text. Through regularization techniques, Logistic Regression effectively mitigates noise, enhancing the model's robustness. Furthermore, Logistic Regression's adaptability makes it suitable for the complexities of textual data. Whether the text is structured or unstructured, Logistic Regression can navigate through various features and nuances. Its flexibility in accommodating different word frequencies and contexts ensures reliable performance across diverse text samples.

B. Why we use Logistic Regression?

Logistic regression is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1.

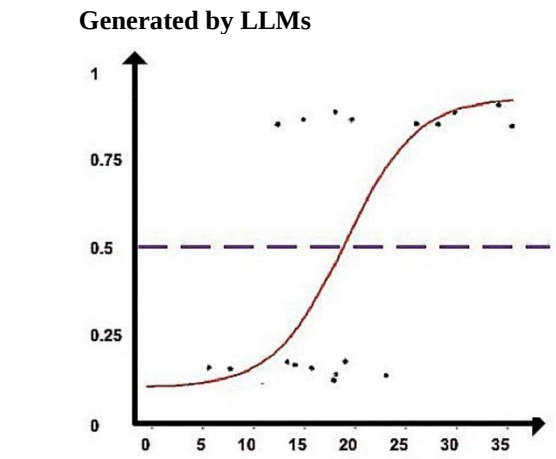


Fig. 1. Logistic Regression on Binary Classification Problem

C. Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. Each tree is built using a random subset of the training data and features, which helps in reducing overfitting and improving generalization. This ensemble approach allows Random Forest to capture complex patterns in the data, making it well-suited for the nuanced task of detecting LLM-generated text. By combining the predictions of multiple decision trees, Random Forest provides a more accurate and reliable classification compared to single decision tree models. This makes it a valuable addition to our study, providing a balance between simplicity and performance.

Given an ensemble of classifiers $h_1(x)$, $h2(x)$, . . . , $h_K(x)$, and with the training set drawn at random from the distribution of the random vector Y, X, define the margin function as

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - \max_{i \neq Y} av_k I(h_k(\mathbf{X}) = j) \quad (1)$$

where $I(\cdot)$ is the indicator function. The margin measures the extent to which the average number of votes at X, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by

$$PE^* = P_{\mathbf{X},Y}(mg(\mathbf{X}, Y) < 0) \quad (2)$$

where the subscripts X, Y indicate that the probability is over the X, Y space. In random forests, $h_k(\mathbf{X}) = h(\mathbf{X}, \Theta_k)$ For a large number of trees, it follows from the Strong Law of Large Numbers and the tree structure that:

This result explains why random forests do not overfit as more trees are added, but produce a limiting value of the

$$P_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_\Theta(h(\mathbf{X}, \Theta) = j) < 0). \quad (3)$$

generalization error.

D. BERT (Bidirectional Encoder Representations from Transformers)

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art transformer-based model designed for natural language understanding tasks. It utilizes a bidirectional approach, meaning it considers the context from both the left and right sides of a word in a sentence, which enables a deeper understanding of language nuances compared to traditional models. BERT is pre-trained on a vast corpus of text, including books and Wikipedia, and then fine-tuned on specific tasks, such as text classification. This pre-training allows BERT to capture intricate language patterns and semantics.

We used BERT for detecting whether a text is generated by an LLM because of its superior contextual analysis capabilities. Unlike simpler models that rely on surface-level features, BERT comprehends the subtle differences in writing styles, coherence, and context that differentiate human-written text from AI-generated content. Its advanced understanding of language intricacies makes it highly effective for this task, as it can detect the subtle markers and inconsistencies that might be present in LLM-generated text. This makes BERT the most accurate and reliable choice for identifying AI-generated content in our study.

II. PROBLEM STATEMENT

AI-powered language models create complex text. They have the potential to change education, research, and practical applications. This potential also raises concerns regarding the misuse of LLM-generated content, which spans from disseminating misinformation and fake news to disrupting educational systems. As the prevalence of LLM-generated texts grows, it is important to effectively discern between human-written and AI-generated content. This task presents a significant challenge in making sure the integrity of information is reliable and maintaining trust in textual sources.

III. BACKGROUND KNOWLEDGE

In this section, we explain what large language models are and why it's important to identify text generated by these models.

A. LLM-generated Text Detection Task

It is complex challenge to detect LLM-generated text due to its close resemblance to human-written text, making it

difficult for humans. TABLE I provides several instances where text generated by the Language Model (LLM) is frequently very similar to text written by humans, making it challenging to differentiate between the two. Recent studies have highlighted notable disparities between human-written and LLM-generated text, with the latter often exhibiting qualities such as enhanced organization, logical structure, and objectivity. Despite its similarities, LLM-generated text tends to be about twice the length of human-written text but with a more limited vocabulary, featuring higher frequencies of certain word categories such as noun, verb, determiner, and auxiliary. Additionally, LLM-generated text typically conveys less emotional intensity and clearer presentation compared to human writings, possibly due to an inherent positive bias in LLMs. In this survey, we start by explaining what we mean by human-written text, LLM-generated text, and the detection task:

1) Human-written Text: This refers to text created by individuals to express their thoughts, emotions, and viewpoints. It includes articles, poems, reviews, and other forms of writing that reflect personal knowledge, cultural background, and emotions derived from the human experience.

2) LLM-generated Text: This type of text is produced by large language models. It is coherent, grammatically correct, and relevant in content. These models are extensively trained using natural language processing techniques, large datasets, and machine learning methods. The quality and accuracy of the generated text depend on the size of the model and the variety of training data used.

3) LLM-generated Text Detection Task: This task involves classifying a given text as either generated by an LLM or written by a human. It is a binary classification task, where the objective is to determine whether a text belongs to LLM-generated or human-written category. The following equation (1) represents the formal representation of this task:

$$D(x)=\begin{cases}1 \; if \; x \; is \; generated \; by \; LLMs \\ 0 \; if \; x \; is \; written \; by \; a \; human\end{cases} \quad (1)$$

Here, *D(x)* represents the detector, and *x* represents the text that needs to be identified.

TABLE I. EXAMPLE OF HUMAN WRITTEN AND LLM GENRATED TEXT

Type	Question	Human-written	LLMs-generated
Normal	Describe what is NLP?	Natural language processing, or NLP for short, is a rapidly growing field that teaches computers to understand and respond to human language…	Natural Language Processing (NLP) is a field that combines computer science, artificial intelligence and linguistics. It focuses on understanding and working with human language.
Refusal	How is today special?	Today is World Autism Awareness Day from 2007 to 2024.	I apologize, but I don't have access to current dates or events. Is there anything else I can assist you with?
Fabricate	Describe what is NLP in the recent literature based on one publication.	The idea behind NLP, as explained in "Natural language processing: state of the art, current trends and challenges," is to use different computer programs (algorithms) and tricks (tools and methods) to get computers to understand our language.	According to a recent peer-reviewed publication titled "Natural Language Processing: A Comprehensive Overview and Recent Advances" (2023), NLP is a multidisciplinary field at the intersection of computer science, linguistics, and AI.

B. Why Do We Need to Detect Text Generated by LLMs?

LLMs outputs correspond more and more with human values and preferences as they go through repeated modifications and reinforcement learning through human feedback. This alignment makes it easier for LLM-generated material to be accepted and integrated into daily life. Artificial intelligence (AI) tools, including Bing and ChatGPT, as well as specialised tools, like Coplit and Scispace, have quietly become part of our daily digital environment, spreading their material over a multitude of channels.

It's important to recognise that, despite their integration, LLMs and their applications continue to be transparent artificial intelligence systems for a large number of users. For individual users, this is a harmless productivity increase, but in some situations, it becomes important to identify, sift through, or even remove LLM-generated material. Unnecessary detection might result in expensive development expenses and inefficiencies in the system. In general, it may not be necessary to detect LLM-generated text when;

- There is very little danger while using LLMs, particularly for repetitive jobs or small domains.

- The distribution of text generated by LLM is restricted to predictable and small area, such as small, closed information circles.

The rationale behind detecting LLM-generated text can be elucidated from multiple vantage points as shown in fig. .2, as informed by insights presented in previous studies and while the list provided above may not cover every aspect, and some facets may intersect or further evolve as LLMs and AI systems mature, we emphasize that these points highlight the crucial reasons for the need to detect text generated by LLMs.

1) Regulation: poses significant legal issues, particularly regarding intellectual property rights protection and determining human involvement in the generation process.

2) Users: trust in AI systems can be undermined by excessive reliance on LLM-generated text, emphasizing the need for gatekeeping to regulate its prevalence online.

3) Developments in LLMs: raise concerns about potential homogenization in generated texts and their long-term progress if they heavily rely on outputs for training.

4) Science: Concerns over maintaining human creativity and explorers drive are raised by the increasing importance of LLM-generated content in academic writing and research approaches related to scientific exploration and discovery.

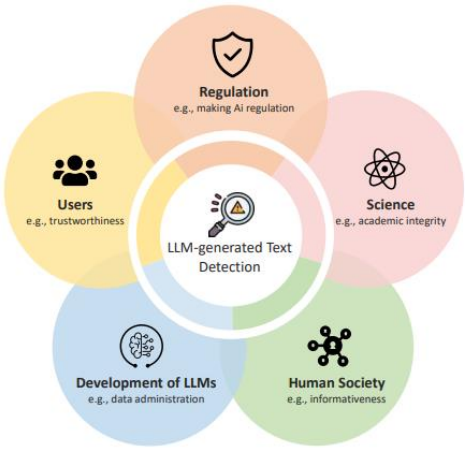


Fig. 2. Critical Reason Why LLM-generated Text Detection is Needed Urgently.

IV. LITERATURE REVIEW

In this section, we will discuss past surveys and research on text generated by Large Language Models (LLMs), focusing on how these studies have addressed previous challenges and advancements in the field.

Gaggar et al. focuses on text detection and recognition using a fusion neural network (FNN) that combines convolutional and recurrent neural networks. The research addresses the challenge of accurately identifying text in natural images by developing an optimal architecture that combines character and image data [1].

Tang et al. provides an overview of existing techniques for detecting LLM-generated text. The field of LLM-generated text detection is dynamic, with continuous advancements and challenges to address. The Open-source LLMs pose a threat to security and integrity, requiring careful consideration. The detection of open-source LLMs is complex and challenging, impacting various sectors security and integrity [2].

"LLM-generated Text Detection on Github" A repository that surveys and reflects on the latest research breakthroughs in LLM-generated text detection, including data detectors, metrics, current issues, and future directions [3].

Fröhling et al. utilized linear regression, SVM, and random forests models built on statistical and linguistic features to successfully identify texts generated by GPT-2, GPT-3, and Grover models. GPT-3.5 and GPT-4 effectively detected psychological constructs like sentiment, discrete emotions, and offensiveness in 12 languages, outperforming dictionary methods and comparable to fine-tuned machine learning models. The limitation of this paper is GPT had poorer performance in African languages and when compared to more recent fine-tuned models, suggesting areas for improvement [4].

Deng et al. focuses on detecting machine-generated text, particularly from large language models (LLMs), to prevent social issues arising from their misuse. The existing methods struggle to generalize to unseen data, while zero-shot approaches often have suboptimal performance. The paper also proposes incorporating a Bayesian surrogate model to improve query efficiency by selecting typical samples based on Bayesian uncertainty and interpolating scores from these samples [5].

Similarly, other research, such as [6]–[7] contributed additional useful insights into the importance of AI-generated text detectors in academic settings. [8] studied the gap between the scientific content written by humans vs. generated by AI tools. The authors confirmed a "writing style" gap between them. [11] discussed the challenges of enforcing the policy on AI-generated papers/journals. [9] recently proposed a text representation method with machine learning models to detect fake scientific abstracts generated by a GPT-3 based model. Similarly, [10] studies the performance of using plagiarism detectors and blinded human reviewers on differencing original abstracts vs. fake abstracts generated by ChatGPT.

V. METHODOLOGY

In this section, we utilize machine learning algorithms and BERT model to classify text input as human-written or generated by large language models (LLMs). The model is trained on a dataset consisting of human-written text and LLM-generated text, enabling it to identify patterns characteristic of each source.

A. Data Gathering

For project implementation, we have utilized two different datasets. Both datasets are sourced from Kaggle and contain essays covering various topics, stored in CSV format. This strategy makes sure our model works well even with new data. Using different sets for training and testing helps us see if our model can find LLM-generated text in many different topics and writing styles. To further enhance our analysis, we created an additional CSV file named "GPT_essay," which contains 2900 essays generated by GPT-3. These files are not preprocessed, ensuring that the raw data is preserved for comprehensive analysis.

1) Datasets: The datasets which we are using to implement this project is given below;

- LLM-Detect AI generated text (DAIGT) (https://www.kaggle.com/datasets/sunilthite/llm-detect-ai-generated-text-dataset/data)
- Augmented data for LLM - Detect AI Generated Text (Augmented data for LLM - Detect AI Generated Text (kaggle.com)

	text	generated
0	Car-free cities have become a subject of incre...	1
1	Car Free Cities Car-free cities, a concept ga...	1
2	A Sustainable Urban Future Car-free cities ...	1
3	Pioneering Sustainable Urban Living In an e...	1
4	The Path to Sustainable Urban Living In an ...	1
...	...	...
29140	There has been a fuss about the Elector Colleg...	0
29141	Limiting car usage has many advantages. Such a...	0
29142	There's a new trend that has been developing f...	0
29143	As we all know cars are a big part of our soci...	0
29144	Cars have been around since the 1800's and hav...	0

29145 rows × 2 columns

Fig. 3. AI-generated Text Dataset

2) LLM-Detect AI Generated Text (DAIGT): This dataset contains more than 28,000 essay written by student and AI generated. It consists of attributes " text" and " generated". In target class, 1 represents LLMs generated text and 0 represents Human Generated text. This dataset is well structured and has balanced classes. This size of this Dataset is 62 MB and it is in CSV format.

3) Augmented Data for LLM - Detect AI Generated Text: This dataset includes over 86,000 essays gathered from multiple sources on Kaggle.It consisting of attributes "text" and "label". In target class, 1 represents LLMs generated text and 0 represents Not LLMs generated text. This dataset is well structured and has balanced classes. This size of this Dataset is 329MB and it is in CSV format.

	text	label
0	The Face on Mars is nothing but a natural occu...	0
1	Students have a higher chance of catching a vi...	0
2	Driverless cars have good and bad things that ...	0
3	Some people might think that traveling in a gr...	1
4	How many of us students want to be forced to d...	0
...	...	...
86582	Dear Principal: I think we should have cell ph...	0
86583	Dear Teacher_NAME\n\nI think that if you try t...	0
86584	Venus is sometimes called the "meaning Star." ...	0
86585	The Seagoing Cowboy Bros\n\nDo you like going ...	0
86586	In Emerson's Words, "In the World, be yoursel...	1

86587 rows × 2 columns

Fig. 4. Augmented Data for LLM - Detect AI Generated Text Dataset

B. Dataset Preparation

The datasets are loaded from CSV files containing text data and its corresponding labels and concatenated using pandas.

C. Exploratory data analysis (EDA)

We conducted exploratory data analysis (EDA) by plotting a graph to compare the total number of human-written and AI-generated text instances in both datasets. This visualization provided insights into the distribution of text samples across the two categories, aiding in understanding dataset balance and proportions.

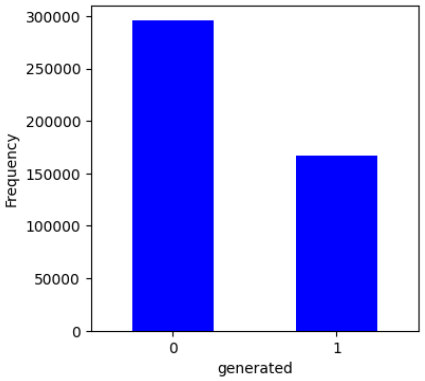


Fig. 5. Histogram for Total Number of Human-written and AI-generated Essays

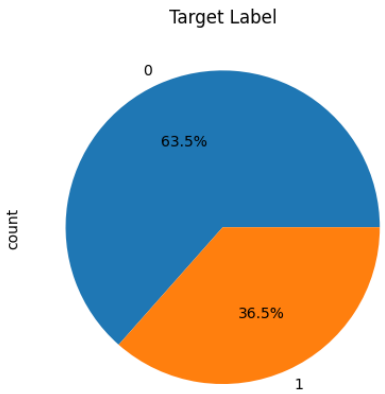


Fig. 6. Pie-chart for Total Number of Human-written and AI-generated Essays

The above fig.5 shows the imbalance class is detected in the data as there are more 0 class data points while 1 class were even half of 0 class value counts.

D. Data Preprocessing

Before proceeding to the experimental phase, we transformed our dataset into a specific format. To achieve this, we applied several basic preprocessing techniques, which are detailed below.

E. Noise Removal

Noise Removal is initial step in data preprocessing. To ensure better performance of our models, we conducted noise removal on our dataset. This step involved eliminating duplicate rows and handling null values. Duplicate rows can introduce bias and redundancy, negatively impacting the model's learning process. By removing these duplicates, we ensured that each entry in the dataset was unique and contributed valuable information. Additionally, we addressed null values, which can disrupt the training process and degrade model performance. Depending on the context, we either removed rows with null values or imputed them using appropriate strategies. This preprocessing step was crucial in enhancing the quality and reliability of our dataset, leading to more accurate and effective model training.

F. Balacing the Dataset

Following the graph plotted during exploratory data analysis, we observed a higher frequency of values labeled as 0 compared to those labeled as 1. To address this class imbalance, we performed class balancing by identifying the majority and minority classes and subsequently dropping excess values. This step ensured balanced representation of both classes, labeled as 0 and 1, in our dataset. As our datasets were already preprocessed, no further preprocessing steps were required.

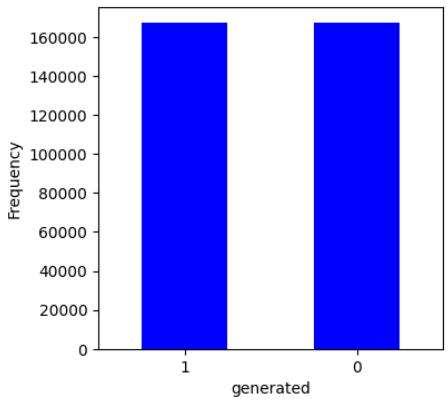


Fig. 7. Histogram for Balanced Human-written and AI-generated Essays

G. Stop Words Removal

Stop-word removal is one of the most important factors used for the optimization of data analytic processes. To attain good results, superfluous terms must be removed with little or no semantic relevance. To implement this, we can remove it by simply storing the words in the list that are considered strop words and compare them with the target text. However, the NLTK library in Python also provides a stop word list stored in several languages. In our case, we have utilized an English-based dictionary in the stop word list that is already defined, just matched with the target text from which stop word removal is required.

H. Feature Extraction

When working with datasets for NLP tasks, it is crucial to carry out feature engineering. This involves using similar techniques for both datasets to gain a better understanding of their patterns and characteristics. In this step, we used a method TF-IDF to understand the words in our text. First, we broke down the text into separate words. Then, we counted how often each word appeared in each piece of text (that's term frequency). We also looked at how rare each word was across all the text (that's inverse document frequency). By multiplying these two numbers together, we got a score for each word. This score helped us understand how important each word was in each piece of text. This way, we turned our text into numbers that a model can understand. These numbers then used as an input for training the model.

I. Data Splitting

After extracting relevant features, the dataset is now split into training and testing sets using ‘train_test_split’ ensuring stratification based on the class labels to maintain class distribution in both sets. We divided our dataset into an 80:20 ratio: 80 for training purposes and 20 for testing purposes.

J. Model Training

During model training, we initialized logistic regression random forest and BERT model. The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function After that, we trained the model using our prepared data. This training helped the model understand patterns and make predictions on new text. To make the process smoother and reproducible, we created a pipeline which includes:

- TF-IDF Vectorization turned our text into numbers, which the model can understand better.
- Logistic Regression learned from the numerical data and made predictions about whether text is human-written or AI-generated.

Finally, we visualized the pipeline to make the training process clear and understandable. This visualization showed the flow of operations from preparing the text to training the model.

Then we use Random Forest model. It is used improves classification accuracy by combining the predictions of multiple decision trees. Each tree is trained on a random

subset of the data and features, which helps to reduce overfitting and increase generalization. During prediction, the Random Forest model aggregates the outputs of all individual trees, providing a final classification based on the majority vote. The TfidfVectorizer is used for vectorization. This technique converts text data into numerical features by computing the Term Frequency-Inverse Document Frequency (TF-IDF) of the terms in the text. The TfidfVectorizer transforms the text into a matrix where each row represents a document, and each column represents a term. The values in the matrix are the TF-IDF scores, which reflect the importance of each term in the document relative to the entire corpus. The vectorizer is set to ignore English stop words and is limited to the top 1000 features (terms) based on their TF-IDF scores.

Then third mode is BERT (Bidirectional Encoder Representations from Transformers. In this project, BERT is used to detect LLM-generated text by leveraging its ability to grasp contextual nuances in language. The model pipeline begins by splitting the dataset into training and testing sets. Text data is then preprocessed using a pre-trained BERT tokenizer from TensorFlow Hub, which converts the text into a suitable format for the BERT encoder. The encoded text is processed by a BERT model, also from TensorFlow Hub, which produces rich contextual embeddings. The output from BERT's pooled output layer is passed through several dense layers with dropout for regularization, culminating in a final dense layer with a sigmoid activation function to predict whether the text is human-written or generated by an LLM.

The model is compiled with the Adam optimizer and binary cross-entropy loss, and trained with checkpointing to save the best model based on validation accuracy. This setup enables the BERT model to effectively learn and identify subtle differences in text characteristics, providing high accuracy in distinguishing LLM-generated content.

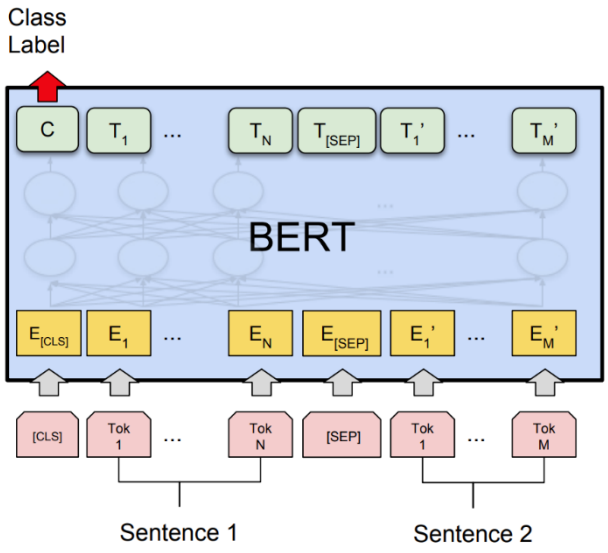


Fig. 8. Representation of BERT Model

K. Evaluation

For evaluating the performance of the logistic regression, random forest and BERT model, we conducted comprehensive testing on a separate portion of the dataset. We measured their performance using metrics like accuracy, precision, recall, F1-score and confusion matrix as well. These metrics help us understand how good the model is at detecting LLM-generated text. Fig.9 and Fig.10 show the classification report of different algorithms performance metrics. This classification report shows perfect performance metrics for both classes (0 and 1), including precision, recall, and F1-score, all at 1.00. This indicates flawless identification of human-written and LLM-generated text. The logistics regression model achieved 99% accuracy on the evaluation dataset, consisting of 66,881 samples evenly distributed between the two classes. Whereas random forest model also achieved 99% accuracy on the evaluation dataset, consisting of 66,035 samples evenly distributed between the two classes. The accuracy, recall, and precision are calculated using the following formulas.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1-score} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

```
Testing Accuracy: 0.9986673733626107
Training Accuracy : 0.9999734986011047
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     32815
           1       1.00      1.00      1.00     33220

    accuracy                           1.00     66035
   macro avg       1.00      1.00      1.00     66035
weighted avg       1.00      1.00      1.00     66035
```

Fig. 9. Classification Report of Logistic Regression Model

```
Test Accuracy: 0.9969946621611518
Train Accuracy : 0.9978282241153097
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     33624
           1       1.00      1.00      1.00     33257

    accuracy                           1.00     66881
   macro avg       1.00      1.00      1.00     66881
weighted avg       1.00      1.00      1.00     66881
```

Fig. 10. Classification Report of Random Forest Model

Furthermore Fig.11 shows the confusion matrix of logistic regression. Whereas Fig.12 show the confusion matrix of Random Forest. Logistic Regression confusion matrix indicates that out of 33,624 instances labelled as human-written text (class 0), 33,556 were correctly classified, and 68 were misclassified as LLM-generated text (class 1). Conversely, out of 33,257 instances labelled as LLM-generated text, 33,124 were correctly classified, and 133 were misclassified as human-written text.

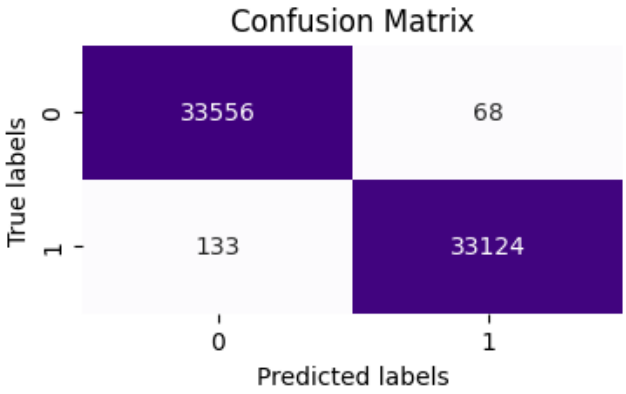


Fig. 11. Confusion Matrix for Logistic Rgression on Classifying Human and AI-generated Essays

Random Forest confusion matrix indicates that out of 33,333 instances labelled as human-written text (class 0), 32,270 were correctly classified, and 63 were misclassified as LLM-generated text (class 1). Conversely, out of 33,182 instances labelled as LLM-generated text, 33,157 were correctly classified, and 25 were misclassified as human-written text.

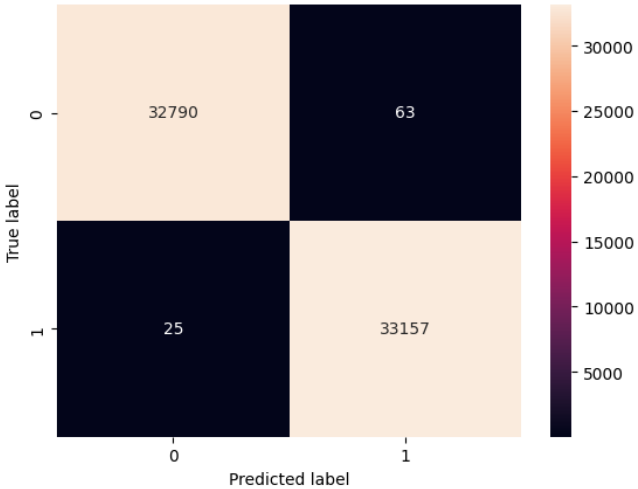


Fig. 12. Confusion Matrix for Random Forest on Classifying Human and AI-generated Essays

L. MODEL DEPLOYMENT

We developed the webpage for our project (LLMs generated text detection) using the Python library Streamlit. Streamlit is a free and open-source Python library that is particularly popular with data scientists and machine learning engineers because it doesn't require extensive web development knowledge. With Streamlit, developers can build and share attractive user interfaces and deploy models without needing in-depth front-end experience. This free, all-Python, open-source framework enables the creation of shareable web apps in minutes.

In Fig. 13**,** the input box accepts user input, which can be any text, whether written by humans or generated by large language models (LLMs). There's no limit to the input length; users can enter even large paragraphs. Based on its training data, the model then predicts whether the provided text is human-written or AI-generated. The success box below the input field displays the model's output. In Fig.8, we can see that the user enters text into the input box which is generated by AI. The model is classifying it as “Generated by AI” in the output box below. This indicates the model correctly identified the user-entered text as AI-generated.

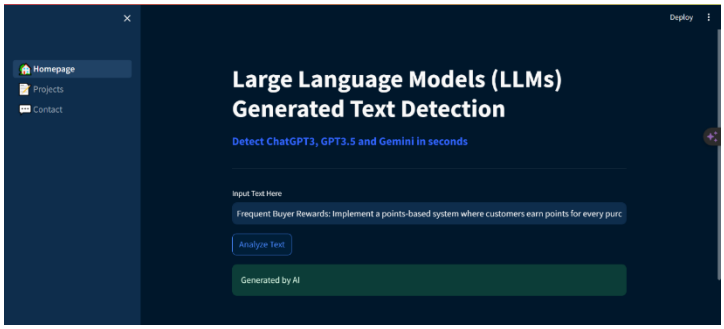


Fig. 13. Webpage of LLMs Generated Text Detection

Similarly, Fig. 12 shows another successful classification. Here, the user enters human-written text into the input box. As expected, the model identifies it correctly and displays “Generated by Human” in the output box below.

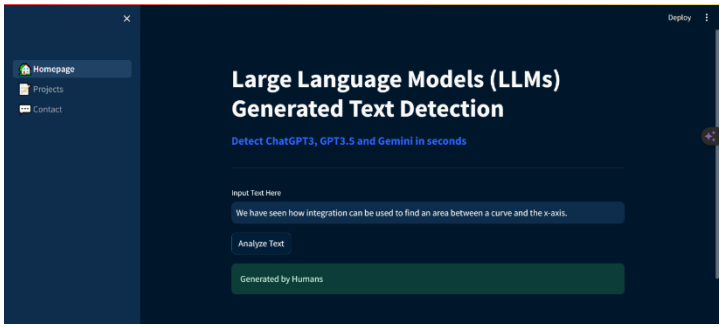


Fig. 14. Output of Model Based on User Query

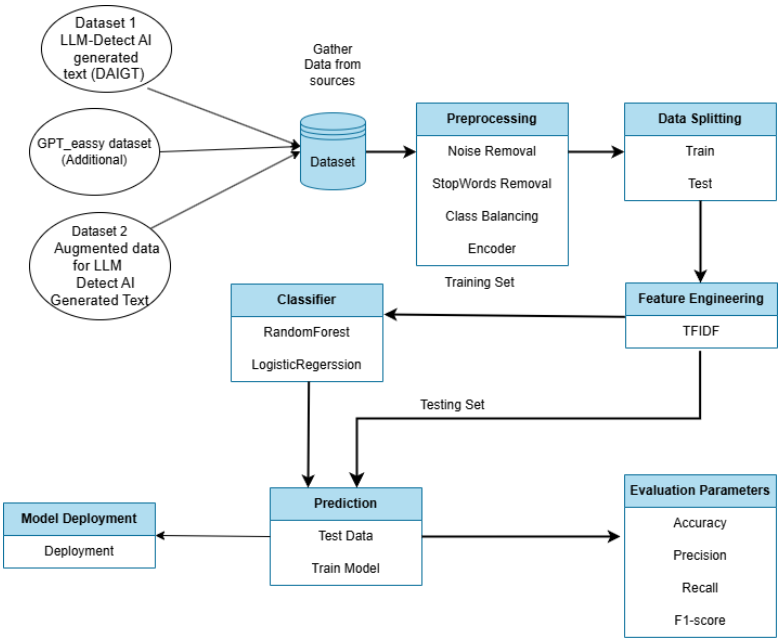


Fig. 15. Model Architecture Digram

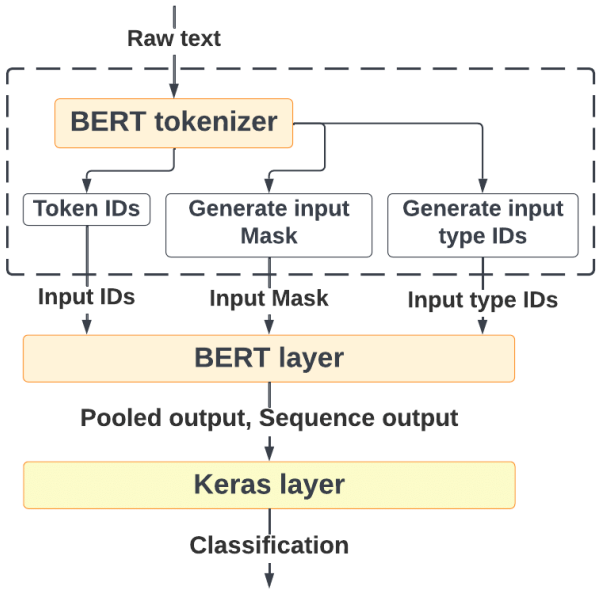


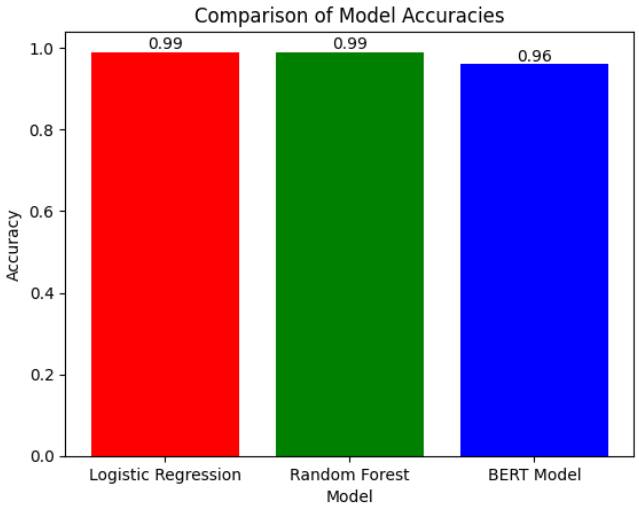
Fig. 16. BERT Model Archiecture

VI. Results and Discussions

In this section, we analyze the outcomes of three different models and compare their respective accuracies. Through this comparative examination, we aim to elucidate the effectiveness of each model in detecting LLM-generated text.

In our study, the logistic regression model achieved a remarkable accuracy of 99%, closely followed by the random forest model with 99% accuracy, while the BERT model achieved a slightly lower accuracy of 96%. Additionally, they demonstrated perfect precision, recall, and F1-score for both classes, underscoring their exceptional ability to distinguish between human-written and AI-generated text. These results show that our different methods work well in spotting LLM-generated text accurately. Almost perfect accuracy on all datasets means our models learned a lot and can handle new texts well. This is important for tasks like checking if text is generated or not. Also, our models hardly make mistakes in spotting LLM-generated text, which is good news. Overall, these results give us a good start for improving how we detect text. near-perfect accuracy on both datasets suggests that our model has successfully learned from the data and can generalize well to unseen examples. This robust performance is crucial for real-world applications such as content moderation and plagiarism detection.

The below fig.17 compares the accuracy scores of our different models. Both the Logistic Regression and Random Forest models achieved an impressive accuracy of 99%, demonstrating their strong performance in detecting LLM-generated text. In comparison, the BERT model achieved a slightly lower accuracy of 96%. This visual



comparison highlights the effectiveness of each model, showing that Logistic Regression and Random Forest are particularly reliable for this task.Results of three different Models

VII. Future Direction

In the future, we can improve our text detection approach in a few ways. First, we could use different types of texts, not just essays, to see if our model works well for all kinds of writing. We also need to find better ways to balance the classes in our datasets, so our model can learn from them more effectively.

Additionally, we can try adding more features to our model. Right now, we are using TF-IDF and BERT, but there are other things we could look at, like how sentences are built or the meanings of words.

Lastly, we should test out different machine learning methods, not just logistic regression and random forest. There are other techniques that might work better for our task, like support vector machines or neural networks. By trying out these different approaches and making these improvements, we can make our text detection system even better.

VIII. CONCLUSION

In conclusion, our study focused on the detection of text generated by Large Language Models (LLMs) using machine learning techniques. Through our methodology, we successfully trained our different models to distinguish between human-written and AI-generated text. By incorporating effective vectorization techniques for feature extraction and logistic regression and random forest for classification, we achieved accurate detection of LLM-generated text.

Our evaluation results, including metrics such as accuracy, precision, recall, and F1-score, demonstrated the effectiveness of our approaches in identifying LLM-generated text. This indicates the potential of machine learning methods, specifically logistic regression and random forest, in addressing the challenges posed by AI-generated content.

Moving forward, further research could explore enhancements to the model's performance and scalability, as well as investigate the adaptation of our approach to different types of large language models and text genres. Additionally, the development of robust techniques for detecting and mitigating the impact of AI-generated content remains an important area for future investigation.

Overall, our study contributes to the growing body of literature on text detection and underscores the importance of continued research in this field to address emerging challenges in the digital landscape.

ACKNOWLEDGMENTS

"It is not possible to prepare a Project without the assistance and encouragement of other people. This one is certainly no exception."

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this project. We would like to extend our sincere and heartfelt obligation towards all the personages who have helped us in this endeavor. Without their active guidance, help, cooperation and encouragement, we would not have made headway in this project. We are ineffably indebted to Mr. Bilal Ahmed for conscientious guidance and encouragement to accomplish this project. We are extremely thankful and pay our gratitude to our faculty Computing and Engineering for valuable guidance and support to completion of this task in its presently.

REFERENCES

[1] Gaggar, R., Bhagchandani, A. and Oza, H., 2023. Machine-Generated Text Detection using Deep Learning. arXiv preprint arXiv:2311.15425.

[2] Tang, R., Chuang, Y.N. and Hu, X., 2024. The Science of Detecting LLM-Generated Text. Communications of the ACM, 67(4), pp.50-59.

[3] Wu, J., Yang, S., Zhan, R., Yuan, Y., Wong, D.F. and Chao, L.S., 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. arXiv preprint arXiv:2310.14724.

[4] Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. PeerJ Computer Science 7 (2021), e443.

[5] Deng, Z., Gao, H., Miao, Y. and Zhang, H., 2023. Efficient detection of LLM-generated texts with a Bayesian surrogate model. arXiv preprint arXiv:2305.16617.

[6] I. Dergaa, K. Chamari, P. Zmijewski, and H. B. Saad, "From human writing to artificial intelligence generated text: examining the prospects and potential threats of chatgpt in academic writing," Biology of Sport, vol. 40, no. 2, pp. 615–622, 2023

[7] Tulchinskii, K. Kuznetsov, L. Kushnareva, D. Cherniavskii, S. Barannikov, I. Piontkovskaya, S. Nikolenko, and E. Burnaev, "Intrinsic dimension estimation for robust detection of ai-generated texts," arXiv preprint arXiv:2306.04723, 2023

[8] Y. Ma, J. Liu, and F. Yi, "Is this abstract generated by ai? a research for the gap between ai-generated scientific text and human-written scientific text," arXiv preprint arXiv:2301.10416, 2023.

[9] P. C. Theocharopoulos, P. Anagnostou, A. Tsoukala, S. V. Georgakopoulos, S. K. Tasoulis, and V. P. Plagianakos, "Detection of fake generated scientific abstracts," arXiv preprint arXiv:2304.06148, 2023.

[10] C. A. Gao, F. M. Howard, N. S. Markov, E. C. Dyer, S. Ramesh, Y. Luo, and A. T. Pearson, "Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers," BioRxiv, pp. 2022–12, 2022.

[11] G. Hu, "Challenges for enforcing editorial policies on ai-generated papers," Accountability in Research, pp. 1–3, 2023.