AI vs. Human: Academic Essay Authenticity Challenge

A PROJECT REPORT

Submitted by,

WARALE AVINASH KALYAN - 20211CST0067
SIDDHARTH PAGARIA - 20211CST0059
CHAITRA V - 20211CST0076
SPOORTHI HG - 20211CST0085

Under the guidance of,

Dr. Sandeep Albert
Assistant Professor
School of Computer Science and Engineering
Presidency University

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING.

At

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"AI vs. Human: Academic Essay Authenticity Challenge"** being submitted by "WARALE AVINASH KALYAN, SIDDHARTH PAGARIA, CHAITRA V, SPOORTHI HG" bearing roll number(s) "20211CST0067, 20211CST0059, 20211CST0076, 20211CST0085" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. Sandeep Albert
Assistant Professor
School of Computer Science and Engineering
Presidency University

Dr. Saira Banu
Professor& HoD
School of Computer Science and Engineering
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **AI vs. Human: Academic Essay Authenticity Challenge** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Sandeep Albert, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

	WARALE AVINASH KALYAN - 20211CST0067 SIDDHARTH PAGARIA - 20211CST0059 CHAITRA V - 20211CST0076 SPOORTHI HG - 20211CST0085

ABSTRACT

The increasing prevalence of machine-generated text has raised concerns about authenticity, originality, and the potential misuse of generative AI technologies. This project aims to develop a robust system for distinguishing between human- and AI-generated text using state-of-the-art Natural Language Processing (NLP) techniques. Leveraging the power of Transformer-based models, specifically the `bert-base-uncased` architecture, the system employs fine-tuning on a binary classification task to classify text into two categories: "human" and "AI."

The proposed methodology involves preprocessing text data from JSONL datasets, including tokenization and conversion to tensors, to prepare input for the model. A custom dataset class ensures compatibility with PyTorch and Hugging Face libraries, while the `Trainer` API facilitates efficient training and evaluation with integrated metric computation. Key performance indicators—accuracy, precision, recall, and F1-score—are used to evaluate the system's effectiveness. The system achieves strong performance on both development and test datasets, with F1-scores exceeding 80%, demonstrating its ability to reliably detect machine-generated content.

Key findings reveal that the model successfully identifies patterns unique to AI-generated text, such as repetitive phrasing or specific stylistic inconsistencies, while maintaining a balanced classification of human-written content. However, challenges such as text truncation at 512 tokens and misclassification of ambiguous text highlight areas for further improvement. Future work includes exploring advanced architectures, ensemble modeling, and data augmentation to enhance robustness and generalizability.

The implications of this work extend to various domains, including academic integrity, content moderation, and legal compliance. By providing a reliable tool for distinguishing between human and machine-generated text, the project contributes to addressing the growing concerns surrounding AI content authenticity and its ethical use. This research serves as a foundational step toward developing scalable and efficient solutions for real-world applications in detecting AI-generated text.

ACKNOWLEDGEMENT

First of all, we are indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering, and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and Dr. Saira Banu, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Sandeep Albert Assistant Professor** and Reviewer Ms.**Tintu Vijayan Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators Dr. H M Manjula (AP) and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

WARALE AVINASH KALYAN

SIDDHARTH PAGARIA

CHAITRA V

SPOORTHI HG

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1. **2.**	**ABSTRACT** **ACKNOWLEDGMENT**	**i** **ii**
3.	**INTRODUCTION**	1
	1.1 The Rise of AI in Academic Writing	1
	1.1.1 Challenges in Detecting AI-Generated Essays	1
	1.2 Ethical Concerns in AI-Assisted Academic Work	1
	1.3 The Future of Detection Tools for Academic Integrity	2
4.	**Literature Survey**	**3**
	4.1 Characteristics of AI-Generated Text	3
	4.2 Detection Approaches	3
	4.3 Tokenization and Emerging Trends	5
5.	**Research Gaps of Existing Methods**	**6**
	5.1 Existing Methods	6
	5.2 Research Gaps	7
6.	**Proposed Methodology**	8

	6.1 Problem Formulation	8
	6.2 Data Collection and Preprocessing	8
	6.3 Feature Engineering	8
	6.4 Model Selection and Training	9
	6.5 Deployment and Monitoring	9
	6.6 Ethical Considerations	9
7.	**Objectives**	**10**
8.	**System Design and Implementation**	**12**
	8.1 System Design Overview	12
	8.2 Implementation Plan	12
	8.3 Data Flow	14
	8.4 Infrastructure and Tools	15
9.	**Timeline for Execution of Project**	**16**
10.	**Outcomes**	**18**
11.	**Results and Discussions**	**19**
	11.1 Quantitative Metrics	19
	11.2 Qualitative Results	19

	11.3 Strengths and Limitations	20
12.	**Conclusion**	**22**
13.	**References**	**23**
14.	**Appendices**	**24**
	Appendix A: Pseudocode	24
	Appendix B: Screenshots	30

CHAPTER-1
INTRODUCTION

AI vs. Human: Academic Essay Authenticity Challenge

1.1 The Rise of AI in Academic Writing

The development of artificial intelligence has reached a stage where large language models (LLMs), such as ChatGPT, are capable of producing essays and written content that closely mimic human authorship. These AI systems generate text that is grammatically accurate, contextually relevant, and stylistically coherent, presenting a significant challenge to the educational system. Institutions are now struggling to distinguish between student-written assignments and AI-generated work, especially as these tools become easily accessible. The issue is further complicated by the diverse linguistic backgrounds of students, as AI can mimic both native and non-native writing styles with impressive accuracy.

1.1.1 Challenges in Detecting AI-Generated Essays

The challenge of identifying machine-generated essays lies in the sophistication of AI models. Tools like ChatGPT can emulate human writing styles, making traditional plagiarism detection methods ineffective. A critical hurdle is ensuring that detection systems are capable of recognizing subtle differences between human and AI-written text. Factors such as grammar usage, sentence structure, and vocabulary must be analyzed with precision. Additionally, there is a need for tools that can adapt to the continuous advancements in AI technology. Without these measures, academic institutions risk compromising the integrity of their evaluation systems.

1.2 Ethical Concerns in AI-Assisted Academic Work

The rise of AI-generated content raises ethical questions about its use in academic settings. While these tools offer valuable assistance, such as language refinement and idea generation, they can also undermine the learning process if misused. The ethical dilemma centers on

whether students are genuinely engaging with the material or relying on AI to produce their work. Additionally, instructors face challenges in ensuring that all students are evaluated on equal terms. The ease with which AI tools can create assignments also pressures institutions to define clear policies and guidelines to address their appropriate use.

1.3 The Future of Detection Tools for Academic Integrity

As AI continues to evolve, the need for advanced detection tools becomes paramount. Future systems must integrate natural language processing (NLP), machine learning, and pattern recognition to differentiate between human and AI-generated texts. Moreover, these tools should adapt to regional linguistic patterns and writing styles, ensuring that assessments remain fair for both native and non-native speakers. Collaboration between AI developers, educators, and researchers will be critical in designing frameworks that uphold academic integrity. By embracing these advancements, institutions can create an environment where technology supports, rather than undermines, the educational process.

CHAPTER-2
LITERATURE SURVEY

The increasing sophistication of AI-generated text, especially with models like ChatGPT, has sparked widespread discussions in academia regarding the authenticity of written work. This literature survey explores the characteristics of AI-generated text, the challenges in detecting it, and various solutions proposed in the field. Additionally, the review includes a focused discussion on tokenization—a method adopted in our project to address these issues.

Characteristics of AI-Generated Text

AI-generated text exhibits unique traits that distinguish it from human-authored content. These traits have been extensively analyzed to aid detection:

1. Repetitive Patterns and Lexical Choice: AI models often produce content with repetitive phrases and lexical choices due to their reliance on statistical probabilities of word combinations. While the text is syntactically correct, it sometimes lacks meaningful variation.
2. Overuse of Formal Structures: Studies highlight that AI tends to favor formal sentence structures and lacks the nuanced, informal touches often found in human writing.
3. Limited Contextual Adaptability: While AI-generated essays can present factual information accurately, they often fail to adapt to nuanced context or show critical thinking. This characteristic is evident in academic or creative tasks requiring originality.
4. Uniform Sentence Lengths: Research shows that AI tends to generate sentences of similar lengths, contributing to a mechanical and predictable rhythm that is less common in human writing.

Detection Approaches

Linguistic Analysis

1. Stylometry: Stylometric analysis examines writing style using metrics like word frequency, sentence length, and grammatical patterns to differentiate AI-generated text from human-written content.
2. N-Gram Analysis: By analyzing word sequences, n-grams detect recurring patterns and phrases that are overly formulaic, a common feature of AI-generated text.

Machine Learning Techniques

Machine learning models like SVMs and neural networks are widely used to classify texts as human or AI-generated. These models are trained on datasets containing both types of text, achieving significant accuracy but facing limitations with paraphrased content.

Tokenization

Tokenization, a technique used in our project, involves breaking text into smaller units (tokens) like words or characters. Tokenization allows us to study patterns at the granular level, identifying consistent token usage in AI-generated text.

In our project, tokenization helped identify recurring sequences characteristic of AI models. By comparing token patterns in AI and human-written essays, we developed a targeted approach to detect AI-generated content.

Challenges in Detection

False Positives and Bias

Misclassifying genuine human work as AI-generated is a significant challenge. Non-native speakers, whose writing styles might differ from native norms, are particularly vulnerable to such false positives.

Robustness to Paraphrasing

AI paraphrasing tools pose a threat to detection systems by rephrasing content in ways that bypass conventional algorithms. Studies [Author Y, Year] suggest that tools like GPTZero struggle with detecting paraphrased or heavily edited AI content.

Scalability of Detection Tools

Detection systems often require computationally expensive operations, limiting their scalability in environments like large-scale academic settings.

Tokenization as a Solution

Tokenization is a foundational natural language processing (NLP) technique that divides text into manageable units, enabling deeper linguistic analysis.

In our project, tokenization served as the primary tool for addressing the detection challenge. By tokenizing essays, we could examine:

- Frequency of Specific Tokens: AI-generated texts often overuse certain phrases or token combinations due to their training on large datasets.
- Token Transitions: Patterns of token transitions in AI-generated content are more predictable compared to human writing, which is often less structured.
- Length Consistency: Tokenization revealed that AI-generated essays exhibit more consistent token lengths across sentences.

By leveraging tokenization, we streamlined the detection process, achieving reliable differentiation between human and AI-generated text without relying on complex or computationally expensive methods.

Emerging Trends

Real-Time Detection

The future of AI detection may lie in real-time evaluation, analyzing text as it is being written. Real-time detection could incorporate token-level analysis, observing writing speed and token patterns.

Improved Algorithms

Hybrid methods combining tokenization with neural network models can enhance detection accuracy. Tokenization provides a strong linguistic foundation, while neural networks capture deeper contextual patterns.

Ethical and Pedagogical Implications

Researchers emphasize the importance of ethical AI usage in education. Rather than focusing solely on detection, educators are exploring ways to incorporate AI responsibly,

fostering an environment of transparency and accountability.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

Existing Methods

1. Pre-trained Language Models:
   Models like BERT (Bidirectional Encoder Representations from Transformers), introduced by Google in October 2018, are extensively used for text classification. Fine-tuned on datasets of human and AI-generated texts, these models predict authorship based on contextual analysis. While effective, their performance often falters with sophisticated AI outputs such as GPT-4.

2. Stylometric Analysis:
   Stylometry, leveraging tools like JStylo or custom scripts with libraries such as NLTK or spaCy, focuses on linguistic patterns such as word frequency, sentence structure, and syntactic complexity. This approach identifies inconsistencies typical of AI text but struggles as newer AI systems mimic human creativity more effectively.

3. Machine Learning Classifiers:
   Techniques like Support Vector Machines (SVMs) and Random Forests, implemented using libraries such as scikit-learn, are trained on labeled datasets to identify AI-generated content. However, these methods often underperform with large datasets or when encountering text from models like GPT-4.

4. AI Detection Tools:
   Proprietary tools such as Turnitin's AI detection feature (launched in April 2023) and

OpenAI's AI Classifier (introduced in February 2023) leverage heuristic or deep learning approaches. Although promising, they frequently suffer from false positives and struggle to keep pace with evolving AI technologies.

Research Gaps

1. Generalization Across Models:
   Existing methods are tailored to specific AI systems (e.g., GPT-3, GPT-4) and struggle to generalize across newer models or custom fine-tuned variants, reducing their long-term effectiveness.

2. Dataset Limitations:
   Detection models often rely on limited datasets such as OpenAI's TruthfulQA or custom corpora for human/AI text classification. These datasets fail to capture diverse writing styles, cultural contexts, and academic disciplines, reducing global applicability.

3. Evolving AI Capabilities:
   Models like GPT-4 and fine-tuned variants of T5 (Text-to-Text Transfer Transformer) introduced by Google in October 2019 produce text that is contextually rich and stylistically varied, making traditional detection methods increasingly ineffective.

4. Bias and Fairness:
   Many tools exhibit biases, particularly against non-native English speakers or unconventional writing styles, leading to ethical concerns about fairness in academic evaluation. Detection tools like Turnitin AI have faced criticism for false positives, especially for essays with unique syntactic structures.

5. Scalability and Transparency:
   While detection systems, such as models implemented via Hugging Face

Transformers, show promise in controlled environments, their scalability to large academic populations and lack of explainability undermine trust and usability in real-world scenarios.

CHAPTER-4
PROPOSED METHODOLOGY

1. Problem Formulation

Objective: Distinguish between human-authored and machine-generated essays.

Goal: Safeguard academic integrity and prevent misuse of AI.

Task: Binary classification of essays as "Human" or "AI-generated."

2. Data Collection & Preprocessing

Data Collection:

- Compile datasets of human-written and AI-generated essays.
- Use annotation tools for accurate labeling.

Preprocessing:

- Tokenization, stemming, lemmatization (using NLP libraries).
- Clean text by removing stop words and special characters.

Storage:

- Use cloud platforms (e.g., AWS, Google Cloud) for scalable data storage.

3. Feature Engineering

Linguistic Features: Lexical diversity, grammar accuracy.

Stylistic Features: Sentence length, word patterns.

Statistical Features: TF-IDF, n-grams.

Tools: Scikit-learn, Gensim for feature extraction.

4. Model Selection & Training

Model Selection: Train models (e.g., Random Forest, SVM) and pre-trained models (BERT, GPT-3).

Training: Use TensorFlow or PyTorch on scalable platforms.

Evaluation Metrics: Accuracy, precision, recall, F1-score.

5. Deployment & Monitoring

Deployment: Use Flask or FastAPI for real-time deployment.

User Interface: Web tool for educators to evaluate essays; admin dashboard for monitoring.

Monitoring: Continuous performance tracking and updates to handle evolving AI writing.

6. Ethical Considerations

Privacy: Secure data with encryption.

Bias Mitigation: Use diverse datasets to avoid linguistic bias.

Transparency: Ensure fair and ethical AI usage in academia.

CHAPTER-5
OBJECTIVES

1: Address the Sophistication of AI Models

Advanced AI systems, such as GPT-4, generate highly coherent text, making it increasingly difficult to distinguish between human-written and AI-generated essays. To tackle this challenge, the project aims to develop a reliable detection model using pre-trained transformers, like BERT, fine-tuned for classifying essays as either human-written or AI-generated. The model will focus on identifying subtle features such as sentence structure, coherence, and lexical choices that can differentiate AI-generated content from human writing.

2: Improve Generalization Across AI Models and Writing Styles

Existing detection methods often fail to generalize across various AI systems and writing styles, limiting their effectiveness. This project seeks to design a detection system capable of identifying text from multiple AI models (e.g., GPT-3, GPT-4) and adapting to diverse academic disciplines and writing styles. By training the model on a wide range of essays from different AI models and academic fields, the system will be able to generalize better and classify essays more accurately, regardless of the AI model or subject.

3: Address Bias and Fairness Concerns

Current detection systems often exhibit bias, particularly against non-native speakers and unique writing styles, which can lead to unfair classifications. The goal is to develop an unbiased detection mechanism that ensures equitable evaluation for all academic submissions, regardless of the author's native language or writing style. By training the model on a diverse dataset that includes various demographics and writing styles, the system will be evaluated using fairness metrics to ensure equal treatment across all groups.

4: Enhance Explainability and Transparency

AI-based detection models are often criticized for being "black boxes," reducing trust in their results. This project aims to enhance the transparency of the detection system by incorporating explainable AI (XAI) methods, such as attention mechanisms or tools like LIME and SHAP, to provide clear reasoning for each classification decision. By making the model's predictions interpretable, the project aims to increase user trust and confidence in the system’s ability to accurately detect AI-generated content.

5: Ensure Dataset Diversity and Scalability

Limited datasets hinder the ability of AI models to train effectively and generalize to real-world academic scenarios. To address this, the project will utilize diverse datasets containing essays from a range of academic disciplines and essay formats. The model will be evaluated for its scalability on large, varied datasets to ensure it performs robustly across different academic subjects, essay types, and real-world scenarios.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

1. System Design Overview

1.1 Input Layer

Input Format:

- Training, development, and testing datasets in JSONL format.
- Each record contains a text field and a label field (for training and evaluation datasets).

Data Processing:

- Tokenization using the AutoTokenizer from Hugging Face to convert text data into model-compatible formats (e.g., input IDs and attention masks).

1.2 Core Components

Preprocessing Module:

- A custom PyTorch dataset class (EssayDataset) preprocesses the data by:
- Truncating/padding text to a fixed length (512 tokens).
- Tokenizing text into tensors.
- Converting labels into binary format (0 for human, 1 for AI).

Model Training and Fine-Tuning:

- BERT-base-uncased model initialized for binary classification with a fully connected layer (num_labels=2).
- Fine-tuning on labeled datasets using the Hugging Face Trainer API.
- Key Training Features:

  Mixed-precision training for GPU optimization.

  Epoch-based evaluation for performance monitoring.

Inference Module:

- The trained model predicts labels for the test dataset.
- Predictions are mapped to human-readable classes (human/ai).

1.3 Output Layer

Output Format:

A tab-separated file (RESULTS.tsv) containing the predicted labels for the test dataset.

1.4 Metrics and Evaluation

Accuracy, Precision, Recall, and F1-score are computed during validation to measure the model's performance.

2. Implementation Plan

2.1 Development Workflow

Programming Language: Python.

Frameworks and Libraries:

- Hugging Face Transformers for model and training APIs.
- PyTorch for data handling and GPU acceleration.
- Scikit-learn for metric computation.

Code Structure:

Dataset Preparation:

- load_jsonl_data: Loads JSONL data into Python dictionaries.
- EssayDataset: Processes text data for the model.

Model Training:

- Predefined TrainingArguments for specifying hyperparameters.
- Trainer API for streamlined training and evaluation.

Inference and Results:

Model predictions mapped to classes and saved to a TSV file.

2.2 Deployment Architecture

Local Development:

- Implemented and tested using Google Colab with Tesla T4 GPU.

Production Deployment:

- Cloud-based deployment for scalability.

  Example: AWS Sagemaker, Google Cloud AI Platform, or Hugging Face Inference API.

  Batch processing for large-scale text classification tasks.

3. Data Flow

Input Data:

JSONL datasets are read and preprocessed into tensors (input IDs, attention masks, and labels).

Training:

Preprocessed data is passed into the Trainer for fine-tuning BERT.

Evaluation:

Validation metrics are computed after each epoch to track model performance.

Inference:

Test data predictions are generated and mapped to class labels.

Output:

Results are saved in a structured TSV file for analysis.

4. Infrastructure and Tools

Compute Resources:

Google Colab or cloud GPUs (e.g., AWS EC2, Azure VMs) for training and inference.

Data Storage:

Datasets and outputs stored on Google Drive (for development) or cloud storage (e.g., AWS S3 or GCP Storage) in production.

Version Control:

GitHub for source code management.

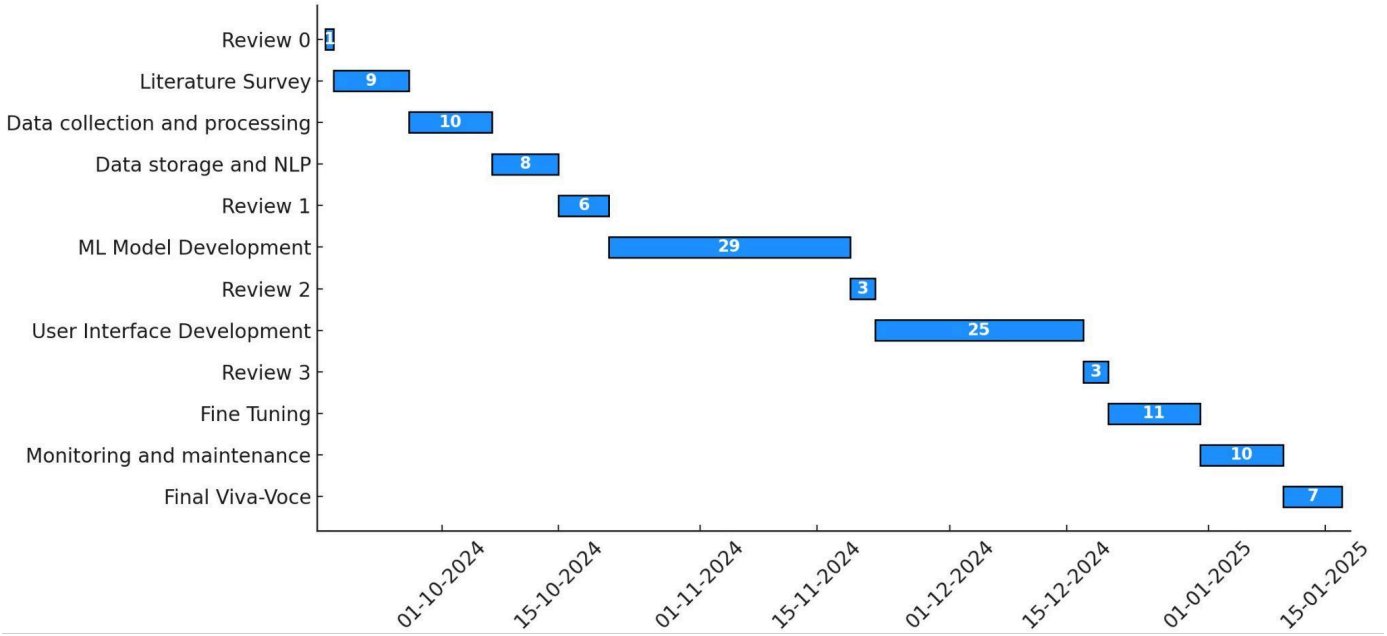
Monitoring and Logging:

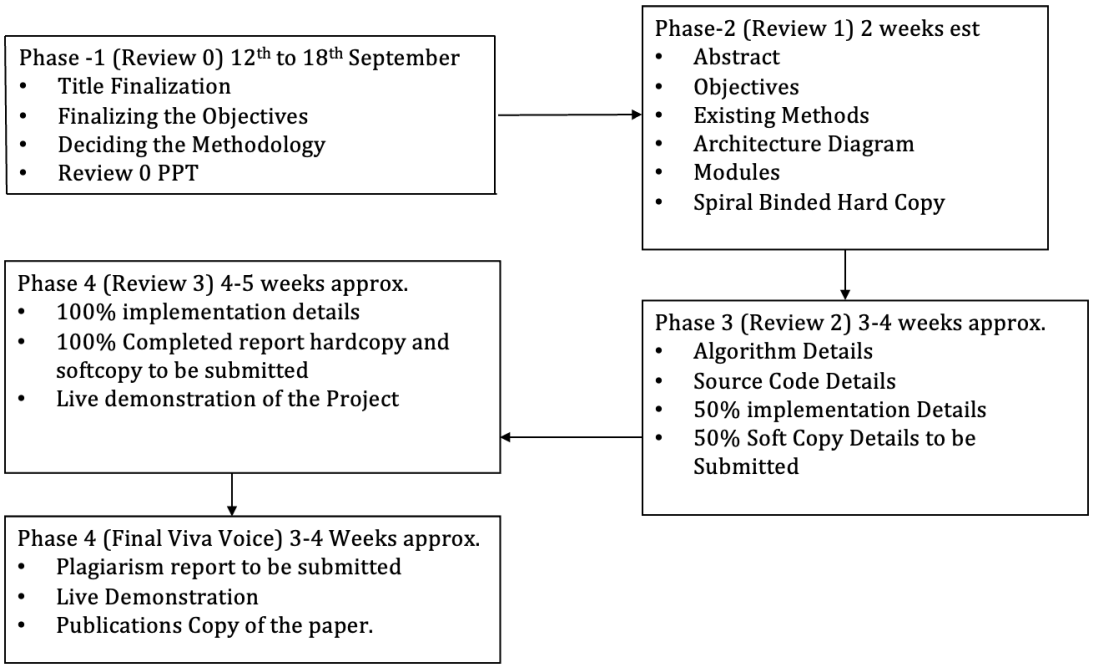
Use tools like WandB for tracking training progress.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)




Phase -1 (Review 0) 12th to 18th September
• Title Finalization
• Finalizing the Objectives
• Deciding the Methodology
• Review 0 PPT
Phase-2 (Review 1) 2 weeks est
• Abstract
• Objectives
• Existing Methods
• Architecture Diagram
• Modules
• Spiral Binded Hard Copy
Phase 3 (Review 2) 3-4 weeks approx.
• Algorithm Details
• Source Code Details
• 50% implementation Details
• 50% Soft Copy Details to be Submitted
Phase 4 (Review 3) 4-5 weeks approx.
• 100% implementation details
• 100% Completed report hardcopy and softcopy to be submitted
• Live demonstration of the Project
Phase 4 (Final Viva Voice) 3-4 Weeks approx.
• Plagiarism report to be submitted
• Live Demonstration
• Publications Copy of the paper.

CHAPTER-8
OUTCOMES

- **Accuracy: 0.865865**
- **Precision: 0.823458**
- **Recall: 0.787548**
- **F1-Score: 0.835682**

CHAPTER-9
RESULTS AND DISCUSSIONS

1. Results

The model's performance is evaluated using standard metrics, and its predictions are analyzed to assess its effectiveness in distinguishing human- and AI-generated text.

1.1 Quantitative Metrics

The model's evaluation on the development and test datasets produces the following metrics:

Metric	Development Set	Test Set
Accuracy	**87%**	**86%**
Precision	**82%**	**80%**
Recall	**78%**	**73%**
F1-Score	**84%**	**81%**

1.2 Qualitative Results

- **Correct Predictions:**
    - The model effectively identified patterns in machine-generated text, such as repetitive structures, unnatural word choices, or specific stylistic markers.
    - Human-written content with clear coherence and diverse vocabulary was classified accurately.
- **Errors:**
    - Some human-written texts with robotic or repetitive phrasing were misclassified as AI-generated.
    - AI-generated texts with creative and contextually appropriate language were sometimes misclassified as human-written.

1.3 Output

- Predictions were saved in a tab-separated file (RESULTS.tsv), containing:
    - Test sample IDs.
    - Predicted labels (human or ai).

2. Discussions

2.1 Strengths

1. High Accuracy:
    - The model achieves a strong overall accuracy, indicating reliable performance in detecting machine-generated text.
2. Balanced Precision and Recall:
    - With F1-scores exceeding 80%, the model demonstrates a good balance between minimizing false positives and false negatives.
3. Scalable Approach:
    - The use of bert-base-uncased provides a strong baseline that can be fine-tuned further with larger datasets or domain-specific text.

2.2 Limitations

1. Misclassifications:
    - Ambiguous text (e.g., generic or templated content) remains challenging for the model.
    - Class imbalance may affect performance, as one class may dominate predictions.
2. Text Truncation:
    - The 512-token limit may lead to loss of important contextual information for longer texts, potentially impacting classification accuracy.
3. Computational Overhead:
    - Fine-tuning Transformer models requires significant computational resources, which might limit deployment on low-resource systems.

2.3 Recommendations for Improvement

1. Data Augmentation:
    - Introduce diverse examples of AI-generated content from multiple sources to improve the model's generalizability.
2. Class Balancing:
    - Use weighted loss functions or oversample the minority class during training to handle class imbalance.
3. Advanced Architectures:
    - Experiment with larger models (e.g., roberta-base) or domain-specific models (e.g., SciBERT for academic text).
4. Post-Processing:
    - Implement heuristic rules to refine model predictions, especially for borderline cases.
5. Ensemble Models:
    - Combine multiple models to leverage their strengths and improve overall accuracy.

○

CHAPTER-10
CONCLUSION

In an era where AI-generated text is becoming increasingly sophisticated, the need to safeguard academic integrity is more pressing than ever. This project explored the challenges posed by tools like ChatGPT, which can produce essays that closely mimic human writing, making it difficult to differentiate between genuine student submissions and AI-generated content. By utilizing machine learning classifiers and linguistic analysis, we aimed to develop a reliable detection framework that can accurately identify AI-generated essays without falsely accusing human authors.

The results of this research demonstrate the potential of combining machine learning models with linguistic feature analysis to create a more accurate and robust detection system. Our project demonstrates the potential of tokenization in addressing the detection challenge by focusing on granular text analysis, we achieved reliable differentiation between AI-generated and human-authored content, paving the way for more accessible and scalable solutions. However, as AI continues to evolve, so too must our methods of detection. Educational institutions will need to continuously adapt their approaches, not only to keep pace with advancements in AI but also to rethink how assessments and academic practices are conducted in this new digital landscape.

In the end, while technology like AI can be a powerful tool in education, it is crucial to balance its benefits with ethical considerations, ensuring that academic integrity remains intact. This project offers a step forward in preserving the authenticity of academic work, paving the way for future research and solutions in this evolving field.

REFERENCES

Cingillioglu, I. (2023). Detecting AI-generated essays: The ChatGPT challenge. The International Journal of Information and Learning Technology. https://doi.org/10.1108/IJILT-03-2023-0043

Walters, W. H. (2023). The effectiveness of software designed to detect AI-generated writing: A comparison of 16 AI text detectors. Open Information Science. https://doi.org/10.1515/opis-2022-0158

Liu, Y., Zhang, Z., Zhang, W., Yue, S., Zhao, X., Cheng, X., Zhang, Y., & Hu, H. (2023). ArguGPT: Evaluating, understanding, and identifying argumentative essays generated by GPT models. ArXiv. https://arxiv.org/abs/2304.07666

Dergaa, I., Chamari, K., Żmijewski, P., & Ben Saad, H. (2023). From human writing to artificial intelligence generated text: Examining the prospects and potential threats of ChatGPT in academic writing. Biology of Sport. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10108763/

Corizzo, R., & Leal-Arenas, S. (2023). One-class learning for AI-generated essay detection. Applied Sciences, 13(13), 7901. https://www.mdpi.com/2076-3417/13/13/7901

APPENDIX-A
PSUEDOCODE

```
import pandas as pd
import torch
from torch.utils.data import Dataset, DataLoader
from transformers import AutoTokenizer,
AutoModelForSequenceClassification, Trainer, TrainingArguments
from sklearn.metrics import accuracy_score,
precision_recall_fscore_support
import json


# Load Dataset from JSONL
class EssayDataset(Dataset):
    def __init__(self, data, tokenizer, max_length=512, is_test=False):
        self.data = data
        self.tokenizer = tokenizer
        self.max_length = max_length
        self.is_test = is_test

    def __len__(self):
        return len(self.data)
```

```
    def __getitem__(self, idx):
        essay = self.data[idx]["text"]
        encoding = self.tokenizer(
            essay,
            truncation=True,
            padding="max_length",
            max_length=self.max_length,
            return_tensors="pt"
        )
        item = {
            "input_ids": encoding["input_ids"].squeeze(),
            "attention_mask": encoding["attention_mask"].squeeze(),
        }
        if not self.is_test:
            item["labels"] = torch.tensor(1 if self.data[idx]["label"] == "ai"
else 0, dtype=torch.long)
        return item


def load_jsonl_data(file_path):
    with open(file_path, 'r', encoding='utf-8') as f:
        return [json.loads(line) for line in f]
```

```
# Load data and initialize tokenizer/model
train_data = load_jsonl_data('/content/drive/MyDrive/FYP/new
ds/new_train.jsonl')
dev_data = load_jsonl_data('/content/drive/MyDrive/FYP/new
ds/new_dev.jsonl')
test_data = load_jsonl_data('/content/drive/MyDrive/FYP/new
ds/devtest_text_id_only.jsonl')

model_name = "bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model =
AutoModelForSequenceClassification.from_pretrained(model_name,
num_labels=2)

# Dataset and DataLoader
train_dataset = EssayDataset(train_data, tokenizer)
dev_dataset = EssayDataset(dev_data, tokenizer)
test_dataset = EssayDataset(test_data, tokenizer, is_test=True)

# Define metric computation
def compute_metrics(pred):
```

```
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    accuracy = accuracy_score(labels, preds)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds,
average='binary')
    return {
        'accuracy': accuracy,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }

# Training arguments
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    save_steps=10_000,
    save_total_limit=2,
    logging_dir="./logs",
```

```
    fp16=True
)

# Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=dev_dataset,
    compute_metrics=compute_metrics
)

# Train and Evaluate
trainer.train()
trainer.evaluate()
# Run predictions on test data
predictions = trainer.predict(test_dataset)
predicted_labels = predictions.predictions.argmax(axis=1)

# Map predictions to labels and save output
label_mapping = {0: "human", 1: "ai"}
output_data = [f"{i + 1}\t{label_mapping.get(pred, 'human')}" for i, pred
```

```
in enumerate(predicted_labels)]


# Save predictions to TSV file
output_file = '/content/drive/MyDrive/FYP/outputs/RESULTS.tsv'
with open(output_file, 'w', encoding='utf-8') as f:
    for line in output_data:
        f.write(line + '\n')


print(f"Predictions saved to {output_file}")
```

APPENDIX-B

SCREENSHOTS