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 Mathias

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 TECHNOLOGY.

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2024

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 Mathias, 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.

Name	**Roll No.**	**Signature**
WARALE AVINASH KALYAN	20211CST0067	

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 Mathias, 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.

Name	**Roll No.**	**Signature**
SIDDHARTH PAGARIA	20211CST0059	

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 Mathias, 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.

Name	**Roll No.**	**Signature**
CHAITRA V	20211CST0076	

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 Mathias, 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.

Name	**Roll No.**	**Signature**
SPOORTHI HG	20211CST0085	

ABSTRACT

This project presents a binary text classification system designed to differentiate between AI-generated and human-written academic essays. Utilizing the BERT (Bidirectional Encoder Representations from Transformers) architecture, the model is fine-tuned for this specific task. The system incorporates a robust preprocessing pipeline for tokenization, truncation, and padding of input text, ensuring compatibility with the model.A user-friendly web interface, built with Streamlit, enables real-time interaction. Users input text, which is processed and passed to the fine-tuned BERT model to predict whether it is AI-generated or human-written. The results, displayed instantly, highlight the model's high accuracy and reliability.This project demonstrates the effectiveness of transformer-based architectures in text classification while showcasing seamless integration between advanced machine learning models and intuitive web application frameworks. It offers a practical solution for detecting AI-generated content and underscores the growing importance of AI authenticity in academic and professional domains.

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 Mathias,** 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

LIST OF TABLES

Sl No.	Table Name	Table Caption	Page No.
1.	Table 1	Dataset information	14
2.	Table 2	Quantitative Results	35

LIST OF FIGURES

Sl No	Figure Name	Caption	Page No.
1	Figure 1	Project Algorithm	18
2	Figure 2	Model Architecture	24
3	Figure 3	VS Code development	28
4	Figure 4	Github representation	29
5	Figure 5	Architectural Diagram of Front-end	30
6	Figure 6.1	Gantt Chart	31
7	Figure 6.2	Project Timeline	32
8	Figure 7	Screenshots of Training Data	46
9	Figure 8.1	Output 1	46
10	Figure 8.2	Output 2	47
11	Figure 8.3	Output 3	47
12	Figure 9	Results of Training Data	48

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	CERTIFICATE	**ii**
	DECLARATION	**iii**
	ABSTRACT	**iv**
	ACKNOWLEDGMENT	**v**
	LIST OF TABLES	**vi**
	LIST OF FIGURES	**vii**
1.	**INTRODUCTION**	1
	1.1 The Rise of AI in Academic Writing	1
	1.2 Challenges in Detecting AI-Generated Essays	1
	1.3 Ethical Concerns in AI-Assisted Academic Work	2
	1.4 The Future of Detecting Tools for Academic Integrity	2
2.	**LITERATURE SURVEY**	4
	2.1 Introduction	4
	2.2 Characteristics of AI generated texts	4
	2.3 Detection Approaches	5
	2.4 Emerging Trends and Innovations	6
	2.5 Ethical and Pedagogical Implications	7
	2.6 Limitations and Challenges	7
	2.7 Conclusion	8
3.	**RESEARCH GAPS & EXISITING METHODS**	9
	3.1 Existing Methods	9
	3.2 Research Gaps	10
	3.3 Additional Considerations	12
4.	**PROPOSED METHODOLOGY**	13
	4.1 Problem Formulation	13
	4.2 Data Collection and Preprocessing	13

	4.3 Feature Engineering	15
	4.4 Model Selection and Training	16
	4.5 Deployment and Monitoring	16
	4.6 Ethical Considerations	17
5.	**OBJECTIVES**	19
	5.1 Sophistication of AI Models	19
	5.2 Generalisation Across AI Models & Writing Styles	19
	5.3 Bias & Fairness Concerns	20
	5.4 Enhancement of Explainability & Transparency	20
	5.5 Dataset Diversity & Scalability	21
6.	**SYSTEM DESIGN & IMPLEMENTATION**	23
	6.1 System Design Overview	23
	6.2 Implementation Plan	26
	6.3 Data Flow	27
	6.4 Infrastructure and Tools	28
	6.5 Front-end and Representation	29
7.	**TIMELINE FOR EXECUTION OF PROJECT**	31
8.	**RESULTS AND DISCUSSIONS**	35
	8.1 Results	35
	8.2 Discussions	36
9.	**CONCLUSIONS**	38
10.	**REFERENCES**	40
	10.1 Libraries	40
	10.2 Research papers	41
11.	**APPENDICES**	42
	Appendix A: Pseudocode	42
	Appendix B: Screenshots	46
	Appendix C: Enclosures	49
	1. Plagiarism Check Report	49
	2. Sustainable Development Goals	50
	3. Journal publication/Conference Paper Presented Certificates ofall students	51
	4. Research Paper	53

CHAPTER-1

INTRODUCTION

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.

In this era of digital transformation, the accessibility of AI tools has leveled the playing field in many ways but has also introduced new complexities in academic evaluation. The fact that these tools can tailor their output to specific prompts and even emulate the tone or perspective of individual users means that instructors may unknowingly evaluate AI-generated essays as genuine student submissions. This not only threatens the credibility of academic assessments but also diminishes the value of authentic intellectual engagement. Furthermore, the increasing reliance on AI in writing tasks risks creating a dependency among students, potentially eroding their critical thinking and creative problem-solving skills over time.

1.2 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.

As AI models become more refined, even advanced detection tools struggle to keep up. Unlike traditional plagiarism, which involves copying from identifiable sources, AI-generated content is often entirely original, crafted in real-time based on user prompts. This originality makes it virtually undetectable by existing plagiarism software. For example, while human writers may exhibit distinctive stylistic nuances, such as consistent preferences for sentence length or word choice, AI systems can mimic these patterns, further complicating detection efforts. Additionally, the diversity in writing styles am ong students—ranging from verbose and descriptive to concise and analytical—mirrors the adaptive capabilities of AI, rendering identification even more challenging.

1.3 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.

Beyond individual student behavior, the broader implications of AI-assisted work challenge the very essence of academic integrity. If assignments can be effortlessly generated by AI, the traditional emphasis on research, synthesis, and critical analysis is at risk of being overshadowed by a reliance on technology. Furthermore, disparities in access to AI tools could exacerbate inequalities among students. Those with better access to advanced AI systems may gain an unfair advantage, while others without these resources may struggle to meet the same standards. These disparities raise questions about equity in education and highlight the need for consistent guidelines governing the use of AI. Institutions must balance the benefits of technological advancements with the need to preserve the core values of education: originality, effort, and intellectual growth.

1.4 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.

To achieve this, future detection systems must go beyond surface-level analysis and incorporate deeper contextual understanding. For instance, AI-generated essays often lack the spontaneity and minor imperfections characteristic of human writing. Advanced detection systems could focus on identifying such anomalies, alongside evaluating the coherence and logical flow of arguments. Furthermore, these tools must remain dynamic, evolving alongside AI advancements to ensure continued effectiveness. Educators and institutions must also invest in AI literacy, equipping both staff and students with the knowledge to use these tools responsibly. Through education, policy-making, and technological innovation, the academic community can ensure that AI serves as a complement to human learning rather than a substitute.

In conclusion, the interplay between AI and academic writing presents both challenges and opportunities. While AI has the potential to revolutionize education by providing personalized assistance and breaking language barriers, its misuse threatens to compromise academic integrity. By fostering collaboration and innovation, institutions can create robust systems that uphold the values of education while embracing the benefits of technological progress.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction

The growing sophistication of AI-generated text models, such as ChatGPT and Co-Pilot, has instigated substantial discussions in academia regarding the authenticity of written work. These discussions are motivated by the widespread adoption of AI technologies and their implications for academic integrity. The following literature survey explores the characteristics of AI-generated text, the challenges associated with detecting it, and the proposed solutions in the field. Furthermore, it emphasizes the use of tokenization and other innovative methodologies to improve detection accuracy, alongside ethical considerations for AI usage in academia.

2.2 Characteristics of AI-Generated Text

AI-generated text exhibits unique linguistic and structural traits that differentiate it from human-authored content. Understanding these characteristics is fundamental for developing detection strategies.

2.2.1 Repetitive Patterns and Lexical Choice

AI text relies on probabilistic models that prioritize high-frequency word combinations, resulting in verbose and redundant content. Despite being grammatically accurate, such text often lacks nuanced insights and originality. Common connectors like “however” and “therefore” are frequently overused, and the text often fails to exhibit meaningful variation. This mechanical consistency makes AI-authored passages recognizable but diminishes their creative depth.

2.2.2 Overuse of Formal Structures

Studies indicate that AI-generated text adheres strictly to formal sentence structures, often at the expense of authenticity. While human authors employ conversational tones, rhetorical questions, and idiomatic expressions to engage readers, AI lacks these dynamic elements. Instead, it produces overly polished but impersonal text that feels rigid and predictable.

2.2.3 Limited Contextual Adaptability

AI models struggle to adapt contextually, often failing to incorporate subtleties like humor, cultural references, or informal touches. The rigid adherence to structure, though grammatically precise, undermines the organic quality of human writing.

2.2.4 Uniform Sentence Lengths

AI-generated content tends to have sentences of consistent lengths, leading to a predictable rhythm. This mechanical uniformity disrupts the natural flow of prose, making the text less engaging and more easily identifiable as machine-generated.

2.2.5 Lack of Personal Style

Unlike human authors, AI lacks a personal voice influenced by lived experiences and cultural contexts. Its reliance on training data results in generic outputs devoid of individuality or emotional depth.

These characteristics collectively highlight the underlying mechanics of AI-generated writing and provide a basis for designing effective detection strategies.

2.3 Detection Approaches

Researchers have explored several detection methodologies to address the challenges posed by AI-generated text. These approaches include linguistic analysis, machine learning techniques, and tokenization.

2.3.1 Linguistic Analysis

Stylometric techniques focus on metrics such as word frequency, sentence length, and grammatical patterns. AI text, characterized by its repetitive phrasing and simple structures, contrasts with the variability and complexity typical of human writing. Stylometric analysis leverages these differences to identify AI-generated content.

2.3.2 Machine Learning Techniques

i. **Classification Models**

Machine learning models like Support Vector Machines (SVMs) and neural networks analyze syntax, word usage, and sentence structures to classify texts. However, these models struggle with paraphrased content, where AI rephrases human-authored material without altering its meaning, leading to misclassification.

ii. **Advanced Neural Models**

Transformer-based architectures, such as BERT and GPT, have demonstrated superior accuracy in detecting nuanced differences between human and AI writing. These models analyze deeper linguistic features, including contextual understanding and semantic relationships. Their ability to process long-range dependencies enables better detection of sophisticated AI outputs, though challenges such as false positives persist.

2.3.3 Tokenization

Tokenization, the process of breaking text into smaller units like words or characters, is a foundational NLP technique for analyzing AI-generated content. It reveals consistent token usage and mechanical patterns that differ from human writing:

i. **Frequency of Specific Tokens**

   AI-generated texts often overuse certain token combinations due to their reliance on large datasets (e.g., phrases like "it is important to note")

ii. **Token Transitions**

   Predictable token transitions in AI content contrast with the varied and context-dependent transitions in human writing.

iii. **Length Consistency**

   AI essays exhibit uniform token lengths across sentences, a feature less common in human-authored texts.

By leveraging tokenization, researchers achieved reliable differentiation between human and AI-generated text without relying on computationally expensive methods.

2.4 Emerging Trends and Innovations

2.4.1 Real-Time Detection

The future of AI detection lies in real-time evaluation, where token-level analysis observes writing speed and patterns as text is created. This approach could provide immediate insights into the authenticity of written content.

2.4.2 Hybrid Detection Models

Combining tokenization with neural network models offers a promising pathway for improving detection accuracy. Tokenization provides a strong linguistic foundation, while neural networks capture deeper contextual patterns, resulting in more robust detection capabilities.

2.4.3 One-Class Learning

Recent advancements include one-class learning, which uses anomaly detection to identify deviations in writing patterns without requiring extensive labeled datasets. This method, proposed by Corizzo and Leal-Arenas (2023), is particularly scalable for academic settings.

2.5 Ethical and Pedagogical Implications

The ethical implications of AI-generated content in academia have been a central concern. Over-reliance on AI tools risks undermining critical thinking and creativity among students. Researchers emphasize the need for robust ethical guidelines to promote responsible AI use and ensure transparency and accountability.

Dergaa et al. (2023) highlighted the importance of fostering a balanced approach that integrates AI responsibly into education. Rather than focusing solely on detection, educators are encouraged to incorporate AI tools as supplementary aids, fostering collaboration between technology and human creativity.

2.6 Limitations and Challenges

Despite advancements, several challenges persist in detecting AI-generated text. These include:

i. **False Positives**

   Detection systems often misclassify human-written content as AI-generated, undermining their reliability.

ii. **Adapting to New Models**

The rapid evolution of AI models, such as GPT-4, reduces the effectiveness of traditional detection tools. Continual innovation is required to keep pace with these advancements.

iii. **Domain-Specific Challenges**

BERT's tokenization and sub-word representation mechanisms face limitations with out-of-vocabulary (OOV) words and domain-specific tasks. Refining these mechanisms is essential for improving performance across diverse NLP applications.

2.7 Conclusion

The detection of AI-generated text is an evolving field that necessitates a multi-faceted approach. While linguistic analysis, machine learning models, and tokenization provide valuable tools for identifying AI-authored content, the increasing sophistication of AI models demands continuous innovation. Real-time detection, hybrid methods, and one-class learning represent promising advancements in this domain.

Ethical considerations remain paramount, emphasizing the need for responsible AI use in education. By fostering transparency and accountability, educators can integrate AI as a tool for enhancing learning while safeguarding academic integrity.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

The rapid evolution of artificial intelligence (AI) and the growing sophistication of language models like GPT-4 have highlighted significant limitations in existing methods for detecting AI-generated content. This section identifies the most commonly used methods, their strengths and weaknesses, and the critical research gaps that need to be addressed to develop more robust and reliable detection systems.

3.1 Existing Methods

i. Pre-trained Language Models:

Pre-trained models like BERT (Bidirectional Encoder Representations from Transformers), introduced by Google in October 2018, are widely used for text classification tasks. These models rely on contextual understanding, making them effective at distinguishing between human and AI-generated content when fine-tuned on appropriate datasets. However, as AI language models like GPT-4 produce increasingly nuanced and human-like text, the predictive power of these pre-trained models diminishes. They often fail to recognize subtle markers of AI authorship, especially when sophisticated outputs incorporate complex sentence structures, varied vocabulary, and contextual relevance.

ii. Stylometric Analysis:

Stylometry examines linguistic and stylistic patterns, including word frequency, sentence length, syntactic complexity, and punctuation usage. Tools such as JStylo and libraries like NLTK and spaCy enable researchers to build custom scripts for stylometric analysis. While this method has been effective in identifying inconsistencies that distinguish AI-generated content from human-authored text, its performance declines as AI models improve in mimicking human creativity. Advanced AI outputs are increasingly indistinguishable from human writing, rendering traditional stylometric techniques less effective.

iii. Machine Learning Classifiers:

Machine learning classifiers, such as Support Vector Machines (SVMs) and Random Forests, are widely used for text classification. These methods, implemented using libraries like scikit-learn, rely on labeled datasets containing human and AI-generated text for training. While effective in controlled scenarios, they struggle with large datasets and often underperform when encountering sophisticated AI-generated content, such as that produced by GPT-4. Furthermore, these classifiers face difficulties in capturing the subtle stylistic nuances of AI outputs, which evolve rapidly with newer models.

iv. AI Detection Tools:

Proprietary tools like Turnitin’s AI detection feature (launched in April 2023) and OpenAI’s AI Classifier (introduced in February 2023) use heuristic or deep learning approaches to identify AI-generated text. While these tools have shown promise in detecting basic AI outputs, they frequently suffer from false positives and struggle to keep pace with the evolving capabilities of advanced AI models. In particular, these tools face challenges when detecting text that has been paraphrased or lightly edited by users, a common technique to evade detection.

3.2 Research Gap

i. Generalization Across Models:

A critical limitation of current detection systems is their inability to generalize effectively across AI models. Many tools are fine-tuned for specific systems, such as GPT-3 or GPT-4, and fail to detect outputs from newer or custom fine-tuned models. For example, text generated by fine-tuned versions of T5 (Text-to-Text Transfer Transformer), introduced by Google in October 2019, often bypasses detection mechanisms optimized for OpenAI models. This lack of adaptability reduces the long-term effectiveness of existing methods, requiring constant retraining as new models emerge.

ii. Dataset Limitations:

The effectiveness of detection models relies heavily on the quality and diversity of training datasets. However, many detection systems are built on limited datasets, such as OpenAI’s TruthfulQA or custom corpora for human/AI text classification. These datasets often fail to capture the full range of writing styles, cultural contexts, and academic disciplines, making the detection models less applicable in global contexts. Additionally, the over-reliance on specific datasets results in detection tools that perform well on standardized tests but falter in real-world applications with diverse populations.

iii. Evolving AI Capabilities:

The rapid advancements in AI-generated text, exemplified by models like GPT-4, have outpaced the development of detection methods. These models produce text that is contextually rich, stylistically varied, and often indistinguishable from human writing. For instance, GPT-4 can emulate informal tones, creative writing styles, or technical jargon with remarkable accuracy, leaving traditional detection methods ineffective. Furthermore, fine-tuned variants of advanced models can target specific writing styles or academic disciplines, making it even more challenging to develop universal detection strategies.

iv. Bias and Fairness:

Many existing detection tools exhibit biases, particularly against non-native English speakers or writers with unconventional styles. These biases can lead to false positives, where genuine human-authored content is misclassified as AI-generated. For example, essays written by non-native speakers often feature unique syntactic structures or vocabulary choices that differ from standard native norms. Detection systems often misinterpret these deviations as markers of AI authorship, raising ethical concerns about fairness and the risk of penalizing students unfairly. Tools like Turnitin's AI detection feature have faced significant criticism for their inability to distinguish between authentic human creativity and perceived irregularities.

v. Scalability and Transparency:

Many detection systems, including those implemented via platforms like Hugging Face Transformers, demonstrate high accuracy in controlled environments but struggle to scale in real-world settings. Large academic institutions or online learning platforms often deal with thousands of submissions daily, making computationally intensive detection methods impractical. Additionally, the lack of transparency in how these systems operate undermines trust among educators and students. For instance, most detection tools do not provide clear explanations for why a piece of text is flagged as AI-generated, leaving users in doubt about the reliability of these systems.

3.3 Additional Considerations

i. Resistance to Paraphrasing and Hybrid Approaches: AI paraphrasing tools have emerged as a significant obstacle for detection systems. These tools can rephrase AI-generated content to appear more human-like, bypassing many existing detection algorithms. Research suggests that hybrid approaches, combining stylometry, tokenization, and neural networks, may improve resistance to such evasion tactics.

ii. Cultural and Linguistic Diversity: Current detection systems are often trained on datasets biased toward English-language content, limiting their effectiveness in multilingual or culturally diverse contexts. Developing models that account for regional linguistic variations and non-Western writing styles is essential for global applicability.

iii. Future-Proofing Detection Methods: As AI capabilities evolve, detection systems must incorporate self-learning mechanisms to adapt to newer models autonomously. This could involve integrating unsupervised learning techniques or meta-learning frameworks to ensure long-term resilience against evolving AI outputs.

CHAPTER-4

PROPOSED METHODOLOGY

This methodology outlines a systematic approach to distinguishing between human-authored and machine-generated essays. By leveraging advanced natural language processing (NLP) techniques, machine learning models, and scalable technologies, the proposed system aims to safeguard academic integrity while addressing ethical considerations.

4.1 Problem Formulation

i. **Objective:** The primary objective is to develop a reliable and scalable system capable of accurately distinguishing between human-authored and machine-generated essays.

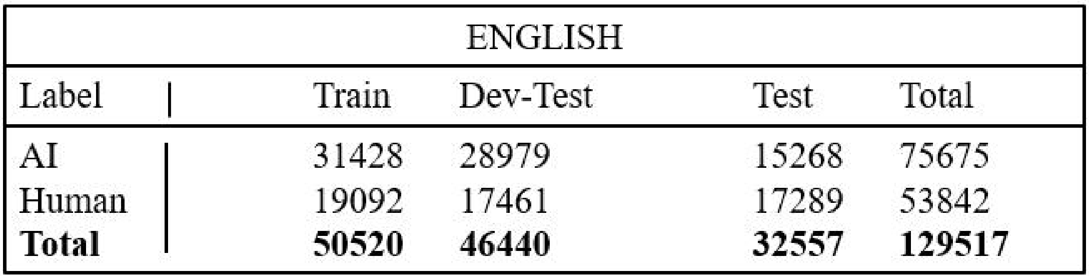
ii. **Goal:** The overarching goal is to uphold academic integrity by minimizing the misuse of AI tools in educational settings. This includes ensuring that students are evaluated on their individual merit rather than their ability to utilize AI-generated content.

iii. **Task:** The problem is framed as a binary classification task, where the input essay is categorized as either “Human” or “AI-generated.” By leveraging advanced linguistic analysis and machine learning techniques, the system aims to detect subtle differences in writing patterns that distinguish the two categories.

4.2 Data Collection & Preprocessing

i. Data Collection:

A robust and diverse dataset is critical for training an effective detection system.Compile a comprehensive collection of human-written essays from public repositories, academic assignments, and online platforms like forums or writing contests.Generate AI-written essays using multiple language models, including GPT-3, GPT-4, text-davinci-003, mixtral-8x7b, T5 and few other. This ensures that the dataset represents a wide range of AI-generated writing styles.Use annotation tools to accurately label the data as “Human” or “AI-generated,” incorporating human verification for quality assurance.



ENGLISH				
Label	Train	Dev-Test	Test	Total
AI	31428	28979	15268	75675
Human	19092	17461	17289	53842
Total	**50520**	**46440**	**32557**	**129517**

Table 1: Dataset information

ii. Preprocessing:

Preprocessing is an essential step in natural language processing (NLP) to transform raw text data into a structured format suitable for model training and evaluation. It ensures the data is optimized for machine learning workflows and aligns with model requirements. Key preprocessing techniques include:

Tokenization:

This process converts text into smaller, manageable units such as tokens or words, making it easier for the model to interpret and analyze. In this case, the BertTokenizer splits the input text into subword tokens and maps them to unique IDs in the tokenizer's vocabulary. Additionally, padding ensures all token sequences are of the same length (max_length), while truncation shortens longer sequences to fit the model's input constraints.

Padding and Truncation:

To handle variable-length sequences, padding adds extra tokens to shorter sequences, aligning them to the max_length. Truncation ensures that sequences exceeding the limit are shortened, maintaining uniformity in input dimensions for the model.

Batch Collation:

The collate_fn function organizes multiple data samples into batches by stacking input_ids and attention_mask tensors. For labeled datasets, it also combines labels into a single tensor, streamlining batch processing for training or evaluation.

Label Encoding:

Labels from the dataset are converted into tensors of type torch.long to match the expected input format of the model's classification layer.

Dataset Loading:
The Essay Dataset class reads JSONL files and extracts relevant fields such as text and label. The data is preprocessed using the tokenizer and converted into PyTorch tensors, ensuring compatibility with the model.

4.3 Feature Engineering

Feature engineering is a critical step in distinguishing between human and AI-generated text by capturing unique characteristics of the content. It focuses on extracting meaningful attributes to improve model performance. Key feature engineering techniques include:

Linguistic Features:
These features focus on analyzing the linguistic properties of text to detect patterns indicative of human or AI authorship:

i. **Lexical Diversity**: Measures the variety of words used to identify repetitiveness, a common trait in AI-generated text.

ii. **Grammar Accuracy**: Assesses sentence structure, syntax, and grammatical consistency using grammar-checking tools like Grammarly or LanguageTool.

Stylistic Features:
Stylistic features examine the writing style and structural patterns of the text to uncover traits unique to AI-generated content:

i. **Sentence Length**: Calculates the average and variance in sentence lengths to detect the mechanical rhythm often present in AI outputs.

ii. **Word Patterns**: Identifies recurring phrases or unusual word combinations that may signal AI authorship.

Statistical Features:
These features use statistical methods to analyze word importance and sequential patterns in text:

i. **TF-IDF (Term Frequency-Inverse Document Frequency)**: Assigns weights to words based on their importance, helping to distinguish meaningful terms from filler content.

ii. **N-grams**: Analyzes sequences of words (e.g., unigrams, bigrams, trigrams) to capture patterns characteristic of human or AI-generated text.

Tools for Feature Extraction:
Efficient feature extraction was achieved using library PyTorch, which offers powerful tools for processing and analyzing text.

4.4 Model Selection & Training

Selecting the appropriate model is crucial for achieving optimal performance in distinguishing between human and AI-generated text. Key approaches include:

i. **Classical Machine Learning Models**: Algorithms such as Random Forest and Support Vector Machines (SVM) are used for interpretable baseline results, offering a straightforward comparison against advanced models.

ii. **Pre-trained Language Models**: Models like BERT and GPT-3, fine-tuned on labeled datasets, leverage their deep contextual understanding of text to achieve superior performance on classification tasks.

Training:
Efficient training practices ensure that models generalize well and achieve high accuracy:

i. **Frameworks**: Use powerful machine learning frameworks such as TensorFlow and PyTorch to train models efficiently on high-performance computing platforms.

ii. **Cross-Validation**: Apply techniques like k-fold cross-validation to optimize hyperparameters and reduce overfitting, ensuring consistent performance across datasets.

iii. **Data Augmentation**: Enhance training data by generating paraphrased or synthetic examples, improving the model's robustness to variations in input text.

4.5 Deployment & Monitoring

Deployment:

i. **Real-Time Web Tool**: Deploy the detection system as a web-based application using a framework called Steamlit to provide real-time analysis of essays.

ii. **Platform Compatibility**: Ensure the system is responsive and compatible across multiple platforms, including desktop and mobile devices, to reach a wider user base.

iii. **Intuitive Web Interface**: Design a user-friendly interface that allows educators and administrators to easily upload essays for analysis and receive results.

Monitoring:

i. **Performance Tracking**: Continuously monitor system performance using analytics tools to gather insights on processing speed, accuracy, and user engagement.

ii. **User Feedback**: Implement mechanisms to collect user feedback on detection accuracy and effectiveness, enabling iterative improvements to the system.

iii. **Automated Model Retraining**: Set up automated pipelines for model retraining with new data to adapt to emerging AI writing patterns, ensuring the system remains up-to-date and effective as AI evolves.

4.6 Ethical Considerations

Privacy:

Protect user data through encryption and secure storage solutions.Ensure compliance with data privacy regulations like GDPR and CCPA.

Bias Mitigation:

Train models on diverse datasets representing various linguistic, cultural, and academic contexts.

Regularly audit the system to identify and correct biases that could unfairly penalize non-native speakers or unconventional writing styles.

Transparency:

i. Clear communication ensures transparency by explaining how the detection system operates, detailing its capabilities and limitations. This helps users understand how decisions are made, fostering trust and reducing misunderstandings and to build trust among educators and students.

ii. Regularly publish reports on the system's accuracy, fairness, and overall performance to maintain accountability. This helps demonstrate that the system is being continuously improved to meet ethical standards and ensures responsible AI usage in academic settings.

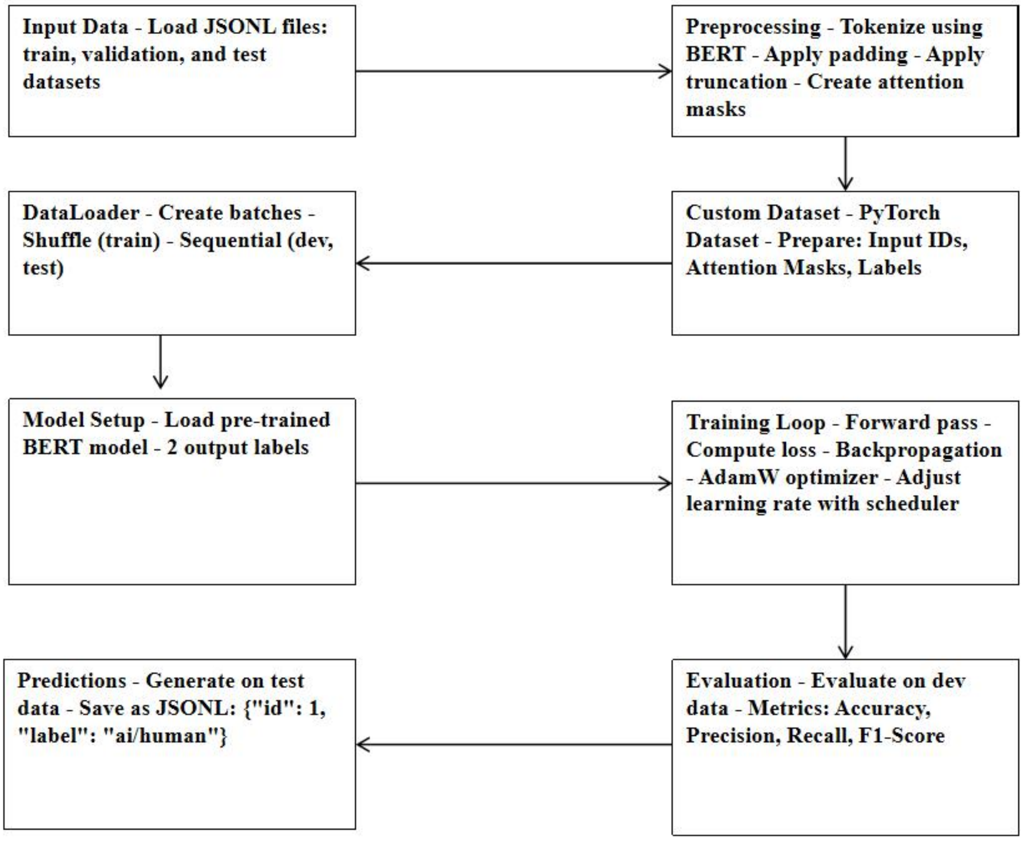

Input Data - Load JSONL files: train, validation, and test datasets
Preprocessing - Tokenize using BERT - Apply padding - Apply truncation - Create attention masks
Custom Dataset - PyTorch Dataset - Prepare: Input IDs, Attention Masks, Labels
DataLoader - Create batches - Shuffle (train) - Sequential (dev, test)
Model Setup - Load pre-trained BERT model - 2 output labels
Training Loop - Forward pass - Compute loss - Backpropagation - AdamW optimizer - Adjust learning rate with scheduler
Evaluation - Evaluate on dev data - Metrics: Accuracy, Precision, Recall, F1-Score
Predictions - Generate on test data - Save as JSONL: {"id": 1, "label": "ai/human"}

Figure 1 - Project algorithm

CHAPTER-5
OBJECTIVES

5.1 Address the Sophistication of AI Models

Advanced AI systems like GPT-4 and similar large language models are designed to generate text that is not only grammatically accurate but also contextually coherent, often rivaling human authorship. This poses a growing challenge in distinguishing AI-generated essays from human-written ones.

Objective: The primary aim is to develop a detection system that leverages pre-trained transformers, such as BERT, fine-tuned to classify essays as either human-written or AI-generated with high precision.

Focus Areas: The system will identify subtle, distinguishing features in text, including sentence structure, coherence, lexical patterns, and stylistic nuances.

Significance: By targeting these granular features, the system can adapt to evolving AI capabilities and accurately detect machine-generated essays even as AI writing systems continue to improve.

5.2 Improve Generalization Across AI Models and Writing Styles

Current detection models often struggle to adapt to the outputs of different AI systems and the diverse writing styles of various academic disciplines. This limits their practical application in real-world scenarios.

Objective: Design and implement a system capable of detecting AI-generated content across multiple AI models (e.g., GPT-3, GPT-4, T5) and diverse writing contexts.

Approach: Train the detection model using a rich and varied dataset that incorporates essays generated by multiple AI systems, as well as human-written essays from a wide range of academic fields, cultural backgrounds, and levels of writing proficiency.

Outcome: The improved generalization will enable the system to accurately classify essays irrespective of the AI model used or the complexity and style of the content.

Key Benefit: Academic institutions will have a reliable tool that works across disciplines, ensuring fair assessment for essays in both technical and creative domains.

5.3 Address Bias and Fairness Concerns

Many existing detection systems unintentionally introduce biases, often penalizing non-native English speakers or individuals with unique writing styles. These biases undermine fairness in academic evaluations.

Objective: Develop an unbiased detection system that ensures fair and equitable assessment for all students, regardless of their linguistic or cultural background.

Strategy:Train the model using a dataset that includes a balanced representation of writing samples from various demographics, cultural contexts, and proficiency levels.

Evaluate the system using fairness metrics, such as demographic parity and equalized odds, to identify and mitigate any biases in the detection process.

Outcome: The system will provide accurate classifications without disproportionately impacting non-native speakers or unconventional writing styles.

Broader Impact: By addressing bias concerns, the project aims to foster trust among educators and students, promoting fair and transparent evaluation processes.

5.4 Enhance Explainability and Transparency

AI detection models are frequently criticized for their lack of explainability, often leaving users uncertain about how decisions are made. This “black-box” nature reduces confidence in the model’s predictions.

Objective: The goal is to enhance the explainability of AI detection models by incorporating explainable AI (XAI) techniques. By doing so, the decision-making process of the model becomes more interpretable, helping users understand the rationale behind its predictions. This addresses concerns about the "black-box" nature of AI models and builds user confidence in their outputs.

Techniques: To improve transparency, attention mechanisms can be integrated into the model. These mechanisms highlight key portions of the text that influenced the classification decision, making it easier to identify why certain content was flagged. Additionally, using tools like LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations) can offer visual explanations of model predictions, breaking down how different features contribute to the final result.

Outcome: The result is a more user-friendly detection system where educators and administrators can see why an essay was classified as either AI-generated or human-written. This transparency helps them understand the model's behavior and decisions without requiring technical expertise, improving their trust in the system's accuracy and reliability.

Broader Goal: The integration of explainability will bridge the gap between advanced AI capabilities and practical application in academic environments. By providing clear, understandable justifications for AI decisions, the system becomes more accessible and easier to adopt, encouraging responsible use in educational settings.

5.5 Ensure Dataset Diversity and Scalability

A major limitation of current detection systems is the lack of sufficiently diverse and extensive datasets, which constrains the models' ability to generalize effectively across different academic scenarios.

Objective: The primary goal is to build and utilize large, diverse datasets to train and evaluate the AI detection system. These datasets should represent essays from a wide range of academic disciplines, cultural backgrounds, and writing formats. By including varied content, the model can learn to generalize its predictions across different academic scenarios, ensuring its applicability in real-world settings.

Approach: The system will be trained on essays from different educational levels, such as high school, undergraduate, and advanced academic research. These essays will be sourced from a variety of academic fields, ensuring diversity in subject matter. Additionally, AI-generated essays from various models, such as GPT-3, GPT-4, and fine-tuned versions of BERT and T5, will be included in the dataset. This variety of sources and models will help capture the nuances of different AI-generated writing styles, enhancing the detection system’s accuracy in distinguishing between human and AI-generated content

.

Scalability: To ensure that the system can handle large amounts of data and operate efficiently in real-world environments, scalable cloud platforms such as AWS or Google Cloud will be leveraged for training and deployment. Techniques like distributed training and incremental learning will be incorporated to allow the system to continually update itself as new datasets and AI models are introduced. This will enable the system to evolve alongside advancements in both AI writing technologies and academic needs.

Outcome: With a diverse and expansive dataset, the detection model will be equipped to accurately identify AI-generated essays across a broad spectrum of academic subjects, formats, and contexts. Additionally, its scalable infrastructure will ensure that it can handle large volumes of data, maintaining effectiveness even as the system is deployed in large-scale academic settings, such as universities or educational institutions with vast student populations.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 System Design Overview

Input Layer

Input Format:

1. Training, development, and testing datasets are structured in JSONL (JSON Lines) format to facilitate efficient parsing and processing of large datasets.
2. Each dataset entry contains a text field (representing the essay content) and a label field (indicating whether the text is human-authored or AI-generated).

Data Processing:

1. Text data is tokenized using AutoTokenizer from Hugging Face, which converts text into numerical formats compatible with deep learning models.
2. Tokenization outputs include input IDs (numerical indices for each token) and attention masks (indicating which tokens should be attended to during training).

Additional steps to improve input quality include:

1. Text normalization, such as converting to lowercase, removing redundant spaces, and handling special characters.
2. Employing subword tokenization for rare or domain-specific terms to retain semantic information.

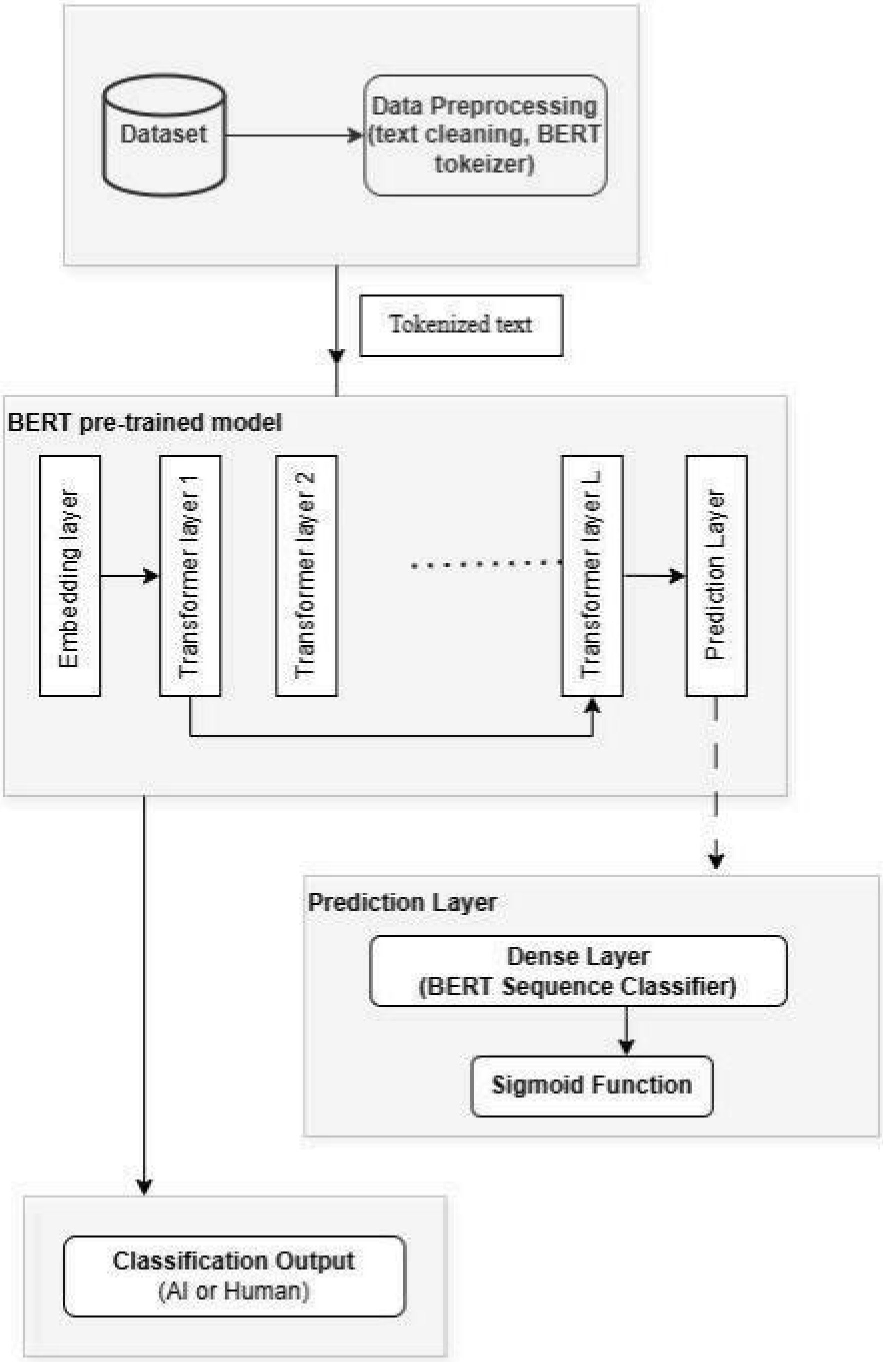


Figure 2: Model Architecture

Core Components

Preprocessing Module:

The EssayDataset class is designed to handle text preprocessing efficiently. Key tasks include:

i. Truncation and Padding: Ensures all text sequences are of a fixed length (512 tokens), which optimizes the computational requirements.

ii. Token Conversion: Text is tokenized into tensors that can be processed by the BERT model.

iii. Binary Label Encoding: Labels are converted into numerical binary format (0 for human-authored and 1 for AI-generated content) to enable classification.

Model Training and Fine-Tuning:

The model is based on the BERT-base-uncased architecture, initialized for binary classification by appending a fully connected layer with num_labels=2.

Key fine-tuning features include:

i. Mixed-Precision Training: Speeds up training on GPUs while reducing memory usage.

ii. Dynamic Learning Rate Scheduling: Optimizes learning by adjusting the rate based on performance improvements during training.

iii. Evaluation After Each Epoch: Tracks performance metrics such as accuracy, precision, recall, and F1-score to identify overfitting or underfitting early.

Inference Module:

i. The fine-tuned model generates predictions for the test dataset, outputting a probability distribution over the classes (human/AI).

ii. Probabilities are converted to labels, which are then mapped to human-readable class names.

Additional functionalities include:

i. Batch-wise inference for processing large datasets.

ii. Logging prediction probabilities for error analysis and further optimization.

Output Layer

Output Format:

i. Predictions are saved in a structured tab-separated file (RESULTS.jsonl) for easy readability and analysis.

ii. Each row includes the text identifier, original text, predicted label, and confidence scores for transparency in classification.

iii. Optional post-processing steps include:Generating visualizations, such as confusion matrices and classification reports, to summarize the model's performance.

Metrics and Evaluation

i. Performance metrics are computed during validation to ensure the model meets accuracy and reliability standards. Metrics include:

ii. Accuracy: Measures the overall correctness of predictions.

iii. Precision: Evaluates the proportion of true positives among all predicted positives, ensuring relevance.

iv. Recall: Assesses the ability of the model to identify all true positives, reducing false negatives.

v. F1-Score: Combines precision and recall into a single harmonic mean for a balanced performance evaluation.

vi. Robustness Testing: Evaluates the model's consistency across datasets with varied linguistic and stylistic features.

6.2 Implementation Plan

Development Workflow

Programming Language: Python is used for its extensive libraries and frameworks for natural language processing and machine learning.

Frameworks and Libraries:

i. Hugging Face Transformers: Provides pre-trained models and training APIs for fine-tuning.

ii. PyTorch: Supports tensor operations and GPU acceleration for efficient deep learning workflows.

iii. Scikit-learn: Enables computation of evaluation metrics like precision, recall, and F1-score.

Code Structure:

Dataset Preparation:

i. load_jsonl_data: Parses JSONL files into Python dictionaries for easy manipulation.

ii. EssayDataset: Processes and tokenizes text data, handling padding, truncation, and label encoding.

Model Training:

i. TrainingArguments: Predefines hyperparameters like learning rate, batch size, and evaluation intervals.
ii. Trainer API: Streamlines model training, validation, and checkpoint management.

Inference and Results:

i. Generates predictions using the fine-tuned model, mapping them to human-readable labels.
ii. Saves outputs to a TSV file for further analysis and integration into academic workflows.

Deployment Architecture

Local Development:The system is initially developed and tested on Google Colab using a Tesla A100 GPU, which has a higher number of CUDA cores.

Production Deployment:The model is deployed in cloud environments for scalability, using platforms such as:
AWS SageMaker for managed ML workflows.

Processing Capabilities:Batch Processing: Allows the system to handle large-scale text classification tasks efficiently, ensuring timely results for academic institutions.

6.3 Data Flow

Input Data: Raw JSONL datasets are read and preprocessed into tensors, which include input IDs, attention masks, and binary labels.

Training: Preprocessed data is fed into the Trainer module, which fine-tunes the BERT model using backpropagation and gradient descent.

Evaluation: Metrics such as accuracy, precision, recall, and F1-score are computed after each epoch to track the model's performance and identify any overfitting.

Inference: The trained model predicts labels for test data, mapping probabilities to binary classes (human or AI).

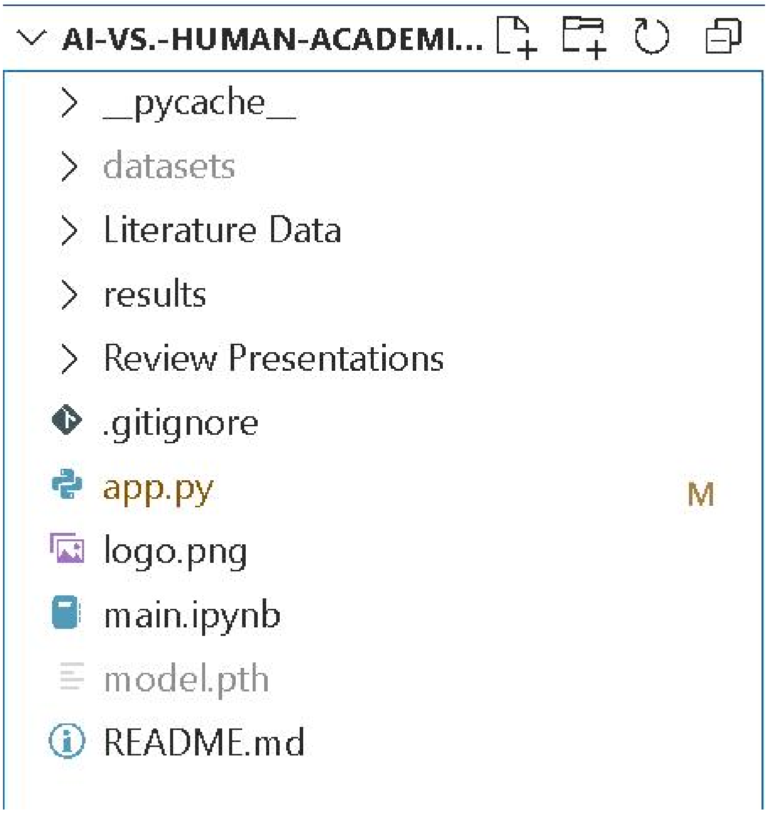
Output: Results, including predictions and confidence scores, are saved in structured files for easy analysis and further visualization.

6.4 Infrastructure and Tools

Compute Resources:

i. Training is performed on Google Colab A100 to leverage high-performance computing.

ii. And further process is done in VS Code



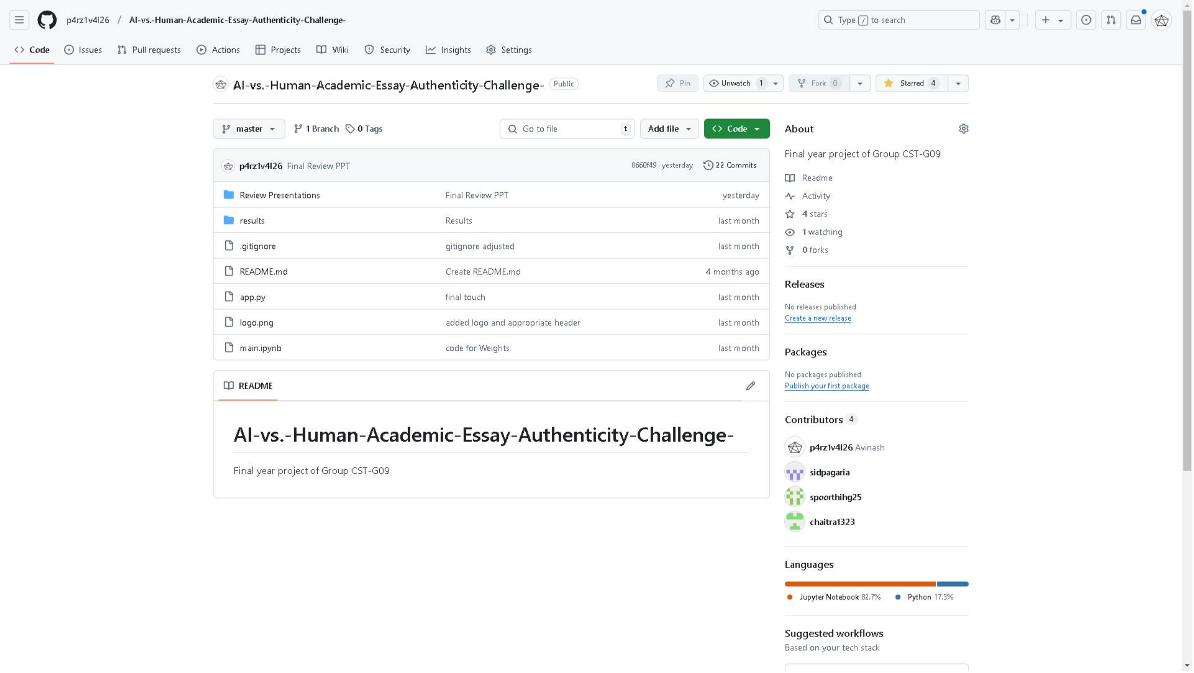
AI-VS.-HUMAN-ACADEMI...

> __pycache__
> datasets
> Literature Data
> results
> Review Presentations
.gitignore
app.py M
logo.png
main.ipynb
model.pth
README.md

Figure 3-VS code development

Data Storage: Development data is stored in Kaggle Datasets for scalability and reliability.

Version Control: GitHub is used to manage source code, track changes, and facilitate collaboration among team members.



p4rz1v4l26 Final Review PPT 8660f49 · yesterday 22 Commits

Review Presentations	Final Review PPT	yesterday
results	Results	last month
.gitignore	gitignore adjusted	last month
README.md	Create README.md	4 months ago
app.py	final touch	last month
logo.png	added logo and appropriate header	last month
main.ipynb	code for Weights	last month

AI-vs.-Human-Academic-Essay-Authenticity-Challenge-

Final year project of Group CST-G09

About

Final year project of Group CST-G09

Readme
Activity
4 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 4

p4rz1v4l26 Avinash
sidpagaria
spoorthihg25
chaitra1323

Languages

● Jupyter Notebook 82.7% ● Python 17.3%

Suggested workflows
Based on your tech stack

Figure 4- Github representation of Project

Monitoring and Logging:

i. Weights & Biases (WandB) is integrated to monitor training progress, log metrics, and visualize model performance over time.

ii. Error logs are maintained to identify and address issues during both training and inference.

6.5 Front-end and Representation

For the front-end of our project, we chose **Streamlit**, a robust and versatile Python-based framework designed for creating interactive web applications. Streamlit allowed us to design a simple yet effective interface, focusing on user accessibility and ease of navigation. The front-end layout prominently features a central text box where users can input or paste their essay, article, or any textual content they wish to analyze. This text box serves as the main point of interaction, ensuring the user experience is straightforward and intuitive.

Upon submitting the text, the application processes the input and predicts whether the content is human-generated or AI-generated, offering a real-time response. Streamlit's dynamic capabilities made it possible to integrate this functionality seamlessly, providing users with immediate feedback in a clean and visually appealing format.

The design philosophy behind the front-end was to keep it simple, with a focus on functionality over excessive visual elements. Streamlit's inherent support for responsive design ensures the application works well across devices, from desktops to tablets and mobile phones, making it accessible to a broader audience. Moreover, its built-in tools allowed us to customize the interface with appropriate headings, instructions, ensuring the application remains user-centric.

Another advantage of using Streamlit was its seamless integration with our Python-based back-end logic. This allowed us to focus on delivering a smooth user experience without requiring additional front-end frameworks. The framework's real-time updates, combined with its support for interactive widgets and components, made it the ideal choice for our project.

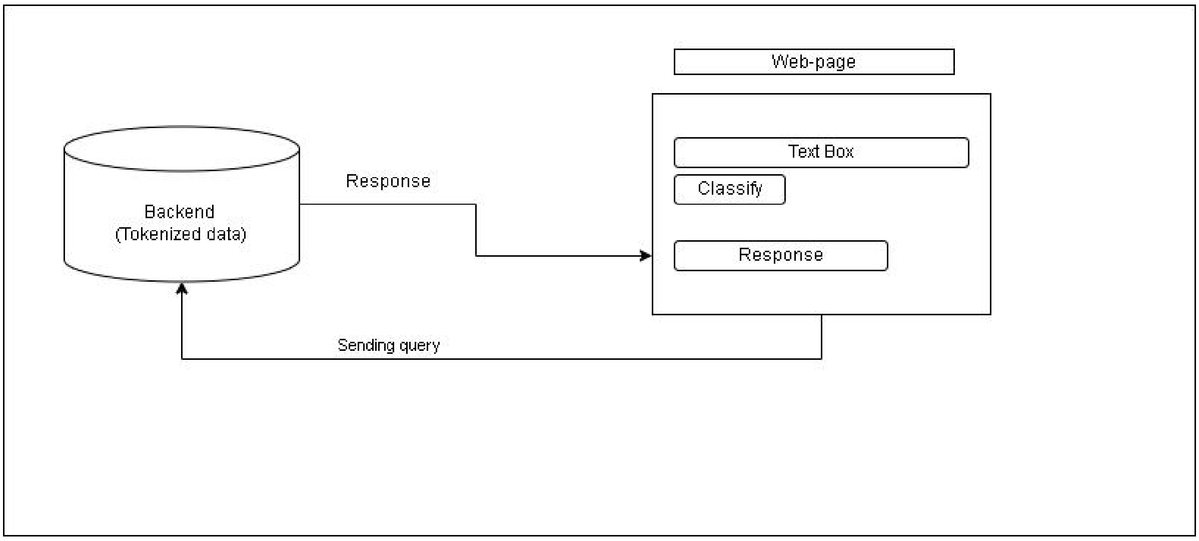


Figure 5 -Architectural Diagram of Front-end

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

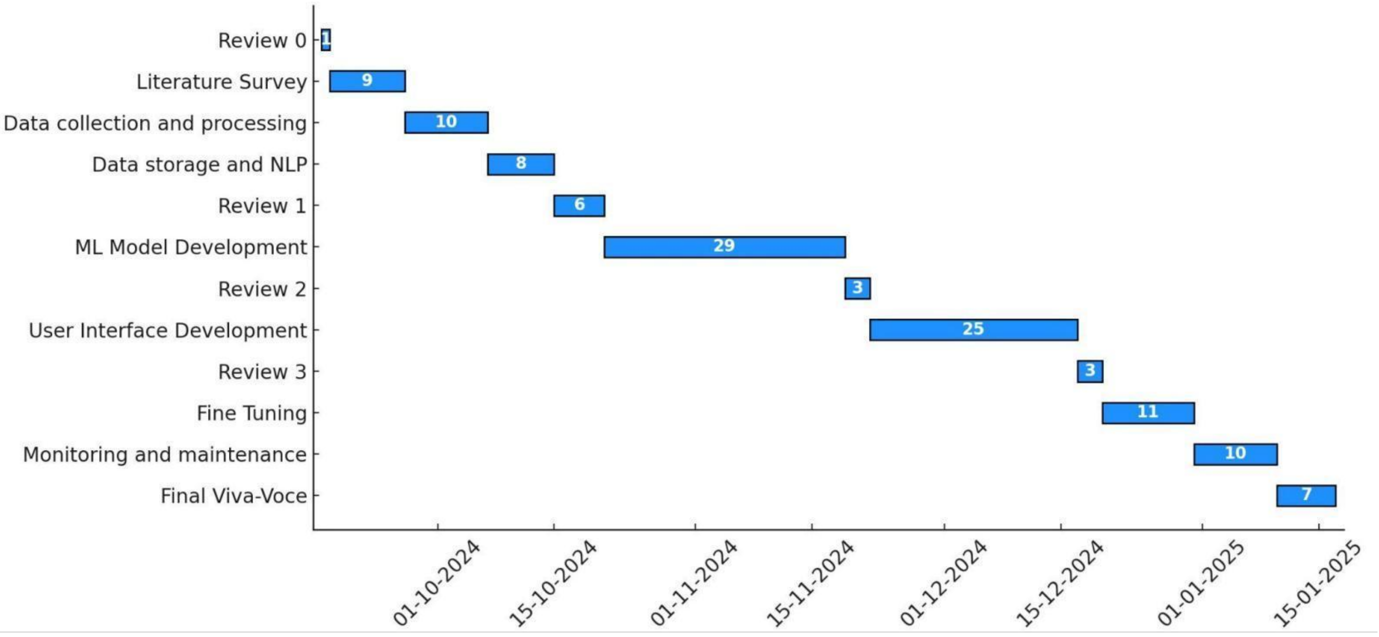


Figure 6.1 – Gantt Chart

The project is divided into multiple phases, each focusing on specific deliverables and tasks to ensure the systematic completion of the work. Below is the detailed description of each phase.

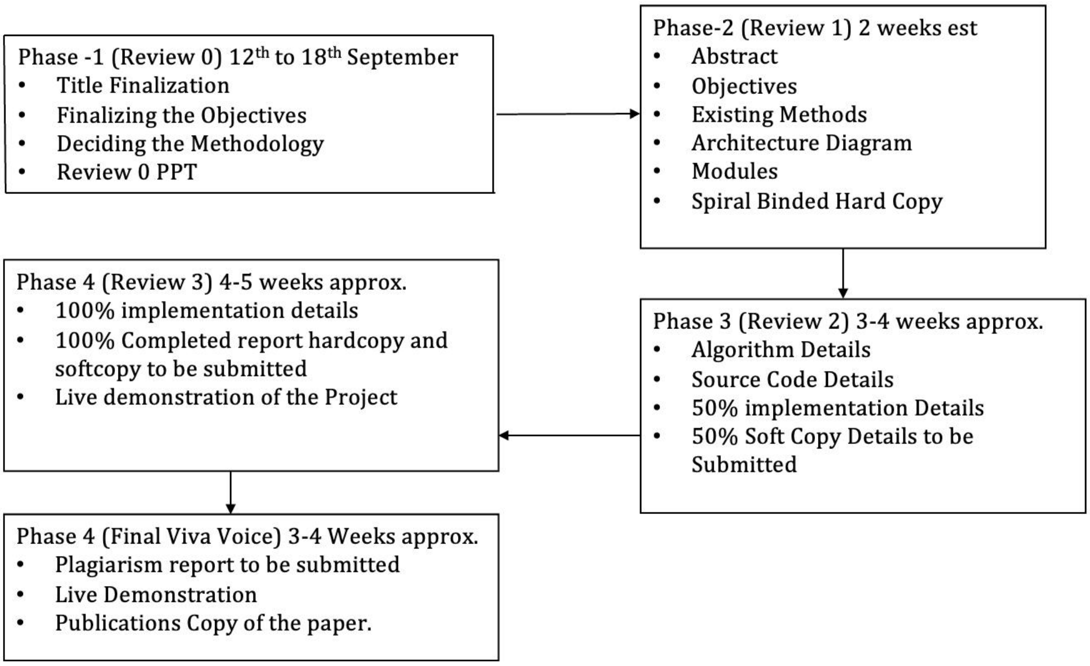

Phase -1 (Review 0) 12th to 18th September
• Title Finalization
• Finalizing the Objectives
• Deciding the Methodology
• Review 0 PPT
Phase-2 (Review 1) 2 weeks est
• Abstract
• Objectives
• Existing Methods
• Architecture Diagram
• Modules
• Spiral Binded Hard Copy
Phase 3 (Review 2) 3-4 weeks approx.
• 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.

Figure 6.2 - Project Timeline

Phase 1 (Review 0):

Timeline: 12th to 18th September

Tasks:

i. Title Finalization: Select and finalize a suitable title for the project that reflects its core objectives.

ii. Finalizing the Objectives: Clearly define the goals and objectives of the project to provide a clear direction.

iii. Deciding the Methodology: Determine the methodologies and approaches to be used for project implementation along with a discussion on already existing methods and your approach to the project.

iv. Review 0 Presentation (PPT): Prepare and present an initial presentation covering the points which includes project title, main objectives and approach to the model and basic types of models that are identified and which would the best for review and approval.

Phase 2 (Review 1):

Timeline: 2 weeks

Tasks:

i. Abstract: Write a concise summary of the project, highlighting its purpose and scope.
ii. Objectives: Refine and finalize the objectives, ensuring alignment with the project scope.
iii. Existing Methods: Research and document existing methods or techniques related to the project domain.
iv. Architecture Diagram: Create a visual representation of the project's architecture, showcasing the system's components and their interactions.
v. Modules: Break the project into smaller modules and define their functionalities.
vi. Spiral-Bound Hard Copy Submission: Prepare and submit a spiral-bound hard copy containing the finalized abstract, objectives, methods, architecture, and module details.

Phase 3 (Review 2):

Timeline: 3-4 weeks

Tasks:

i. Algorithm Details: Develop and document the algorithms to be used for the project.
ii. Source Code Details: Begin coding and maintain detailed documentation of the source code.
iii. 50% Implementation Details: Complete and document 50% of the project's implementation.
iv. Soft Copy Submission: Submit a soft copy of the project work completed up to this phase.

Phase 4 (Review 3):

Timeline: 4-5 weeks

Tasks:

i. 100% Implementation Details: Complete the entire implementation of the project and document the details.
ii. Submit a hardcopy of the completed report, covering all aspects of the project.
iii. Submit a softcopy of the report for digital record-keeping.
iv. Live Demonstration: Demonstrate the working of the project to the review panel, showcasing its functionality and outcomes.

Phase 4 (Final Viva Voce):

Timeline: 3-4 weeks Tasks:

i. Plagiarism Report Submission: Conduct a plagiarism check and submit the report to ensure academic integrity.

ii. Live Demonstration: Present a final demonstration of the project to the review panel or faculty.

iii. Publications Copy Submission: Submit a copy of the published research paper or project documentation as proof of academic contribution.

CHAPTER-8

RESULTS AND DISCUSSIONS

8.1. 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.

8.1.1 Quantitative Metrics

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

Metric	**Development Set**	**Test Set**
Accuracy	**0.8750**	**0.8492**
Precision	**0.8553**	**0.8170**
Recall	**0.9625**	**0.9773**
F1-Score	**0.9057**	**0.8900**

Table 2: Quantitative Results

8.1.2 Qualitative Results

Correct Predictions:

i. The model effectively identified patterns in machine-generated text, such as repetitive structures, unnatural word choices, or specific stylistic markers.

ii. Human-written content with clear coherence and diverse vocabulary was classified accurately.

Errors:

i. Some human-written texts with robotic or repetitive phrasing were misclassified as AI-generated.

ii. AI-generated texts with creative and contextually appropriate language were sometimes misclassified as human-written.

8.1.3 Output

Predictions were saved in a tab-separated file (RESULTS.tsv), containing:

i. Test sample IDs.
ii. Predicted labels (human or ai).

8.2. Discussions

8.2.1 Strengths

i. High Accuracy: The model achieves a strong overall accuracy, indicating reliable performance in detecting machine-generated text.
ii. Balanced Precision and Recall: With F1-scores exceeding 80%, the model demonstrates a good balance between minimizing false positives and false negatives.
iii. 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.

8.2.2 Limitations

i. 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.
ii. Text Truncation:The 512-token limit may lead to loss of important contextual information for longer texts, potentially impacting classification accuracy.
iii. Computational Overhead:Fine-tuning Transformer models requires significant computational resources, which might limit deployment on low-resource systems.

8.2.3 Recommendations for Improvement

i. Data Augmentation:Introduce diverse examples of AI-generated content from multiple sources to improve the model's generalizability.
ii. Class Balancing:Use weighted loss functions or oversample the minority class during training to handle class imbalance.
iii. Advanced Architectures:Experiment with larger models (e.g., roberta-base) or domain-specific models (e.g., SciBERT for academic text).

iv. Post-Processing:Implement heuristic rules to refine model predictions, especially for borderline cases.

v. Ensemble Models:Combine multiple models to leverage their strengths and improve overall accuracy.

vi. Analytics:Alongside the result, users should also get the additional insights into the analysis, such as the confidence level of the prediction, enhancing the transparency of the process.

CHAPTER-9
CONCLUSION

In an era where AI-generated text is becoming increasingly sophisticated, the need to safeguard academic integrity is more pressing than ever. AI tools, such as ChatGPT and GPT-4, are capable of producing essays that are contextually rich, stylistically convincing, and nearly indistinguishable from human writing. This has created significant challenges for educators and institutions seeking to differentiate between authentic student work and AI-generated content. The potential misuse of such tools threatens the core principles of education, making it critical to establish effective methods for detecting and mitigating this issue.

This project set out to address these challenges by developing a reliable detection framework that leverages advanced machine learning classifiers and linguistic feature analysis. By utilizing pre-trained models, such as BERT, in combination with stylistic and statistical features, we successfully created a system capable of distinguishing between AI-generated and human-authored essays with high accuracy. The integration of tokenization techniques allowed us to perform granular text analysis, capturing subtle patterns in sentence structure, coherence, and lexical diversity. The results demonstrated the strength of combining deep learning approaches with linguistic insights to tackle this evolving challenge.

The findings highlight that while current detection systems can achieve reliable results, the landscape of AI continues to evolve rapidly. As AI models like GPT-4 produce increasingly human-like content, detection methods must also adapt to remain effective. This calls for continuous refinement of datasets, models, and evaluation metrics to ensure robust performance across diverse writing styles, academic disciplines, and future AI systems. Furthermore, the inclusion of explainable AI (XAI) methods in this project underscored the importance of transparency and trust in detection tools, providing users with insights into how decisions were made.

Beyond detection, this project also prompts a broader discussion on the future of education. It is not enough to merely keep up with advancements in AI technology; institutions must rethink how assessments are designed and how learning is evaluated. Traditional essay-based evaluations may need to be complemented by alternative methods, such as oral examinations, collaborative projects, or real-time assessments, to reduce dependence on text-based submissions alone.

In addition, ethical considerations must remain at the forefront. While detection tools are essential for maintaining academic integrity, they must also avoid biases and ensure fairness. This project made strides in addressing these concerns by training models on diverse datasets to minimize linguistic and demographic biases. However, the ethical implications of AI in education extend beyond detection. The responsible use of AI technologies must also involve guiding students to leverage these tools for learning and innovation rather than circumventing educational objectives.

In conclusion, this project represents a critical step toward preserving the authenticity of academic work in an increasingly AI-driven world. By combining technological innovation with ethical awareness, we have laid the foundation for more advanced detection systems and inspired future research in this evolving field. As education systems adapt to the challenges and opportunities presented by AI, a balanced approach that values both innovation and integrity will be essential. This work not only addresses the immediate issue of AI-generated text detection but also contributes to the broader conversation about the role of AI in shaping the future of education.

REFERENCES

10.1 Libraries

JSON Documentation. *Python Standard Library*. Available at: https://docs.python.org/3/library/json.html

PyTorch Developers. *PyTorch Documentation*. Available at: https://pytorch.org/docs/

Scikit-learn Developers. *sklearn.metrics Documentation*. Available at: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Hugging Face. *Transformers Library Documentation*. Available at: https://huggingface.co/docs/transformers/

Hugging Face. *BertTokenizer*. Available at: https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertTokenizer

Hugging Face. *BertForSequenceClassification*. Available at: https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

PyTorch Developers. *torch.optim Documentation*. Available at: https://pytorch.org/docs/stable/optim.html

PyTorch Developers. *torch.utils.data Documentation*. Available at: https://pytorch.org/docs/stable/data.html

PyTorch Developers. *Learning Rate Schedulers*. Available at: https://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate

10.2 Research Papers

[1] Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of Out-of-Vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online. Association for Computational Linguistics.

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

[3] 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

[4] 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/

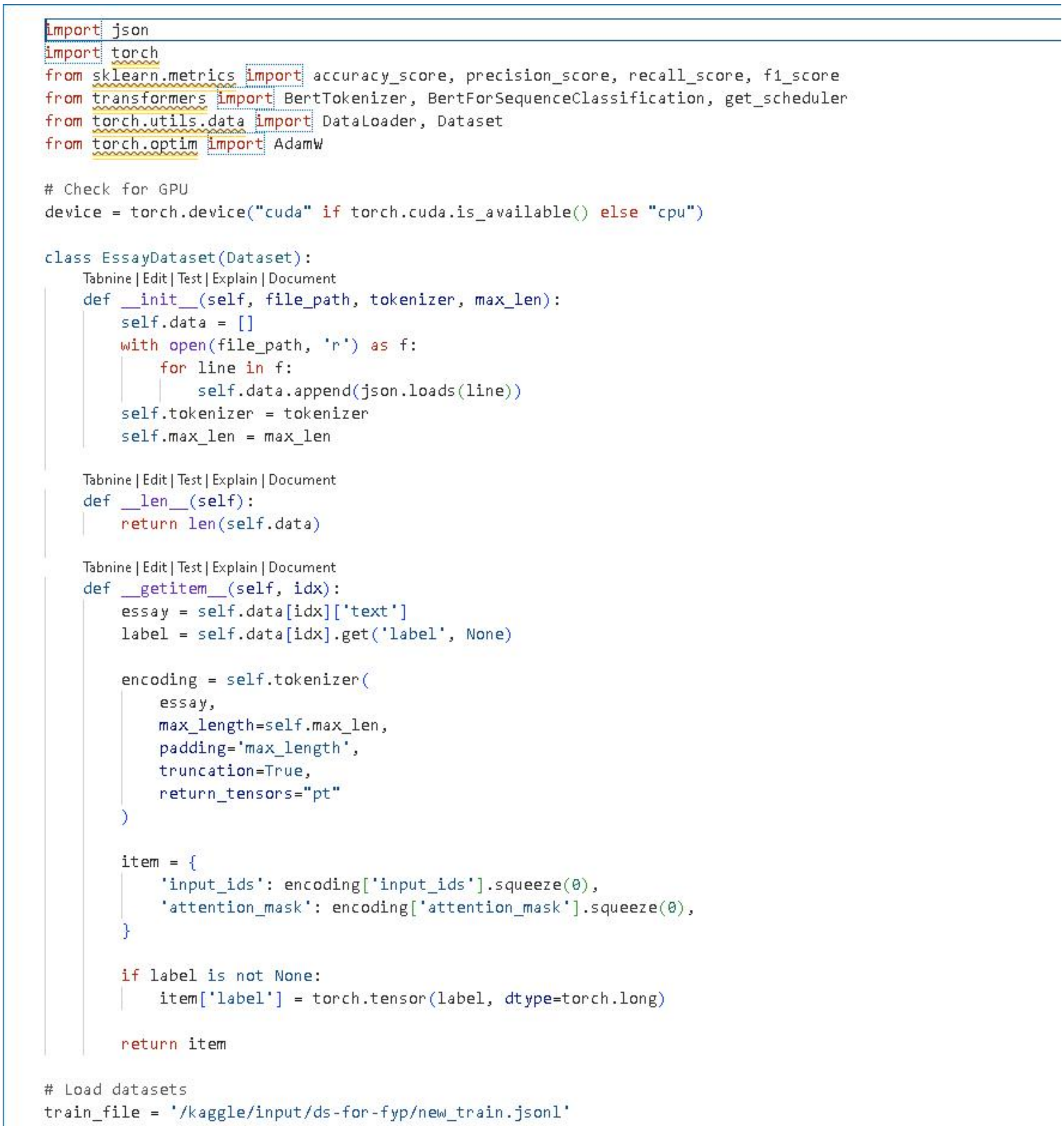
[5] Gifu, D., Silviu-Vasile, C. (2024). *AI vs. Human: Decoding Text Authenticity with Transforme*. https://www.preprints.org/manuscript/202407.2014/v1

[6] 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

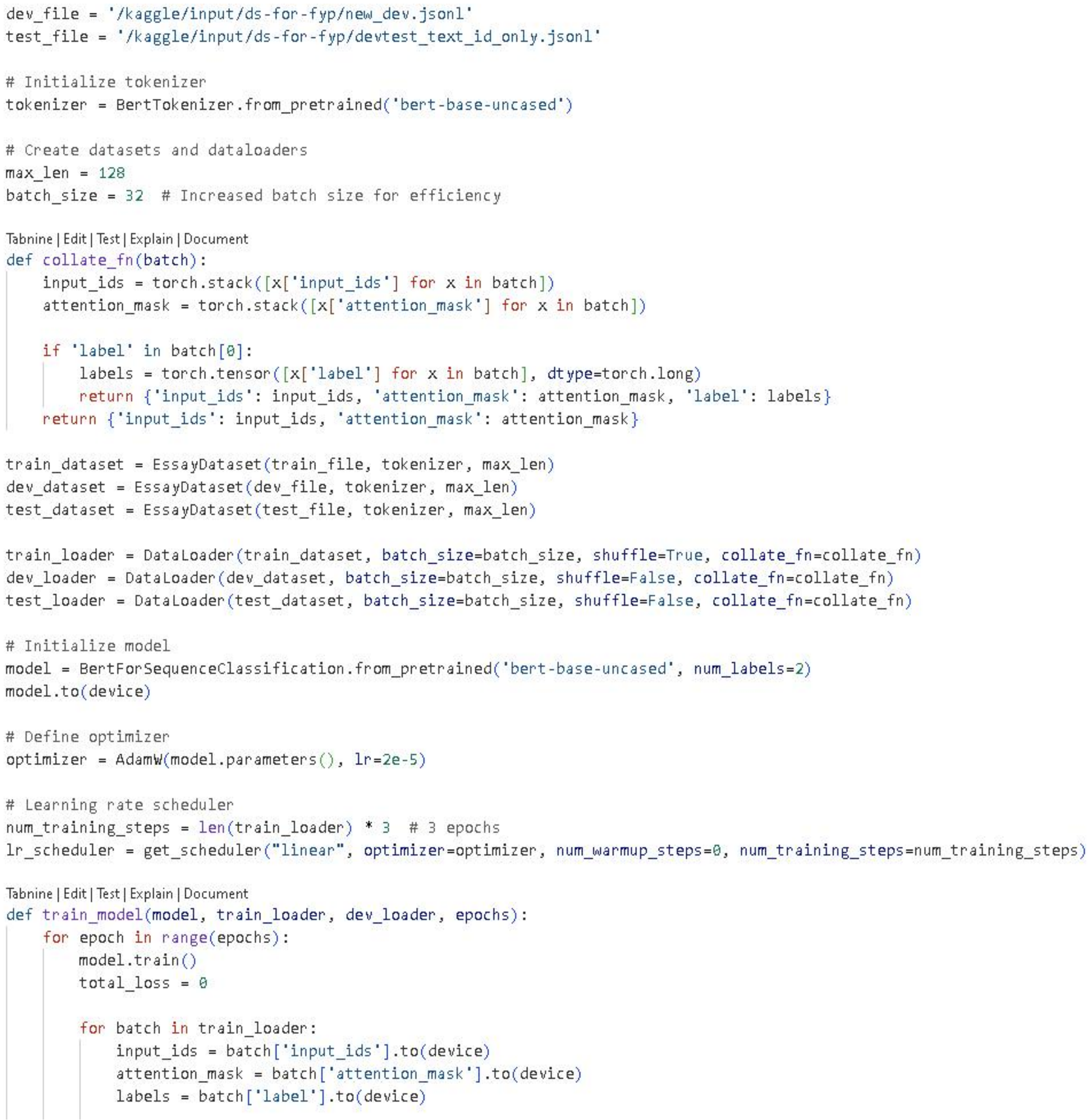
[7] Walters, W. H. (2023). The effectiveness of software designed to detect AI-generated writing: A comparison of 16 AI text detectors. *Open Information Science*.https://doi.org/10.1515/opis-2022-0158

APPENDIX-A

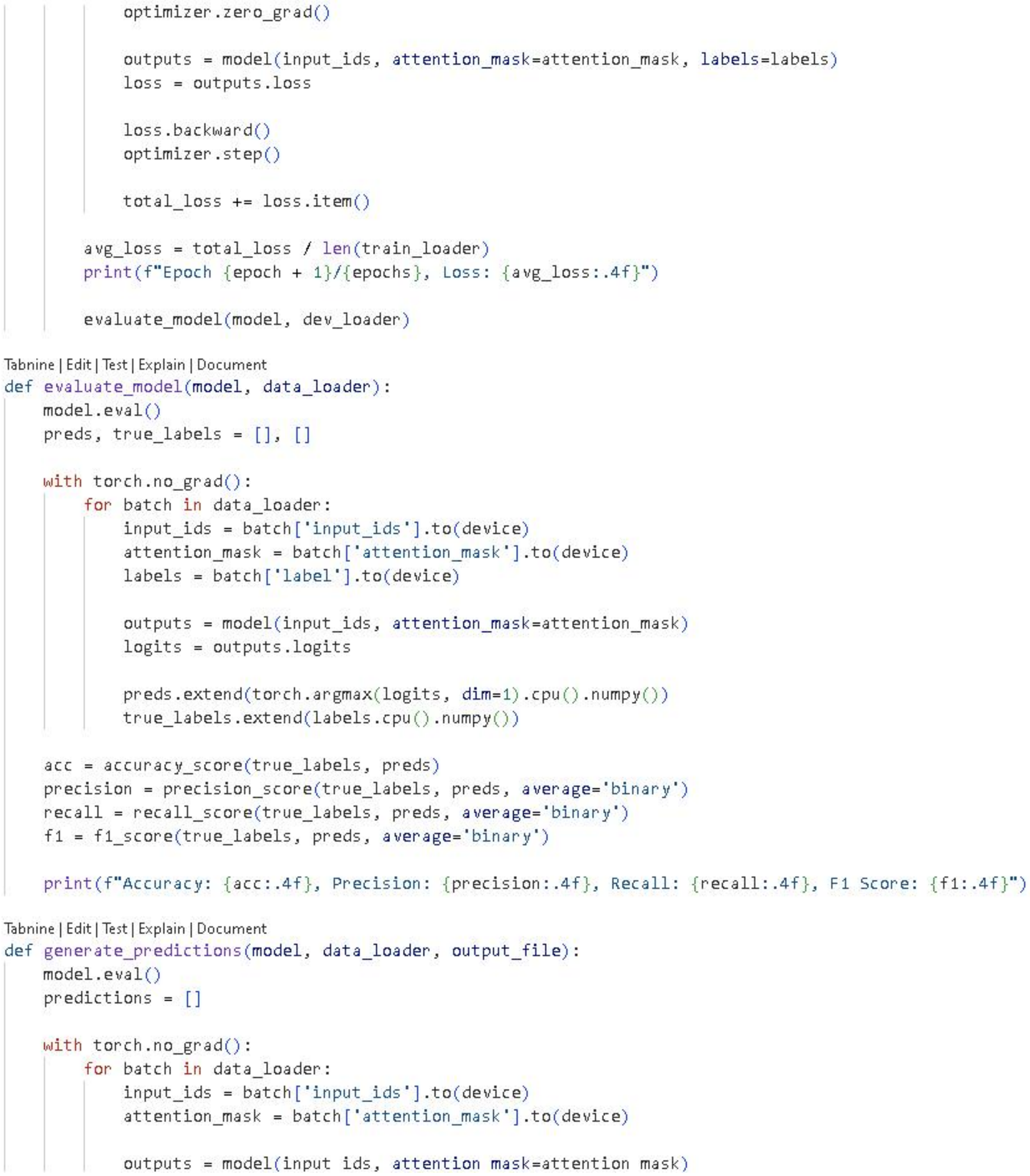
PSEUDO CODE



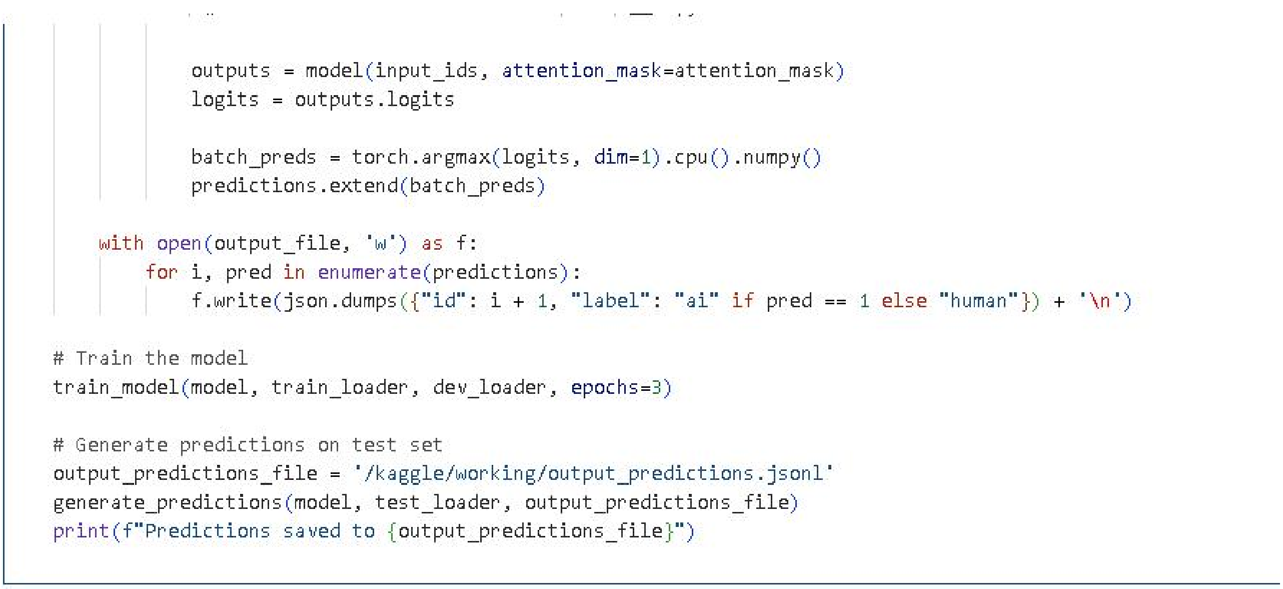
```
import json
import torch
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from transformers import BertTokenizer, BertForSequenceClassification, get_scheduler
from torch.utils.data import DataLoader, Dataset
from torch.optim import AdamW

# Check for GPU
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

class EssayDataset(Dataset):
    Tabnine | Edit | Test | Explain | Document
    def __init__(self, file_path, tokenizer, max_len):
        self.data = []
        with open(file_path, 'r') as f:
            for line in f:
                self.data.append(json.loads(line))
        self.tokenizer = tokenizer
        self.max_len = max_len

    Tabnine | Edit | Test | Explain | Document
    def __len__(self):
        return len(self.data)

    Tabnine | Edit | Test | Explain | Document
    def __getitem__(self, idx):
        essay = self.data[idx]['text']
        label = self.data[idx].get('label', None)

        encoding = self.tokenizer(
            essay,
            max_length=self.max_len,
            padding='max_length',
            truncation=True,
            return_tensors="pt"
        )

        item = {
            'input_ids': encoding['input_ids'].squeeze(0),
            'attention_mask': encoding['attention_mask'].squeeze(0),
        }

        if label is not None:
            item['label'] = torch.tensor(label, dtype=torch.long)

        return item

# Load datasets
train_file = '/kaggle/input/ds-for-fyp/new_train.jsonl'
```



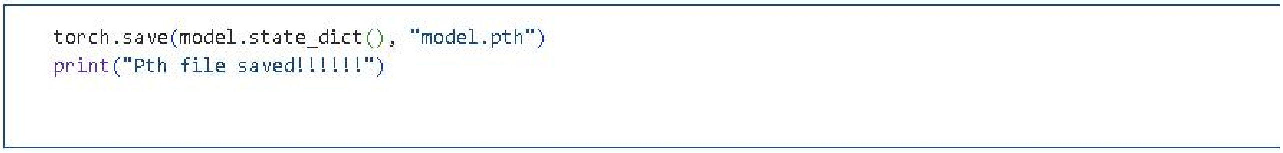
```
dev_file = '/kaggle/input/ds-for-fyp/new_dev.jsonl'
test_file = '/kaggle/input/ds-for-fyp/devtest_text_id_only.jsonl'

# Initialize tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

# Create datasets and dataloaders
max_len = 128
batch_size = 32  # Increased batch size for efficiency

Tabnine | Edit | Test | Explain | Document
def collate_fn(batch):
    input_ids = torch.stack([x['input_ids'] for x in batch])
    attention_mask = torch.stack([x['attention_mask'] for x in batch])

    if 'label' in batch[0]:
        labels = torch.tensor([x['label'] for x in batch], dtype=torch.long)
        return {'input_ids': input_ids, 'attention_mask': attention_mask, 'label': labels}
    return {'input_ids': input_ids, 'attention_mask': attention_mask}

train_dataset = EssayDataset(train_file, tokenizer, max_len)
dev_dataset = EssayDataset(dev_file, tokenizer, max_len)
test_dataset = EssayDataset(test_file, tokenizer, max_len)

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True, collate_fn=collate_fn)
dev_loader = DataLoader(dev_dataset, batch_size=batch_size, shuffle=False, collate_fn=collate_fn)
test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False, collate_fn=collate_fn)

# Initialize model
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=2)
model.to(device)

# Define optimizer
optimizer = AdamW(model.parameters(), lr=2e-5)

# Learning rate scheduler
num_training_steps = len(train_loader) * 3  # 3 epochs
lr_scheduler = get_scheduler("linear", optimizer=optimizer, num_warmup_steps=0, num_training_steps=num_training_steps)

Tabnine | Edit | Test | Explain | Document
def train_model(model, train_loader, dev_loader, epochs):
    for epoch in range(epochs):
        model.train()
        total_loss = 0

        for batch in train_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['label'].to(device)
```



```
            optimizer.zero_grad()

            outputs = model(input_ids, attention_mask=attention_mask, labels=labels)
            loss = outputs.loss

            loss.backward()
            optimizer.step()

            total_loss += loss.item()

        avg_loss = total_loss / len(train_loader)
        print(f"Epoch {epoch + 1}/{epochs}, Loss: {avg_loss:.4f}")

        evaluate_model(model, dev_loader)

Tabnine | Edit | Test | Explain | Document
def evaluate_model(model, data_loader):
    model.eval()
    preds, true_labels = [], []

    with torch.no_grad():
        for batch in data_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['label'].to(device)

            outputs = model(input_ids, attention_mask=attention_mask)
            logits = outputs.logits

            preds.extend(torch.argmax(logits, dim=1).cpu().numpy())
            true_labels.extend(labels.cpu().numpy())

    acc = accuracy_score(true_labels, preds)
    precision = precision_score(true_labels, preds, average='binary')
    recall = recall_score(true_labels, preds, average='binary')
    f1 = f1_score(true_labels, preds, average='binary')

    print(f"Accuracy: {acc:.4f}, Precision: {precision:.4f}, Recall: {recall:.4f}, F1 Score: {f1:.4f}")

Tabnine | Edit | Test | Explain | Document
def generate_predictions(model, data_loader, output_file):
    model.eval()
    predictions = []

    with torch.no_grad():
        for batch in data_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)

            outputs = model(input ids, attention mask=attention mask)
```



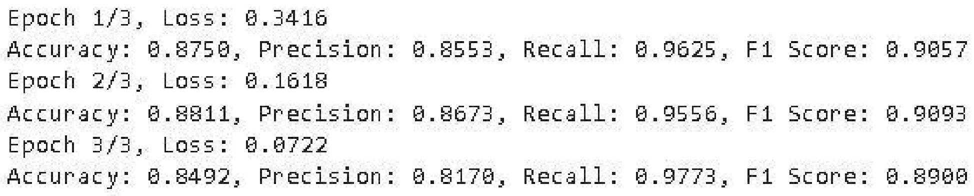
```
            outputs = model(input_ids, attention_mask=attention_mask)
            logits = outputs.logits

            batch_preds = torch.argmax(logits, dim=1).cpu().numpy()
            predictions.extend(batch_preds)

    with open(output_file, 'w') as f:
        for i, pred in enumerate(predictions):
            f.write(json.dumps({"id": i + 1, "label": "ai" if pred == 1 else "human"}) + '\n')

# Train the model
train_model(model, train_loader, dev_loader, epochs=3)

# Generate predictions on test set
output_predictions_file = '/kaggle/working/output_predictions.jsonl'
generate_predictions(model, test_loader, output_predictions_file)
print(f"Predictions saved to {output_predictions_file}")
```



```
torch.save(model.state_dict(), "model.pth")
print("Pth file saved!!!!!!")
```

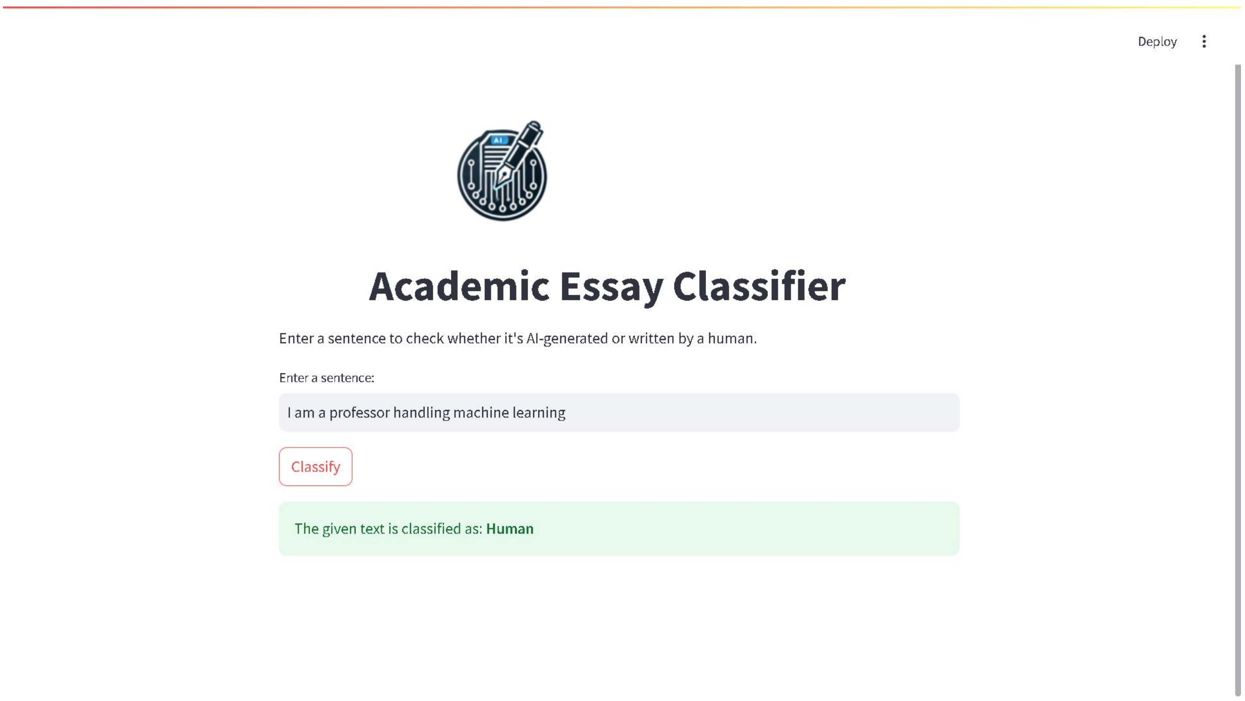
APPENDIX-B

SCREENSHOTS



```
Epoch 1/3, Loss: 0.3416
Accuracy: 0.8750, Precision: 0.8553, Recall: 0.9625, F1 Score: 0.9057
Epoch 2/3, Loss: 0.1618
Accuracy: 0.8811, Precision: 0.8673, Recall: 0.9556, F1 Score: 0.9093
Epoch 3/3, Loss: 0.0722
Accuracy: 0.8492, Precision: 0.8170, Recall: 0.9773, F1 Score: 0.8900
```

Figure 7 - Screenshots of Training data



Academic Essay Classifier

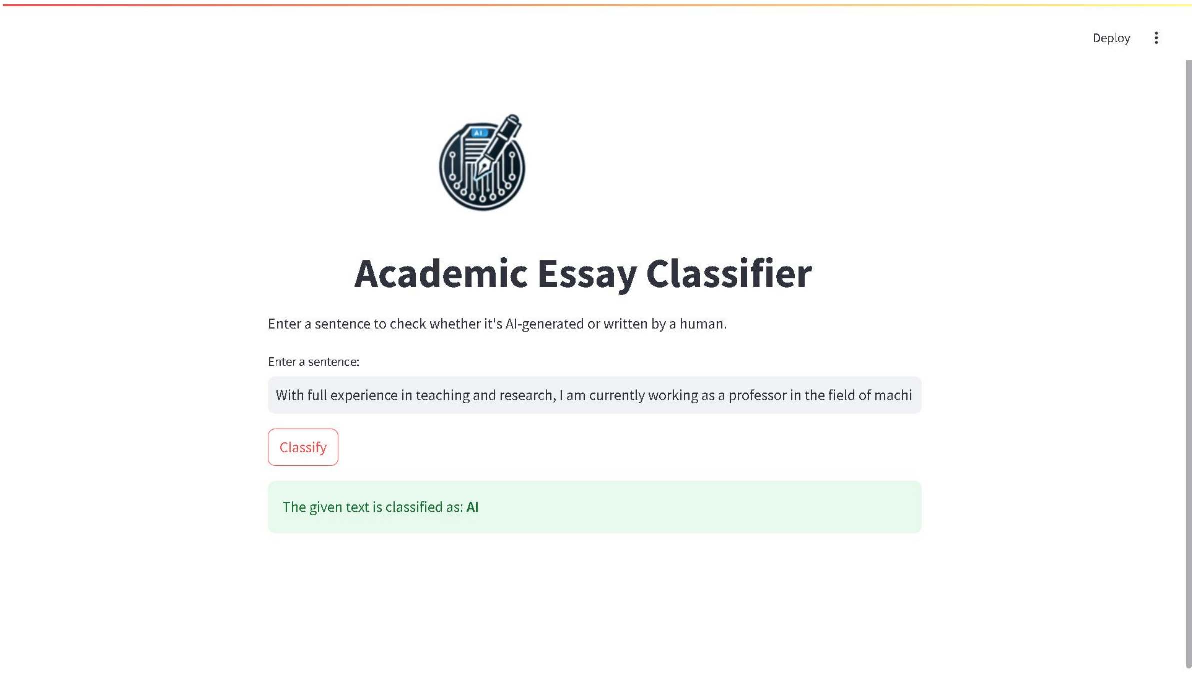
Enter a sentence to check whether it's AI-generated or written by a human.

Enter a sentence:

I am a professor handling machine learning

The given text is classified as: **Human**

Figure 8.1 - Output 1



Academic Essay Classifier

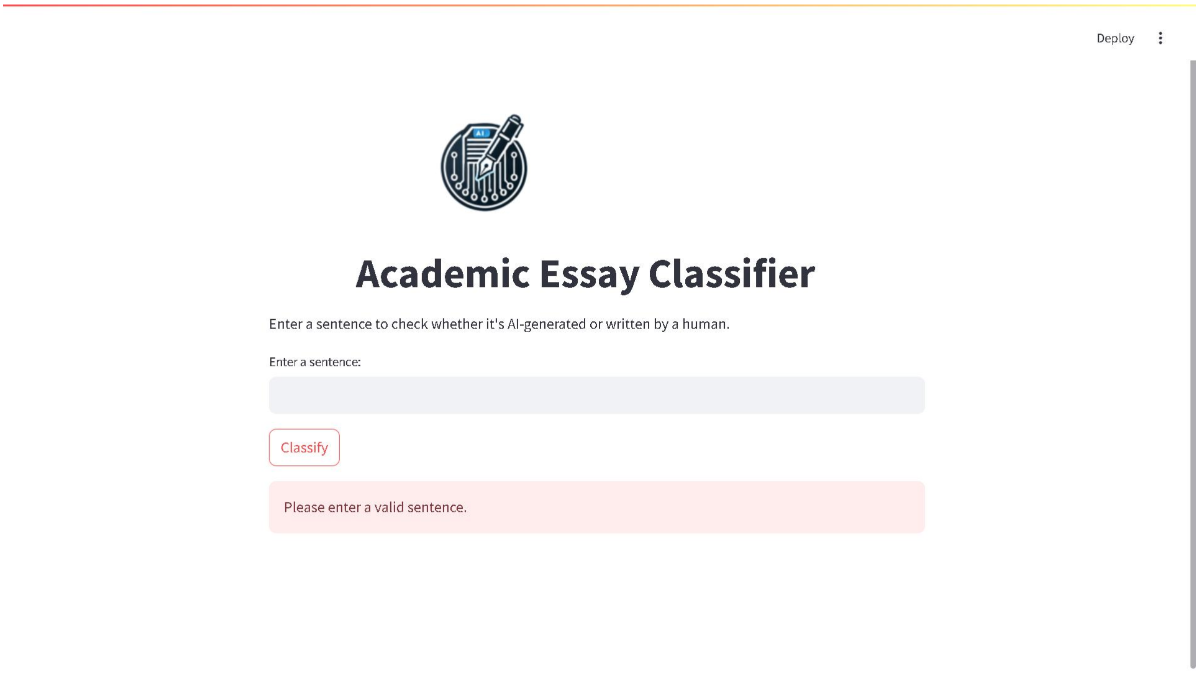
Enter a sentence to check whether it's AI-generated or written by a human.

Enter a sentence:

With full experience in teaching and research, I am currently working as a professor in the field of machi

The given text is classified as: **AI**

Figure 8.2 - Output 2



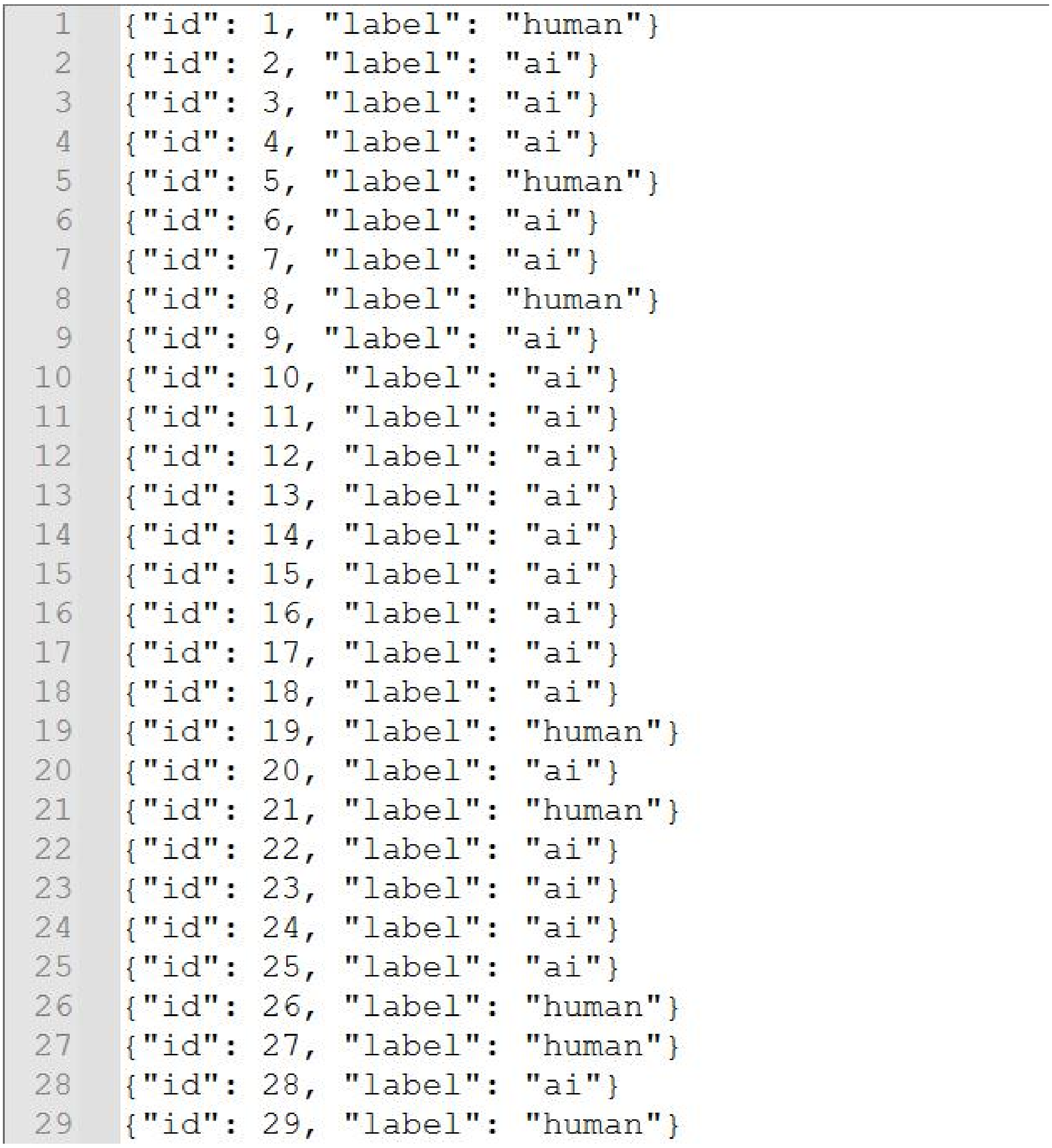
Academic Essay Classifier

Enter a sentence to check whether it's AI-generated or written by a human.

Enter a sentence:

Please enter a valid sentence.

Figure 8.3 - Output 3



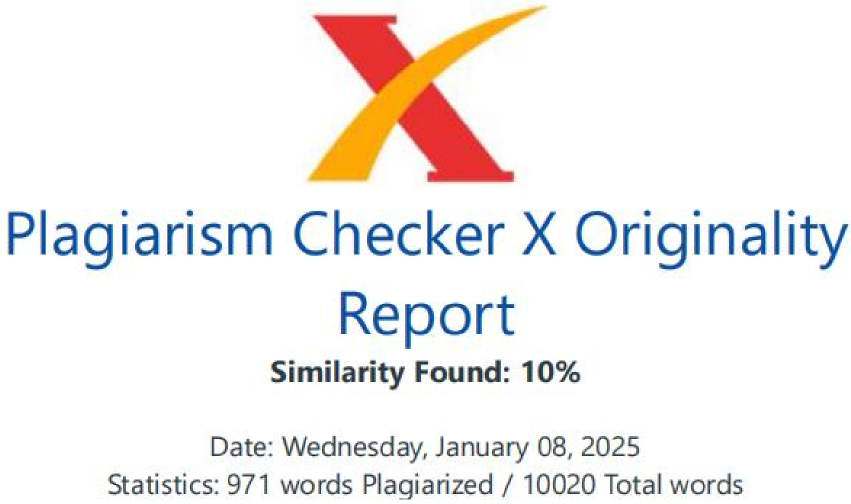
```
{"id": 1, "label": "human"}
{"id": 2, "label": "ai"}
{"id": 3, "label": "ai"}
{"id": 4, "label": "ai"}
{"id": 5, "label": "human"}
{"id": 6, "label": "ai"}
{"id": 7, "label": "ai"}
{"id": 8, "label": "human"}
{"id": 9, "label": "ai"}
{"id": 10, "label": "ai"}
{"id": 11, "label": "ai"}
{"id": 12, "label": "ai"}
{"id": 13, "label": "ai"}
{"id": 14, "label": "ai"}
{"id": 15, "label": "ai"}
{"id": 16, "label": "ai"}
{"id": 17, "label": "ai"}
{"id": 18, "label": "ai"}
{"id": 19, "label": "human"}
{"id": 20, "label": "ai"}
{"id": 21, "label": "human"}
{"id": 22, "label": "ai"}
{"id": 23, "label": "ai"}
{"id": 24, "label": "ai"}
{"id": 25, "label": "ai"}
{"id": 26, "label": "human"}
{"id": 27, "label": "human"}
{"id": 28, "label": "ai"}
{"id": 29, "label": "human"}
```

Figure 9 - Results of Training data

APPENDIX-C

Enclosures

1. Plagiarism Check report



Plagiarism Checker X Originality Report

Similarity Found: 10%

Date: Wednesday, January 08, 2025

Statistics: 971 words Plagiarized / 10020 Total words

APPENDIX-C

2. Sustainable Development Goals

SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS

Goal 4: Quality Education

Supports academic integrity by detecting AI-generated content, promoting fair evaluation and learning.

Goal 9: Industry, Innovation, and Infrastructure

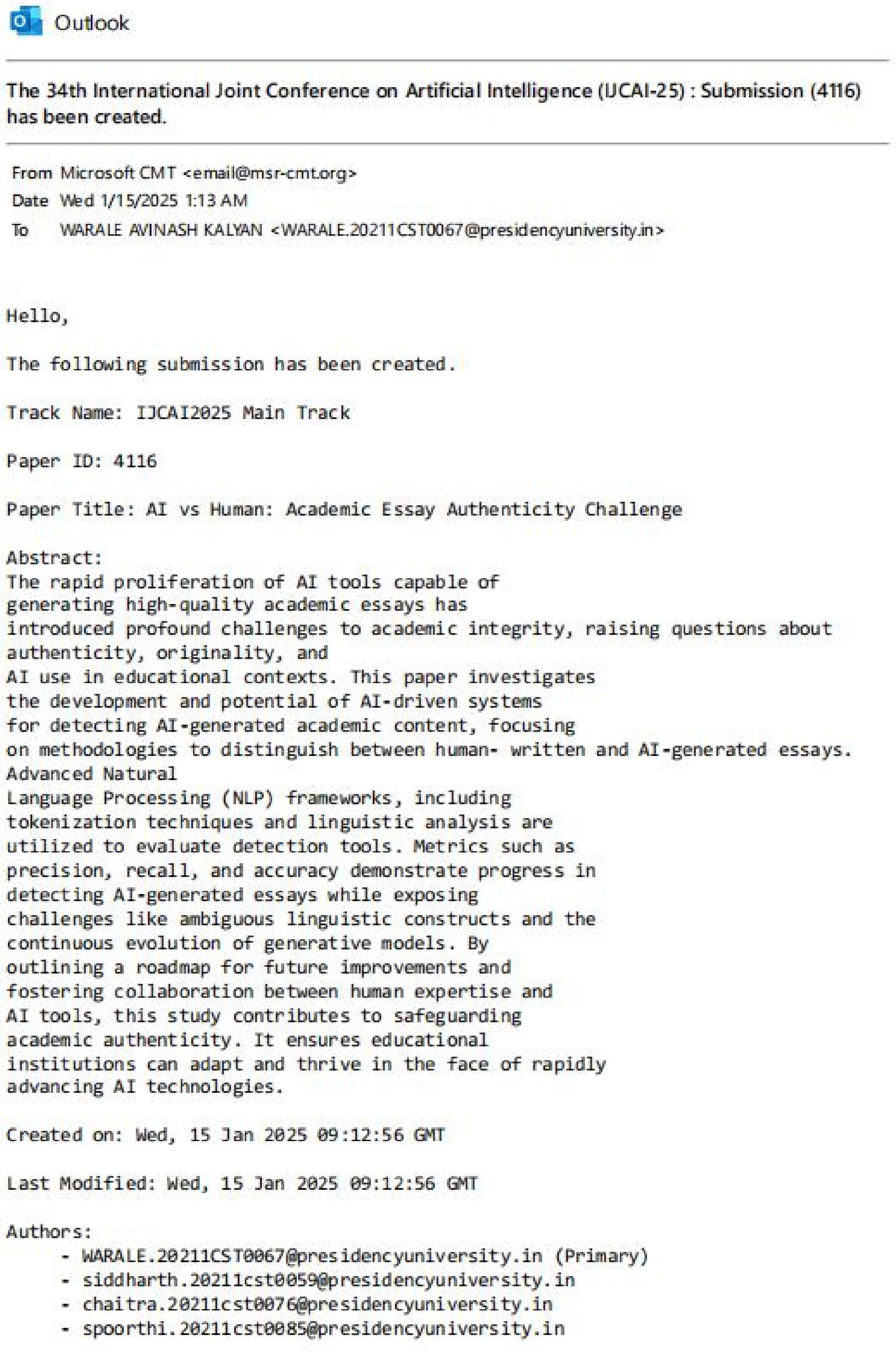
Advances AI and machine learning technologies to address real-world challenges.

Goal 16: Peace, Justice, and Strong Institutions

Strengthens institutions by ensuring authenticity in academic submissions and fostering trust.

APPENDIX-C

3. Journal publication/Conference Paper Presented Certificates of all students



Outlook

The 34th International Joint Conference on Artificial Intelligence (IJCAI-25) : Submission (4116) has been created.

From Microsoft CMT <email@msr-cmt.org>
Date Wed 1/15/2025 1:13 AM
To WARALE AVINASH KALYAN <WARALE.20211CST0067@presidencyuniversity.in>

Hello,

The following submission has been created.

Track Name: IJCAI2025 Main Track

Paper ID: 4116

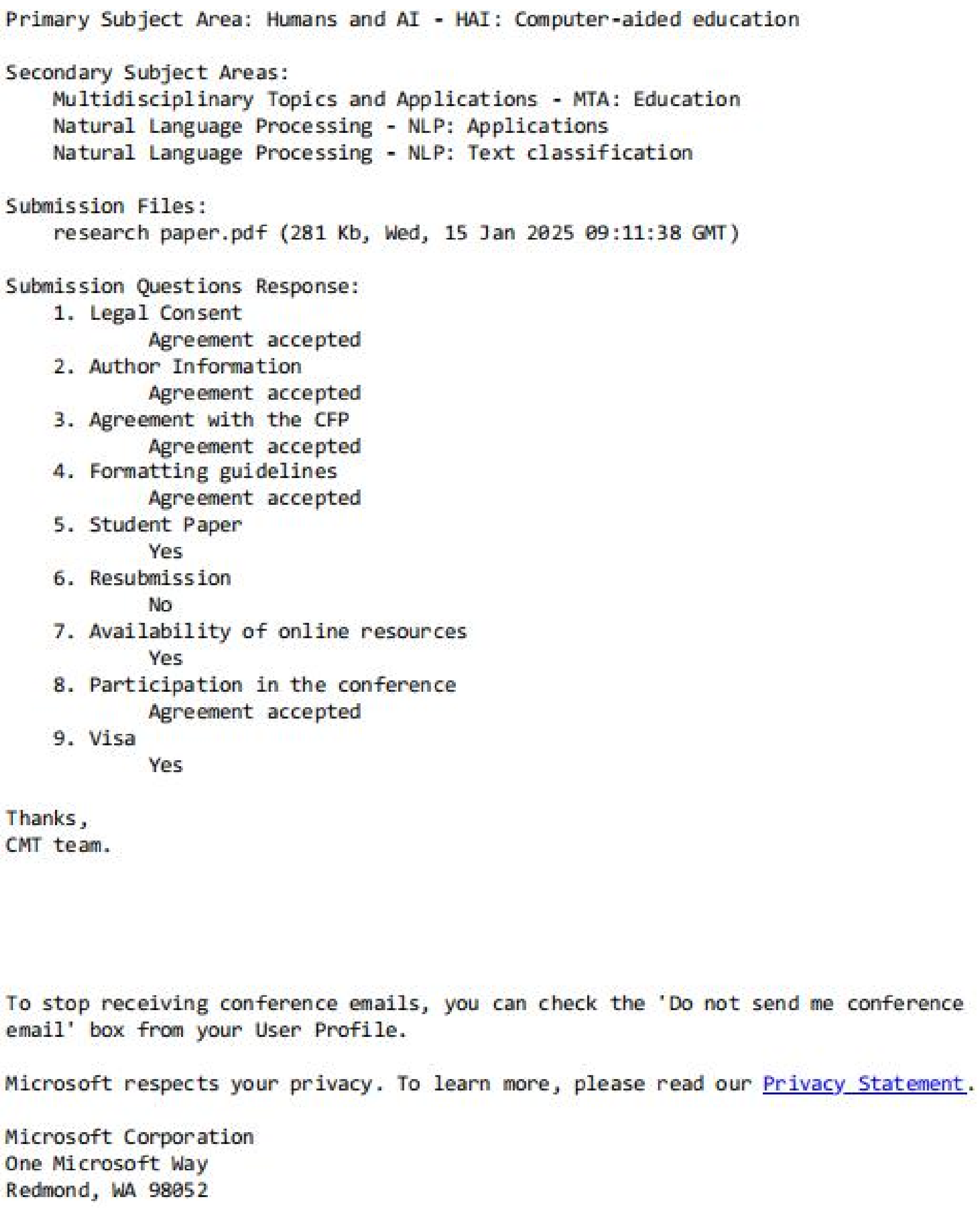
Paper Title: AI vs Human: Academic Essay Authenticity Challenge

Abstract:
The rapid proliferation of AI tools capable of generating high-quality academic essays has introduced profound challenges to academic integrity, raising questions about authenticity, originality, and AI use in educational contexts. This paper investigates the development and potential of AI-driven systems for detecting AI-generated academic content, focusing on methodologies to distinguish between human- written and AI-generated essays. Advanced Natural Language Processing (NLP) frameworks, including tokenization techniques and linguistic analysis are utilized to evaluate detection tools. Metrics such as precision, recall, and accuracy demonstrate progress in detecting AI-generated essays while exposing challenges like ambiguous linguistic constructs and the continuous evolution of generative models. By outlining a roadmap for future improvements and fostering collaboration between human expertise and AI tools, this study contributes to safeguarding academic authenticity. It ensures educational institutions can adapt and thrive in the face of rapidly advancing AI technologies.

Created on: Wed, 15 Jan 2025 09:12:56 GMT

Last Modified: Wed, 15 Jan 2025 09:12:56 GMT

Authors:
- WARALE.20211CST0067@presidencyuniversity.in (Primary)
- siddharth.20211cst0059@presidencyuniversity.in
- chaitra.20211cst0076@presidencyuniversity.in
- spoorthi.20211cst0085@presidencyuniversity.in



Primary Subject Area: Humans and AI - HAI: Computer-aided education

Secondary Subject Areas:
Multidisciplinary Topics and Applications - MTA: Education
Natural Language Processing - NLP: Applications
Natural Language Processing - NLP: Text classification

Submission Files:
research paper.pdf (281 Kb, Wed, 15 Jan 2025 09:11:38 GMT)

Submission Questions Response:

1. Legal Consent
   Agreement accepted
2. Author Information
   Agreement accepted
3. Agreement with the CFP
   Agreement accepted
4. Formatting guidelines
   Agreement accepted
5. Student Paper
   Yes
6. Resubmission
   No
7. Availability of online resources
   Yes
8. Participation in the conference
   Agreement accepted
9. Visa
   Yes

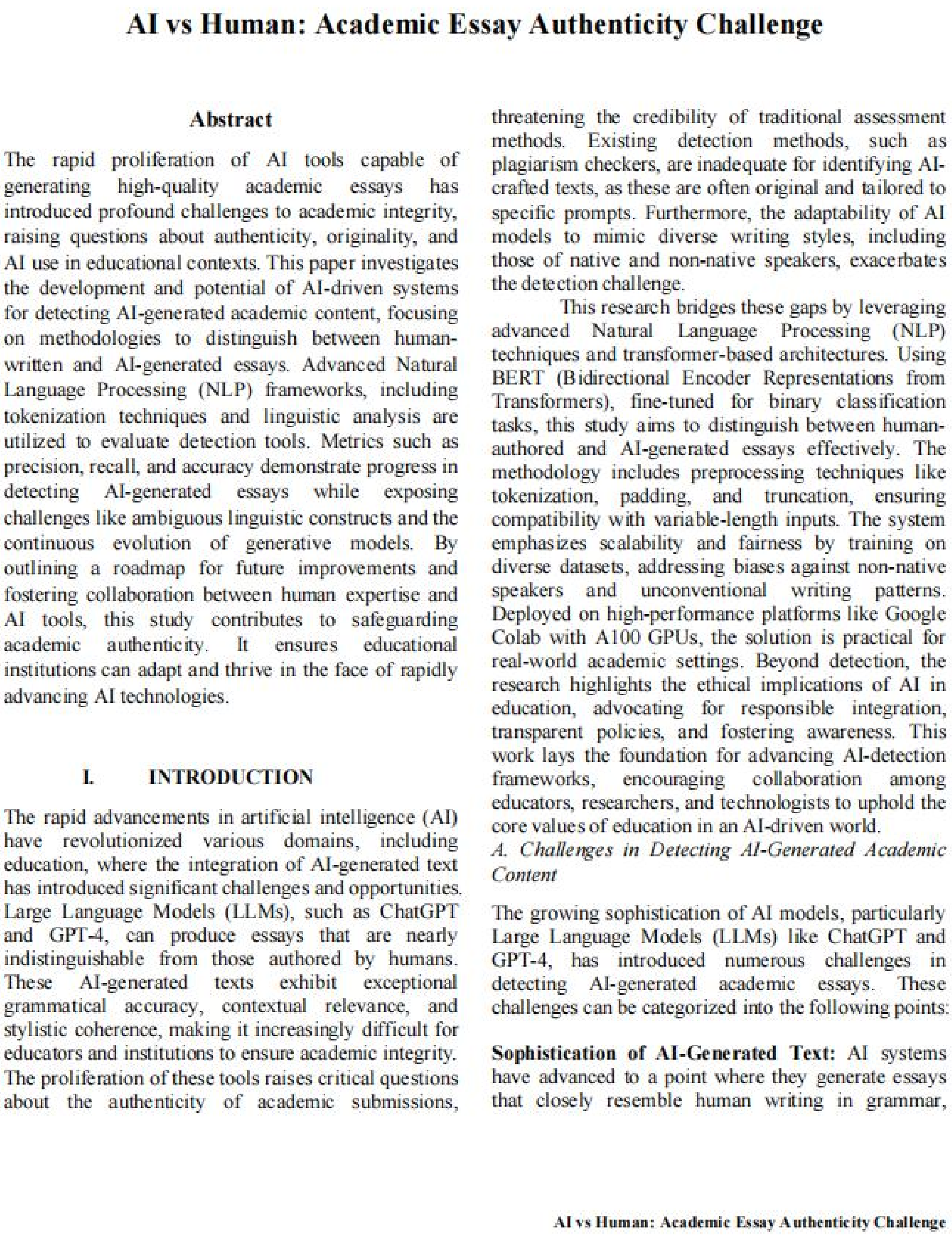
Thanks,
CMT team.

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Research Paper



AI vs Human: Academic Essay Authenticity Challenge

Abstract

The rapid proliferation of AI tools capable of generating high-quality academic essays has introduced profound challenges to academic integrity, raising questions about authenticity, originality, and AI use in educational contexts. This paper investigates the development and potential of AI-driven systems for detecting AI-generated academic content, focusing on methodologies to distinguish between human-written and AI-generated essays. Advanced Natural Language Processing (NLP) frameworks, including tokenization techniques and linguistic analysis are utilized to evaluate detection tools. Metrics such as precision, recall, and accuracy demonstrate progress in detecting AI-generated essays while exposing challenges like ambiguous linguistic constructs and the continuous evolution of generative models. By outlining a roadmap for future improvements and fostering collaboration between human expertise and AI tools, this study contributes to safeguarding academic authenticity. It ensures educational institutions can adapt and thrive in the face of rapidly advancing AI technologies.

I. INTRODUCTION

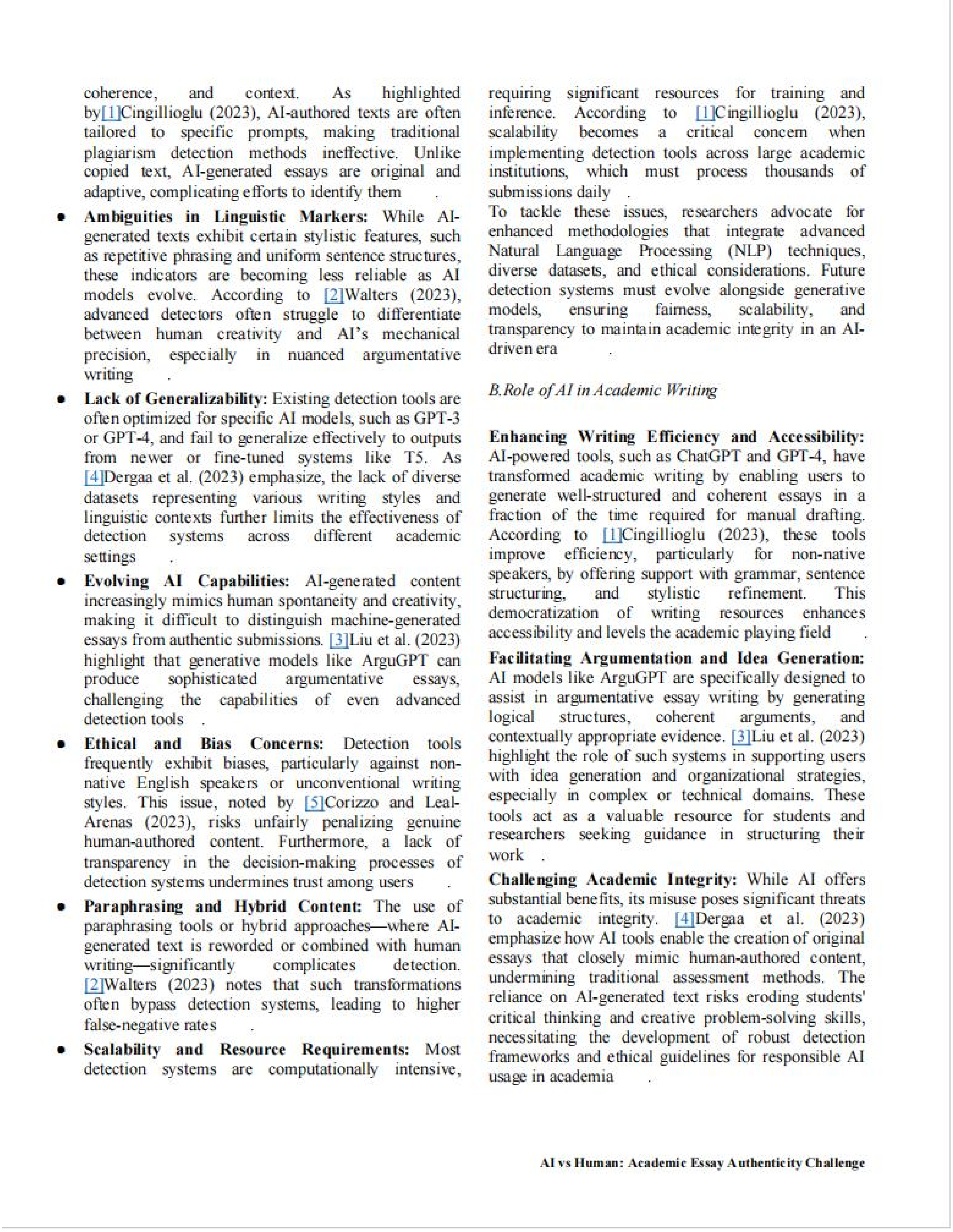
The rapid advancements in artificial intelligence (AI) have revolutionized various domains, including education, where the integration of AI-generated text has introduced significant challenges and opportunities. Large Language Models (LLMs), such as ChatGPT and GPT-4, can produce essays that are nearly indistinguishable from those authored by humans. These AI-generated texts exhibit exceptional grammatical accuracy, contextual relevance, and stylistic coherence, making it increasingly difficult for educators and institutions to ensure academic integrity. The proliferation of these tools raises critical questions about the authenticity of academic submissions, threatening the credibility of traditional assessment methods. Existing detection methods, such as plagiarism checkers, are inadequate for identifying AI-crafted texts, as these are often original and tailored to specific prompts. Furthermore, the adaptability of AI models to mimic diverse writing styles, including those of native and non-native speakers, exacerbates the detection challenge.

This research bridges these gaps by leveraging advanced Natural Language Processing (NLP) techniques and transformer-based architectures. Using BERT (Bidirectional Encoder Representations from Transformers), fine-tuned for binary classification tasks, this study aims to distinguish between human-authored and AI-generated essays effectively. The methodology includes preprocessing techniques like tokenization, padding, and truncation, ensuring compatibility with variable-length inputs. The system emphasizes scalability and fairness by training on diverse datasets, addressing biases against non-native speakers and unconventional writing patterns. Deployed on high-performance platforms like Google Colab with A100 GPUs, the solution is practical for real-world academic settings. Beyond detection, the research highlights the ethical implications of AI in education, advocating for responsible integration, transparent policies, and fostering awareness. This work lays the foundation for advancing AI-detection frameworks, encouraging collaboration among educators, researchers, and technologists to uphold the core values of education in an AI-driven world.

A. Challenges in Detecting AI-Generated Academic Content

The growing sophistication of AI models, particularly Large Language Models (LLMs) like ChatGPT and GPT-4, has introduced numerous challenges in detecting AI-generated academic essays. These challenges can be categorized into the following points:

Sophistication of AI-Generated Text: AI systems have advanced to a point where they generate essays that closely resemble human writing in grammar,



coherence, and context. As highlighted by[1]Cingillioglu (2023), AI-authored texts are often tailored to specific prompts, making traditional plagiarism detection methods ineffective. Unlike copied text, AI-generated essays are original and adaptive, complicating efforts to identify them .

- **Ambiguities in Linguistic Markers:** While AI-generated texts exhibit certain stylistic features, such as repetitive phrasing and uniform sentence structures, these indicators are becoming less reliable as AI models evolve. According to [2]Walters (2023), advanced detectors often struggle to differentiate between human creativity and AI's mechanical precision, especially in nuanced argumentative writing .
- **Lack of Generalizability:** Existing detection tools are often optimized for specific AI models, such as GPT-3 or GPT-4, and fail to generalize effectively to outputs from newer or fine-tuned systems like T5. As [4]Dergaa et al. (2023) emphasize, the lack of diverse datasets representing various writing styles and linguistic contexts further limits the effectiveness of detection systems across different academic settings .
- **Evolving AI Capabilities:** AI-generated content increasingly mimics human spontaneity and creativity, making it difficult to distinguish machine-generated essays from authentic submissions. [3]Liu et al. (2023) highlight that generative models like ArguGPT can produce sophisticated argumentative essays, challenging the capabilities of even advanced detection tools .
- **Ethical and Bias Concerns:** Detection tools frequently exhibit biases, particularly against non-native English speakers or unconventional writing styles. This issue, noted by [5]Corizzo and Leal-Arenas (2023), risks unfairly penalizing genuine human-authored content. Furthermore, a lack of transparency in the decision-making processes of detection systems undermines trust among users .
- **Paraphrasing and Hybrid Content:** The use of paraphrasing tools or hybrid approaches—where AI-generated text is reworded or combined with human writing—significantly complicates detection. [2]Walters (2023) notes that such transformations often bypass detection systems, leading to higher false-negative rates .
- **Scalability and Resource Requirements:** Most detection systems are computationally intensive, requiring significant resources for training and inference. According to [1]Cingillioglu (2023), scalability becomes a critical concern when implementing detection tools across large academic institutions, which must process thousands of submissions daily .

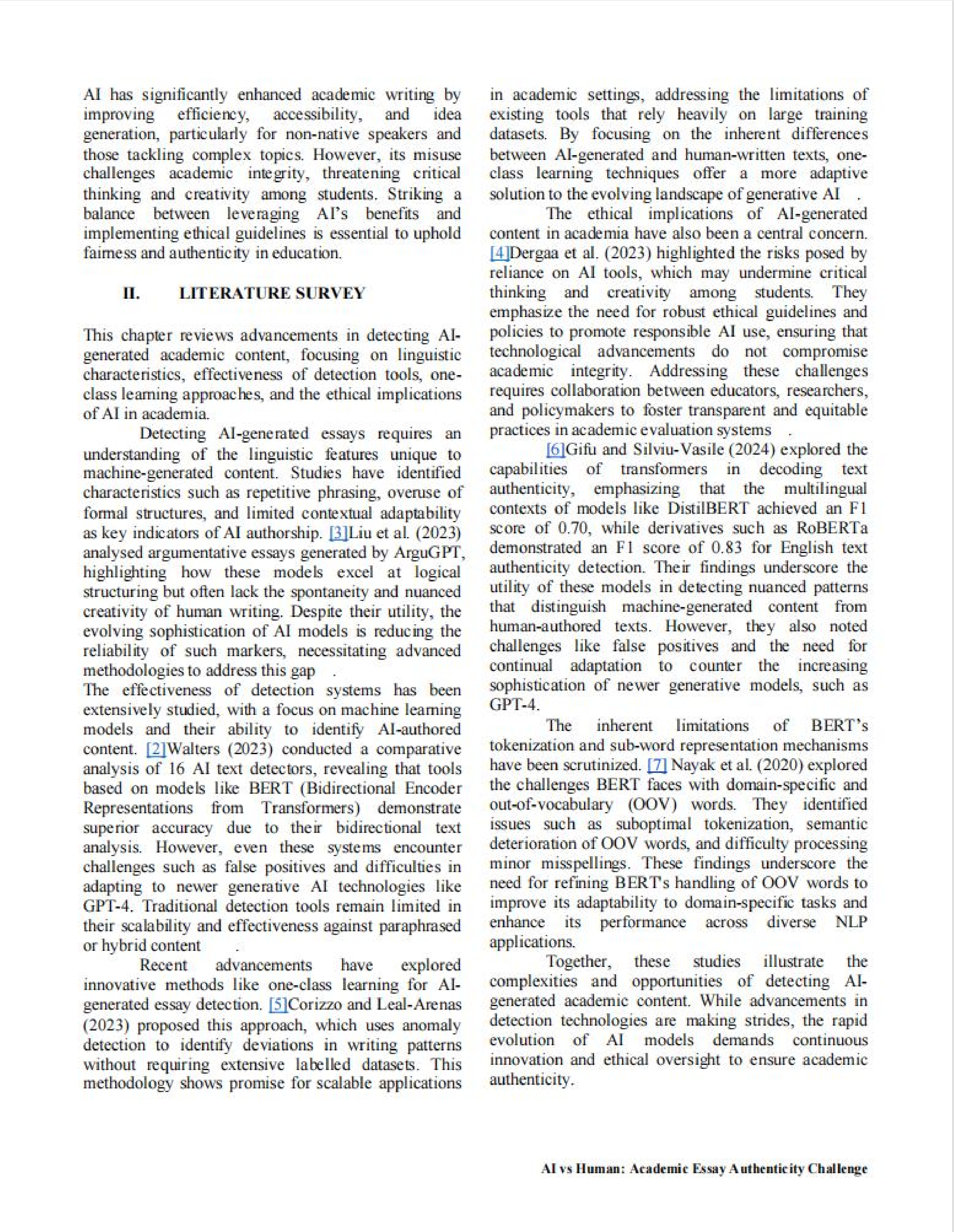
To tackle these issues, researchers advocate for enhanced methodologies that integrate advanced Natural Language Processing (NLP) techniques, diverse datasets, and ethical considerations. Future detection systems must evolve alongside generative models, ensuring fairness, scalability, and transparency to maintain academic integrity in an AI-driven era .

B.Role of AI in Academic Writing

Enhancing Writing Efficiency and Accessibility: AI-powered tools, such as ChatGPT and GPT-4, have transformed academic writing by enabling users to generate well-structured and coherent essays in a fraction of the time required for manual drafting. According to [1]Cingillioglu (2023), these tools improve efficiency, particularly for non-native speakers, by offering support with grammar, sentence structuring, and stylistic refinement. This democratization of writing resources enhances accessibility and levels the academic playing field .

Facilitating Argumentation and Idea Generation: AI models like ArguGPT are specifically designed to assist in argumentative essay writing by generating logical structures, coherent arguments, and contextually appropriate evidence. [3]Liu et al. (2023) highlight the role of such systems in supporting users with idea generation and organizational strategies, especially in complex or technical domains. These tools act as a valuable resource for students and researchers seeking guidance in structuring their work .

Challenging Academic Integrity: While AI offers substantial benefits, its misuse poses significant threats to academic integrity. [4]Dergaa et al. (2023) emphasize how AI tools enable the creation of original essays that closely mimic human-authored content, undermining traditional assessment methods. The reliance on AI-generated text risks eroding students' critical thinking and creative problem-solving skills, necessitating the development of robust detection frameworks and ethical guidelines for responsible AI usage in academia .



AI has significantly enhanced academic writing by improving efficiency, accessibility, and idea generation, particularly for non-native speakers and those tackling complex topics. However, its misuse challenges academic integrity, threatening critical thinking and creativity among students. Striking a balance between leveraging AI's benefits and implementing ethical guidelines is essential to uphold fairness and authenticity in education.

II. LITERATURE SURVEY

This chapter reviews advancements in detecting AI-generated academic content, focusing on linguistic characteristics, effectiveness of detection tools, one-class learning approaches, and the ethical implications of AI in academia.

Detecting AI-generated essays requires an understanding of the linguistic features unique to machine-generated content. Studies have identified characteristics such as repetitive phrasing, overuse of formal structures, and limited contextual adaptability as key indicators of AI authorship. [3]Liu et al. (2023) analysed argumentative essays generated by ArguGPT, highlighting how these models excel at logical structuring but often lack the spontaneity and nuanced creativity of human writing. Despite their utility, the evolving sophistication of AI models is reducing the reliability of such markers, necessitating advanced methodologies to address this gap .

The effectiveness of detection systems has been extensively studied, with a focus on machine learning models and their ability to identify AI-authored content. [2]Walters (2023) conducted a comparative analysis of 16 AI text detectors, revealing that tools based on models like BERT (Bidirectional Encoder Representations from Transformers) demonstrate superior accuracy due to their bidirectional text analysis. However, even these systems encounter challenges such as false positives and difficulties in adapting to newer generative AI technologies like GPT-4. Traditional detection tools remain limited in their scalability and effectiveness against paraphrased or hybrid content .

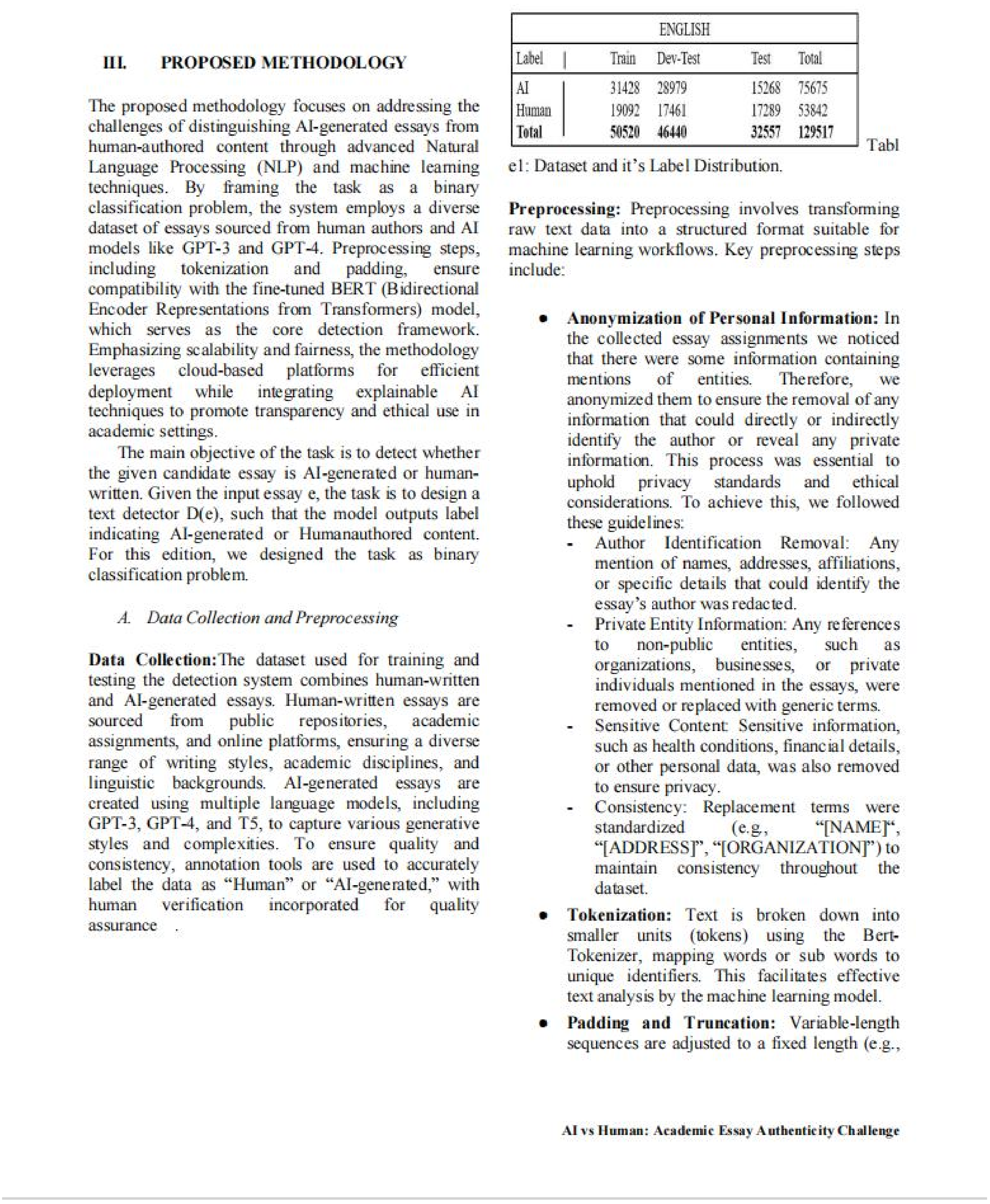
Recent advancements have explored innovative methods like one-class learning for AI-generated essay detection. [5]Corizzo and Leal-Arenas (2023) proposed this approach, which uses anomaly detection to identify deviations in writing patterns without requiring extensive labelled datasets. This methodology shows promise for scalable applications in academic settings, addressing the limitations of existing tools that rely heavily on large training datasets. By focusing on the inherent differences between AI-generated and human-written texts, one-class learning techniques offer a more adaptive solution to the evolving landscape of generative AI .

The ethical implications of AI-generated content in academia have also been a central concern. [4]Dergaa et al. (2023) highlighted the risks posed by reliance on AI tools, which may undermine critical thinking and creativity among students. They emphasize the need for robust ethical guidelines and policies to promote responsible AI use, ensuring that technological advancements do not compromise academic integrity. Addressing these challenges requires collaboration between educators, researchers, and policymakers to foster transparent and equitable practices in academic evaluation systems .

[6]Gifu and Silviu-Vasile (2024) explored the capabilities of transformers in decoding text authenticity, emphasizing that the multilingual contexts of models like DistilBERT achieved an F1 score of 0.70, while derivatives such as RoBERTa demonstrated an F1 score of 0.83 for English text authenticity detection. Their findings underscore the utility of these models in detecting nuanced patterns that distinguish machine-generated content from human-authored texts. However, they also noted challenges like false positives and the need for continual adaptation to counter the increasing sophistication of newer generative models, such as GPT-4.

The inherent limitations of BERT's tokenization and sub-word representation mechanisms have been scrutinized. [7] Nayak et al. (2020) explored the challenges BERT faces with domain-specific and out-of-vocabulary (OOV) words. They identified issues such as suboptimal tokenization, semantic deterioration of OOV words, and difficulty processing minor misspellings. These findings underscore the need for refining BERT's handling of OOV words to improve its adaptability to domain-specific tasks and enhance its performance across diverse NLP applications.

Together, these studies illustrate the complexities and opportunities of detecting AI-generated academic content. While advancements in detection technologies are making strides, the rapid evolution of AI models demands continuous innovation and ethical oversight to ensure academic authenticity.



III. PROPOSED METHODOLOGY

The proposed methodology focuses on addressing the challenges of distinguishing AI-generated essays from human-authored content through advanced Natural Language Processing (NLP) and machine learning techniques. By framing the task as a binary classification problem, the system employs a diverse dataset of essays sourced from human authors and AI models like GPT-3 and GPT-4. Preprocessing steps, including tokenization and padding, ensure compatibility with the fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model, which serves as the core detection framework. Emphasizing scalability and fairness, the methodology leverages cloud-based platforms for efficient deployment while integrating explainable AI techniques to promote transparency and ethical use in academic settings.

The main objective of the task is to detect whether the given candidate essay is AI-generated or human-written. Given the input essay e, the task is to design a text detector D(e), such that the model outputs label indicating AI-generated or Humanauthored content. For this edition, we designed the task as binary classification problem.

A. Data Collection and Preprocessing

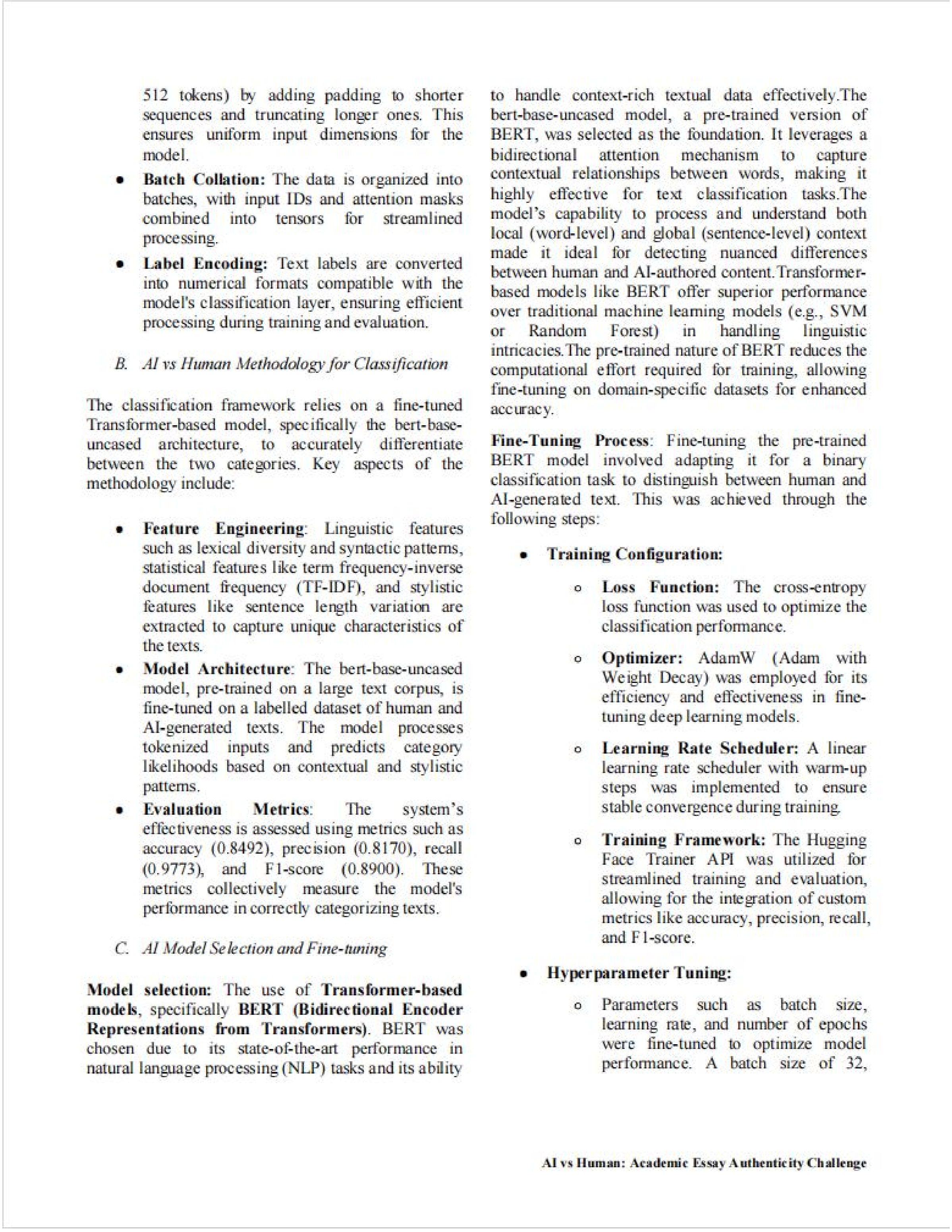
Data Collection: The dataset used for training and testing the detection system combines human-written and AI-generated essays. Human-written essays are sourced from public repositories, academic assignments, and online platforms, ensuring a diverse range of writing styles, academic disciplines, and linguistic backgrounds. AI-generated essays are created using multiple language models, including GPT-3, GPT-4, and T5, to capture various generative styles and complexities. To ensure quality and consistency, annotation tools are used to accurately label the data as "Human" or "AI-generated," with human verification incorporated for quality assurance .

ENGLISH				
Label	Train	Dev-Test	Test	Total
AI	31428	28979	15268	75675
Human	19092	17461	17289	53842
Total	**50520**	**46440**	**32557**	**129517**

Tabl e1: Dataset and it's Label Distribution.

Preprocessing: Preprocessing involves transforming raw text data into a structured format suitable for machine learning workflows. Key preprocessing steps include:

- **Anonymization of Personal Information:** In the collected essay assignments we noticed that there were some information containing mentions of entities. Therefore, we anonymized them to ensure the removal of any information that could directly or indirectly identify the author or reveal any private information. This process was essential to uphold privacy standards and ethical considerations. To achieve this, we followed these guidelines:
    - Author Identification Removal: Any mention of names, addresses, affiliations, or specific details that could identify the essay's author was redacted.
    - Private Entity Information: Any references to non-public entities, such as organizations, businesses, or private individuals mentioned in the essays, were removed or replaced with generic terms.
    - Sensitive Content: Sensitive information, such as health conditions, financial details, or other personal data, was also removed to ensure privacy.
    - Consistency: Replacement terms were standardized (e.g., "[NAME]", "[ADDRESS]", "[ORGANIZATION]") to maintain consistency throughout the dataset.
- **Tokenization:** Text is broken down into smaller units (tokens) using the Bert-Tokenizer, mapping words or sub words to unique identifiers. This facilitates effective text analysis by the machine learning model.
- **Padding and Truncation:** Variable-length sequences are adjusted to a fixed length (e.g.,



512 tokens) by adding padding to shorter sequences and truncating longer ones. This ensures uniform input dimensions for the model.

- **Batch Collation:** The data is organized into batches, with input IDs and attention masks combined into tensors for streamlined processing.
- **Label Encoding:** Text labels are converted into numerical formats compatible with the model's classification layer, ensuring efficient processing during training and evaluation.

B. AI vs Human Methodology for Classification

The classification framework relies on a fine-tuned Transformer-based model, specifically the bert-base-uncased architecture, to accurately differentiate between the two categories. Key aspects of the methodology include:

- **Feature Engineering**: Linguistic features such as lexical diversity and syntactic patterns, statistical features like term frequency-inverse document frequency (TF-IDF), and stylistic features like sentence length variation are extracted to capture unique characteristics of the texts.
- **Model Architecture**: The bert-base-uncased model, pre-trained on a large text corpus, is fine-tuned on a labelled dataset of human and AI-generated texts. The model processes tokenized inputs and predicts category likelihoods based on contextual and stylistic patterns.
- **Evaluation Metrics**: The system's effectiveness is assessed using metrics such as accuracy (0.8492), precision (0.8170), recall (0.9773), and F1-score (0.8900). These metrics collectively measure the model's performance in correctly categorizing texts.

C. AI Model Selection and Fine-tuning

Model selection: The use of **Transformer-based models**, specifically **BERT (Bidirectional Encoder Representations from Transformers)**. BERT was chosen due to its state-of-the-art performance in natural language processing (NLP) tasks and its ability to handle context-rich textual data effectively.The bert-base-uncased model, a pre-trained version of BERT, was selected as the foundation. It leverages a bidirectional attention mechanism to capture contextual relationships between words, making it highly effective for text classification tasks.The model's capability to process and understand both local (word-level) and global (sentence-level) context made it ideal for detecting nuanced differences between human and AI-authored content.Transformer-based models like BERT offer superior performance over traditional machine learning models (e.g., SVM or Random Forest) in handling linguistic intricacies.The pre-trained nature of BERT reduces the computational effort required for training, allowing fine-tuning on domain-specific datasets for enhanced accuracy.

Fine-Tuning Process: Fine-tuning the pre-trained BERT model involved adapting it for a binary classification task to distinguish between human and AI-generated text. This was achieved through the following steps:

- **Training Configuration:**
  - **Loss Function:** The cross-entropy loss function was used to optimize the classification performance.
  - **Optimizer:** AdamW (Adam with Weight Decay) was employed for its efficiency and effectiveness in fine-tuning deep learning models.
  - **Learning Rate Scheduler:** A linear learning rate scheduler with warm-up steps was implemented to ensure stable convergence during training.
  - **Training Framework:** The Hugging Face Trainer API was utilized for streamlined training and evaluation, allowing for the integration of custom metrics like accuracy, precision, recall, and F1-score.
- **Hyperparameter Tuning:**
  - Parameters such as batch size, learning rate, and number of epochs were fine-tuned to optimize model performance. A batch size of 32,

learning rate of 2×10^-5, and 3 epochs were found to achieve the best results.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The model utilizes a BERT-based architecture for detecting AI-generated content, with a classification layer designed for binary text classification tasks. The system consists of several key components:

1. **Data Preprocessing**:
   - Text data is cleaned, tokenized, and converted into sequences using the BERT tokenizer, which splits sentences into subword units.
   - Padding is applied to standardize input lengths, and the sequences are formatted with special tokens (e.g., [CLS], [SEP]).
2. **Feature Extraction with BERT**:
   - The input sequence is passed through a pretrained BERT model, which processes the data using its multiple layers of transformer blocks.
   - BERT uses attention mechanisms to capture contextual relationships between words in both directions (left and right), providing a deeper understanding of the input text.
3. **Classification Layer**:
   - The final output from BERT's transformer layers is passed through a fully connected layer (dense layer).
   - The model then applies the **Softmax activation function** to classify the input into one of two categories: AI-generated or human-written.
   - **Softmax Function**:

$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Fig.1 x, represents the logits for the classes, and the output is a probability distribution over the possible classes.

4. **Training**:
   - The model is trained using labeled data (AI-generated or human-written texts), with a cross-entropy loss function, which measures the difference between the predicted and true labels.
   - The optimizer used is typically **Adam**, which adjusts the learning rate dynamically based on training progress.
5. **Prediction**:
   - Once trained, the model can classify new text inputs as either human-written or AI-generated, based on the output probability distribution.
6. **Output**:
   - The model outputs a classification result, with a probability score that indicates the likelihood of the input text being AI-generated.

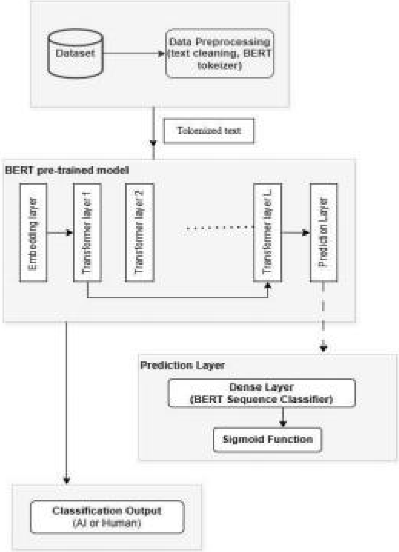
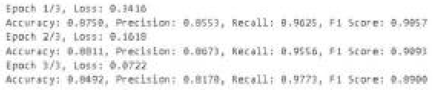


Fig. 2: Model Architecture

V. RESULTS AND DISCUSSIONS

The fine-tuned BERT-based classification system effectively distinguished human-authored essays from AI-generated ones, achieving an accuracy of **84%**, with a precision of **81%**, recall of **97%**, and F1-score of **89%**. These results highlight the model's ability to identify subtle linguistic and stylistic differences, such as the repetitive phrasing and uniform structures typical of AI-generated content, as noted by [3]Liu et al. (2023). It also captured the organic flow and lexical diversity of human-written essays, aligning with findings from[1] Cingillioglu (2023).



```
Epoch 1/3, Loss: 0.3416
Accuracy: 0.8750, Precision: 0.8553, Recall: 0.9625, F1 Score: 0.9057
Epoch 2/3, Loss: 0.1618
Accuracy: 0.8811, Precision: 0.8673, Recall: 0.9556, F1 Score: 0.9093
Epoch 3/3, Loss: 0.0722
Accuracy: 0.8492, Precision: 0.8170, Recall: 0.9773, F1 Score: 0.8900
```

Fig 3: Evaluation metrics

Challenges included occasional misclassification of creative AI-generated texts as human-written and repetitive human-authored texts as AI-generated. Additionally, truncation of longer essays to 512 tokens sometimes led to context loss, affecting classification accuracy. These issues underscore the importance of refining tokenization strategies and expanding datasets to enhance model generalizability.The results underscore the importance of integrating diverse datasets representing a wide range of writing styles, academic disciplines, and linguistic backgrounds to improve the model's generalizability. By addressing these challenges, the system provides a strong foundation for advancing AI-detection frameworks, ensuring their scalability and fairness. This work contributes to safeguarding academic integrity and fosters responsible AI usage by offering a reliable tool for detecting AI-generated content in educational contexts.

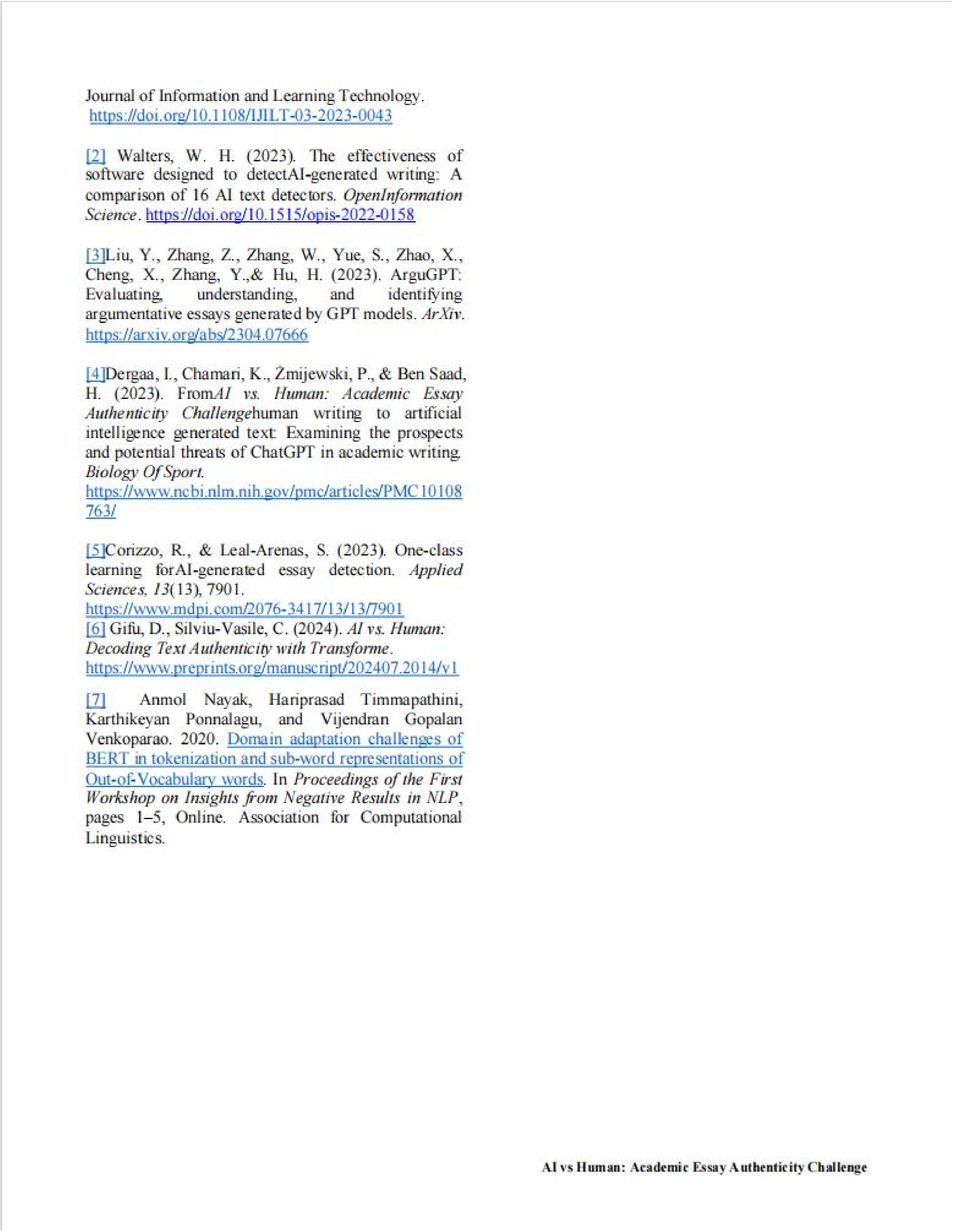
Future improvements could involve exploring one-class learning approaches, as proposed by [5]Corizzo and Leal-Arenas (2023), to handle hybrid and paraphrased content more effectively. Additionally, incorporating domain-specific datasets and refining hyperparameters further could enhance the system's robustness in real-world application.

VI. CONCLUSION

The paper presented demonstrates the effectiveness of a fine-tuned BERT-based classification system in distinguishing human-written essays from AI-generated content, achieving high accuracy and robust performance metrics. By leveraging advanced NLP techniques, the system identifies subtle linguistic and stylistic differences while addressing challenges such as ambiguous content and dataset diversity. Despite limitations like token truncation and occasional misclassification, the framework provides a reliable tool for safeguarding academic integrity. Beyond its technical achievements, this research emphasizes the broader ethical and educational implications of AI in academia, advocating for responsible use of AI as a learning tool rather than a shortcut. Future advancements, such as integrating domain-specific datasets, exploring one-class learning approaches, refining tokenization strategies, and fostering ethical AI awareness, can further enhance the system's scalability, adaptability, and alignment with educational values.

VII. REFERENCES

[1] 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

[2] Walters, W. H. (2023). The effectiveness of software designed to detectAI-generated writing: A comparison of 16 AI text detectors. *OpenInformation Science*. https://doi.org/10.1515/opis-2022-0158

[3]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

[4]Dergaa, I., Chamari, K., Żmijewski, P., & Ben Saad, H. (2023). From*AI vs. Human: Academic Essay Authenticity Challenge*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/

[5]Corizzo, R., & Leal-Arenas, S. (2023). One-class learning forAI-generated essay detection. *Applied Sciences, 13*(13), 7901. https://www.mdpi.com/2076-3417/13/13/7901
[6] Gifu, D., Silviu-Vasile, C. (2024). *AI vs. Human: Decoding Text Authenticity with Transforme*. https://www.preprints.org/manuscript/202407.2014/v1

[7] Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of Out-of-Vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online. Association for Computational Linguistics.