Project Report: AI-Generated Content Detection Platform

Contents

1 Introduction

As fraud methods evolve, from financial fraud, identity theft, credit card misuse, and even plagiarism, modern technology uses advanced techniques such as machine learning and artificial intelligence to detect these frauds.

This project involves the development of a website or mobile application capable of identifying whether a given text or script is generated by AI tools. It combines natural language processing (NLP) and web development.

2 Model Description

2.1 Chosen AI Model

The core of the detection system is based on transformer-based models such as BERT. BERT is a pre-trained deep learning model designed to process and understand text by capturing context and meaning in a bidirectional manner. This model is trained to recognize patterns in text that are often indicative of AI-generated content.

2.2 Design and Development Process

2.2.1 Data Collection

- **Gather Text Data:**
  - *Sources for AI-generated texts:* GPT models (e.g., ChatGPT), open-source datasets (Kaggle `AI_Human.csv`), GitHub.
  - *Sources for human-written texts:* Books (e.g., *Rich Dad Poor Dad*), articles (e.g., news from News in Levels).
- **Label Data:** Ensure that the collected texts are properly labeled as *AI-generated* or *human-written*.

2.2.2 Data Preprocessing

- **Text Cleaning:** Remove any irrelevant information (e.g., HTML tags, special characters).
- **Tokenization:** Split the text into words or sentences for easier analysis.
- **Normalization:** Convert text to a consistent format (e.g., lowercasing, stemming).

3 User Interface Development Tools

3.1 Front-end Development

- HTML/CSS/JavaScript: For creating the web UI.

3.2 Back-end Development

- Python: For the backend logic and model implementation.
- Flask: For building APIs and managing model requests.

4 Front-end UI

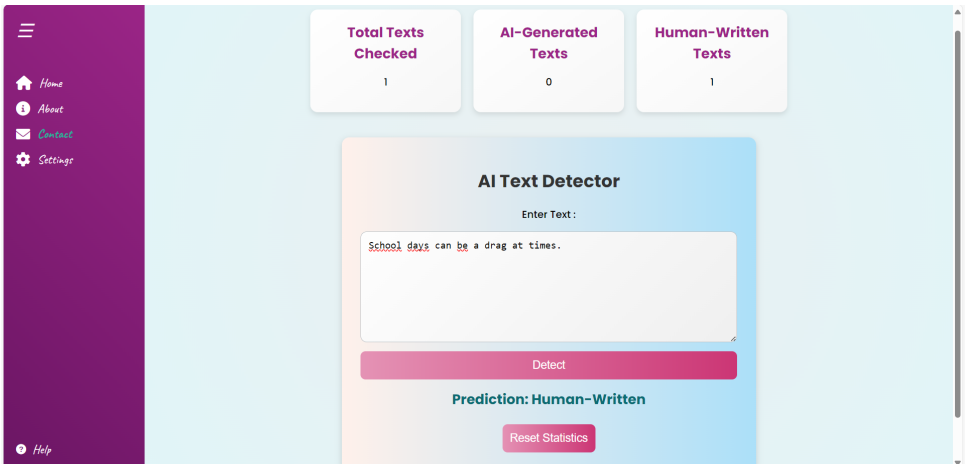

Figure 1: Home page UI design

5 Back-end and model

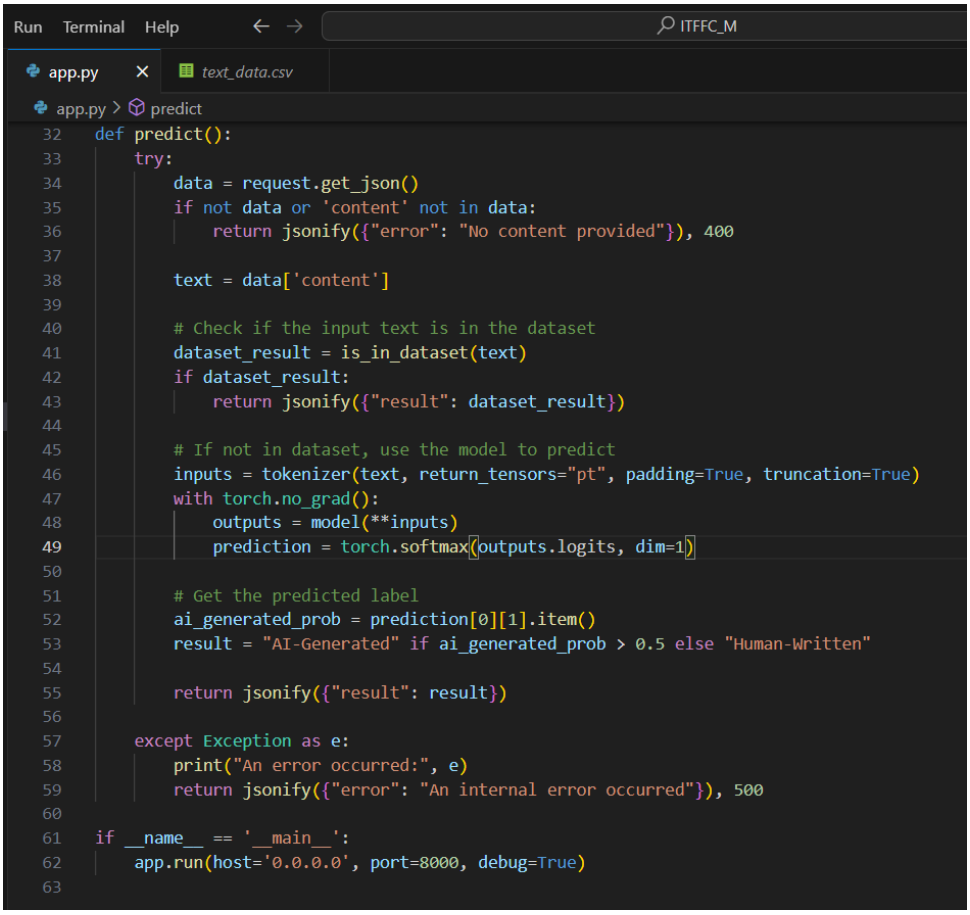

Figure 2: flask

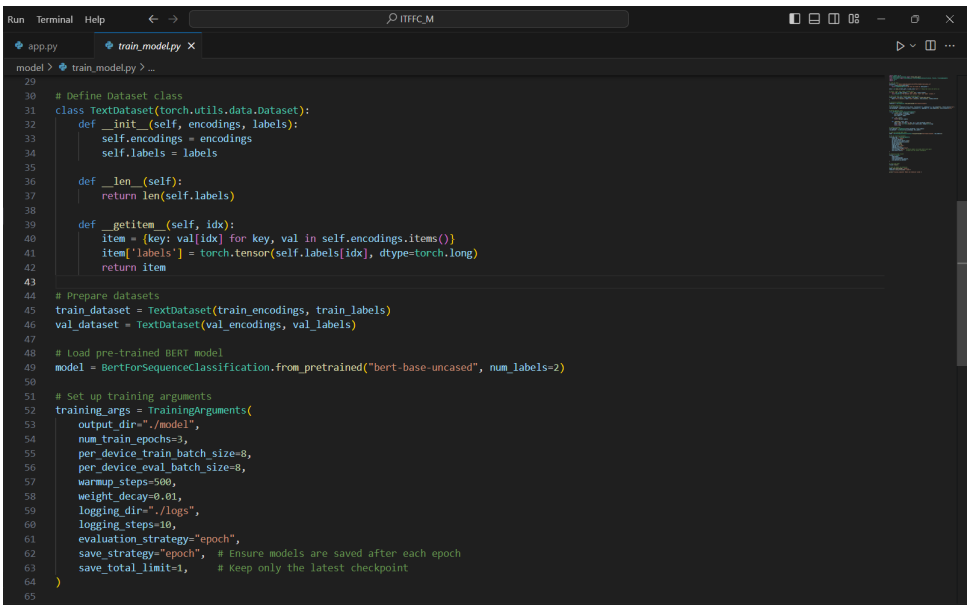

Figure 3: Model training

6 Workflow diagram

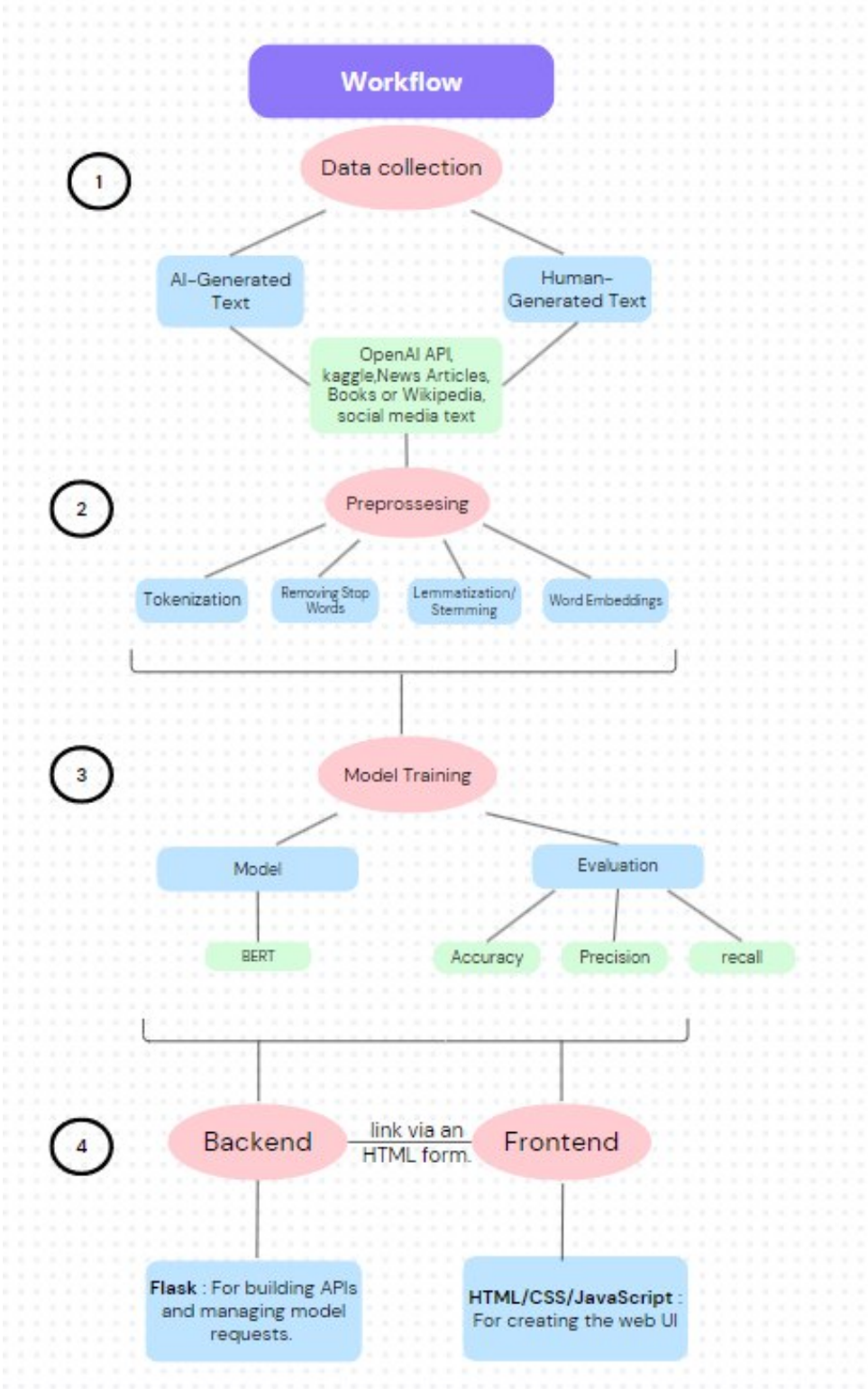

Figure 4: Workflow diagram modeling

7 Comparative Analysis Table

Study	Approach/Method	Model/Algorithm	Dataset	Key Metrics	Findings
GPT-2 Output Detector	Uses fine-tuned transformers	GPT-2 fine-tuned model	OpenAI dataset (Reddit)	Accuracy,Precision, Recall	Fine-tuning GPT-2 improves detection of its own outputs, achieving high accuracy on similar models
RoBERTa for Detection	Fine-tunes RoBERTa on labeled text data	RoBERTa	Custom dataset of AI-generated text	F1 Score, Accuracy	Outperforms traditional methods with a balanced accuracy on longer texts
GLTR (Giant Language Model Test Room)	Statistical anomaly detection based on token likelihoods	BERT-based model	Custom dataset from diverse models	Probability distribution, PPL	Highlights statistical anomalies in AI text, visual aid for human-assisted verification
Turing-NLG Detection	Uses Turing Natural Language Generation model	Turing-NLG-based detection	OpenAI GPT-2 and GPT-3 datasets	Accuracy, Specificity	Provides high specificity in distinguishing text generated by large language models
Grover Detector	Fine-tuned language model specifically for AI-generated news	Grover	AI-generated news articles (Grover dataset)	Accuracy, F1 Score	High accuracy on news detection, emphasizes importance of training on AI-generated news
DetectGPT	Utilizes log probability-based features	Log probability comparison method	GPT-3 generated texts	F1 Score, Accuracy	Employs probability mismatches to flag AI text, works well across different GPT models
OpenAI Classifier	Fine-tuned on diverse text corpora	Transformer-based classifier	Custom dataset from AI and human texts	Precision, Recall, Accuracy	Effective in general scenarios, but less accurate on short and complex texts

Table 1: Comparative Table of Previous Work in Detecting AI-Generated vs. Human-Written Text