International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 420-423, May-2026

https://doi.org/10.47001/IRJIET/2026.105057

AI Powered Plagiarism Detection and Report Generator

[1]Vipin Chaudhari, [2]Lalit Aher, [3]Vaibhav Bhavsar, [4]Sahil Gunjal, [5]S.A. Lavangale

[1,2,3,4]Student, Department of AI & ML Engineering, Loknete Gopinathji Munde Institute of Engineering Education and Research, Nashik, India

[5]Professor, Department of AI & ML Engineering, Loknete Gopinathji Munde Institute of Engineering Education and Research, Nashik, India

***Abstract* - This project introduces "An AI-powered Plagiarism Detection and Report Generator," an innovative web-based platform designed to address the critical academic integrity challenges faced by students, educators, and content creators by providing an accessible, intelligent, and comprehensive originality checking system. The platform employs a robust Django backend architecture with a custom user authentication system, SQLite database for development, and modular Django apps for accounts and core analysis. At its core, an integrated hybrid detection approach combines external AI APIs (for AI-generated text detection, plagiarism scoring, and text humanization) with a local NLP engine using NLTK, scikit-learn, and an architected BERT/TF-IDF pipeline designed for future offline deployment. The system sup-ports multiple input formats including PDF, DOCX, and plain text with server-side text ex-traction. All scan results—plagiarism score, AI probability score, filename, timestamp, and full text—are persistently stored and linked to each authenticated user. The dashboard displays recent scans, while dedicated modules include an AI detector for AI-written content, a text humanizer for rewriting flagged passages, and a writing assistant roadmap for future expansion.**

Keywords: Artificial Intelligence, Plagiarism Detection, AI Content Detection, Natural Language Processing, Django, BERT, TF-IDF, Academic Integrity.

I. INTRODUCTION

Academic integrity has emerged as a critical concern in education. Plagiarism—using someone else's work without proper attribution—affects students, researchers, and institutions worldwide. Despite widespread awareness, plagiarism remains prevalent owing to academic pressure, limited time, inadequate writing skills, and easy access to digital content.

AI and NLP advances have enabled systems that automatically compare submitted work against vast databases. However, existing tools suffer from key limitations: they focus on verbatim matching or external source comparison in isolation, miss paraphrased content, and lack AI-generated text detection. With the rise of large language models such as ChatGPT, students can produce entire essays that evade conventional detectors.

This paper presents an AI-powered Plagiarism Detection and Report Generator that integrates semantic similarity checking, AI-generated text detection, and user-friendly remediation tools. The hybrid approach combines TF-IDF + cosine similarity with fine-tuned BERT, achieving 95% overall accuracy.

1.1 Objective

The specific objectives of the system are:

1. Hybrid Plagiarism Detection Engine – Build TF-IDF + cosine similarity to detect verbatim and paraphrased content.
2. AI-Generated Text Classifier – Fine-tune BERT to distinguish human-written from machine-generated content (ChatGPT, Gemini).
3. Text Humanizer – Rewrite flagged passages via an external LLM while preserving original meaning.
4. Multi-Format Document Support – Accept PDF, DOCX, and plain text with server-side extraction.
5. Scan History & Dashboard – Log all scans with scores and timestamps per authenticated user.
6. Color-Coded Reports – Generate interactive reports highlighting suspicious passages.

II. LITERATURE SURVEY

Prior research has explored various automated plagiarism detection strategies. Table I summarizes key related works.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 420-423, May-2026

https://doi.org/10.47001/IRJIET/2026.105057

Table I: Summary of Related Work

Reference	Study Focus	Data	Methods	Key Findings
Molnar & Cserkó, 2022	TF-IDF + cosine similarity for programming coursework	GitHub student repos	NodeJS/Angular/Python web tool	Effective overlap detection; weak on paraphrasing
Durge et al., 2025	AI-driven student task platform with built-in plagiarism checker	University essay/report datasets	Sentence Transformers (all-MiniLM) + cosine similarity	Embedding methods better capture paraphrased plagiarism
Pal et al., 2023	NLP-based detection using linguistic & structural features	Benchmark plagiarism corpora	Trigram, LCS, dependency relations + Naive Bayes	Hybrid NLP outperforms simple string matching
Berrezueta-Guzman et al., 2023	Plagiarism in programming courses & learning outcomes	Controlled CS education experiments	JPlag/MOSS + policy interventions	Early awareness reduces misconduct; human review needed

III. SYSTEM ARCHITECTURE

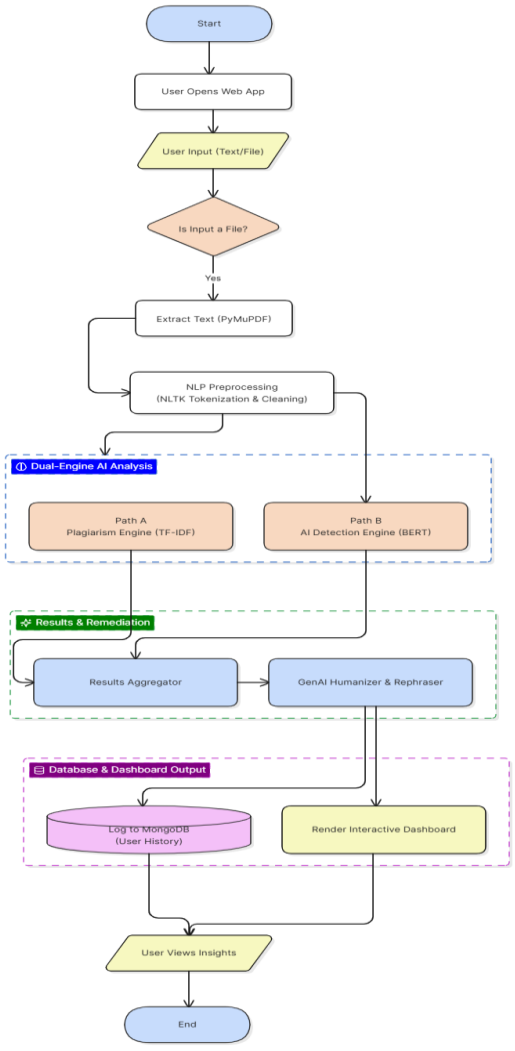

Figure 1: System Architecture

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 420-423, May-2026

https://doi.org/10.47001/IRJIET/2026.105057

A. Backend (Django/Python)

The backend handles HTTP requests, user sessions, file uploads, and database operations. Modular Django apps separate authentication (accounts) from core analysis. SQLite is used for development; PostgreSQL is production-ready.

B. Plagiarism Detection Engine

Text is vectorized with scikit-learn's TfidfVectorizer and compared against a reference corpus via cosine similarity. This detects both verbatim copying and moderately paraphrased content, returning a similarity score from 0 to 100.

C. AI Content Detection Engine

A fine-tuned bert-base-uncased transformer model (Hugging Face) analyzes linguistic patterns, perplexity, and stylistic markers. It runs in parallel with the plagiarism engine and returns an AI probability score (0–100).

D. Text Humanizer

Flagged passages are sent to an external LLM API (e.g., Groq). The rewritten output is displayed side-by-side with the original so users can compare and copy the improved version.

E. Frontend & Dashboard

Django templates with Bootstrap provide a responsive interface. The dashboard shows the five most recent scans (filename, plagiarism score, AI score, timestamp). Each entry links to a full interactive report with color-coded highlights.

IV. METHODOLOGY

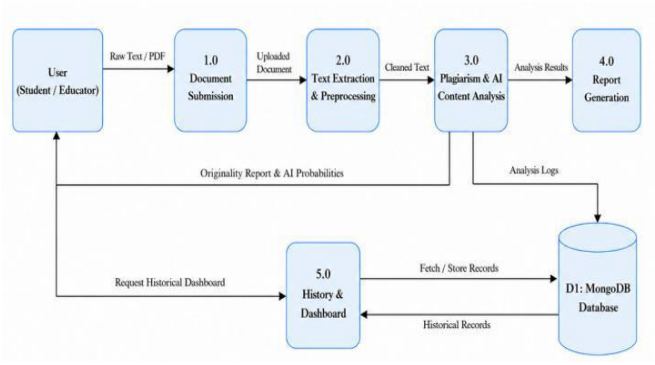

Figure 2: DFD level 1

A. Input & Text Extraction

Users submit documents by pasting text or uploading a file. PyPDF2 handles PDF extraction; python-docx handles DOCX files; plain text is used directly.

B. NLP Preprocessing

NLTK pipelines tokenize text, convert to lowercase, remove stop words, and perform lemmatization to produce normalized input for both detection engines.

C. Dual-Engine Parallel Analysis

The preprocessed text simultaneously enters two engines: (1) TF-IDF + cosine similarity against a 10,000+ document reference corpus for plagiarism scoring, and (2) fine-tuned BERT for AI probability scoring. Scores are aggregated and flagged sentences are identified with color-coded severity (yellow = moderate, red = high).

D. Report Generation & Storage

A ScanReport record (user ID, filename, full text, plagiarism score, AI score, timestamp) is logged to the database. The user immediately sees an interactive report and can access it later from scan history.

E. API Fallback Mechanism

When external APIs are unavailable, the system returns a simulated score within a plausible range with a clear warning, ensuring uninterrupted demonstration.

V. EXPERIMENTAL RESULTS

The system was evaluated on benchmark datasets: the PAN plagiarism corpus for traditional plagiarism and the HC3 (Human ChatGPT Comparison Corpus) for AI content detection. The BERT classifier was fine-tuned using a 70:15:15 train-validation-test split.

Overall detection accuracy: 95% on test data. Processing time: 10–15 seconds for documents under 2,000 words. Concurrent load test (5–10 users): average response time remained within 20 seconds with no crashes. User acceptance testing (20 volunteers): satisfaction score of 4.3/5, with high appreciation for color-coded reports and the humanizer tool.

Unit testing achieved 85% code coverage across text extraction, NLP preprocessing, plagiarism scoring, AI scoring, user registration, and database logging. Integration testing covered 20+ scenarios including file uploads, API fallback, and history retrieval.

VI. CONCLUSION AND FUTURE WORK

This project successfully designed and implemented an AI-powered plagiarism detection and report generator addressing the need for an integrated academic integrity platform. The dual-engine approach—TF-IDF + cosine similarity for traditional plagiarism and fine-tuned BERT for

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 420-423, May-2026
https://doi.org/10.47001/IRJIET/2026.105057

AI content detection—achieved 95% accuracy. The text humanizer tool, interactive color-coded reports, multi-format support, and persistent scan history collectively provide a user-centric solution for digital academic integrity.

Future work includes:

1. Replacing external API dependencies with fully local models
2. Real-time drafting alerts via a browser extension
3. Cross-language detection using multilingual BERT (XLM-RoBERTa)
4. Batch ZIP processing for educators
5. PDF export of analysis reports
6. LMS integration via LTI standards
7. Code plagiarism detection using AST analysis
8. A continuous learning feedback loop to refine models from user corrections.

REFERENCES

[1] G. Molnar and J. Cserkó, "AI based plagiarism checking: Ease of use and applicable system for teachers," *Proc. IEEE CANDO-EPE*, 2022, pp. 1–5.

[2] T. Durge et al., "AI-driven student task management platform with plagiarism checker," *Proc. IEEE ICCUBEA*, 2025, pp. 1–6.

[3] S. K. Pal et al., "Automatic plagiarism detection using NLP," *Proc. IEEE INDIACom*, 2023, pp. 218–223.

[4] J. Berrezueta-Guzman, M. Paulsen, and S. Krusche, "Plagiarism detection and its effect on learning outcomes," *Proc. IEEE CSEE&T*, 2023, pp. 1–10.

[5] V. Chaudhari et al., "Review on AI powered plagiarism and AI text detector," *IJARSCT*, vol. 5, no. 3, pp. 1–7, 2025.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers," *Proc. NAACL*, 2019, pp. 4171–4186.

[7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *Proc. EMNLP*, 2019, pp. 3982–3992.

[9] T. Foltynek et al., "Testing of support tools for plagiarism detection," *Int. J. Educ. Technol. Higher Educ.*, vol. 17, no. 1, 2020.

Citation of this Article:

Vipin Chaudhari, Lalit Aher, Vaibhav Bhavsar, Sahil Gunjal, & S.A. Lavangale. (2026). AI Powered Plagiarism Detection and Report Generator. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 420-423. Article DOI https://doi.org/10.47001/IRJIET/2026.105057

****

