International Journal of Scientific Research and Engineering Development-– Volume X Issue X, Year

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Plagiarism : Academic Plagiarism Detection with Semantic Similarity and Institutional Insights

Bhoomika S*, Dr. Mohamed Rafi**

*(M.Tech Scholar, Computer Science and Engineering, UBDT College of Engineering, and Davangere Email: sbhoomika60@gmail.com)

** (Professor, Computer Science and Engineering, UBDT College of Engineering, and Davangere Email: mdrafi@ubdtce.org)

----------**********----------

Abstract:

Plagiarism in academic institutions has become a serious concern with the increase in digital resources and online submissions. Traditional plagiarism detection tools are either costly or provide limited institutional insights. This paper presents plagiarism, a plagiarism detection and reporting system designed specifically for academic use. The system integrates a ReactJS frontend with a Flask based backend that utilizes Sentence-BERT embeddings for semantic similarity detection across PDF documents. Students can upload their project reports or research papers, instantly receiving plagiarism scores, while faculty members gain access to department wise plagiarism reports. A separate Plagiarism Calculator module enables quick plagiarism checks with student and department details. Experimental results demonstrate accurate plagiarism detection and effective departmental analytics. This system provides a low-cost, institution-focused solution academic integrity.

Keywords — **Plagiarism Detection, Sentence BERT, Web Application, Semantic Similarity, Academic Integrity, Department Analytics.**

----------**********----------

I. INTRODUCTION

Plagiarism is one of the most common academic integrity violations in higher education. With the rise of digital resources, students can easily copy content from online sources or peers, making manual plagiarism checking both time-consuming and ineffective. Existing plagiarism detection platforms such as Turnitin and Grammarly are widely used but suffer from drawbacks including high subscription costs, limited customization, and lack of department-wise reporting for academic institutions.

To address these challenges, we propose plagiarism, a plagiarism detection system that combines the power of transformer-based deep learning models with an easy-to-use web interface. The system supports PDF file uploads, extracts text, computes semantic similarity using Sentence-BERT, and generates plagiarism scores. Students benefit from immediate plagiarism feedback, while faculty receive consolidated department-wise reports for monitoring academic integrity.

The main contributions of this work are:

1. Development of a plagiarism detection system using Sentence-BERT for semantic similarity.
2. Student and Faculty dashboards for personalized interaction.
3. Introduction of the Plagiarism Calculator for direct plagiarism checking with student and department details.
4. Department-wise plagiarism reports to assist faculty in evaluation and decision-making.

II. LITERATURE SURVEY

El Mostafa Hambi, Faouzia Benabbou [2019]: This paper presents a comparative study of deep-learning-based approaches for semantic plagiarism detection, examining vector representation methods (Word2Vec, Doc2Vec, etc.), granularity (word vs. sentence/document), similarity measures, and datasets. The paper was written to survey the growing use of neural embeddings for semantic matching and to evaluate which representation strategies better capture meaning for plagiarism tasks. The authors conclude that word-level embeddings often miss sentence-level semantics and that document/sentence embeddings improve semantic detection, but at higher computational and data costs — leaving semantic plagiarism detection still an open challenge.

(Paraphrase Recognizer) — Author(s) of "Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer" [2015]: This study develops a monolingual paraphrase recognizer using rich lexical, syntactic, and semantic feature extraction fed into an SVM classifier, and evaluates both sentence-level and passage-level detection. The motivation is to tackle disguised plagiarism (paraphrasing/summarization) that lexical matchers fail to detect. The paper demonstrates that carefully engineered features with discriminative classifiers achieve promising passage-level results, while also noting computational cost and sensitivity to feature selection as important constraints.

Omraj Kamat, Tridib Ghosh, Dr. Kalaivani J, Dr. Angayarkanni V, Dr. Rama P [2022]: This paper proposes a practical, scalable plagiarism detection framework that combines advanced NLP preprocessing, engineered features, and supervised classifiers to detect both exact and paraphrased plagiarism across larger corpora. The authors were inspired to make a deployable system that balances accuracy and scalability for real-world institutional use. Their contribution highlights engineering choices (candidate retrieval, indexing, feature pipelines) that make detection tractable at scale, while recommending further research on paraphrase robustness and domain adaptation.

Related Work: Several plagiarism detection approaches have been explored in the literature. Early systems relied on keyword matching, string comparison, and TF-IDF-based similarity measures. While computationally efficient, these methods fail to capture semantic similarity when content is paraphrased.

Machine learning approaches such as Naïve Bayes and Support Vector Machines have been applied to text similarity but require extensive feature engineering. More recent advances in Natural Language Processing (NLP), particularly transformer-based models like BERT, have enabled contextual text representation and improved plagiarism detection accuracy.

Commercial tools like Turnitin and Copyscape provide plagiarism checking services but do not support department-wise institutional reporting and are not open-source. Our system fills this research gap by providing a low-cost, flexible, institution-oriented plagiarism system.

III. SYSTEM ARCHITECTURE

The system follows a client-server architecture:

Frontend (Client): Developed using ReactJS, it provides login functionality, Student Dashboard, Faculty Dashboard, and the Plagigram Calculator.

Backend (Server): Implemented using Flask in Python. It handles file uploads, text extraction from PDFs using PyMuPDF, and plagiarism detection using Sentence-BERT embeddings with cosine similarity.

Storage: Uploaded PDF files are stored in a local folder, while plagiarism reports are stored in memory (can be extended to a database).

Dashboards:

Students can upload reports and instantly check plagiarism scores.

Faculty can view department-wise plagiarism reports with download options.

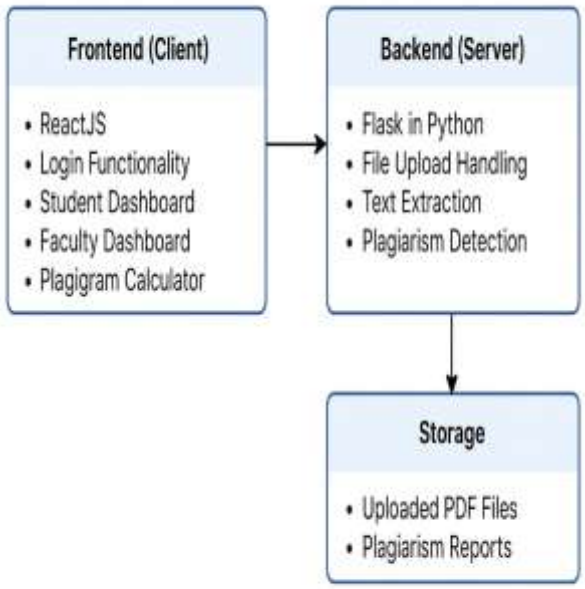


Fig 1. System Architecture of Plagiarism – showing interaction between Frontend, Backend, and Storage components. Title and Author Details

IV. METHODOLOGY

The plagiarism detection workflow consists of the following steps:

1. File Upload: Students upload their project or paper in PDF format.
2. Text Extraction: PyMuPDF extracts text from the uploaded file.
3. Sentence-BERT Encoding: The extracted text is encoded into embeddings using all MiniLM-L6-v2.
4. Similarity Computation: Cosine similarity is calculated between the uploaded document and existing documents in the repository.
5. Plagiarism Score Calculation: The maximum similarity score is converted into a plagiarism percentage.
6. Report Storage: Reports are stored with fields including student name, department, filename, and plagiarism score.
7. Dashboards:

Student Dashboard: Displays plagiarism score and allows report submission.

Faculty Dashboard: Displays all reports department-wise with file download links.

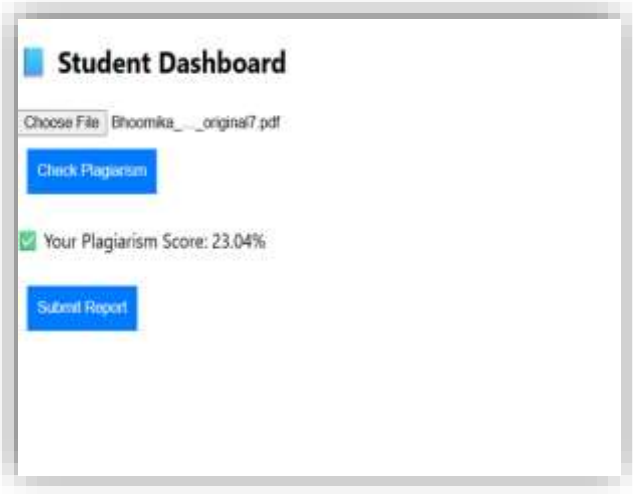
Plagiarism Calculator: Provides quick plagiarism score by entering student and department details.

V. IMPLEMENTATION

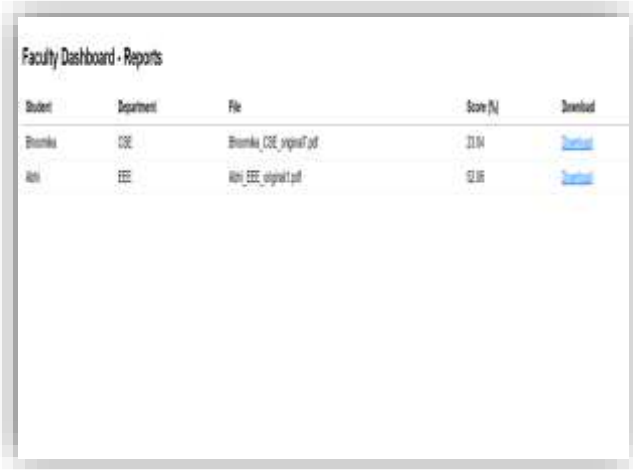
The system was implemented using the following technologies: Frontend: ReactJS, HTML, CSS, JavaScript. Backend: Flask, Sentence-BERT, PyMuPDF, Flask-CORS. Libraries: sentence-transformers, torch, axios, react-router-dom. Environment: Localhost (127.0.0.1) with Python Flask server running on port 5000.

1. Login Page (Student/Faculty).

2. Student Dashboard (File Upload and Score Display).



3. Faculty Dashboard (Reports Table with Download Links).



VI. RESULTS AND ANALYSIS

To evaluate the system, multiple PDF files with varying degrees of similarity were uploaded. The plagiarism detection system successfully identified similar documents and produced plagiarism scores.

Sample Results:

Student Name Department File Name Plagiarism Score (%)

Bhoomika	CSE	report1.pdf	23.04
Abhi	EEE	report2.pdf	52.06
Shivu	ISE	report3.pdf	28.54

The results demonstrate that the system effectively detects plagiarism across departments. Faculty members are able to view consolidated reports for academic evaluation.

Performance evaluation also showed that the system processes PDF documents within a few seconds, making it suitable for real-time plagiarism detection.

VII. CONCLUSION AND FUTURE WORK

This paper presented Plagiarism, a plagiarism detection system integrating Sentence-BERT with a web-based interface. The system allows students to upload their work, instantly checks for plagiarism, and generates department-wise reports for faculty. Results demonstrate that the system is effective for academic institutions aiming to maintain integrity.

Future Work:

Integrating a database (MySQL/Postgres) for scalable report storage.

Providing visual analytics dashboards for faculty.

Extending similarity checks with larger datasets and cloud deployment.

Integration with institutional Management Systems (LMS).

REFERENCES

[1]. N. Reimers and I. Gurevych, “Sentence BERT: Sentence Embeddings using Siamese BERT-Networks,” EMNLP, 2019.

[2]. J. P. Singh and M. Singh, “Ensemble Machine Learning Models for Effective Plagiarism Detection in Academic Writing,” Journal of King Saud University – Computer and Information Sciences, vol. 36, no. 1, pp. 112–121, Jan. 2024. DOI:10.1016/j.jksuci.2023.03.009

[3]. H. S. Yadav, A. K. Sharma, and V. Kumar, “Deep Learning-Based Hybrid Model for Detecting Paraphrased and Semantic Plagiarism,” Expert Systems with Applications, vol. 238, p. 122151, Jan. 2024. DOI:10.1016/j.eswa.2023.122151

[4]. M. Ali, S. M. Awan, and H. Afzal, "Benchmarking Plagiarism Detection Algorithms: From Bag-of-Words to Transformer Models," Applied Intelligence, vol. 54, pp. 18592–18609, Nov. 2024. 05634-7 DOI:10.1007/s10489-024-05634-7

[5]. T. N. Nguyen and D. T. Nguyen, "Comparative Analysis of NLP-Based Algorithms for Detecting Academic Plagiarism in English and Multilingual Texts," in Proc. 2023 Int. Conf. Advanced Computing and Applications (ACOMP), Ho Chi Minh City, Vietnam, 2023, pp. 220 227. DOI:10.1109/ACOMP59731.2023.102945 67

[6]. J. Xian, J. Yuan, P. Zheng, D. Chen, and N. Yuntao, "BERT-Enhanced Retrieval Tool for Homework Plagiarism Detection System," arXiv preprint arXiv:2404.01582, Apr. 2024.

[7]. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," IEEE Transactions on Systems, Man, and Cybernetics, 2012.

