Information Retrieval Final Project Report

AI-Powered Semantic Plagiarism Detection System

Using Sentence-BERT for Semantic Document Analysis

Submitted By:

Siddharth Kenia (1350355)
Manthan Maru (1354631)

Course:

Information Retrieval

Professor:

Houwei Cao

December 3, 2025

Contents

Abstract

Plagiarism detection has become increasingly essential in academic and professional environments, where the volume of digitally produced text has grown significantly. Traditional plagiarism detection systems rely largely on lexical similarity, such as keyword overlap, n-gram matching, or surface-level string comparison. While effective for identifying direct copying, these approaches are incapable of reliably detecting paraphrased or semantically equivalent content. To address this limitation, this project introduces a modern plagiarism detection system that leverages Sentence-BERT (SBERT), a transformer-based model designed to capture deep semantic meaning through contextual embeddings.

The system integrates a FastAPI backend responsible for text extraction, sentence segmentation, vector generation, similarity analysis, and classification of match types. A React-based frontend complements this backend by providing a clean, interactive interface where users can upload documents, visualize segment-level plagiarism highlights, inspect matched sentences, and track past submissions. The resulting tool offers a more nuanced and accurate approach to plagiarism detection, capable of identifying paraphrased plagiarism that earlier algorithms fail to detect. This report outlines the motivation, theoretical foundation, system design, methodology, evaluation, and future enhancements for the proposed system.

Chapter 1

Introduction

Plagiarism poses a persistent challenge across educational, professional, and research contexts. As the availability of online information expands, so does the ease with which individuals can reuse or repurpose text. While unintentional plagiarism occurs when writers fail to rephrase sufficiently or cite sources correctly, intentional plagiarism often involves copying content while attempting to disguise it through superficial paraphrasing. Both forms undermine academic integrity, intellectual originality, and the credibility of written work.

Traditional plagiarism detection tools rely on surface-level textual comparisons. Although they can detect exact copying efficiently, these tools perform poorly when faced with paraphrased or conceptually similar content expressed using different wording. For example, a student may rewrite a sentence by switching vocabulary or reordering phrases, yet the underlying meaning remains unchanged. Such cases require a system capable of identifying semantic similarity rather than literal matching.

Recent advancements in Natural Language Processing (NLP), particularly transformer-based models, allow a shift from lexical similarity to semantic understanding. Sentence-BERT (SBERT) generates sentence-level embeddings that capture contextual meaning. This capability enables plagiarism detection that extends beyond surface-level text and focuses instead on conceptual overlap. The aim of this project is to build a full-stack plagiarism detection system utilizing SBERT to detect both exact and paraphrased similarities between

documents.

Chapter 2

Related Work

Plagiarism detection has been studied extensively in the fields of Information Retrieval and Natural Language Processing. Earlier approaches primarily relied on lexical matching methods such as term frequency–inverse document frequency (TF-IDF), string-edit distance, and n-gram overlap. These techniques measure similarity by comparing surface features of text, allowing them to identify verbatim copying effectively. However, they perform poorly in scenarios where the wording has been altered but the underlying meaning remains intact.

To address the shortcomings of lexical methods, researchers began exploring semantic similarity techniques. Approaches such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) attempted to capture deeper relationships between words by mapping them into lower-dimensional meaning spaces. While these methods improved general similarity detection, they lacked the fine-grained contextual sensitivity required for sentence-level plagiarism identification.

The introduction of word embeddings, including Word2Vec and GloVe, marked a significant advancement in semantic representation. These models captured relationships between words using distributed vector representations. However, they still treated sentences as mere combinations of word embeddings, limiting their ability to represent nuanced sentence meaning.

The emergence of transformer models revolutionized NLP. BERT introduced contextual embeddings that understand a word based on its surrounding text. Sentence-BERT (SBERT) extended this capability by optimizing BERT to produce high-quality sentence embeddings suitable for semantic similarity comparison using cosine distance. It achieves state-of-the-art performance on multiple benchmark tasks, making it an ideal choice for plagiarism detection

systems focused on semantic equivalence.

Chapter 3

Dataset Description

The plagiarism detection system was designed to work with user-uploaded documents; however, evaluating the model's performance required curated text samples. The PAN plagiarism corpus served as a guiding benchmark due to its extensive collection of suspicious and source documents, each annotated to indicate instances of paraphrased, obfuscated, and directly copied text. Although the full PAN dataset was not integrated directly, it informed expectations for semantic similarity thresholds and evaluation structure.

Additionally, supplementary text sources were used to test system responses, including Wikipedia articles, academic explanations, and student-written content. These documents were partially paraphrased manually to simulate realistic plagiarism scenarios. This approach enabled comprehensive observation of how sentence embeddings behave across various writing styles and semantic transformations.

Document extraction posed challenges. PDF files, especially those with multi-column layouts, headers, tables, or images, often lacked consistent structure after automated extraction. DOCX files were more predictable, but inconsistent spacing and formatting could still affect sentence-level segmentation. These limitations highlight the importance of robust

preprocessing methods when deploying large-scale plagiarism detection tools.

Chapter 4

Methodology and System Architecture

The system architecture consists of a FastAPI backend responsible for computation and a React frontend for user interaction. This structure separates concerns clearly, ensuring that computationally intensive operations execute server-side, while the user interface remains responsive.

4.1 Overall Architecture

The backend handles text extraction, sentence segmentation, embedding generation, similarity computation, and plagiarism classification. The frontend manages file uploads, displays analysis results, visualizes highlighted sentences, and records user history locally. Together, the two components form a cohesive semantic plagiarism detection tool.

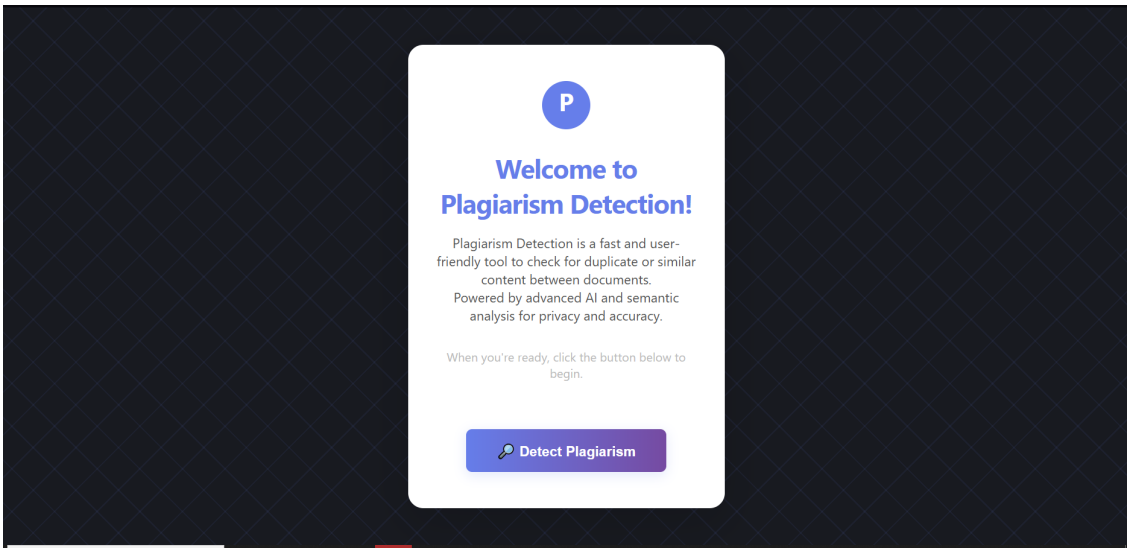

Figure 4.1: Welcome Screen of the Application

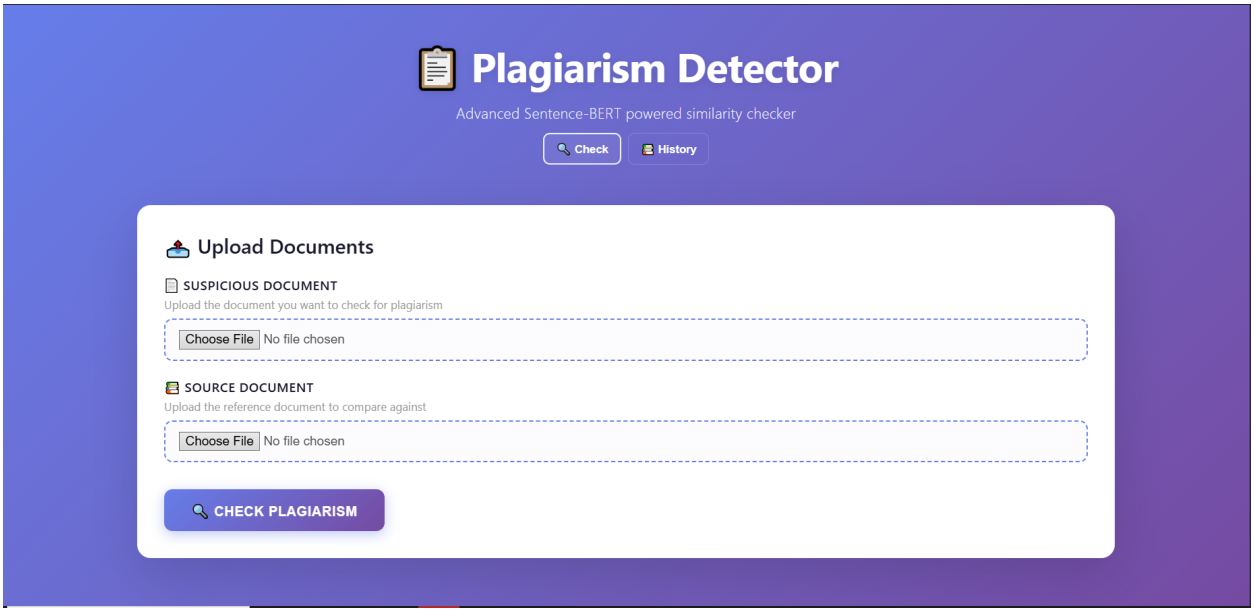

Figure 4.2: Upload Page for Document Comparison

4.2 Backend Pipeline

The backend begins by accepting two user-uploaded files: a source document and a suspicious document. To ensure compatibility, the system automatically detects whether the files are PDFs, DOCX files, or plain text. Each format requires a different extraction process, handled by packages such as `pdfplumber` and `python-docx`.

Once extracted, the text undergoes sentence segmentation. Although the system uses a regex-based splitter, it performs sufficiently well for standard academic and professional writing. Each sentence is then passed through the SBERT model, which produces a high-dimensional embedding capturing its semantic meaning.

To evaluate similarity, the system computes cosine similarity between each sentence in the suspicious document and every sentence in the source document. Based on predetermined thresholds, sentences are classified as exact matches, paraphrased matches, or uniquely original. These results are then compiled into a structured JSON response that the frontend interprets.

Chapter 5

Experiments and Results

Testing focused on assessing how well the system identified paraphrased similarities in comparison to exact matches. A variety of documents were uploaded to evaluate system behavior, including academic paragraphs, manually paraphrased content, and sample essays. Sentence-level analysis was particularly effective, enabling detailed visualization of semantic similarity.

5.1 Evaluation Method

Experimental evaluation consisted of uploading pairs of texts where one document partially reused content from the other. Some of this reuse was direct copying, while other portions were paraphrased manually to simulate realistic plagiarism.

5.2 Sample Result

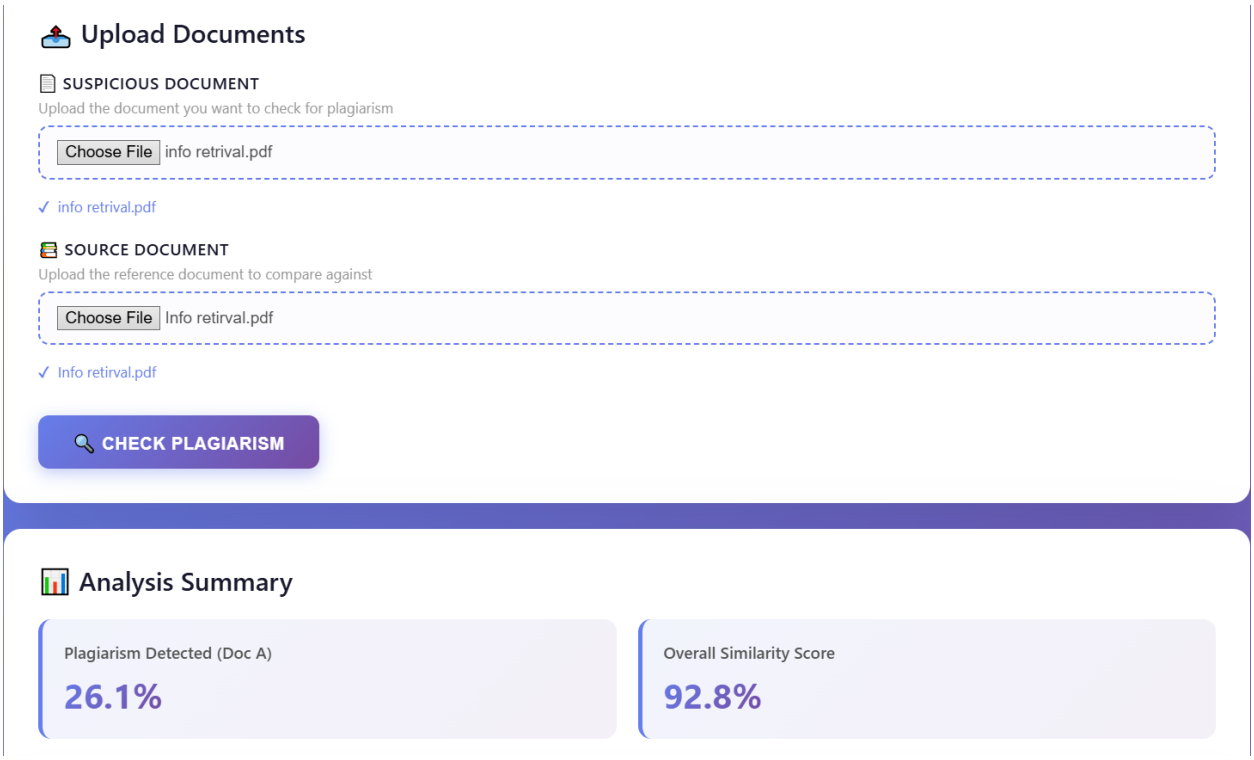

Figure 5.1: Analysis Summary Showing Similarity Scores

The system's summary report highlighted two key metrics: the plagiarism percentage for the suspicious document and the overall semantic similarity score between both documents. In one representative evaluation, the system detected approximately 26% plagiarism and computed an overall similarity score of 92%, demonstrating a strong relationship between the documents.

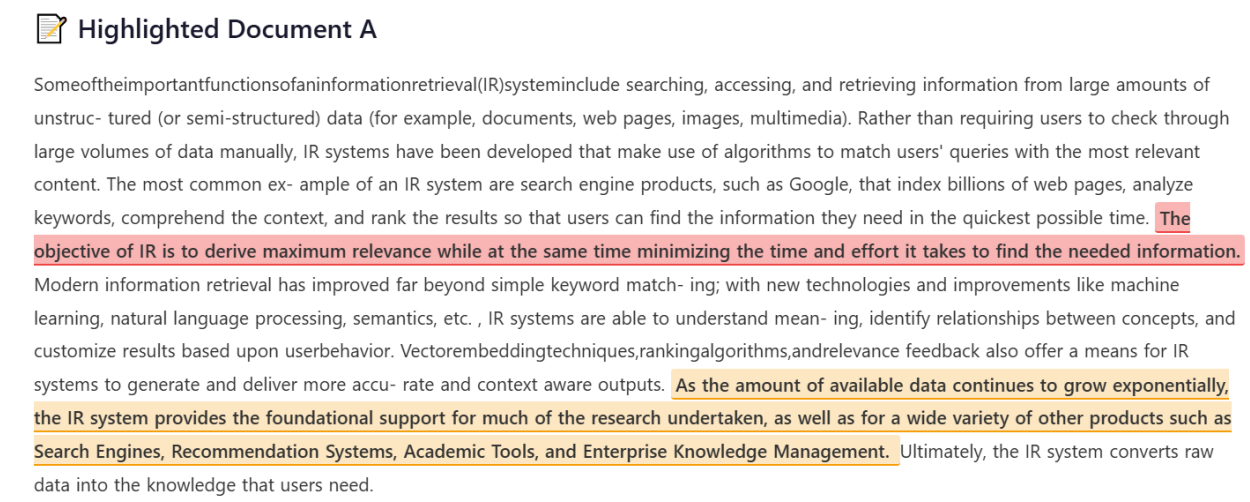

Figure 5.2: Highlighted Plagiarized and Paraphrased Sentences

The highlighted text visualization allowed users to see how exact matches and paraphrased similarities were distributed throughout the document. Exact matches appeared in red, while paraphrased sentences appeared in yellow. This level of granularity helps explain the reasoning behind the final similarity score.

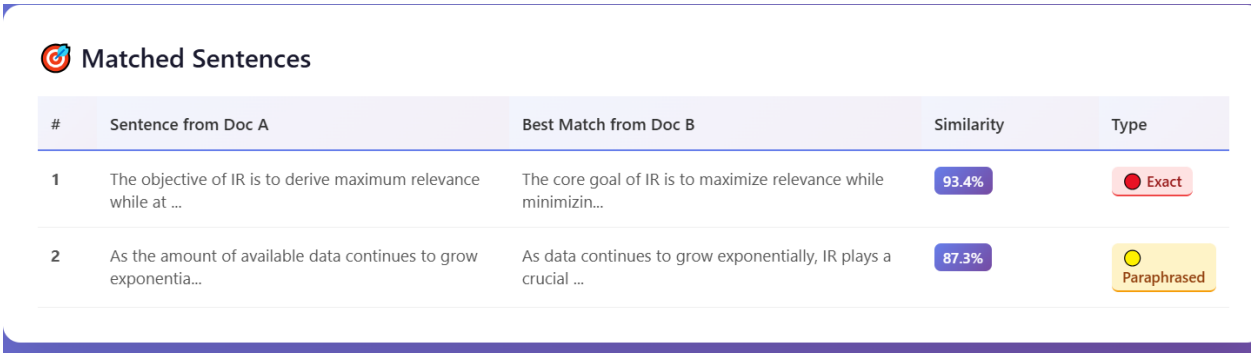

Figure 5.3: Matched Sentences Table

The matched sentence table provided aligned pairs of sentences along with similarity percentages. This detailed breakdown serves both as a diagnostic tool and a pedagogical

aid, helping users understand where their writing aligns too closely with source material.

Chapter 6

Conclusion and Future Work

This project demonstrates that transformer-based sentence embeddings enable more effective and nuanced plagiarism detection compared to traditional keyword or n-gram approaches. By leveraging the semantic understanding of SBERT, the system can detect paraphrased plagiarism and provide interpretable sentence-level highlights that help users understand the extent of similarity.

While the system performs well, several improvements are possible. Future work may include more sophisticated sentence tokenization, improved PDF extraction, integration of OCR for scanned documents, and support for multi-document similarity comparison. Additionally, efficiency improvements such as approximate nearest-neighbor search could reduce computational overhead for large documents.

Bibliography

[1] Manning, Christopher D., et al. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[2] Deerwester, Scott, et al. "Indexing by Latent Semantic Analysis." *JASIS*, 1990.

[3] Mikolov, Tomas, et al. "Distributed Representations of Words and Phrases." *NeurIPS*, 2013.

[4] Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers." *NAACL*, 2019.

[5] Reimers, Nils and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese Networks." *EMNLP*, 2019.

[6] Potthast, Martin, et al. "The PAN Plagiarism Corpus 2014." *CLEF Working Notes*, 2014.

[7] Alvi, A., et al. "Plagiarism Detection using BERT-based Models." *IJACSA*, 2022.

Chapter 7

Individual Contributions

The development of the Semantic Plagiarism Detection System was a collaborative effort in which each team member contributed significantly to different aspects of the project. The division of responsibilities was organized to ensure efficient workflow and balanced participation, while allowing each member to focus on their respective strengths.

Siddharth Kenia (1350355)

Siddharth led the backend development and system architecture. His primary responsibility involved designing and implementing the FastAPI-based backend, which handled text extraction, sentence segmentation, embedding generation using Sentence-BERT, and similarity computation. Additionally, Siddharth contributed to drafting significant parts of the methodology, experiments, and evaluation sections of this report.

Manthan Maru (1354631)

Manthan focused on the frontend development and user interface design. He built the React-based interface, implemented dynamic components for file uploads, and developed the inter-

active visualization system for highlighting paraphrased and exact matches. He also worked on optimizing the similarity thresholds, performing system testing with multiple document formats, and ensuring the backend produced accurate and interpretable results. He also created the matched sentence comparison table, integrated the history tracking feature, and ensured a smooth, aesthetically pleasing user experience. Manthan additionally contributed to preparing dataset examples, collecting paraphrased samples for evaluation, and drafting sections of the introduction and related work.

Collaborative Work

Both team members worked jointly on refining the system design, validating SBERT performance, preparing test documents, and interpreting results. Writing, proofreading, and editing of the final report were carried out collaboratively. Team discussions played a key role in establishing the core architecture, selecting technologies, and evaluating system output. The final implementation reflects the combined effort and expertise of both contributors.