

Adaptive AI-Based Plagiarism Detection System

Ayush Kumar, V. Deepak Kumar, Saurav Das, N. N. Deepti

Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Visvesvaraya Technological University, Bengaluru, India

Published October 2026

Abstract

Traditional plagiarism detection systems normally compute a separate similarity score applying a corrected combination of detection techniques, regardless of the type of plagiarism presence. This approach performs poorly when the nature of text-based overlap varies; for instance, verbatim copying produces high scores across all word-related measures, whereas rephrasing produces low word-related similarity but high meaning-related similarity. This paper presents an adaptive plagiarism detection system that first computes four unaffiliated similarity measures; exact sequence correspondence, TF-IDF cosine similarity, n-gram overlap, and Sentence-BERT meaning-related similarity; then classifies the anticipated plagiarism pattern utilizing a customizable, rule-based classifier, and ultimately re-weights the four measures according to the detected pattern before computing a last and ultimate adaptive score. The system moreover offers sentence-level evidence and a traceable, non-templated explanation.

Keywords: plagiarism detection; natural language processing; adaptive weighting; semantic similarity; Sentence-BERT; TF-IDF; explainable AI

1 Introduction

Because of the increasing amount of digital content being produced in educational and inquiry environments, there is now a need for automated plagiarism detection. Yet the majority of plagiarism detection tools mainly rely on exact or near-exact string matching and thus are unable to pick up on material that has been rephrased or semantically restructured in such a way that the same meaning is conveyed using different terms [1]. Even those systems that combine a number of similarity techniques generally do so by using a single correction formula that is applied in the same way to all pairs of documents, thereby assuming that all kinds of plagiarism produce the same numerical fingerprint. This assumption is not valid in reality: direct copying leads to high similarity scores across all three types of measures—exact, word-related, and meaning-related—while deliberate rewording results in low word-based similarity but still maintains high meaning-related similarity. The present paper proposes and investigates a system that

1. A rule-based, configuration-driven classifier that categorizes document-pair similarity into six patterns (Direct Copy, Partial Copy, Paraphrased Plagiarism, Structural Similarity, Mixed or Complex Similarity, and Low or No Significant Plagiarism) without requiring labelled training data.
2. An adaptive weighting engine that maps each detected pattern to an empirically motivated weight vector across four detection techniques, confirmed to invariably sum to unity.
3. A direct, controlled comparison between adaptive and fixed-weight scoring that isolates and shows the effect of adaptive weighting.

4. An explainability layer that generates natural-language justifications traceable to precise and particular computed values, and sentence-level evidence identifying which origin sentences contributed to a flagged score.
5. An experimental calibration of classification thresholds against measured baseline similarity between separate documents, alternatively than supposed values.

2 Related work

Early versions of plagiarism detection systems like MOSS used string and token matching to identify plagiarised contents [1]. They failed to detect and compare similar contents if it was written in a different way. New models tried to compare semantically similar text that was written in different way. They represented text using dense terms like GloVe [2], and instead of comparing the symbols used in the text, they compared their related meanings. These models formed the basis of sentence and document level meaning comparison. Sentence-BERT uses transformers and is able to create meaningful sentence representations [3]. It uses cosine-similarity to compare semantically similar documents. Models that rely on a single approach for detection have become way more easy to bypass. This approach inspired the use of multiple detectors in the current system. These models focused on adversarial attacks to copyright and plagiarism detection systems [4].

3 Proposed system architecture

3.1 System architecture

The system uses a multi-stage pipeline approach for adaptive plagiarism detection and for providing explanations in its reports. It accepts documents in PDF, DOCX, and TXT format. The document processing section extracts the textual content from the uploaded files and at the same time keeps the original sentence-level text ready for use when generating evidence. When carrying out pre-processing, the text is divided into tokens and cleaned. In order to carry out lexical comparison, the processed text is changed to lowercase, the stop words are deleted, and the tokens are lemmatized. The original sentence-level text is kept separately so that it can be used for sentence-level analysis and for exact verbatim comparisons.

The system then applies four methods for detecting plagiarism separately to both the suspicious document and the reference documents, normalising their outputs to a common scale from 0 to 100. The first of these is an exact-match detector which makes use of Python's `difflib.SequenceMatcher` to compare suspicious sentences with reference sentences and determines the most matching sentence for each comparison. The second technique is a TF-IDF detector that generates vector representations using `scikit-learn` and then calculates cosine similarity to assess lexical similarity. The third is an n-gram overlap detector that combines the unigram, bigram and trigram overlap coefficients in order to detect similarities in sequences of words. The fourth method is a semantic similarity detector that employs the Sentence-BERT model `all-MiniLM-L6-v2` and computes cosine similarity in order to identify text that is semantically similar, including paraphrased material. If the transformer model cannot be loaded, for instance because of no internet connection or because the model resources are not available, the system falls back on a character n-gram TF-IDF as a substitute, thus enabling the pipeline to keep running rather than failing entirely.

The results from the four detectors are then fed into the plagiarism type classification module, where a rule-based classifier assigns the document to one of six plagiarism categories by means of threshold

rules that are defined in an external configuration file. These rules are checked in order of priority until a matching category is found. Once the type of plagiarism has been detected, the adaptive weight selection mechanism chooses the appropriate weight profile. The weights selected take into account the expected importance of each similarity measure for the particular plagiarism pattern in question; for instance, exact matching is given the highest weight in the case of direct copying, while semantic similarity is assigned a higher weight when dealing with paraphrased plagiarism. The adaptive plagiarism score is then computed as a weighted sum of the four detector scores.

The weight profiles associated with each type of plagiarism detected are given in Table 1, and the system also has a fixed baseline weight profile for comparison with the adaptive weighting method. When the plagiarism level is low or none is detected, equal weights are given to all four detectors in order not to favour any particular similarity measure in the absence of a predominant plagiarism pattern.

The module that carries out the sentence-level analysis then finds, for each suspicious sentence, the reference sentence that matches it most closely and keeps the corresponding similarity sub-scores as evidence. This information is passed on to the explainable AI reporting module, which creates natural-language justifications that are traceable and are based entirely on the evidence and the numerical values calculated in the previous stages. The final results—comprising the plagiarism category identified, the adaptive plagiarism score, the sentence-level evidence, and the explanations—are displayed via a Streamlit dashboard and can also be exported in the form of a downloadable PDF report.

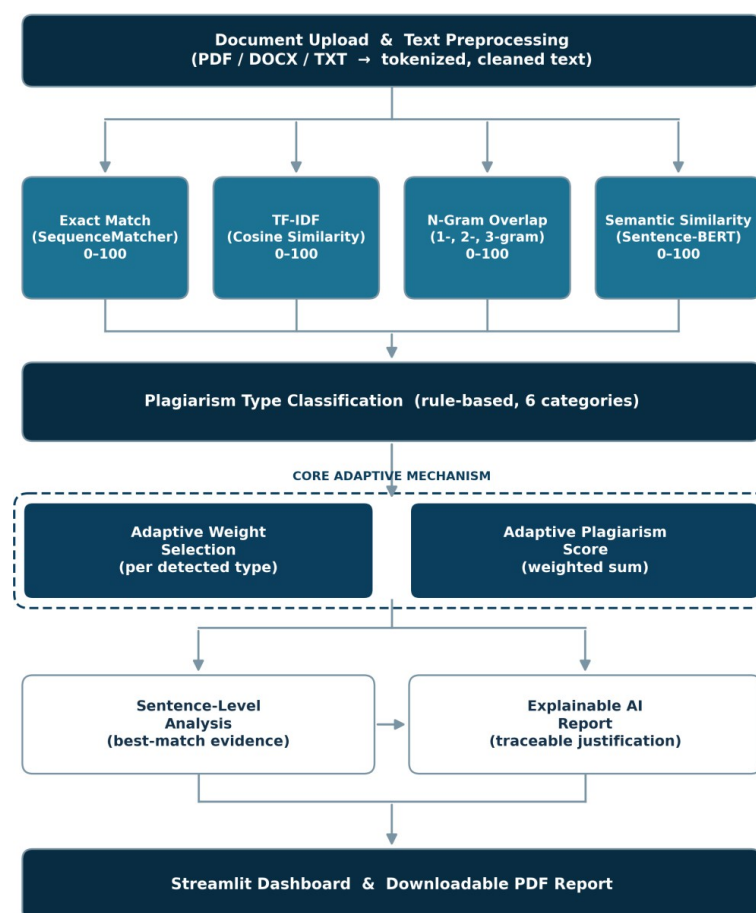


Figure 1. System architecture of the proposed adaptive plagiarism detection and explainable reporting framework.

Table 1. Adaptive weight profiles by detected plagiarism type.

Plagiarism Type	Exact	TF-IDF	N-Gram	Semantic
Direct Copy	40%	30%	20%	10%
Partial Copy	30%	25%	25%	20%
Paraphrased Plagiarism	10%	20%	10%	60%
Structural Similarity	10%	20%	20%	50%
Mixed or Complex Similarity	25%	25%	20%	30%
Low / No Significant Plagiarism	25%	25%	25%	25%
Fixed baseline (traditional)	20%	25%	20%	35%

4 Implementation

The system was built using Python 3.12, with Streamlit used as the front-end framework. The TF-IDF vectorizer and cosine similarity were implemented using scikit-learn, while NLTK was exploited for tokenization and lemmatization. Additionally, sentence-transformers was used for semantic similarity scores. Moreover, results are stored in a SQLite database, while ReportLab was used to generate the reports in PDF format. Finally, thresholds, weight vectors, and classification rules are stored separately in a YAML file, rather than being hardcoded. More specifically, each weight vector's elements sum up to 1.0, which allows for immediate rejection of invalid configurations, thus failing fast and hard, instead of potentially generating invalid results later during analysis.

The thresholds were defined heuristically. However, it is notable that difflib's SequenceMatcher works with sequences of characters, not words. As such, even unrelated English paragraphs will have a certain percentage of similarity due to the similarity of function words and phraseology. Therefore, the lowest threshold was determined by calculating the ratio of three pairs of unrelated documents. The mean value was found to be 32.7%, with individual values being 30.9%, 35.3%, and 31.8%. This value was then used as a threshold for the "Low or No Significant Plagiarism" category, as using a threshold close to 0% would result in most unrelated pairs of documents being classified as having significant plagiarism.

5 Experimental results and discussion

The system is validated in 2 ways, an automated unit-test suite exercising individual components and the controlled document-level experiments comparing adaptive and fixed-weight scorings.

5.1 Unit-level validation

A suite of 25 automated tests (10 of which tested the four detectors in isolation, 7 – the classification-rule behavior, and 8 – the adaptive weighting engine and hybrid scorer) passed successfully. Among the individual tests, one is of specific interest as it creates a paraphrase-shaped score distribution (exact = 15, TF-IDF = 25, n-gram = 15, semantic = 90) and checks that the adaptive score with the Paraphrased Plagiarism weight vector indeed places more emphasis on the semantic similarity than the fixed baseline would, thus proving that the adaptive mechanism favors different scoring distributions. Finally, a similar test was made with the Direct Copy distribution, where all four detectors'

scores were equalized, which showed that adaptive and fixed scoring differ by no more than 5 points in such a scenario.

5.2 Document-level evaluation

Table 2 reports results for four evaluation documents, each compared against a common reference document: a near-verbatim copy, a document interleaving verbatim and original sentences, a substantially reworded paraphrase, and an unrelated document on a different topic.

Table 2. Detector scores and classification outcomes across evaluation documents.

Document	Exact	TF-IDF	N-Gram	Semantic	Classified Type	Adaptive	Fixed	Δ
Direct Copy sample	100.0	88.8	100.0	100.0	Direct Copy (99.0%)	96.6	97.2	-0.6
Partial Copy sample	60.0	32.9	42.4	50.8	Partial Copy (85.5%)	47.0	46.5	+0.5
Paraphrased sample	43.4	35.6	15.2	36.6	Mixed / Complex (62.0%)	33.8	33.5	+0.3
Low-similarity sample	31.5	2.4	1.7	11.1	Low / No Plagiarism (89.5%)	11.7	11.1	+0.6

The Direct Copy sample was correctly classified with high confidence (99.0%); its adaptive and fixed scores are almost identical (respectively, 96.6% and 97.2%), as would be expected given all four detectors already agree strongly, leaving little room for re-weighting to change the outcome. The interleaved-content sample was correctly classified as Partial Copy (85.5% confidence), consistent with its intermediate exact-match score (60.0%) combined with moderate n-gram and semantic overlap. The unrelated document was correctly classified as Low or No Significant Plagiarism (89.5% confidence) and received a low adaptive score (11.7%), correctly avoiding a false positive despite the 31.5% exact-match baseline noise discussed in the Implementation section.

The substantially reworded paraphrase sample was classified as mixed or complex similarity (62.0% confidence) instead of Paraphrased Plagiarism. This reflects a known limitation of the evaluation environment rather than the classification logic itself, semantic scoring for this run used the character n-gram fallback described in the Proposed system architecture section, because the full Sentence-BERT model requires a one-time internet-dependent download that was unavailable in the evaluation environment. The fallback captures shared character sequences and word roots but not deep meaning equivalence, and consequently under-estimates semantic similarity for genuine paraphrases compared to the full transformer model. We report this result transparently as a limitation of the offline fallback path, and expect — consistent with the semantic detector's behaviour on identical and near-identical sentences during unit testing — that evaluation with the full Sentence-BERT model would raise the semantic score sufficiently to trigger the intended Paraphrased Plagiarism classification. Validating this directly on a larger corpus using the full model is identified as future work.

6 Conclusion and future work

This paper presents a unique adaptive plagiarism detection system that classifies likely types of textual overlaps before weighing four independent measures of similarity with varying weights, rather than applying a global fixed formula on all the documents. Automated testings confirms that

this adaptive system yields significantly different scores specifically where detection methods disagree on content, such as paraphrasing, and converge with a traditional fixed-weight baseline where detection methods already agree, such as direct copying. Document-level evaluations on four representative cases correctly identified direct copying, partial copying, and unrelated content with high confidence; evaluating substantially reworded paraphrases with the full Sentence-BERT model, rather than the offline fallback as used in this evaluation environment, is identified as an immediate future work. Future directions include evaluating on a larger, publicly-available plagiarism corpus (e.g., PAN plagiarism detection datasets), extension to languages besides English, and a user study evaluating whether the system's generated explanations improve reviewer trust and review speed compared to a bare similarity.

Acknowledgments

The authors would like to thank Mrs. Deepti N N for her guidance and supervision of this project at Rajiv Gandhi Institute of Technology, Bengaluru, under the Visvesvaraya Technological University Major Project Phase II curriculum (BCSP786).

References

- [1] Aiken A. *MOSS: A system for detecting software plagiarism*. Stanford University. 1994.
- [2] Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. In: *Proc. 2014 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. 2014. p. 1532–1543.
- [3] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proc. 2019 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. 2019. p. 3982–3992.
- [4] Saadatpanah P, Shafahi A, Goldstein T. Adversarial attacks on copyright detection systems. In: *Proc. 2nd Workshop Adversarial Machine Learning in Real-World Computer Vision Applications*. 2019. p. 1–8.
- [5] Qamar, Ali Z, Malik S, Khan AQ, Ahmar M. AI-based plagiarism detection system. *International Journal of Innovations in Science & Technology*. 2025;7(10):184–189.
- [6] Shaikh F, Siddavatam S. AI-based plagiarism detection system. *International Journal of Engineering Research & Technology (IJERT)*. 2026;14(2):1–5.
- [7] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*. 1988;24(5):513–523.
- [8] Cavnar WB, Trenkle JM. N-gram-based text categorization. In: *Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*. 1994. p. 161–169.
- [9] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proc. NAACL-HLT*. 2019. p. 4171–4186.
- [10] Alzahrani MA, Palade V, Salim N. On retrieving intelligently plagiarized documents using semantic similarity. *Engineering Applications of Artificial Intelligence*. 2015;45:246–258.