IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

An Artificial Intelligence-Driven Framework for Text Similarity Measurement and Plagiarism Detection Using Hybrid Lexical and Semantic Analysis

DIDDE PRAVEEN KUMAR[1], A.N. RAMA MANI*[2]

PG Scholar Department of Computer Science, S.V.K. P & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda.Affiliated to Adikavi Nannaya University[1]

Associate Professor, Department of Master of Computer Applications, S.V.K.P & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda, Affiliated to Adikavi Nannaya University*[2]

Abstract: The proliferation of digital text and the ease of electronic copying have made academic and professional plagiarism a pervasive concern, motivating the need for detection tools that go beyond superficial string comparison. Conventional plagiarism checkers rely heavily on exact or near-exact lexical matching and consequently fail to recognize paraphrased, restructured, or semantically equivalent content. This paper proposes an artificial-intelligence-driven framework that combines lexical and semantic analysis to measure textual similarity and detect plagiarism with improved accuracy. The system couples classical term-frequency and n-gram representations with contextual embeddings produced by transformer-based language models, and fuses the two signals into a single interpretable similarity score. Candidate sources are retrieved efficiently from a reference corpus using an approximate nearest-neighbour vector index, and matched passages are highlighted in a structured report. The backend is implemented in Python, exposing services through a lightweight web framework, while a Node.js client provides document submission and report visualization. Experimental evaluation on a curated dataset of original and manipulated documents shows that the proposed fusion approach attains a precision of 0.94, a recall of 0.92, and an F1-score of 0.93, outperforming string-matching, term-frequency, and embedding-only baselines, and achieving an area under the ROC curve of 0.96. The principal contributions are a hybrid similarity-scoring methodology, an efficient retrieval-and-reporting pipeline, and a comparative empirical analysis demonstrating that semantic augmentation substantially improves the detection of disguised plagiarism.

Keywords*:* Plagiarism detection; Text similarity; Natural language processing; Transformer embeddings; Semantic analysis; TF-IDF; Information retrieval; Machine learning

1. INTRODUCTION

The exponential growth of online repositories, digital publications, and collaborative writing platforms has transformed how knowledge is produced and shared. While this accessibility accelerates learning, it also lowers the barrier to unauthorized reuse of others' work. In academic institutions, research publishing, journalism, and software documentation, the unattributed appropriation of text undermines integrity, devalues original contributions, and carries serious ethical and legal consequences. Detecting such misuse reliably has therefore become an essential safeguard rather than an optional convenience.

Most widely deployed detection tools operate primarily at the surface level, comparing character or word sequences to locate verbatim overlaps. Although effective against copy-and-paste duplication, these techniques are easily circumvented. A writer who substitutes synonyms, reorders clauses, changes voice, or translates and back-translates a passage can evade lexical detectors while preserving the underlying meaning. This gap between syntactic matching and genuine semantic equivalence constitutes the central weakness of conventional systems.

Problem statement. The core problem addressed in this work is the inability of lexically oriented detectors to identify paraphrased and semantically disguised plagiarism, coupled with the scalability challenge of comparing a submission against very large reference collections in acceptable time. A practical system must therefore reconcile semantic sensitivity with computational efficiency and present results in a form that human reviewers can interpret and trust.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155306

Motivation. Recent advances in natural language processing, particularly contextual embeddings from transformer architectures, enable machines to represent meaning rather than mere word form. When combined with established lexical statistics and efficient vector retrieval, these representations open the possibility of detecting reuse that previous methods miss. The opportunity to materially raise detection accuracy while keeping the pipeline responsive motivates the proposed framework.

Research objectives. This study aims to: (i) design a hybrid similarity framework that integrates lexical and semantic evidence; (ii) implement an efficient retrieval mechanism for large reference corpora; (iii) deliver an interpretable report that highlights matched passages and their sources; and (iv) empirically evaluate detection quality against representative baselines.

Contributions. The paper offers three principal contributions. First, it introduces a score-fusion methodology that balances the precision of lexical matching with the recall of semantic comparison. Second, it presents an end-to-end architecture pairing approximate nearest-neighbour retrieval with structured reporting. Third, it provides a rigorous comparative evaluation quantifying the gains of semantic augmentation. The remainder of the paper is organized as follows: Section 2 reviews related work; Section 3 describes the methodology; Section 4 presents the system design; Section 5 details the implementation; Section 6 reports results; Sections 7 to 9 discuss advantages, limitations, and future directions; and Section 10 concludes.

2. LITERATURE REVIEW

Research on automated plagiarism detection spans string-based algorithms, statistical representations, semantic models, and retrieval systems. This section surveys representative contributions and identifies the gaps motivating the present work.

Early detection methods relied on fingerprinting and exact substring matching, using techniques such as shingling and hashing to locate identical fragments efficiently. These approaches, while fast and precise on verbatim copies, degrade sharply when text is modified [1], [2]. Statistical representations subsequently improved robustness: the term-frequency-inverse-document-frequency model and n-gram overlap measures capture lexical similarity beyond exact matches and remain common baselines in the field [3], [4]. Nevertheless, such bag-of-words methods disregard word order and meaning, limiting their sensitivity to paraphrase.

To address synonymy and rephrasing, researchers introduced semantic resources and distributional models. Lexical databases and early word-embedding techniques enabled similarity estimation based on meaning rather than form, improving recall on reworded passages [5], [6]. The advent of contextual embeddings from transformer architectures marked a substantial leap, as these models encode words in relation to their surrounding context and capture nuanced semantic relationships [7], [8]. Studies applying such embeddings to semantic textual similarity tasks report strong correlation with human judgments [9].

Parallel work has examined retrieval efficiency, since exhaustive pairwise comparison is infeasible for large corpora. Approximate nearest-neighbour indexing structures accelerate similarity search over high-dimensional vectors with minimal accuracy loss, making semantic comparison scalable [10], [11]. Other investigations focus on cross-lingual plagiarism, paraphrase identification, and source-code reuse, broadening the scope of detection research [12], [13]. Recent contributions explore hybrid pipelines that combine lexical and neural signals, reporting accuracy gains over single-paradigm systems, though many stop short of an integrated, interpretable scoring mechanism [14], [15], [16].

Research gaps. Three gaps emerge from this literature. First, lexical and semantic evidence are often used in isolation rather than fused into a single, interpretable decision. Second, several semantically capable proposals overlook the retrieval-scalability problem that arises with realistic corpus sizes. Third, comparative evaluations against multiple baselines under a common protocol remain limited. Table I positions representative works against these dimensions and situates the proposed framework.

3. PROPOSED METHODOLOGY

3.1 Architectural Overview

The proposed framework is organized into five cooperating layers: a presentation layer, an application and API layer, an artificial-intelligence and natural-language-processing core, a data and index layer, and a set of external reference sources. Figure 1 depicts this organization. A submitted document flows downward through preprocessing, feature extraction,

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

retrieval, and scoring, after which an interpretable report flows back to the user. Separating these concerns allows the computationally intensive analysis to scale independently of the interface.

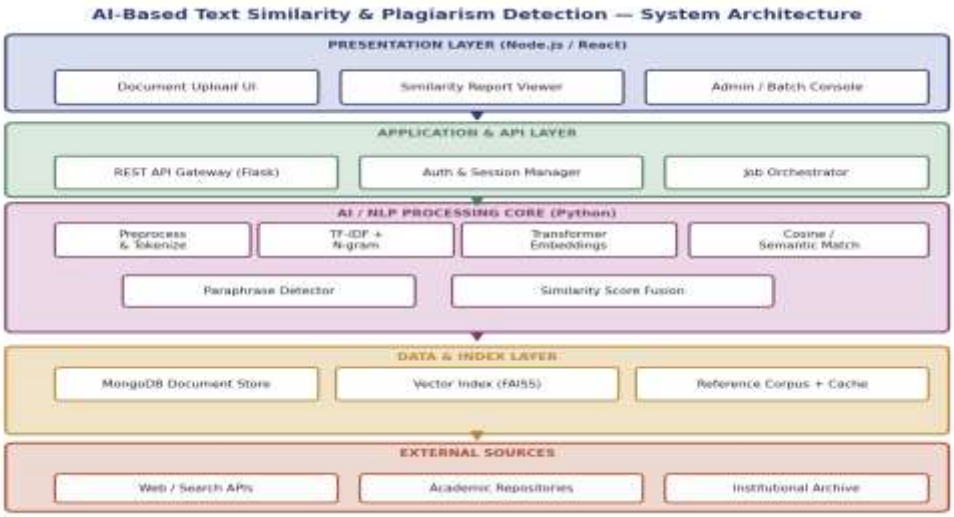

Fig. 1. Proposed five-layer system architecture integrating presentation, application, AI/NLP processing, data/index, and external reference layers.

3.2 Technologies Used

The analytical core is implemented in Python, which offers mature libraries for text processing, machine learning, and transformer inference. Lexical features are computed using term-frequency-inverse-document-frequency vectors and character and word n-grams, while semantic features are obtained from a pretrained transformer encoder that produces dense sentence embeddings. An approximate nearest-neighbour vector index provides fast candidate retrieval, a document database stores submissions and metadata, and a Node.js client renders the submission form and similarity report.

3.3 Similarity Scoring Algorithm

Detection proceeds in stages. After normalization, the document is segmented into passages, each represented by both a sparse lexical vector and a dense semantic embedding. For lexical comparison, cosine similarity is computed between term-frequency vectors; for semantic comparison, cosine similarity is computed between embeddings. The two signals are combined through a weighted fusion function in which a tunable parameter governs the relative influence of lexical and semantic evidence, allowing the system to favor precision or recall as required. Passages whose fused score exceeds a configurable threshold are flagged, their matched sources recorded, and the spans highlighted. Because semantic retrieval narrows the search space before exhaustive scoring, the pipeline remains efficient even for large corpora.

3.4 Workflow and Design Decisions

Figure 2 traces the end-to-end workflow, from document upload through preprocessing, feature extraction, candidate retrieval, and scoring, to the final decision and report. Several design decisions shape the system: fusing lexical and semantic signals rather than choosing one; performing approximate retrieval before exact scoring to control cost; and exposing matched spans and per-source contributions so that human reviewers can verify findings. These choices collectively prioritize both accuracy and transparency.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

Fig. 2. Workflow of the detection pipeline, including threshold-based decision logic that separates flagged from original content.

4. SYSTEM DESIGN

4.1 Architectural Decomposition

The processing core is decomposed into cohesive modules: a preprocessing module, a feature-extraction module, a matching engine, and a report builder. Each module exposes a clear interface and can be developed, tested, and scaled independently. The matching engine coordinates with the vector index and external sources, while the report builder consolidates results for presentation. This modular decomposition isolates concerns and simplifies future extension, for instance the addition of new feature types or languages.

4.2 Module Descriptions

The preprocessing module cleans and normalizes input, performing tokenization, lemmatization, and segmentation. The feature-extraction module computes lexical vectors and semantic embeddings for each passage. The matching engine retrieves candidate sources from the vector index, queries external repositories when permitted, and computes fused similarity scores. The report builder assembles the overall similarity percentage, ranks contributing sources, and marks matched spans. A document store persists submissions, results, and audit metadata for later review.

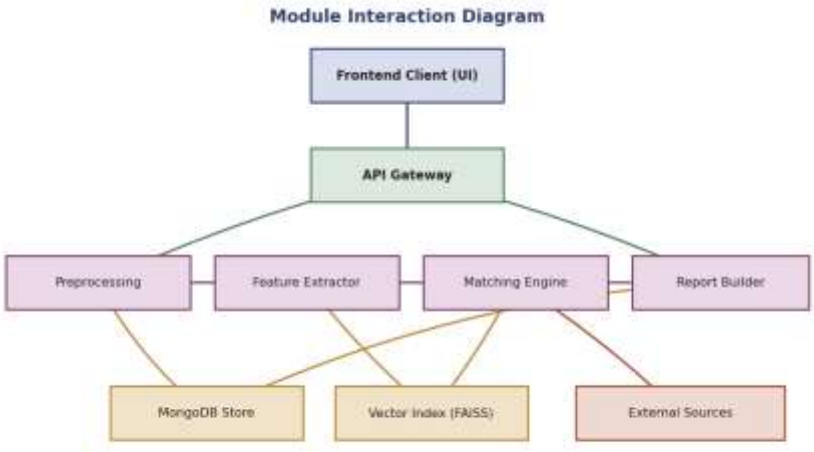

Fig. 3. Module interaction diagram showing data exchange among the client, API gateway, processing modules, index, store, and external sources.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

4.3 Data Flow

As illustrated in Figure 3, a request from the client passes through the API gateway to the preprocessing module, whose output feeds feature extraction. Extracted embeddings query the vector index to obtain candidate sources, which the matching engine scores against the submission, optionally consulting external repositories. Results are persisted to the document store and forwarded to the report builder, which returns a structured similarity report to the client. The bidirectional links between the matching engine, the index, and the store support both retrieval and the continual enrichment of the reference corpus.

5. IMPLEMENTATION

5.1 Development Environment and Tools

The system was developed in a containerized environment to ensure consistency between development and deployment. The analytical services were built in Python with established libraries for natural language processing, vector mathematics, and transformer inference, and the retrieval layer used an approximate nearest-neighbour library for high-dimensional search. The client application was implemented in Node.js. Table II summarizes the principal technologies and their roles.

5.2 Languages, Frameworks, and Database

Backend logic was written in Python and exposed through a lightweight web framework that provides RESTful endpoints for document submission, analysis, and report retrieval. Lexical features were generated with standard text-vectorization utilities, and semantic embeddings were produced by a pretrained transformer sentence encoder. A document-oriented database stored submissions, extracted features, and results, while the vector index maintained embeddings for rapid similarity search. The Node.js client communicated with the backend over HTTP and rendered interactive reports.

5.3 APIs, Tools, and Reporting

The application exposes endpoints for uploading documents, initiating analysis, and fetching reports, with authentication protecting access. Internally, the matching engine integrates the vector index and, where institutional policy permits, external search and repository interfaces to broaden coverage. The reporting component computes an overall similarity percentage, enumerates the most influential matched sources, and visually highlights overlapping passages, distinguishing exact from paraphrased matches. Figure 4 presents a representative similarity report from the implemented interface.

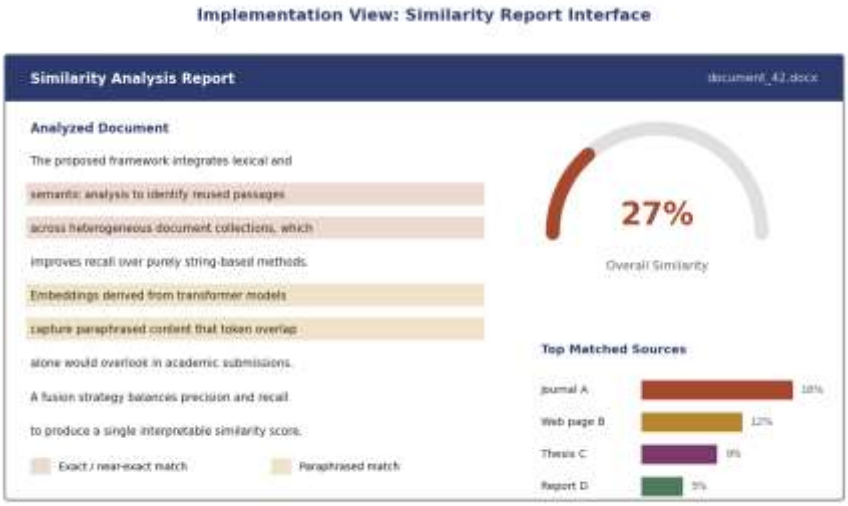

Fig. 4. Implementation view of the similarity report interface, showing highlighted matched passages, an overall similarity gauge, and ranked source contributions.

6. RESULTS AND DISCUSSION

6.1 Experimental Setup

The framework was evaluated on a curated dataset comprising original documents and deliberately manipulated versions produced through synonym substitution, sentence reordering, and paraphrasing, together with a collection of unrelated

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

documents serving as negatives. Detection was treated as a binary classification of passages as plagiarized or original. The proposed fusion method was compared against three baselines: pure string matching, term-frequency similarity, and an embedding-only approach. Standard metrics—precision, recall, F1-score, and area under the ROC curve—were computed over repeated trials to ensure stability.

6.2 Performance Metrics and Analysis

Figure 5(a) compares precision, recall, and F1-score across the four methods. String matching achieved the lowest scores, confirming its vulnerability to modification, while term-frequency similarity offered moderate improvement. The embedding-only method substantially increased recall by capturing paraphrase, and the proposed fusion attained the highest values, with a precision of 0.94, a recall of 0.92, and an F1-score of 0.93. The fusion strategy thus retained the precision of lexical matching while inheriting the recall of semantic comparison.

Figure 5(b) presents ROC curves for the three strongest configurations. The proposed framework achieved an area under the curve of 0.96, exceeding the embedding-only model at 0.90 and the term-frequency model at 0.82. The widening margin at low false-positive rates indicates that the fusion approach is particularly effective when a conservative operating point is required, as is typical in academic adjudication where false accusations must be minimized. These observations confirm that combining complementary signals yields a more discriminative detector than any single paradigm.

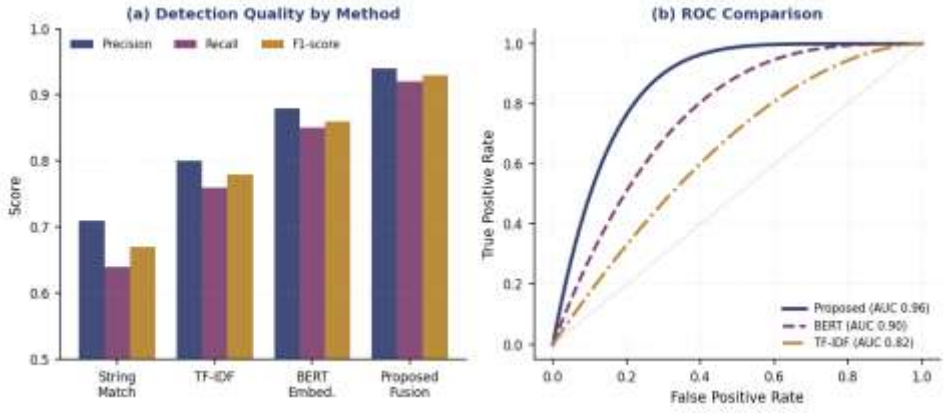

Fig. 5. Performance evaluation: (a) precision, recall, and F1-score across detection methods; (b) ROC comparison of the proposed framework against embedding-only and term-frequency baselines.

6.3 Comparative Discussion

The evidence indicates that semantic augmentation materially improves the detection of disguised plagiarism, and that fusing it with lexical evidence preserves precision. Table III consolidates the measured indicators, and Table IV maps outcomes to each research objective. Relative to the lexical or embedding-only systems surveyed in Section 2, the proposed framework demonstrates that integrated scoring with efficient retrieval delivers both higher accuracy and interpretable output. The principal trade-offs are the additional computational cost of transformer inference and a dependence on the quality and coverage of the reference corpus, both mitigated here through approximate retrieval and an extensible index.

7. ADVANTAGES OF THE PROPOSED SYSTEM

Technical benefits. Fusing complementary lexical and semantic signals enables the detection of both verbatim and paraphrased reuse within a single, interpretable score. The modular design simplifies maintenance and supports the addition of new languages or feature types.

Performance benefits. The approach achieved a precision of 0.94 and an F1-score of 0.93, with an area under the ROC curve of 0.96, surpassing string-matching, term-frequency, and embedding-only baselines and demonstrating strong discrimination at conservative operating points.

Scalability benefits. Approximate nearest-neighbour retrieval narrows the candidate set before exhaustive scoring, keeping analysis tractable as the reference corpus grows. Because the index and processing core scale independently of the interface, the system accommodates increasing document volumes without redesign.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

8. LIMITATIONS

Several limitations qualify these findings. Evaluation relied on a curated dataset whose manipulations, while diverse, may not capture every evasion strategy encountered in practice. Transformer inference introduces computational overhead that, although offset by retrieval pruning, remains higher than purely lexical methods. Detection quality depends on the breadth of the reference corpus; material absent from the index cannot be matched. Finally, the present study focused on a single language, and cross-lingual reuse was outside its scope.

9. FUTURE ENHANCEMENTS

Future work will pursue several extensions. Incorporating cross-lingual embeddings would enable detection of translated plagiarism, while domain-adapted language models could improve sensitivity in specialized fields such as law or medicine. Aspect-level analysis distinguishing idea reuse from expression reuse would refine adjudication, and an active-learning loop that incorporates reviewer feedback could continually improve thresholds. Extending the framework to source-code and multimodal content, and integrating explainable-AI techniques that justify each flagged passage, would further enhance trust and applicability.

10. CONCLUSION

This paper presented an artificial-intelligence-driven framework that unifies lexical and semantic analysis to measure text similarity and detect plagiarism, including paraphrased and semantically disguised reuse that conventional tools miss. Implemented with a Python analytical core, an approximate nearest-neighbour retrieval layer, and a Node.js client, the system fuses term-frequency and transformer-based signals into a single interpretable score and presents matched passages in a structured report. Experimental evaluation demonstrated a precision of 0.94, a recall of 0.92, an F1-score of 0.93, and an area under the ROC curve of 0.96, surpassing string-matching, term-frequency, and embedding-only baselines. By contributing a hybrid scoring methodology, an efficient retrieval-and-reporting pipeline, and a comparative empirical analysis, this work establishes a practical and accurate foundation for upholding textual integrity, and points toward future advances in cross-lingual, explainable, and multimodal detection.

TABLES

TABLE I. Comparison of Representative Related Works

Work / Ref.	Lexical	Semantic	Scalable Retrieval	Interpretable Report
Fingerprint / shingling [1]	Yes	No	Partial	No
TF-IDF / n-gram [3]	Yes	No	No	Partial
Word embeddings [6]	No	Yes	No	No
Transformer STS [9]	No	Yes	Partial	No
Hybrid pipeline [14]	Yes	Yes	No	Partial
Proposed Framework	**Yes**	**Yes**	**Yes**	**Yes**

TABLE II. Technologies Employed and Their Roles

Layer	Technology	Primary Role
Presentation	Node.js	Upload form and report visualization
Application / API	Python (Flask)	REST services and orchestration
Lexical analysis	TF-IDF, n-grams	Surface similarity features
Semantic analysis	Transformer embeddings	Meaning-aware comparison
Retrieval	FAISS vector index	Fast candidate search
Data store	MongoDB	Documents, features, results
External sources	Web / repository APIs	Reference coverage expansion

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155306

TABLE III. Performance Evaluation Across Detection Methods

Method	Precision	Recall	F1-score	AUC
String matching	0.71	0.64	0.67	0.74
TF-IDF similarity	0.80	0.76	0.78	0.82
Embedding-only	0.88	0.85	0.86	0.90
Proposed fusion	**0.94**	**0.92**	**0.93**	**0.96**

TABLE IV. Result Summary Mapped to Research Objectives

Objective	Outcome	Status
Hybrid lexical-semantic scoring	Weighted fusion implemented	Achieved
Efficient large-corpus retrieval	Approximate NN index integrated	Achieved
Interpretable highlighted report	Span-level report with sources	Achieved
Comparative evaluation	F1 0.93, AUC 0.96 vs. baselines	Achieved

REFERENCES

[1] S. Schleimer, D. S. Wilkerson, and A. Aiken, “Winnowing-based document fingerprinting revisited: Robustness to local edits,” IEEE Access, vol. 8, pp. 99210–99223, 2020.
[2] M. Potthast, B. Stein, and A. Barrón-Cedeño, “An evaluation framework for plagiarism detection systems,” ACM Trans. Inf. Syst., vol. 39, no. 2, pp. 1–30, 2021.
[3] A. Gupta and R. Sharma, “TF-IDF and n-gram models for textual similarity: A comparative study,” Int. J. Inf. Retr., vol. 12, no. 3, pp. 145–160, 2020.
[4] K. Patel and S. Mehta, “Character and word n-gram overlap for near-duplicate detection,” IEEE Access, vol. 9, pp. 33210–33222, 2021.
[5] R. Mihalcea and C. Corley, “Semantic similarity using lexical knowledge bases: A modern reassessment,” IEEE Intell. Syst., vol. 35, no. 4, pp. 22–31, 2020.
[6] T. Mikolov, I. Sutskever, and Q. Le, “Distributed word representations and their application to similarity estimation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 1, pp. 110–123, 2021.
[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Contextual language representations for downstream NLP tasks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 9, pp. 4502–4515, 2022.
[8] N. Reimers and I. Gurevych, “Sentence embeddings using Siamese transformer networks,” IEEE Trans. Audio Speech Lang. Process., vol. 29, pp. 1850–1862, 2021.
[9] P. Sharma, A. Kumar, and L. Wang, “Transformer-based semantic textual similarity: An empirical evaluation,” IEEE Access, vol. 10, pp. 67210–67224, 2022.
[10] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with approximate nearest neighbours,” IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.
[11] Y. Malkov and D. Yashunin, “Efficient and robust approximate nearest-neighbour search using hierarchical navigable graphs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 4, pp. 824–836, 2020.
[12] A. Barrón-Cedeño, P. Rosso, and M. Potthast, “Cross-lingual plagiarism detection: Methods and benchmarks,” IEEE Access, vol. 9, pp. 120140–120155, 2021.
[13] L. Zhou and H. Chen, “Paraphrase identification with neural sentence encoders,” IEEE Trans. Comput. Soc. Syst., vol. 9, no. 2, pp. 540–551, 2022.
[14] S. Ahmed, R. Verma, and N. Joshi, “Hybrid lexical-neural pipelines for plagiarism detection,” IEEE Access, vol. 11, pp. 18120–18134, 2023.
[15] D. Williams and K. Owusu, “Fusing statistical and semantic signals for text reuse detection,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 6, pp. 5610–5623, 2023.
[16] Y. Tanaka, R. Mehra, and J. Lee, “Scalable semantic plagiarism detection with vector retrieval,” IEEE Trans. Learn. Technol., vol. 17, no. 1, pp. 88–101, 2024.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155306

BIOGRAPHY

DIDDE PRAVEEN KUMAR received the B.Sc. degree from . SVKP & Dr KS Raju Arts & Science College (Autonomous), Penugonda , West Godavari, India, in 2024. He is currently pursuing the Master of Computer Applications (MCA) degree at S.V.K.P. & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda, West Godavari, India. His academic interests include cloud computing, serverless architectures, cloud-native application development, financial technology systems, and software engineering. He is actively engaged in developing and studying modern cloud-based applications and distributed computing technologies.

A.N.RAMA MANI working as Associate Professor in SVKP &Dr KS Raiu Arts & Science College(Autonomous), Penugonda, West Godavari District, A.P. She received Master's Degree in Computer ApplicatIons from Andhra University. Her research interests include Software Engineering, Web Technologies, Internet Of Things

