

AI-Based Plagiarism Checker: A Review of Text Similarity, Machine Learning, Deep Learning, and AI-Obfuscated Plagiarism Detection

Gavad Yash Laxman, Prof. Abhishek Popat Bangar, Prof. S. B. Bhosale, Prof. Karad V. A.

Department of Computer Engineering
Jaihind College of Engineering, Kuran, Maharashtra, India

Abstract — Digital documents, paraphrasing tools and large language models have made plagiarism much harder to catch than plain copy-and-paste matching. This review looks at five research papers that approach academic plagiarism detection from different directions: a systematic literature survey, text-based feature selection, deep learning, a comparison of detection techniques, and the detection of AI-obfuscated plagiarism in modeling assignments. Between them, the studies draw on lexical overlap, sentence structure, semantic similarity, writing style and token sequences taken from structured modeling artifacts. One study pairs Chi-square feature selection with a Support Vector Machine (SVM) and reports F-measure scores of 89.34% on PAN 2013 and 92.95% on PAN 2014; another reports that its LSTM-based system placed first among the systems it was compared with on the same benchmarks. A more recent approach for modeling assignments normalizes tokens, matches them automatically and leaves the final judgement to a human inspector, which makes it more resistant to AI-based obfuscation. Taken together, the papers show that no single similarity measure can handle every kind of plagiarism. A practical AI-based checker should therefore combine fast candidate retrieval, several text and structure features, carefully validated machine-learning models and human review. Open problems include resistance to paraphrasing, coverage of datasets and languages, fair evaluation, and the danger of reading a detector score as conclusive proof.

Keywords— AI-based plagiarism checker, plagiarism detection, natural language processing, machine learning, deep learning, semantic similarity, SVM, LSTM

I. INTRODUCTION

Plagiarism means presenting someone else's words, ideas, results or other original work as one's own, without proper credit. Digital access has made copying and reuse easy, but it has also made automatic detection more practical. Modern plagiarism, however, goes well beyond verbatim copying. Text can be disguised by swapping synonyms, rearranging sentences or rewriting the meaning, and AI tools can blur the link between a source and the document that was handed in. The systematic review by Foltýnek, Meuschke and Gipp traces progress in semantic analysis, non-textual feature analysis and machine learning, and also points out weaknesses in how detection systems are evaluated [1].

A dependable checker therefore has to look for several levels of similarity instead of relying on one matching rule. Lexical methods are good at direct copying, syntactic methods look at how sentences are built, semantic methods compare meaning, and stylometric methods watch for shifts in writing style. In computer science teaching, the problem can also extend to source code and structured modeling artifacts, where ordinary text comparison may fall short [1], [5].

This paper brings together five supplied studies and compares their methods, reported results and limitations, and then draws out what they mean for an AI-based plagiarism checker. The scope is deliberately limited to those sources; it is a focused narrative review, not a fresh systematic search or an independent experiment.

Review approach and scope. We use a focused narrative synthesis of the five supplied publications and make no claim to have run a new systematic search. The sources play different roles: one broad literature review supplies the taxonomy, two studies describe feature-based and deep-learning systems, one compares text-based techniques, and one deals with AI-obfuscated plagiarism in modeling assignments. Findings are given as the original authors state them, and scores from different tasks are not treated as directly comparable.

Each study was read with the same questions in mind: what it tries to detect, how documents are represented, which analytical method is used, what evidence supports it, and what practical limits apply. We did not reproduce the experiments, re-run any model or independently check the original datasets. The architecture and recommendations that follow are thus a

synthesis of the supplied evidence and not a newly tested implementation.

Table I. Detection layers and their complementary roles

Detection layer	Typical signal	Strength / limitation
Lexical	Exact strings, n-grams, fingerprints	Quick on copied text; loses power once the text is rewritten.
Syntactic	Part-of-speech patterns, sentence structure	Catches some structural edits; depends on the language.
Semantic	Embeddings, contextual similarity	Can spot paraphrases that keep the meaning; thresholds need care.
Feature-based ML	Selected similarity features + SVM	Feature contributions are easy to interpret; needs representative labels.
Deep learning	Learned mixes of lexical and semantic cues	Flexible, but data-hungry and in need of independent validation.
Structure-aware	Normalized tokens or trees for models and code	Suits non-prose artifacts; needs domain-specific parsing.

II. REVIEWED STUDIES AND APPROACHES

1. Systematic Review of Academic Plagiarism Detection

Foltýnek, Meuschke and Gipp examined 239 research papers published between 2013 and 2018 and sorted the detection approaches into technically oriented categories [1]. The review spans lexical, syntax-based, semantic and idea-based methods, and also covers non-textual elements such as citations, images and mathematical content. Lexical techniques such as n-gram comparison work well against direct copying, but they struggle with heavily disguised text unless paired with deeper language analysis. The authors see machine learning combined with a mix of analysis methods as a promising path, while stressing that evaluations need to be more methodologically rigorous [1].

This paper gives the broad conceptual footing for our review: a checker should blend different signals and report its performance openly, instead of treating one similarity score as a complete verdict.

2. Feature Selection and SVM-Based Detection

El-Rashidy and colleagues describe a text plagiarism detection system built from preprocessing, feature selection and classification [2]. Their supervised dataset is made of 34

sentence-similarity features meant to capture lexical, syntactic and semantic traits. Chi-square ranking picks out the most discriminative features, and an SVM then builds a hyperplane to separate similar from dissimilar cases. The pipeline has a traditional paragraph-level comparison path and a classifier-based path, followed by post-processing that identifies suspicious segments and merges them [2].

On the full PAN 2013 and PAN 2014 test corpora, the authors report Plagdet scores of 89.12% and 92.91% and F-measure scores of 89.34% and 92.95%, respectively [2]. The results suggest that feature selection can lessen the need for hand-tuned feature weights. Still, these figures belong to the study's own datasets and test setup, and they should not be assumed to hold for every language, subject area or AI-generated paraphrase.

3. Deep Learning-Based Detection

In a related study, the same group builds a database of 42 text-similarity features covering lexical, syntactic and semantic cases, and compares densely connected convolutional neural network approaches with a long short-term memory (LSTM) model [3]. The models are tested on PAN 2013 and PAN 2014 data using precision, recall, F-measure, granularity and Plagdet. The authors report that their LSTM-based system ranked first against the state-of-the-art systems included in their comparison [3].

The study supports the use of neural models for merging several similarity signals and learning decision boundaries. At the same time, results depend on the training data, the way features are built and the test design. Topping a benchmark does not, on its own, show robustness to every current generative model, to cross-language plagiarism or to documents outside the evaluation corpus.

4. Comparative Analysis of Text-Based Techniques

Sajid and colleagues compare text-based plagiarism detection techniques and review 189 research papers published between 2019 and 2024 [4]. They discuss the growing need to tell human-written text from AI-generated text, and they name machine learning, semantic analysis and non-textual content as areas of active development. Their conclusion is that combining analytical methods for textual and non-textual features is a promising way to improve detection [4].

The paper also raises a useful caution for AI-assisted checking: classifying text as AI-written and detecting plagiarism are related but separate tasks. A classifier that guesses whether prose came from an AI does not show that the content was

copied from any particular source. A plagiarism checker needs evidence of a source relationship, and any AI-origin prediction should be shown as a separate, uncertain signal.

5. Detection of AI-Obfuscated Plagiarism in Modeling Assignments

Sağlam and colleagues tackle plagiarism in modeling assignments, where artifacts such as UML models can be altered to hide copying [5]. Their method tokenizes the modeling artifacts, normalizes the token sequences to resist obfuscation, performs pairwise matching with a similarity calculation, and visualizes candidate matches so a person can inspect them. They evaluate it on real assignments and on plagiarism disguised with ChatGPT, and report better detection of AI-generated attacks and broader resilience than the state-of-the-art approaches they compare against [5].

A key contribution is the pairing of automated analysis with human judgement. It also widens the idea of a checker beyond ordinary prose, since model-based and other tree-structured artifacts may call for representations built around their structure. The results belong to the modeling-assignment setting and should not be carried over directly to essays or research papers.

III. COMPARATIVE DISCUSSION

Taken together, the five studies describe complementary layers of a detection system. The systematic review [1] provides the taxonomy; the SVM approach [2] shows feature selection and supervised classification at work; the deep-learning study [3] explores neural architectures; the comparative survey [4] stresses the variety of text-based techniques and the changing AI landscape; and the modeling-assignment study [5] deals with obfuscation in structured artifacts.

Lexical matching is still a quick way to find copied passages, but it is easily defeated by heavy rewriting. Syntactic and semantic features can help expose transformed passages, and machine-learning classifiers can weigh many features together. Deep learning may pick up complex relationships, though it needs representative data and careful validation. For structured submissions, token normalization and domain-specific representations can suit better than prose-only similarity. Across all of this, the reviewed literature supports a hybrid design more than it supports any one universal algorithm.

The reported numbers cannot be lined up directly, because the studies differ in task, dataset, feature definitions and metrics. For instance, the F-measure and Plagdet results in [2] come from PAN corpora, whereas [5] looks at AI-obfuscated

modeling assignments. For that reason, this review does not rank the approaches by setting their scores side by side as though they measured the same thing.

In practice, retrieval and judgement should be kept apart. Retrieval narrows a large repository down to plausible source documents, and a costlier comparison stage then examines the candidate passages closely. A high document-level similarity score can hide whether the overlap comes from a bibliography, a properly quoted passage, standard terminology or unattributed reuse. Reports should therefore show source-linked spans and explain which signals contributed to the score. Threshold selection is a second practical concern. Lower thresholds raise recall by flagging more transformed passages, but they also add false positives in formulaic writing, technical definitions and stock phrases. Higher thresholds cut some false alarms but may miss paraphrased reuse. The right operating point depends on the institution, the language, the discipline and what is at stake in a review; the reviewed papers do not establish a universal threshold.

Table 2. Comparison of the five reviewed sources

Study / method	Evidence reported	Implication
Foltýnek et al. [1] Systematic review	239 papers from 2013–2018 surveyed.	Use layered signals and evaluate openly.
El-Rashidy et al. [2] Feature selection + SVM	PAN 2013: Plagdet 89.12%, F1 89.34%; PAN 2014: Plagdet 92.91%, F1 92.95%.	Selected features can perform well on benchmark corpora.
El-Rashidy et al. [3] Deep learning	LSTM ranked first among the compared systems in the reported experiments.	Neural models can fuse signals; validate independently.
Sajid et al. [4] Comparative analysis	Covers 189 papers published during 2019–2024.	Keep plagiarism evidence apart from AI-text classification.
Sağlam et al. [5] AI-obfuscated modeling plagiarism	Automated matching plus human inspection in the studied task.	Support structured artifacts and keep expert review.

IV. PROPOSED DESIGN FOR AN AI-BASED PLAGIARISM CHECKER

Drawing on the reviewed evidence, a practical checker can be organized into the following stages:

- Document ingestion and preprocessing: pull out text or structured content, normalize the encoding, split the document into paragraphs and sentences, and apply suitable tokenization and linguistic preprocessing.
- Candidate retrieval: use efficient lexical fingerprints or n-gram matching to find likely source documents in an authorized comparison corpus.
- Multi-feature analysis: compute lexical overlap, syntactic similarity, semantic similarity and relevant stylistic signals; for code or modeling artifacts, use structure-aware tokenization and normalization.
- AI/ML classification: combine the selected features with a transparent classifier such as an SVM, or with a trained neural model when enough labelled data exists. The probability that text is AI-generated should stay separate from source-based plagiarism evidence.
- Segment extraction and explanation: locate suspicious spans, link them to possible source passages, and show the reviewer the evidence along with confidence indicators.
- Human review and appeal: have a qualified person weigh context, quotations, citations, common phrases and possible false positives before any academic decision is made.

This is a proposed synthesis of the supplied literature, not a system that has been built or tested in this paper. The architecture follows the evidence in the reviewed studies that mixed signals and human inspection can complement automated similarity scoring [1], [2], [5].

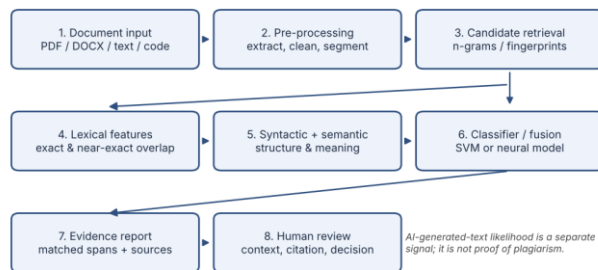


Fig. 1. Proposed hybrid architecture for an AI-based plagiarism checker (conceptual; not experimentally validated).

Evaluation framework. A practical evaluation should report precision, recall and F1-score together with a task-appropriate metric such as Plagdet when source localization matters. Precision is the share of flagged cases that are truly relevant, recall is the share of relevant cases that were found, and F1 summarizes the balance between the two. Results should also state the unit of evaluation (document, passage or source

match), the test corpus, the language, the threshold, and whether the test set is independent of training and tuning.

Recommended validation protocol. First, split the data by source document or author where possible, to reduce leakage. Second, treat direct copying, light edits, paraphrasing, citation-preserving reuse and AI-obfuscated examples as separate categories. Third, measure false positives on quotations, references, templates and common technical language. Fourth, test across subjects and languages, and give confidence intervals or run-to-run variability when feasible. Finally, let trained reviewers inspect a sample of true and false positives before the system is deployed.

Table 3. Suggested evaluation measures for future systems

Measure	What it tells the evaluator	Caution
Precision	How many flagged matches are relevant	Can look high when the system misses many cases.
Recall	How many relevant cases are found	Can climb while false alarms climb too.
F1-score	Balance between precision and recall	Hides differences between the two measures.
Plagdet	Detection quality, including localization	Compare only when definitions and test conditions match.
False-positive review	Whether legitimate overlap is wrongly flagged	Needs human-labelled examples and context.

V. CHALLENGES AND RESEARCH GAPS

First, obfuscation remains a central difficulty: paraphrasing and AI-assisted rewriting can keep the meaning while reducing surface overlap. Second, no single dataset can represent every language, discipline, writing style or new generative system. Third, evaluation should prevent leakage between training and test data and report several measures instead of accuracy alone. Fourth, explanations matter: users should be able to inspect matched source passages and understand why a segment was flagged. Fifth, false positives can unfairly harm students and researchers, so scores must never be treated as proof of misconduct. These concerns agree with the methodological gaps and combined-method recommendations in the review studies [1], [4].

Future work should test hybrid systems on diverse, independently held-out corpora; include human-written paraphrases and AI-obfuscated examples; cover multilingual and cross-language cases; and report precision, recall, F-measure, detection granularity and, where suitable, calibration. Studies should also keep source-based plagiarism detection distinct from AI-authorship detection and set out a clear protocol for human adjudication.

Operational and governance choices affect reliability as well. Institutions need to decide which repositories may be searched, how submitted documents are kept, and who may see similarity reports. The checker should keep an audit trail of candidate sources, model version, thresholds and reviewer decisions. A flagged result should be treated as grounds for investigation and never as an automatic finding of misconduct, and students and authors should be able to see the cited overlap, add context or appeal a decision.

AI-generated-text detection should be reported apart from source-based plagiarism evidence. Generated prose may be original in wording yet still carry unsupported or unattributed ideas; equally, human-written text can legitimately resemble a source because of quotation or standard terminology. Folding both tasks into one unexplained score risks misleading readers. A transparent report should separate matched-source evidence, semantic similarity and any optional authorship-style signal.

Corpus and document handling. A checker is only as good as its comparison corpus. If the likely source documents are missing, even a strong model cannot find where reused material came from. Corpus coverage should therefore be described alongside the model: institutional repositories, open-access articles, student submissions where policy permits, and relevant web sources each come with different access and update constraints. Duplicate records should be merged, document versions tracked and source metadata kept, so that a match can be traced to a stable record instead of an opaque similarity value.

Pre-processing and language handling. PDF extraction can introduce errors through reading order, tables, footnotes, hyphenation and scanned pages, and these errors can either lower similarity or create artificial matches. A practical pipeline should keep the original document, log extraction warnings and let reviewers view the matched source in context. Tokenization, stemming, stop-word handling and sentence segmentation should be chosen to suit the language and document type, since aggressive normalization can erase meaningful distinctions in technical writing, equations, source code or structured models.

Interpretable evidence reports. A useful report names the candidate source, shows aligned passages, separates exact overlap from paraphrase-like similarity, and says whether quotations or references were left out of scoring. When several signals are used, the interface should explain each one separately instead of showing a single opaque score. Confidence values must be calibrated against labelled examples before they are read as probabilities. The report should also state its limits, for example that a source may be absent from the corpus or that semantic similarity can come from a shared subject rather than from copying.

Human-centred workflow. The system should support academic judgement, not replace it. A reviewer can check whether a passage is quoted, whether the source is cited nearby, whether the wording is conventional terminology, and whether the similarity is substantial in context. Reviewers should be trained on both false positives and false negatives, and where the consequences are serious, a second review or an appeal process is appropriate. These safeguards fit the reviewed modeling-assignment study, which pairs automated analysis with human inspection and does not treat the detector as the final authority [5].

Deployment and maintenance. Once deployed, the checker should be monitored for shifts in document formats, writing conventions, source availability and AI-assisted rewriting behaviour. A small, independently labelled sample can be audited from time to time to estimate drift and review quality. Changes to preprocessing, feature selection, model weights or thresholds should be versioned and tested before release, and evaluation should be repeated after major updates, because a result for one benchmark or assignment type does not guarantee the same performance in another discipline or institution.

VI. CONCLUSION

The reviewed research indicates that AI-based plagiarism checking works best as a combination of methods and not as a single detector. Lexical comparison finds direct overlap efficiently, feature-based SVM systems merge diverse similarity measures, deep learning can pick up complex patterns, and structure-aware token methods can handle AI-obfuscated modeling artifacts. The studies report encouraging benchmark results, but differences in tasks and test conditions prevent direct comparison. A dependable checker should combine complementary features, validate on representative data, expose its source evidence and keep human review at the centre of any consequential decision.

REFERENCES

1. T. Foltýnek, N. Meuschke, and B. Gipp, “Academic Plagiarism Detection: A Systematic Literature Review,” *ACM Computing Surveys*, vol. 52, no. 6, Article 112, 2019. doi: 10.1145/3345317.
2. M. A. El-Rashidy, R. G. Mohamed, N. A. El-Fishawy, and M. A. Shouman, “An effective text plagiarism detection system based on feature selection and SVM techniques,” *Multimedia Tools and Applications*, vol. 83, pp. 2609–2646, 2024. doi: 10.1007/s11042-023-15703-4.
3. M. A. El-Rashidy, R. G. Mohamed, N. A. El-Fishawy, and M. A. Shouman, “Reliable plagiarism detection system based on deep learning approaches,” *Neural Computing and Applications*, vol. 34, pp. 18837–18858, 2022. doi: 10.1007/s00521-022-07486-w.
4. M. Sajid, M. Sanaullah, M. Fuzail, T. S. Malik, and S. M. Shuhidan, “Comparative analysis of text-based plagiarism detection techniques,” *PLOS ONE*, vol. 20, no. 4, e0319551, 2025. doi: 10.1371/journal.pone.0319551.
5. T. Sağlam, S. Hahner, L. Schmid, and E. Burger, “Automated Detection of AI-Obfuscated Plagiarism in Modeling Assignments,” in *Proc. 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2024, pp. 297–308. doi: 10.1145/3639474.3640084.