frontiers | Frontiers in Computer Science

TYPE Systematic Review
PUBLISHED 17 March 2025
DOI 10.3389/fcomp.2025.1504725

OPEN ACCESS

EDITED BY
Kamel Barkaoui,
Conservatoire National des Arts et Métiers (CNAM), France

REVIEWED BY
Rocco Zaccagnino,
University of Salerno, Italy
Gerardo Sierra,
National Autonomous University of Mexico, Mexico

*CORRESPONDENCE
Cemil Turan
✉ cemil.turan@sdu.edu.kz

RECEIVED 01 October 2024
ACCEPTED 24 February 2025
PUBLISHED 17 March 2025

CITATION
Amirzhanov A, Turan C and Makhmutova A (2025) Plagiarism types and detection methods: a systematic survey of algorithms in text analysis. *Front. Comput. Sci.* 7:1504725. doi: 10.3389/fcomp.2025.1504725

Plagiarism types and detection methods: a systematic survey of algorithms in text analysis

Altynbek Amirzhanov[1], Cemil Turan[1]* and Alfira Makhmutova[2]

[1]Computer Science, SDU University, Kaskelen, Kazakhstan, [2]General Education, New Uzbekistan University, Tashkent, Uzbekistan

Plagiarism in academic and creative writing continues to be a significant challenge, driven by the exponential growth of digital content. This paper presents a systematic survey of various types of plagiarism and the detection algorithms employed in text analysis. We categorize plagiarism into distinct types, including verbatim, paraphrasing, translation, and idea-based plagiarism, discussing the nuances that make detection complex. This survey critically evaluates existing literature, contrasting traditional methods like string-matching with advanced machine learning, natural language processing, and deep learning approaches. We highlight notable works focusing on cross-language plagiarism detection, source code plagiarism, and intrinsic detection techniques, identifying their contributions and limitations. Additionally, this paper explores emerging challenges such as detecting cross-language plagiarism and AI-generated content. By synthesizing the current landscape and emphasizing recent advancements, we aim to guide future research directions and enhance the robustness of plagiarism detection systems across various domains.

KEYWORDS

plagiarism detection, text analysis, natural language processing, plagiarism types, machine learning, AI-generated content

1 Introduction

Plagiarism, often defined as the uncredited replication or close imitation of someone else's work, remains a persistent threat to academic integrity across various disciplines. The *Office of Research Integrity (ORI)* and foundational studies (Roig, 2006) define plagiarism as the act of using another person's intellectual output without proper acknowledgment, which directly undermines the principles of originality and academic honesty. As digital content continues to expand, the challenge of detecting and preventing plagiarism has become increasingly complex (Gandhi et al., 2024).

Early detection methods, such as *string-matching algorithms*, were effective for identifying verbatim plagiarism. Tools like *Turnitin* and *CopyCatch* employ *Rabin-Karp* and *Knuth-Morris-Pratt string-matching* techniques to efficiently compare text segments and detect direct text overlap. These approaches, widely adopted in educational institutions and publishing platforms, provide high accuracy in detecting exact text matches. However, plagiarism has evolved beyond simple copy-pasting to include *paraphrasing, translation, idea-based plagiarism, and AI-generated content*, making traditional methods increasingly inadequate.

In response, advancements in *machine learning (ML) and natural language processing (NLP)* have significantly enhanced plagiarism detection by incorporating *semantic similarity models, deep learning architectures, and citation-based techniques*. Emerging challenges, such as plagiarism in programming code and *cross-lingual plagiarism*, further

complicate detection efforts. For instance, in programming plagiarism, even minor syntax changes (e.g., variable name alterations or logic restructuring) can obscure copied code. Specialized tools like *Measure of Software Similarity (MOSS)* and *Program Dependence Graphs (PDG)* exemplify approaches tailored to detect such obfuscation. Meanwhile, AI-generated content detection introduces a new frontier, requiring models capable of identifying machine-generated text with high accuracy.

This paper presents a *systematic survey* of plagiarism types and detection algorithms, integrating findings from previous research and highlighting recent advancements in AI-based detection techniques. By categorizing plagiarism into *verbatim, paraphrased, translation-based, conceptual plagiarism, and programming code plagiarism*, this study provides a *comprehensive overview* of the current landscape of plagiarism detection. Additionally, we examine emerging challenges such as *cross-lingual plagiarism* and *AI-generated content detection*, providing insights into future research directions.

2 Research objectives and questions

Plagiarism detection remains a complex challenge due to the increasing sophistication of textual obfuscation techniques. Traditional approaches, including *string-matching and syntactic analysis*, struggle with advanced forms of plagiarism, necessitating the development of more robust AI-driven solutions. To provide a *structured and comprehensive* analysis of plagiarism detection

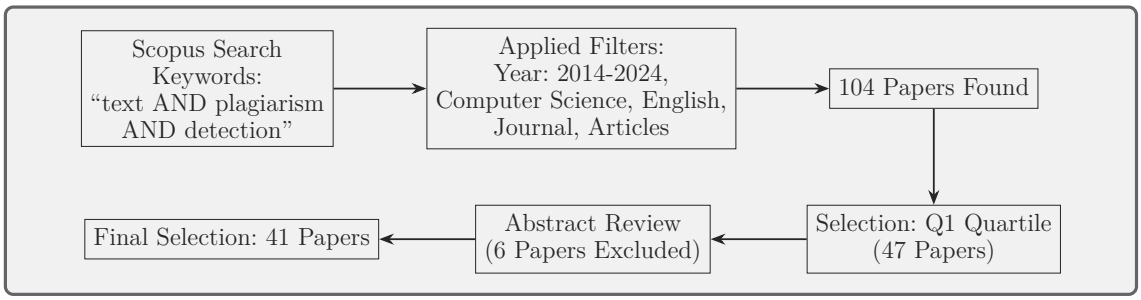

FIGURE 1
Paper selection methodology: flow of search, filters, and selection process.

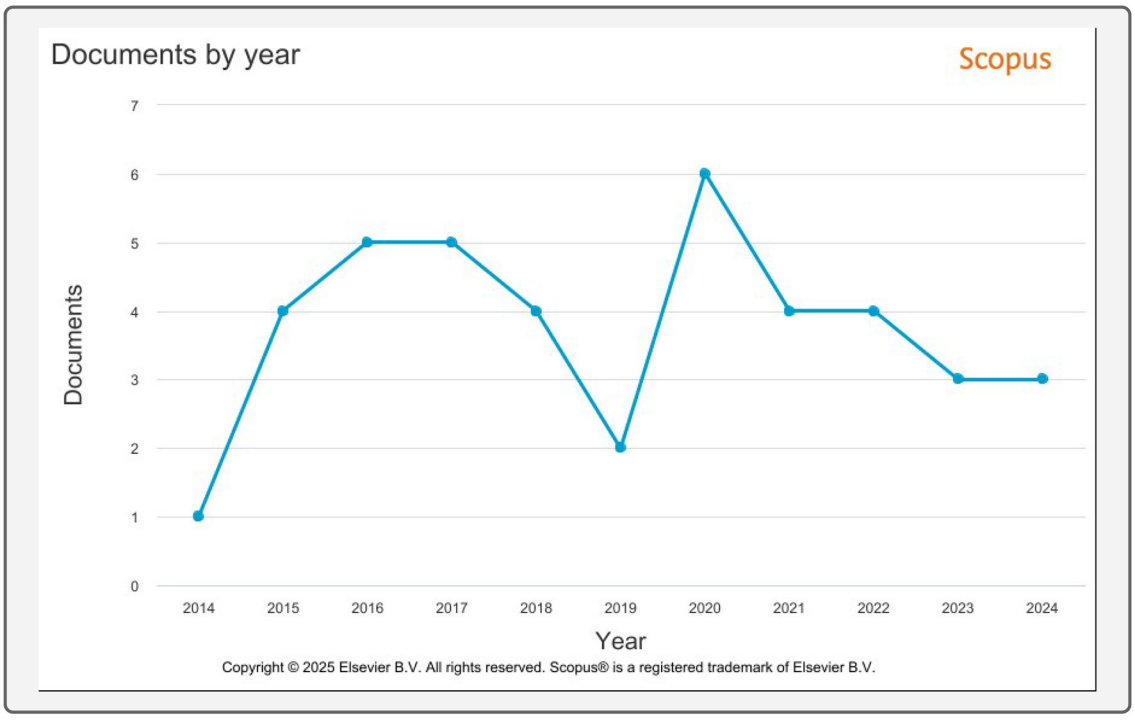

FIGURE 2
Publication trends in reviewed papers (2014–2024).

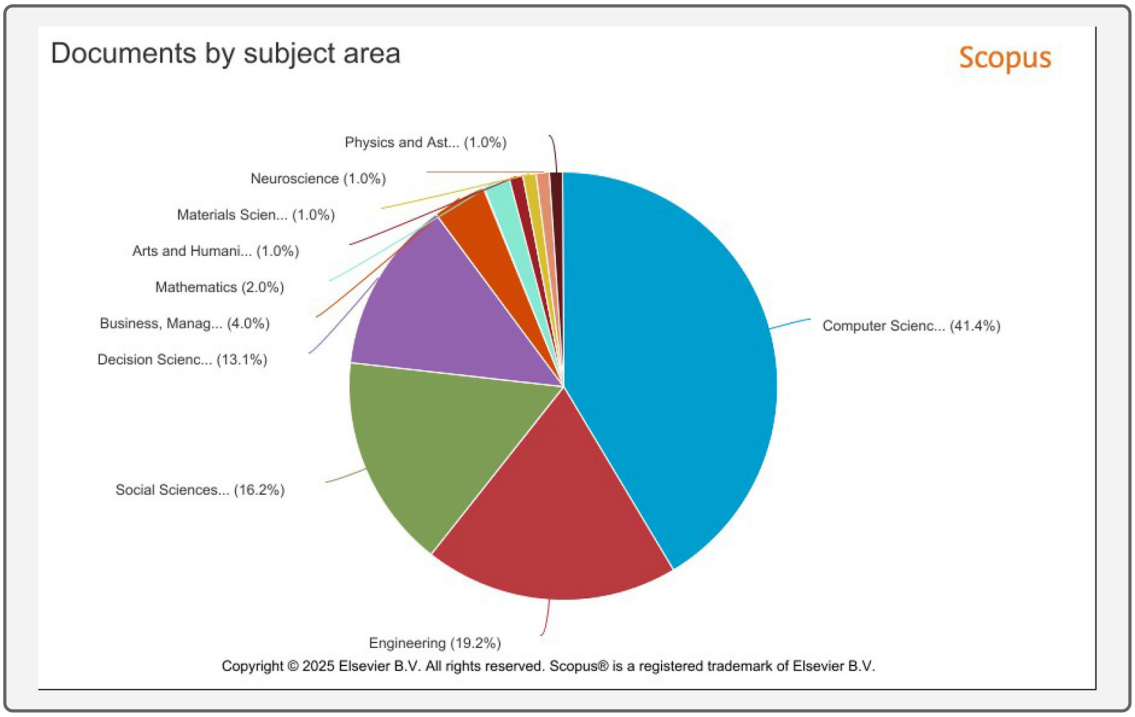

FIGURE 3
Disciplinary distribution of reviewed research papers.

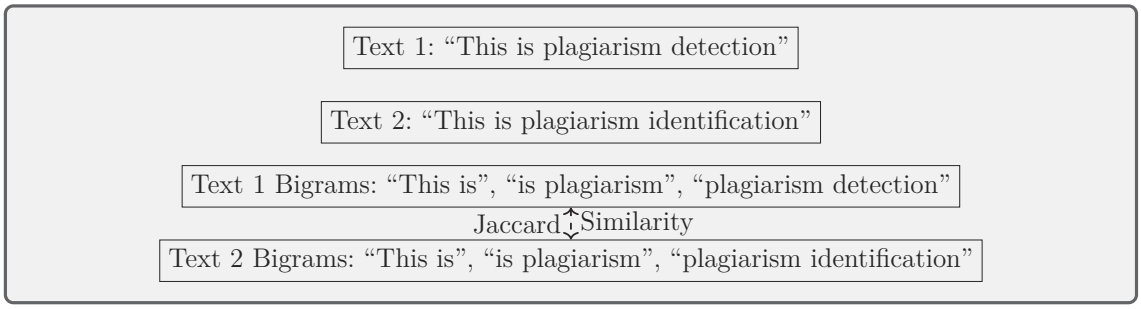

FIGURE 4
Shingle/substring matching: example of dividing text into bigrams and comparing using Jaccard similarity.

methodologies, this study is guided by the following objectives and research questions:

2.1 Research objectives

This paper aims to:

- **Categorize and analyze** the different types of plagiarism, highlighting their detection complexities.
- **Critically evaluate** the methodologies and algorithms currently used in plagiarism detection, comparing traditional approaches with *ML, NLP, and deep learning techniques.*
- **Identify emerging challenges**, including *AI-generated plagiarism and cross-lingual detection*, and propose *future research directions* to enhance detection systems.

2.2 Research questions

This study seeks to answer the following key questions:

1. **What are the distinct types of plagiarism, and how do they differ in terms of detection complexity?**
2. **What are the strengths and limitations of existing plagiarism detection methods?**

TABLE 1 Shingle/substring matching.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Chekhovich and Khazov (2022)	Duplicate publication, text recycling	Russian scientific publications	Shingle index, Antiplagiat	eLIBRARY.RU	F1, threshold 0.66	Detecting duplicate scientific publications in Russian journals	Likely $O(n \log n)$ due to shingle index structures	Efficient for large-scale document comparison; widely used in Russian scientific domain	Shingle methods can fail with highly paraphrased text
Turrado García et al. (2018)	Misspelled names, deduplication	Names datasets	LSH, Damerau-Levenshtein, Jaccard	Synthetic dataset	Pairwise comparisons	Name deduplication, detecting misspellings in databases	LSH reduces complexity to sublinear time; Damerau-Levenshtein is $O(nm)$	Effective for detecting name misspellings and deduplication	Fails in cases where names are completely altered or context is missing
Al-Thwaib et al. (2020)	Verbatim, paraphrasing	Academic dissertations	N-grams, NLP	JUPlag corpus (2,312 dissertations)	No accuracy provided	Detecting verbatim and paraphrased plagiarism in academic writing	N-gram comparison typically runs in $O(n)$ but scales with document size	Handles verbatim and paraphrased plagiarism well with NLP integration	N-gram approaches struggle with deeply obfuscated plagiarism
Velásquez et al. (2016)	External and intrinsic plagiarism	Spanish academic documents	Information fusion, n-grams, writing style	Spanish corpus, PAN-PC 2010, 2011	Precision 85.59%, Recall 55.6%, F1 48.24%	Plagiarism detection in Spanish academic documents	Information fusion and n-grams run in polynomial time, estimated $O(n^2)$	Combines multiple features for better accuracy in Spanish documents	Struggles with short text plagiarism detection and recall rate is low
Malandrino et al. (2022)	Music plagiarism detection	Famous legal cases (MusicXML)	Meta-heuristic, clustering	George Washington & Columbia Law dataset	Spectral clustering 97% accuracy	Detecting music plagiarism in legal cases	Meta-heuristic clustering runs in $O(n \log n)$ for typical cases	High accuracy for music plagiarism detection; adaptive clustering approach	Method is domain-specific and may not generalize well to textual plagiarism

10.3389/fcomp.2025.1504725

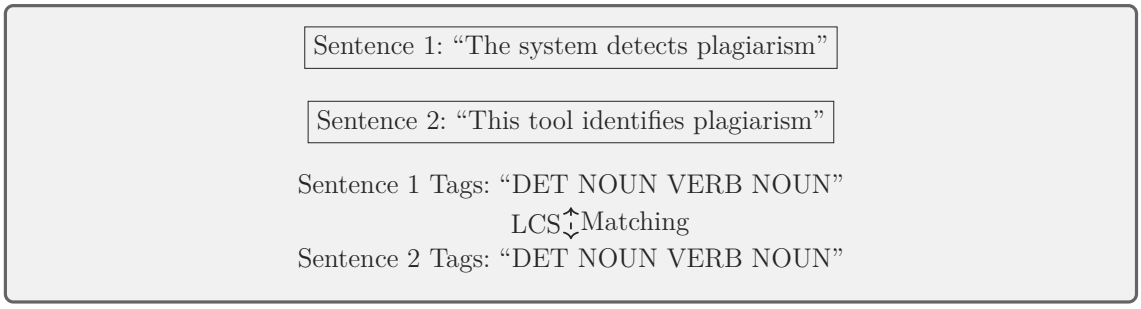

FIGURE 5
Syntax-based approaches: example of POS tagging and LCS matching.

3. **How do advanced ML, NLP, and deep learning techniques enhance plagiarism detection?**
4. **What are the emerging trends and challenges in detecting AI-generated content and cross-language plagiarism?**

By addressing these questions, this study aims to provide *a comprehensive overview* of current detection methodologies while offering *insights into future advancements* in plagiarism detection research.

3 Paper selection methodology

To ensure *methodological rigor* and *clarity*, this study follows the **PICOS framework**, which defines the *Population, Intervention, Comparison, Outcomes, and Study Design* of this systematic review.

- **Population (P):** Academic, educational, and creative communities affected by plagiarism challenges, including researchers, educators, journal editors, and plagiarism detection system developers.
- **Intervention (I):** Various plagiarism detection techniques, including:
  - Traditional methods (e.g., string-matching, syntactic similarity).
  - Semantic similarity models (e.g., word embeddings, deep learning).
  - Machine learning and NLP-based methods (e.g., transformers, BERT-based models).
  - Citation-based approaches and structural analysis.
- **Comparison (C):** A critical evaluation of:
  - Rule-based and string-matching approaches vs. AI-driven methods.
  - Traditional textual similarity techniques vs. deep learning architectures.
  - Monolingual plagiarism detection vs. cross-lingual plagiarism detection methods.
- **Outcomes (O):** Identification of the most effective strategies for plagiarism detection, insights into emerging challenges such as AI-generated content, semantic plagiarism, and cross-lingual text transformation, and evaluation of the role of deep learning, NLP, and citation-based methods in plagiarism detection.
- **Study Design (S):** A *systematic survey* of *peer-reviewed studies* from *high-quality journals (2014–2024)*, focusing on both theoretical advancements and real-world applications of plagiarism detection.

By following the PICOS framework, this study provides a *structured and transparent* review, ensuring reproducibility and guiding future research in plagiarism detection. To ensure a systematic and comprehensive review of the literature, we additionally followed the PRISMA guidelines for identifying, screening, and including relevant papers. Overall the methodology comprised the following steps.

3.1 Database selection and search strategy

We selected *Scopus* as the primary database for retrieving papers due to its extensive coverage of high-quality peer-reviewed journals. The following search query was used to identify relevant studies:

"text AND plagiarism AND detection"

This search query was designed to target research specifically focused on textual plagiarism detection techniques.

To refine the search results and focus on relevant studies, we applied the following filters:

- **Year range**: 2014–2024, ensuring the inclusion of recent advancements.
- **Subject area**: computer science, aligning with technological developments in plagiarism detection.
- **Document type**: only full-length peer-reviewed journal articles.
- **Source type**: journals, prioritizing high-quality research.
- **Language**: only English-language papers were considered for consistency.

TABLE 2 Syntax-based approaches.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Manzoor et al. (2023)	Intrinsic plagiarism detection	Literary, academic texts	Lexical, syntactic, semantic analysis, ML	PAN, Corpus of English Novels, Wikipedia	F1-score, Precision, Recall	Academic and literary intrinsic plagiarism detection	Resource requirements discussed; computational complexity not explicitly provided	Diverse methods including ML and deep learning improve detection robustness	Lack of a reference collection limits robustness and benchmarking
Vani and Gupta (2017b)	Text plagiarism	Academic short answers	POS tagging, chunking, feature selection	PAN12-14, PSA	Accuracy 97.89%, F1 0.979	Detection in academic short answers	POS tagging and chunking are typically $O(n)$; feature selection adds extra overhead	High accuracy in academic short-answer plagiarism detection	Performance depends on feature engineering; domain-specific

After applying the filters, the search yielded a total of 104 papers. To further enhance the quality of the review, we restricted our selection to papers published in **Q1 quartile journals**. Q1 journals are recognized as leading in their fields and ensure high-impact research that meets rigorous peer-review standards. This selection criterion aligns with our objective of synthesizing advanced and reliable methodologies in plagiarism detection. This reduced the number of papers to 47. Following the initial query and filtering, we carefully reviewed the abstracts of the selected papers to assess their relevance to the scope of this review. More six papers were excluded based on the following criteria:

- Focus on plagiarism detection in non-textual domains, such as images or audio.
- Lack of empirical validation or practical application of proposed methods.
- Redundancy with other included studies, offering no additional insights.
- Methodological limitations, such as insufficient sample sizes or incomplete datasets.
- Language mismatch (abstracts or full text not available in English).
- Inaccessibility or incomplete publication details.

As a result, 41 papers were included in the final dataset for this review, providing a solid foundation for analyzing plagiarism detection techniques. The methodology followed is summarized in Figure 1, which outlines the step-by-step paper selection process.

4 Quantitative analysis of reviewed literature on plagiarism detection

To provide quantitative and statistical insights of the literature, we present a statistical overview of the papers included in our systematic review. This analysis provides insights into the publication trends, disciplinary focus, and methodological evolution of plagiarism detection research.

4.1 Publication trends in reviewed papers (2014–2024)

Figure 2 shows a temporal analysis of the papers reviewed in this study as fluctuating research activity in plagiarism detection. Key observations include:

- A notable peak in 2020, reflecting an increased focus on AI-driven detection techniques and the rising concern over AI-generated plagiarism.
- Stable publication activity between 2015 and 2017, indicating sustained interest in refining plagiarism detection methodologies.
- A gradual decline in recent years, potentially due to:
  - The maturity of existing plagiarism detection techniques.

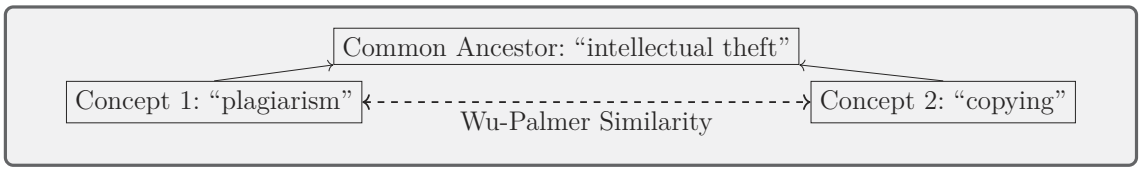

FIGURE 6
Concept-based approaches: Wu-Palmer similarity between two concepts in a semantic hierarchy.

– A shift toward integrating plagiarism detection within broader NLP and AI applications.

These patterns indicate waves of research focus, often aligned with advancements in machine learning (ML), deep learning (DL), and natural language processing (NLP)-based approaches.

4.2 Disciplinary distribution of reviewed research

Figure 3 categorizes the reviewed papers by subject area, highlighting the disciplinary focus within plagiarism detection research:

- **Computer science (41.4%)** remains the dominant field, reflecting its central role in developing text-matching, NLP, and AI-based plagiarism detection algorithms.
- **Engineering (19.2%)** accounts for a significant share, likely due to the development of software tools and algorithmic optimizations.
- **Social sciences (16.2%) and decision sciences (13.1%)** emphasize the increasing interdisciplinary interest in plagiarism detection, particularly in academic integrity, ethics, and policy frameworks.
- Smaller contributions from **Mathematics (2.0%)**, **business management (4.0%)**, and **Neuroscience (1.0%)** highlight the adoption of plagiarism detection methods beyond technical disciplines.

This disciplinary spread reinforces that while plagiarism detection is primarily a computational challenge, there is growing cross-disciplinary engagement, particularly in areas like education, publishing ethics, and AI-driven academic misconduct detection.

4.3 Evolution of detection methodologies in reviewed papers

Our literature analysis reveals distinct shifts in research focus across different periods:

- **Pre-2018**: emphasis on traditional string-matching, n-gram, and citation-based approaches, widely used in early detection tools.
- **Post-2018**: a significant shift toward AI-powered detection, driven by:
  - The rise of deep learning (CNNs, LSTMs, transformers like BERT/GPT).
  - A growing need for cross-language plagiarism detection.
  - Concerns over AI-generated content and its detection.

This transition from surface-level text similarity to deeper semantic analysis highlights the increasing complexity of modern plagiarism cases, requiring more sophisticated detection models.

4.4 Aligning research trends with emerging needs

The reviewed literature reflects the evolving challenges in plagiarism detection:

- The rise of cross-language detection techniques aligns with the globalization of academic publishing.
- The peak in 2020 corresponds with increased awareness of AI-generated text (e.g., GPT-3, BARD), emphasizing the importance of machine-learning-based plagiarism detection.
- The shift to deep learning methods suggests a growing need for adaptive, context-aware plagiarism detection systems.

Overall, the quantitative insights from our reviewed papers provide a broader context for our systematic survey, demonstrating the evolution of research priorities in plagiarism detection. The statistical trends validate the transition from traditional similarity-based approaches to AI-driven, semantic plagiarism detection, highlighting the need for scalable and adaptive detection methodologies.

5 Background and types of plagiarism

Plagiarism is a widespread issue that undermines academic integrity, intellectual honesty, and innovation. With the rapid growth of digital content and access to online information, plagiarism has become increasingly sophisticated, requiring equally advanced methods for detection.

TABLE 3 Concept-based approaches.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Manzoor et al. (2023)	Intrinsic plagiarism detection	Literary, academic texts	Lexical, syntactic, semantic analysis, ML	PAN, Corpus of English Novels, Wikipedia	F1-score, Precision, Recall	Academic and literary intrinsic plagiarism detection	Resource requirements discussed; computational complexity not explicitly provided	Diverse methods including ML and deep learning improve detection robustness	Lack of a reference collection limits robustness and benchmarking
Vani and Gupta (2017b)	Text plagiarism	Academic short answers	POS tagging, chunking, feature selection	PAN12-14, PSA	Accuracy 97.89%, F1 0.979	Detection in academic short answers	POS tagging and chunking are typically $O(n)$; feature selection adds extra overhead	High accuracy in academic short-answer plagiarism detection	Performance depends on feature engineering; domain-specific

5.1 Types of plagiarism

Plagiarism manifests in various sophisticated forms, each posing unique challenges to detection and prevention in academic and research contexts. Understanding these types is crucial for developing effective detection strategies and maintaining academic integrity. Supplementary Image 1 the plagiarism types in a simple diagram which has brief information below:

- **Verbatim plagiarism:** direct copying of text without changes or attribution.
- **Paraphrased plagiarism:** rewriting the original text while retaining the core meaning.
- **Idea-based plagiarism:** appropriating someone else's ideas or arguments without acknowledgment.
- **Translation plagiarism:** translating content from one language to another without citation.
- **Code plagiarism:** reusing source code or program logic with minimal alterations.
- **AI-generated content:** using AI tools like GPT or BARD to generate content without proper disclosure.

Plagiarism can also occur in more subtle and advanced forms:

- **Obfuscated plagiarism:** modifying text structure or replacing key terms while retaining the original meaning (Alzahrani et al., 2015; Gharavi et al., 2019).
- **Cross-language plagiarism:** translating content across languages without credit, making detection more complex (Alzahrani and Aljuaid, 2022; Franco-Salvador et al., 2016a).
- **Multilingual and language-independent plagiarism:** extending plagiarism detection across different languages and linguistic structures (Gharavi et al., 2019).
- **Duplicate and redundant publications:** republishing existing work with minor modifications to increase publication count (Benos et al., 2005; Errami et al., 2008; Lariviere and Gingras, 2010).

5.2 Prevalence and impact

Plagiarism is a significant problem across educational institutions and professional settings. Studies suggest:

- A 2023 survey found that up to 58% of university students admitted to engaging in some form of plagiarism during their academic careers.
- An estimated 1.5% of all published papers involve duplicate content or unethical reuse (Errami et al., 2008).
- In software development, code plagiarism accounts for nearly 20% of all academic misconduct cases reported by universities (Liu et al., 2015).

The consequences include devaluation of academic credentials, intellectual theft, and reputational damage to institutions.

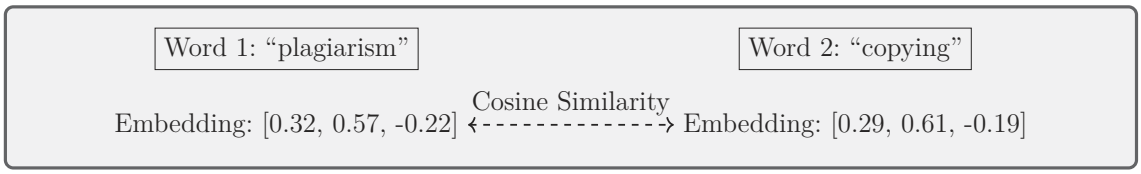

FIGURE 7
Word embedding models: cosine similarity between word vectors in a high-dimensional space.

6 Plagiarism detection methods

Plagiarism detection is a crucial task in academic, professional, and digital environments, safeguarding the integrity of intellectual property. Various methods have been developed to identify plagiarism types, ranging from verbatim copying to complex paraphrasing and idea plagiarism. They have evolved to address challenges, from traditional *string matching techniques* to modern *ML and NLP-based* approaches. The reviewed papers propose a wide range of detection methods, which we have categorized into six primary approaches:

- **Textual similarity-based**,
- **Semantic similarity-based**,
- **Cross-language detection**,
- **Machine learning and deep learning models**,
- **Citation and structural-based approaches**,
- **Code-based detection**.

While conventional methods excel in detecting verbatim plagiarism, they often struggle with paraphrased and conceptual plagiarism. AI-driven techniques, such as *deep learning and citation-based approaches*, are promising but require high computational resources. By understanding these trends, this paper highlights the importance of adopting diverse detection methods tailored to different plagiarism forms and the evolving landscape of digital content creation.

6.1 Textual similarity-based approaches

Textual similarity-based methods focus on detecting overlaps in surface-level textual features. These methods include:

6.1.1 Shingle/substring matching

Shingle-based approaches compare overlapping subsequences of text (e.g., n-grams, q-grams) to detect similarities. Several reviewed works, such as those by Chekhovich and Khazov (2022), Turrado García et al. (2018), Al-Thwaib et al. (2020), employ these methods. Velásquez et al. (2016) and Malandrino et al. (2022) also use n-gram analysis to measure document similarity. A widely used formula is Jaccard similarity:

$$\text{Jaccard Similarity}(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

where A and B represent sets of n-grams from the compared texts. As shown in Figure 4, the division of texts into bigrams for comparison using Jaccard similarity is illustrated. Table 1 offers a summary of the key studies utilizing shingle/substring matching methods, including details on datasets and accuracy metrics.

6.1.2 Syntax-based approaches

Syntax-based methods analyze grammatical structures to detect plagiarism, even when sentences are restructured. Methods proposed by Manzoor et al. (2023), Vani and Gupta (2017a) involve parsing text into syntactic components using POS tagging and algorithms like the Longest Common Subsequence (LCS):

$$LCS(X,Y) = \max \begin{cases} LCS(X_{i-1}, Y_{j-1}) + 1 & \text{if } X_i = Y_j \\ \max(LCS(X_{i-1}, Y_j), LCS(X, Y_{j-1})) & \text{otherwise} \end{cases} \quad (2)$$

where X and Y are sequences (e.g., sentences or phrases) from two documents. Figure 5 illustrates how sentences are parsed into POS tags and analyzed using the LCS algorithm to detect structural similarities. Key studies on syntax-based plagiarism detection methods are summarized in Table 2, highlighting datasets and accuracy results.

6.2 Semantic similarity-based approaches

Semantic similarity methods detect plagiarism by analyzing the meaning behind words, going beyond surface-level text comparison. These methods are crucial for detecting paraphrased content and are divided into concept-based approaches and word embedding models.

6.2.1 Concept-based approaches

Concept-based methods use semantic role labeling (SRL), named entity recognition (NER), and linguistic knowledge to detect idea plagiarism (Taufiq et al., 2023; Vani and Gupta, 2017b; Abdi et al., 2015). These approaches combine semantic and syntactic similarity to detect deeper textual relationships. A common metric used is Wu-Palmer Similarity:

$$\text{Wu-Palmer Similarity}(w_1, w_2) = \frac{2 \times \text{depth}(\text{LCS}(w_1, w_2))}{\text{depth}(w_1) + \text{depth}(w_2)} \quad (3)$$

TABLE 4 Word embedding models.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Mehak et al. (2023)	Text Reuse (Phrasal)	Urdu language content	Sentence Transformer, N-gram, embeddings	UTRD-Phr-23	F1-score ∼0.63	Detecting text reuse in Urdu content	Sentence Transformer runs in $O(n \log n)$, N-grams in $O(n)$	Adapts well to Urdu language-specific text reuse detection	Lower F1-score suggests room for improvement in embedding effectiveness
Alzahrani et al. (2015)	Obfuscated plagiarism	Academic and web texts	Fuzzy semantic similarity, WordNet	PAN-PC-09, PAN-PC-10, Microsoft Paraphrase	Precision 0.9178, Recall 0.6933	Detecting obfuscated plagiarism across different text sources	Fuzzy semantic similarity and WordNet traversal run in $O(n \log n)$	Highly effective for uncovering obfuscated plagiarism	WordNet-based approaches depend on lexicon availability and coverage
Darwish et al. (2023)	Semantic plagiarism	Summary obfuscation	Quantum genetic algorithm, WordNet	PAN13-14 dataset	F-score improved 10%	Handling summary obfuscation in academic plagiarism detection	Quantum genetic algorithm has high computational overhead ($O(n^2)$)	Shows improvement in handling summary obfuscation cases	Quantum methods may require specialized resources for scalability
Sahi and Gupta (2017)	Verbatim, paraphrasing	Academic papers	Semantic-syntactic analysis	PAN-PC-11	F1 0.837, Plagdet 0.836	Detecting verbatim and paraphrased plagiarism in academic texts	Semantic-syntactic analysis runs in $O(n^2)$ for pairwise comparisons	Good balance of semantic and syntactic analysis for plagiarism detection	Computationally expensive for large datasets
Alvi et al. (2021)	Paraphrase plagiarism	Academic short answers	Context matching, embeddings, Smith-Waterman	Corpus of plagiarized short answers	F1 0.905, 0.802	Paraphrased plagiarism detection in short academic answers	Smith-Waterman algorithm runs in $O(nm)$, embeddings in $O(n \log n)$	Performs well on paraphrased plagiarism detection	Accuracy depends on the quality of paraphrase embeddings
Gharavi et al. (2019)	Obfuscation types	Multilingual plagiarism	Embedding-based, cosine, Jaccard	PAN-PC-2013, PersianPlagDet2016, custom Arabic	Plagdet ∼79.9%	Multilingual plagiarism detection with embedding-based methods	Embedding-based similarity runs in $O(n \log n)$, Jaccard in $O(n)$	Handles multilingual plagiarism effectively	Scalability remains a challenge with larger datasets

10.3389/fcomp.2025.1504725

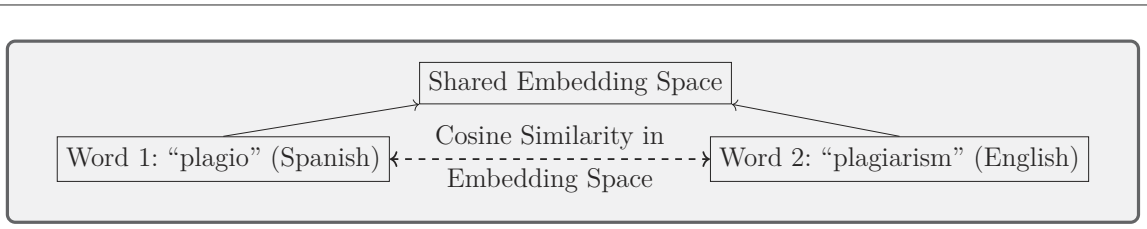

FIGURE 8
Multilingual embedding models: mapping words from different languages into a shared vector space for similarity comparison.

where w_1 and w_2 are two words being compared, and LCS refers to their least common subsumer in a semantic hierarchy. In Figure 6, the comparison of two concepts through Wu-Palmer similarity, with the identification of the least common subsumer, is demonstrated. In Table 3, the major studies on concept-based plagiarism detection approaches are summarized, with attention to the datasets and reported accuracy metrics.

6.2.2 Word embedding models

Word embedding models, such as Word2Vec, BERT, and GPT-based transformers, allow for nuanced detection by capturing contextual meanings (Mehak et al., 2023; Alzahrani et al., 2015; Darwish et al., 2023). Cosine similarity is often used to measure similarity in an embedding space:

$$\text{Cosine Similarity}(A, B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (4)$$

where A_i and B_i are the word embedding vectors from two different texts. Figure 7 demonstrates how cosine similarity is calculated for word embedding vectors to capture semantic relationships. A summary of key studies utilizing word embedding models for plagiarism detection, along with their datasets and accuracy metrics, is provided in Table 4.

6.3 Cross-language and multilingual approaches

Cross-language plagiarism detection methods address the challenge of identifying plagiarism in translated texts. These include:

- **Multilingual embedding models**: these models represent words from different languages into the same space so they can be compared. This approach is seen in the works by Glavaš et al. (2018), Alzahrani and Aljuaid (2022), and Roostaee et al. (2020), where models like BERT and other cross-lingual types which help to detect plagiarism between languages like Spanish and English. As shown in Figure 8, words from different languages, including Spanish and English, are placed in the same space for comparison using cosine similarity. In Table 5, key studies using multilingual embedding models for cross-lingual plagiarism detection are summarized, with information on methods, datasets, and accuracy.
- **Cross-language detection**: methods by Ehsan and Shakery (2016) and Ehsan et al. (2018) use translation-based approaches and dynamic text alignment for detecting plagiarism between different languages, such as German-English and Spanish-English pairs. The procedure for cross-language plagiarism detection, using cosine similarity between Spanish and English texts, is depicted in Figure 9. In Table 6, you will find an overview of the methods, datasets, and performance metrics from various studies on cross-language plagiarism detection.
- **Knowledge graphs and embedding models**: knowledge graphs and embedding models combine the power of structured semantic networks (knowledge graphs) with the flexibility of embedding models to detect plagiarism across languages. These approaches are particularly useful in cross-lingual plagiarism detection where the challenge is to compare texts in different languages. By using knowledge graphs, which model relationships between concepts, and embedding models that represent words or concepts in vector space, these methods can handle cases of paraphrasing or translation-based plagiarism. The knowledge graphs provide a structural representation of concepts and their relationships, while embeddings map those concepts into a continuous vector space, allowing comparison across languages. Franco-Salvador et al. (2016a,b) have pioneered this hybrid approach with their methods like KBSim (Knowledge-Based Similarity) and XCNN for detecting plagiarism between Spanish-English and German-English texts. A detailed overview of key studies using knowledge graphs and embedding models for cross-lingual plagiarism detection can be found in Table 7, where methods, datasets, and accuracy metrics are discussed.

6.4 Machine learning and deep learning approaches

Traditional machine learning models such as SVMs and Random Forest classifiers have been widely used for text classification (Hussain and Suryani, 2015; Polydouri et al., 2018; El-Rashidy et al., 2022). As depicted in Figure 10, the

TABLE 5 Multilingual embedding models.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Glavaš et al. (2018)	Cross-lingual plagiarism detection	Spanish-English academic papers	Cross-lingual word embeddings	PAN-PC-11	R@1 = 89.5%, R@10 = 94%	Detection of cross-lingual plagiarism in academic texts	Cross-lingual word embeddings run in $O(n \log n)$	High recall for cross-lingual plagiarism detection	Limited to language pairs seen during training
Alzahrani and Aljuaid (2022)	Cross-lingual plagiarism detection	Arabic-English academic texts	Deep learning, ML	Custom corpus	Accuracy ~97%	Identifying cross-lingual plagiarism in Arabic-English texts	Deep learning models run in $O(n^2)$ for training, $O(n)$ for inference	Achieves high accuracy for Arabic-English cross-lingual plagiarism detection	Requires large annotated cross-lingual datasets for effective performance
Roostaee et al. (2020)	Cross-lingual plagiarism detection	Multilingual academic texts	Vector space models, embeddings	PAN-PC-11, PAN-PC-12, SemEval	Plagdet 0.720, 0.769	Multilingual plagiarism detection across academic datasets	Vector space models and embeddings operate in $O(n \log n)$	Effective across multiple languages and academic text domains	Plagdet scores indicate potential room for improvement in precision

feature extraction process and SVM classification workflow lead to the final plagiarism detection outcome. Table 8 outlines key studies using traditional machine learning models for plagiarism detection, along with details on their methods and performance metrics.

Deep learning models, including LSTM, CNN, and Transformers, offer powerful tools for detecting more subtle forms of plagiarism, such as paraphrasing. Works by Shahmohammadi et al. (2020), Hayawi et al. (2023), Suman et al. (2021), Agarwal et al. (2018), Romanov et al. (2021), El-Rashidy et al. (2024), Shakeel et al. (2020), and Iqbal et al. (2024) apply these advanced neural networks to plagiarism detection tasks. In Figure 11, an LSTM network is depicted, demonstrating its ability to process input sequences like sentences for plagiarism detection. A concise summary of key studies employing deep learning techniques for plagiarism detection, including methods, datasets, and performance metrics, is provided in Table 9.

6.5 Structural and citation-based approaches

Structural and citation-based approaches focus on how documents are organized or how citations are reused. These methods are particularly effective in academic and research-based plagiarism detection. Gipp et al. (2014), Pertile et al. (2015), and Vani and Gupta (2018) employ citation pattern analysis to track citation reuse and bibliographic coupling, while structural methods look at document organization to detect anomalies. As shown in Figure 12, bibliographic coupling highlights shared references between two documents, helping to visualize the overlap in citation patterns. The key studies that utilize citation-based approaches for plagiarism detection, along with their corresponding datasets and accuracy metrics, are summarized in Table 10.

6.6 Code-based plagiarism detection

Code-based plagiarism detection is designed to handle the unique challenges of source code plagiarism, where syntactic and structural changes can mask copied code. Liu et al. (2015) and Bartoszuk and Gagolewski (2021) use methods like Program Dependence Graph (PDG) and q-grams to detect code similarities, even when the code is restructured or altered in non-obvious ways. Figure 13 shows how q-grams are extracted from tokenized code sequences and compared to assess similarity. In Figure 14, the comparison of Program Dependence Graphs is demonstrated, with structural elements like operations and conditions matched between two programs. A summary of studies focused on code-based plagiarism detection, including the methods used and their performance metrics, is provided in Table 11.

While this section has outlined various plagiarism detection techniques and their operational mechanisms, the next section critically evaluates these approaches, highlighting their

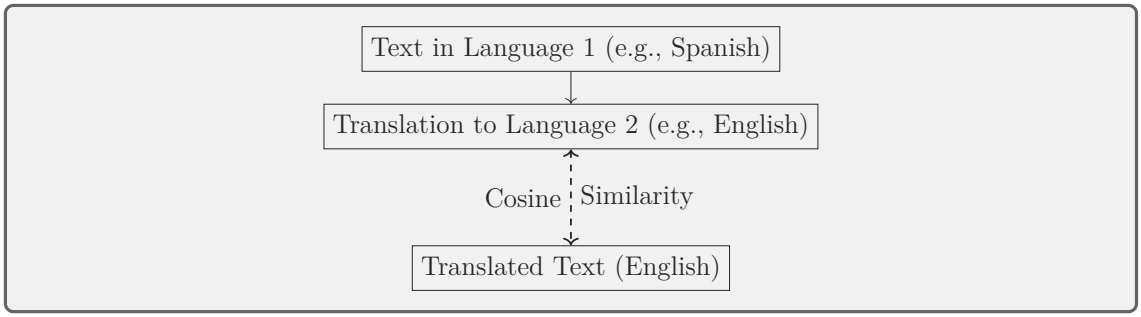

FIGURE 9
Cross-language detection: translating text between languages and detecting plagiarism using similarity methods.

TABLE 6 Cross-language detection.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Ehsan and Shakery (2016)	Cross-lingual plagiarism detection	Cross-lingual documents	Proximity-based retrieval, topic segmentation	PAN-PC-12	F2 score 0.6703	Detecting cross-lingual plagiarism in textual data	Proximity-based retrieval runs in $O(n \log n)$; topic segmentation depends on document length	Effective for detecting cross-lingual similarity using topic segmentation	Performance depends on language-specific topic segmentation accuracy
Ehsan et al. (2018)	Cross-lingual plagiarism detection	Academic papers	Dictionary-based, dynamic alignment	PAN-PC-12	Plagdet 0.863	Identifying cross-lingual plagiarism in academic literature	Dictionary-based methods operate in $O(n)$, dynamic alignment runs in $O(n^2)$	High accuracy in academic cross-lingual plagiarism detection	Relies on the availability of quality bilingual dictionaries

TABLE 7 Knowledge graphs and embedding models.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Franco-Salvador et al. (2016a)	Cross-language plagiarism detection	Cross-lingual academic texts	Hybrid models, knowledge graphs (KBSim, XCNN)	PAN-PC-11 (Spanish-English, German-English)	Plagdet ~0.64	Detecting cross-lingual plagiarism in academic texts	Knowledge graph similarity is $O(n^2)$; hybrid models improve efficiency	Integrates deep learning with structured knowledge for better detection	Performance depends on the completeness of the knowledge graph
Franco-Salvador et al. (2016b)	Cross-language plagiarism (paraphrasing)	Cross-lingual academic texts	Cross-language Knowledge Graph Analysis	PAN-PC-10, PAN-PC-11	Plagdet ~0.663	Detecting paraphrased plagiarism across languages	Knowledge graph analysis operates in $O(n \log n)$ for retrieval, $O(n^2)$ for entity linking	Effective for capturing semantic relationships across languages	Requires extensive multilingual knowledge bases for high accuracy

trade-offs in computational efficiency, detection accuracy, and real-world application.

7 Critical assessment of detection methods

Building on the previous section's discussion of plagiarism detection techniques, this section critically evaluates their effectiveness, computational efficiency, and scalability. Each method has distinct advantages and limitations, necessitating a comparative analysis to determine their applicability in different scenarios.

7.1 Trade-offs between computational efficiency and detection accuracy

Different detection methods present trade-offs between computational efficiency and detection accuracy. Traditional

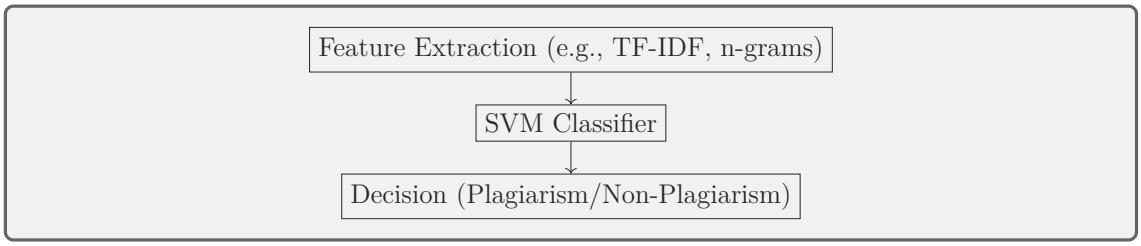

FIGURE 10
Traditional machine learning: feature extraction and classification using an SVM.

TABLE 8 Traditional machine learning approaches.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Hussain and Suryani (2015)	Intelligent plagiarism	Academic papers	χ-Sim, SVM	Custom dataset	PI 48.23%	Detecting text similarity in academic settings	SVM runs in $O(n^2)$ for training, $O(n)$ for inference	Efficient for detecting text similarity in academic settings	Performance is limited by feature extraction quality
Polydouri et al. (2018)	Intrinsic plagiarism	Academic papers	Supervised ML, stylometric features	PAN 2009, 2011, 2016	F1-score 0.43 (Random Forest)	Intrinsic plagiarism detection based on writing style	Random Forest has $O(n \log n)$ training complexity	Can detect intrinsic plagiarism using writing style analysis	Accuracy depends on sufficient stylistic variation in text
El-Rashidy et al. (2022)	Lexical, syntactic, semantic plagiarism	Academic texts	SVM, Chi-square feature selection	PAN 2012, 2013, 2014	F1 89.34%, 92.95%	Plagiarism detection using supervised ML techniques	Chi-square selection is $O(n^2)$, SVM training is $O(n^2)$, inference is $O(n)$	High accuracy across multiple datasets; effective for multiple plagiarism types	Computational overhead can be high for large-scale data

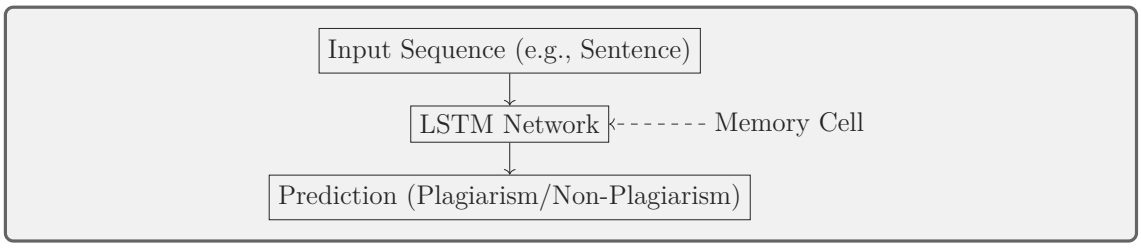

FIGURE 11
Deep learning: LSTM network processing an input sequence and predicting plagiarism.

methods such as n-grams and string matching are computationally efficient but struggle with detecting paraphrased plagiarism. More advanced deep learning methods, while highly effective in semantic analysis, require substantial computational resources and training data.

7.1.1 Comparison of detection methods

Computationally efficient but less accurate methods

- Textual and lexical approaches: traditional methods like string matching and n-gram models excel in computational efficiency due to their simplicity and low resource requirements. For example, Liu et al. (2015) demonstrated a linear complexity algorithm for source code detection with a 0% false alarm rate. However, these methods struggle with paraphrased or obfuscated plagiarism, limiting their robustness.
- Citation-based approaches: techniques analyzing citation patterns (e.g., Velásquez et al., 2016) are computationally efficient but lack the ability to detect nuanced text-level transformations.
- Lexical and string matching techniques: fast but weak against paraphrased content.

TABLE 9 Deep learning approaches.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Shahmohammadi et al. (2020)	Paraphrase detection	Paraphrase in question pairs (NLP)	Bi-LSTM, handcrafted features	MSRP, Quora	Accuracy 79.2%, F1 85.4%	Detecting paraphrase similarity in NLP tasks	Bi-LSTM runs in $O(n^2)$; handcrafted feature extraction adds processing overhead	Effective in handling paraphrased text with deep learning	Handcrafted features require extensive domain expertise
Hayawi et al. (2023)	AI-generated text	Human/AI-generated essays, code	Random Forest, SVM, LSTM	GPT, BARD texts	Accuracy 95.74%	Detecting AI-generated text in academic writing and coding	LSTM runs in $O(n^2)$; SVM and RF scale with dataset size	High accuracy in distinguishing AI-generated text	Performance varies with emerging AI text generators
Suman et al. (2021)	Author profiling	Twitter data (text and images)	BERT, EfficientNet	PAN-2018	Accuracy 89.53%	Social media analytics and multimodal author profiling	No detailed discussion on computational efficiency, but BERT typically runs in $O(n^2)$, EfficientNet in $O(n \log n)$	Use of BERT and EfficientNet enables effective multimodal profiling	Does not address limitations in handling diverse user behaviors
Agarwal et al. (2018)	Paraphrase detection	User-generated texts	CNN + RNN	Microsoft Paraphrase Corpus	F1-score 84.5%	Identifying paraphrased content in online discussions	CNN in $O(n)$, RNN in $O(n^2)$	Effective combination of CNN and RNN for paraphrase detection	Training requires significant computational power
Romanov et al. (2021)	Authorship identification	Russian literary texts	SVM, LSTM, CNN, Transformer	Moshkov library	96% (SVM), 94% (CNN), 87% (LSTM), 93% (Transformer)	Forensic linguistics and author verification	Transformer operates in $O(n^2)$, CNN in $O(n)$, LSTM in $O(n^2)$	High accuracy in authorship identification	Transformer-based models require extensive training data
El-Rashidy et al. (2024)	Lexical, syntactic, semantic plagiarism	Academic texts	LSTM, DenseNet	PAN 2013, PAN 2014	Plagdet 89.81%, 93.92%	Detecting different types of plagiarism in academic texts	LSTM runs in $O(n^2)$, DenseNet operates in $O(n \log n)$	Effective for detecting multiple types of plagiarism	Computationally intensive for large datasets
Shakeel et al. (2020)	Paraphrase detection	Short text paraphrase	CNN, LSTM, data augmentation	Quora, MSRP, SemEval	F1-score 75.4%, 84.8%	Improving paraphrase detection using data augmentation	CNN in $O(n)$, LSTM in $O(n^2)$	Improved accuracy with data augmentation techniques	Requires a large amount of augmented training data
Iqbal et al. (2024)	Paraphrase detection	Urdu texts	DNN (D-TRAPPD, WENGO)	SUSPC, UPPC, USTRC	F1-score 96.80%, 87.85%	Detecting Urdu text reuse and plagiarism	DNN runs in $O(n^2)$ complexity	High accuracy in Urdu paraphrase detection	Deep models require large labeled datasets

10.3389/fcomp.2025.1504725

Methods prioritizing accuracy over efficiency

- Deep learning models: advanced methods, such as LSTM-based approaches (El-Rashidy et al., 2024), achieve superior performance with PlagDet scores surpassing competitors. However, their high computational costs and long training times make them resource-intensive.
- Knowledge graph-based detection: Franco-Salvador et al. (2016b) introduced knowledge graph approaches for cross-language plagiarism, achieving high accuracy but at the cost of significant computational overhead.
- Syntax-based methods: effective for detecting restructured sentences but computationally expensive.
- Word embeddings and semantic models: capture deeper meaning but require large-scale training.

7.2 Scalability challenges and real-world applications

While deep learning methods offer state-of-the-art accuracy, their practical deployment in large-scale systems presents challenges. Institutions and publishers handling vast repositories of documents need hybrid approaches combining efficiency and semantic robustness. Cloud-based parallel processing and selective document screening strategies are potential solutions to balance computational cost with detection performance.

7.2.1 Scalability challenges

Resource-intensive methods face scalability issues, particularly in real-world applications involving large datasets or real-time detection requirements. For instance, Hussain and Suryani (2015) reported exponential increases in training times as dataset sizes grew from 1,000 to 10,000 documents. Similarly, cross-language detection methods relying on extensive semantic analysis often require substantial memory and processing power.

Practical solutions

1. Selective processing: pre-screening techniques, such as text standardization, can reduce the computational load by narrowing the dataset requiring detailed analysis.
2. Distributed computing: leveraging cloud-based systems or parallel processing can improve the scalability of advanced methods.
3. Hybrid techniques: combining traditional methods with advanced semantic approaches provides a balance between efficiency and accuracy. For example, sahi2017novel) integrated syntactic and semantic analysis, achieving scalability and robust detection.

Practical implications

- Real-time applications: systems for educational and publishing environments must prioritize lightweight methods or pre-processing to ensure timely results.
- Large-scale databases: distributed computing and hybrid approaches are essential for managing millions of documents effectively.

By critically assessing the trade-offs between computational efficiency and detection accuracy, this section underscores the need for adaptive and scalable plagiarism detection methods. Future research should focus on hybrid approaches and optimization techniques to achieve a balance suited to diverse real-world applications.

8 Insights, challenges, and future directions

The landscape of plagiarism detection is rapidly evolving due to the increasing complexity of academic writing and the diverse forms of content reuse. Several key insights and challenges have emerged from recent research, providing a foundation for improving plagiarism detection systems.

8.1 Actionable insights

Recent advancements in plagiarism detection have identified key areas for improvement:

1. **Enhanced linguistic models**: incorporating advanced linguistic features such as *semantic role labeling (SRL)* and *dependency parsing* can improve the detection of paraphrased and idea-based plagiarism. Research by Shakeel et al. (2020) demonstrates that fine-tuning large language models (LLMs) like *BERT* and *GPT* for plagiarism detection captures subtle textual variations more effectively.
2. **Cross-lingual plagiarism detection**: multilingual embedding models and *cross-language knowledge graphs* should be further developed to tackle translated plagiarism. Studies such as Franco-Salvador et al. (2016a) show that *word sense disambiguation* enhances semantic equivalence detection across languages, improving multilingual plagiarism detection.
3. **Real-time detection systems**: integrating plagiarism detection tools within writing software can help prevent misconduct at the source. Efficient algorithms are needed for real-time feedback without compromising accuracy (Pertile et al., 2015).
4. **AI-generated content fingerprinting**: the rise of AI-generated content from models like *GPT-4* and *BARD* necessitates the development of classifiers tailored to detect AI-generated text. Hayawi et al. (2023) discuss how AI writing models exhibit unique linguistic “fingerprints" that classifiers can leverage for detection.

8.2 Challenges

Despite these advancements, plagiarism detection systems face several challenges:

1. **Linguistic and discursive variability**: variations in *writing style, tone, and cultural expression* make it difficult to detect

FIGURE 12
Citation pattern analysis: bibliographic coupling between two documents based on shared references.

TABLE 10 Citation-based approaches.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Gipp et al. (2014)	Citation-based plagiarism	Scientific papers	Citation pattern analysis (CbPD), Greedy Tiling	PMC OAS corpus (PubMed)	Fleiss's kappa 0.65	Detecting disguised plagiarism using citation patterns	Greedy Tiling runs in $O(n^2)$; citation pattern analysis in $O(n \log n)$	Effective for detecting disguised plagiarism using citation patterns	Requires high-quality citation data for accuracy
Pertile et al. (2015)	Verbatim, paraphrased, citation plagiarism	Scientific publications	Content-based, citation-based analysis	ACL, PubMed	Precision 0.76, 0.61	Identifying different forms of scientific text plagiarism	Citation-based analysis runs in $O(n \log n)$; content-based varies by method	Strong results for verbatim and paraphrased plagiarism	Citation structure may not always reflect textual similarity
Vani and Gupta (2018)	Paraphrasing, structural plagiarism	Academic papers	POS tagging, WordNet similarity	PAN, PSA	No accuracy provided	Detecting structural plagiarism in academic writing	POS tagging operates in $O(n)$; WordNet similarity in $O(n^2)$	Useful for detecting structural plagiarism	Lacks accuracy benchmarks; WordNet dependency limits scalability

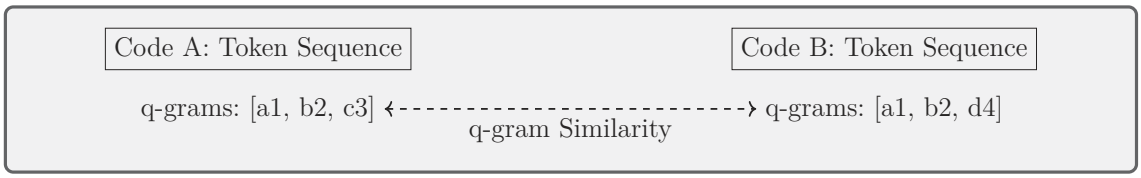

FIGURE 13
Token-based approaches: tokenizing source code into q-grams and comparing them for plagiarism detection.

FIGURE 14
Program dependence graph-based approaches: matching PDGs of two programs to detect plagiarism.

plagiarism, especially in multilingual contexts. Detecting *idea-based plagiarism* requires models that understand discourse-level semantics (Hussain and Suryani, 2015).

2. **Scalability**: many state-of-the-art detection models require significant computational resources, limiting accessibility for smaller institutions. Developing *scalable*

TABLE 11 Code-based plagiarism detection.

References	Plagiarism type	Scope of study	Method	Dataset used	Accuracy	Applications or use cases	Computational complexity	Strengths	Weaknesses
Liu et al. (2015)	Code plagiarism	Programming assignments	Improved LCS, code standardization	Xiangtan University dataset	False alarm 0%, omission 5%	Detecting exact and near-duplicate code plagiarism	LCS operates in $O(nm)$ complexity	High precision in detecting exact and near-duplicate code	May struggle with highly obfuscated code variations
Bartoszuk and Gagolewski (2021)	Source code plagiarism detection	Source code similarity, clone detection	PDG, Levenshtein, q-grams	Simulated R functions	F1-score 0.967	Detecting similarity in code clones and programming assignments	PDG operates in $O(n^2)$, Levenshtein in $O(nm)$, q-grams in $O(n)$	Robust for detecting code similarity across different structures	Computational overhead increases for large-scale codebases

models that maintain high accuracy remains an open research problem.

3. **Adaptability to emerging techniques**: plagiarists increasingly use advanced obfuscation methods such as *automated paraphrasing tools* and *neural translation models*. Detection systems must incorporate *adaptive learning mechanisms* to evolve with these threats (El-Rashidy et al., 2024).

8.3 Recommendations

To address these challenges, future research should focus on:

1. **Integration of discourse analysis**: plagiarism detection systems should incorporate *discourse analysis techniques* to capture nuanced semantic relationships between sentences and paragraphs. This is particularly useful for detecting *idea-based and cross-lingual plagiarism*.
2. **Publicly available benchmarks**: establishing *multilingual datasets* for benchmarking will facilitate consistent evaluation and comparison of plagiarism detection methods. Collaborative initiatives can ensure diverse linguistic and cultural coverage.
3. **Interdisciplinary collaboration**: researchers in *computational linguistics, education, and ethics* should work together to develop holistic solutions that address both the *technical and ethical dimensions* of plagiarism detection.
4. **Education and awareness**: while technological advancements play a crucial role, promoting *academic integrity through education* is equally essential. Institutions should prioritize awareness campaigns alongside detection tool deployment.

8.4 Future research directions

Looking ahead, several opportunities exist to address these challenges:

- Developing **hybrid models** that combine linguistic analysis with deep learning techniques for greater robustness.
- Exploring **multimodal approaches** that integrate text, images, and other data types for comprehensive content analysis (Agarwal et al., 2018).
- Designing **adaptive algorithms** capable of detecting *emerging plagiarism techniques*, including AI-generated content.
- Investigating **user-centric detection tools** that integrate seamlessly into existing workflows, providing *non-intrusive yet effective plagiarism prevention mechanisms*.

By addressing these challenges and pursuing these recommendations, future research can significantly enhance the *effectiveness, fairness, and scalability* of plagiarism detection systems. This will help maintain **academic integrity** in an increasingly complex digital landscape.

9 Conclusion

In this survey, we systematically analyzed various types of plagiarism and the corresponding detection methods, ranging from traditional string-matching techniques to advanced AI-driven approaches. While lexical and shingle-based methods remain effective for detecting verbatim plagiarism, they struggle with more complex cases such as paraphrased, cross-lingual, and AI-generated plagiarism. Recent advancements in deep learning, particularly semantic similarity models and multilingual embeddings, have significantly improved detection accuracy. However, the computational cost and scalability of these approaches remain key challenges.

To enhance plagiarism detection systems, future research should focus on refining cross-language detection using knowledge graphs and multilingual embeddings. The rise of AI-generated content necessitates new techniques, such as linguistic fingerprinting, to differentiate between human and machine-generated text. Additionally, balancing detection accuracy with computational efficiency is crucial for integrating these systems into real-time applications. Hybrid models that combine traditional rule-based methods with AI-driven approaches could offer a scalable solution. Furthermore, developing large-scale, standardized datasets will facilitate better benchmarking and model generalizability, ultimately ensuring more robust and fair plagiarism detection frameworks.

By addressing these challenges, plagiarism detection systems can evolve to meet the growing complexities of academic and digital content integrity. Future efforts should prioritize integrating detection tools into educational and publishing platforms, enabling

real-time feedback mechanisms that help prevent plagiarism at its source. With continued advancements in AI and linguistic analysis, the field is well-positioned to develop more sophisticated, adaptable, and ethical solutions to combat plagiarism in an increasingly digital world.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

AA: Investigation, Methodology, Writing – original draft, Writing – review & editing. CT: Conceptualization, Methodology, Supervision, Writing – review & editing. AM: Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This research was funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan within the framework of project AP23487777.

Acknowledgments

The authors would like to express their sincere gratitude to Dr. Shirali Kadyrov for his invaluable assistance and guidance in the writing of this paper. This work is part of the PhD research of the first author conducted at SDU University.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that Gen AI was used in the creation of this manuscript. We used generative AI tools to enhance the language and assist with proofreading in this paper.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2025.1504725/full#supplementary-material

References

Abdi, A., Idris, N., Alguliyev, R. M., and Aliguliyev, R. M. (2015). Pdlk: plagiarism detection using linguistic knowledge. *Expert Syst. Appl.* 42, 8936–8946. doi: 10.1016/j.eswa.2015.07.048

Agarwal, B., Ramampiaro, H., Langseth, H., and Ruocco, M. (2018). A deep network model for paraphrase detection in short text messages. *Inf. Process. Manage.* 54, 922–937. doi: 10.1016/j.ipm.2018.06.005

Al-Thwaib, E., Hammo, B. H., and Yagi, S. (2020). An academic Arabic corpus for plagiarism detection: design, construction and experimentation. *Int. J. Educ. Technol. High. Educ.* 17, 1–26. doi: 10.1186/s41239-019-0174-x

Alvi, F., Stevenson, M., and Clough, P. (2021). Paraphrase type identification for plagiarism detection using contexts and word embeddings. *Int. J. Educ. Technol. High. Educ.* 18:42. doi: 10.1186/s41239-021-00277-8

Alzahrani, S., and Aljuaid, H. (2022). Identifying cross-lingual plagiarism using rich semantic features and deep neural networks: A study on Arabic-English plagiarism cases. *J. King Saud Univ.-Comput. Inform. Sci.* 34, 1110–1123. doi: 10.1016/j.jksuci.2020.04.009

Alzahrani, S. M., Salim, N., and Palade, V. (2015). Uncovering highly obfuscated plagiarism cases using fuzzy semantic-based similarity model. *J. King Saud Univ.-Comput. Inform. Sci.* 27, 248–268. doi: 10.1016/j.jksuci.2014.12.001

Bartoszuk, M., and Gagolewski, M. (2021). T-norms or t-conorms? How to aggregate similarity degrees for plagiarism detection. *Knowl.-Based Syst.* 231:107427. doi: 10.1016/j.knosys.2021.107427

Benos, D. J., Fabres, J., Farmer, J., Gutierrez, J. P., Hennessy, K., Kosek, D., et al. (2005). Ethics and scientific publication. *Adv. Physiol. Educ.* 29, 59–74. doi: 10.1152/advan.00056.2004

Chekhovich, Y. V., and Khazov, A. V. (2022). Analysis of duplicated publications in Russian journals. *J. Informetr.* 16:101246. doi: 10.1016/j.joi.2021.101246

Darwish, S. M., Mhaimeed, I. A., and Elzoghabi, A. A. (2023). A quantum genetic algorithm for building a semantic textual similarity estimation framework for plagiarism detection applications. *Entropy* 25:1271. doi: 10.3390/e25091271

Ehsan, N., and Shakery, A. (2016). Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information. *Inf. Process. Manage.* 52, 1004–1017. doi: 10.1016/j.ipm.2016.04.006

Ehsan, N., Shakery, A., and Tompa, F. W. (2018). Cross-lingual text alignment for fine-grained plagiarism detection. *J. Inform. Sci.* 45, 443–459. doi: 10.1177/0165551518787696

El-Rashidy, M. A., Mohamed, R. G., El-Fishawy, N. A., and Shouman, M. A. (2022). Reliable plagiarism detection system based on deep learning approaches. *Neural Comput. Appl.* 34, 18837–18858. doi: 10.1007/s00521-022-07486-w

El-Rashidy, M. A., Mohamed, R. G., El-Fishawy, N. A., and Shouman, M. A. (2024). An effective text plagiarism detection system based on feature selection and SVM techniques. *Multimed. Tools Appl.* 83, 2609–2646. doi: 10.1007/s11042-023-15703-4

Errami, M., Hicks, J. M., Fisher, W., Trusty, D., Wren, J. D., Long, T. C., et al. (2008). Déjà vu—a study of duplicate citations in medline. *Bioinformatics* 24, 243–249. doi: 10.1093/bioinformatics/btm574

Franco-Salvador, M., Gupta, P., Rosso, P., and Banchs, R. E. (2016a). Cross-language plagiarism detection over continuous-space- and knowledge graph-based representations of language. *Knowl.-Based Syst*. 111, 87–99. doi: 10.1016/j.knosys.2016.08.004

Franco-Salvador, M., Rosso, P., and Montes-y Gómez, M. (2016b). A systematic study of knowledge graph analysis for cross-language plagiarism detection. *Inf. Process. Manage*. 52, 550–570. doi: 10.1016/j.ipm.2015.12.004

Gandhi, N., Gopalan, K., and Prasad, P. (2024). A support vector machine based approach for plagiarism detection in python code submissions in undergraduate settings. *Front. Comput. Sci*. 6:1393723. doi: 10.3389/fcomp.2024.1393723

Gharavi, E., Veisi, H., and Rosso, P. (2019). Scalable and language-independent embedding-based approach for plagiarism detection considering obfuscation type: no training phase. *Neural Comput. Appl*. 32, 10593–10607. doi: 10.1007/s00521-019-04594-y

Gipp, B., Meuschke, N., and Breitinger, C. (2014). Citation-based plagiarism detection: Practicability on a large-scale scientific corpus. *J. Assoc. Inform. Sci. Technol*. 65, 1527–1540. doi: 10.1002/asi.23228

Glavaš, G., Franco-Salvador, M., Ponzetto, S. P., and Rosso, P. (2018). A resource-light method for cross-lingual semantic textual similarity. *arXiv* [Preprint]. arXiv:1801.06436. doi: 10.48550/arXiv.1801.06436

Hayawi, K., Shahriar, S., and Mathew, S. S. (2023). The imitation game: detecting human and AI-generated texts in the era of chatgpt and bard. *arXiv* [Preprint]. arXiv:2307.12166. doi: 10.48550/arXiv.2307.12166

Hussain, S. F., and Suryani, A. (2015). On retrieving intelligently plagiarized documents using semantic similarity. *Eng. Appl. Artif. Intell*. 45, 246–258. doi: 10.1016/j.engappai.2015.07.011

Iqbal, H. R., Maqsood, R., Raza, A. A., and Hassan, S.-U. (2024). Urdu paraphrase detection: a novel dnn-based implementation using a semi-automatically generated corpus. *Nat. Lang. Eng*. 30, 354–384. doi: 10.1017/S1351324923000189

Lariviere, V., and Gingras, Y. (2010). On the prevalence and scientific impact of duplicate publications in different scientific fields (1980-2007). *J. Document*. 66, 179–190. doi: 10.1108/00220411011023607

Liu, X., Xu, C., and Ouyang, B. (2015). Plagiarism detection algorithm for source code in computer science education. *Int. J. Dist. Educ. Technol*. 13, 29–39. doi: 10.4018/IJDET.2015100102

Malandrino, D., De Prisco, R., Ianulardo, M., and Zaccagnino, R. (2022). An adaptive meta-heuristic for music plagiarism detection based on text similarity and clustering. *Data Min. Knowl. Discov*. 36, 1301–1334. doi: 10.1007/s10618-022-00835-2

Manzoor, M. F., Farooq, M. S., Haseeb, M., Farooq, U., Khalid, S., Abid, A., et al. (2023). Exploring the landscape of intrinsic plagiarism detection: benchmarks, techniques, evolution, and challenges. *IEEE Access* 11, 14706–14729. doi: 10.1109/ACCESS.2023.3338855

Mehak, G., Muneer, I., and Nawab, R. M. A. (2023). Urdu text reuse detection at phrasal level using sentence transformer-based approach. *Expert Syst. Appl*. 234:121063. doi: 10.1016/j.eswa.2023.121063

Pertile, S. L., Moreira, V. P., and Rosso, P. (2015). Comparing and combining content- and citation-based approaches for plagiarism detection. *J. Assoc. Inform. Sci. Technol*. 66, 1976–1991. doi: 10.1002/asi.23593

Polydouri, A., Vathi, E., Siolas, G., and Stafylopatis, A. (2018). An efficient classification approach in imbalanced datasets for intrinsic plagiarism detection. *Evol. Syst*. 11, 503–515. doi: 10.1007/s12530-018-9232-1

Roig, M. (2006). *Avoiding plagiarism, self-plagiarism, and other questionable writing practices: A guide to ethical writing*. Available online at: https://ori.hhs.gov/sites/default/files/plagiarism.pdf

Romanov, A., Kurtukova, A., Shelupanov, A., Fedotova, A., and Goncharov, V. (2021). Authorship identification of a russian-language text using support vector machine and deep neural networks. *Future Internet* 13:3. doi: 10.3390/fi13010003

Roostaee, M., Fakhrahmad, S. M., and Sadreddini, M. H. (2020). Cross-language text alignment: a proposed two-level matching scheme for plagiarism detection. *Expert Syst. Appl*. 160:113718. doi: 10.1016/j.eswa.2020.113718

Sahi, M., and Gupta, V. (2017). A novel technique for detecting plagiarism in documents exploiting information sources. *Cogn. Comput*. 9, 852–867. doi: 10.1007/s12559-017-9502-4

Shahmohammadi, H., Dezfoulian, M., and Mansoorizadeh, M. (2020). Paraphrase detection using LSTM networks and handcrafted features. *Multimed. Tools Appl*. 80, 24137–24155. doi: 10.1007/s11042-020-09996-y

Shakeel, M. H., Karim, A., and Khan, I. (2020). A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts. *Inf. Process. Manage*. 57:102204. doi: 10.1016/j.ipm.2020.102204

Suman, C., Naman, A., Saha, S., and Bhattacharyya, P. (2021). A multimodal author profiling system for tweets. *IEEE Trans. Comput. Soc. Syst*. 8, 1407–1416. doi: 10.1109/TCSS.2021.3082942

Taufiq, U., Pulungan, R., and Suyanto, Y. (2023). Named entity recognition and dependency parsing for better concept extraction in summary obfuscation detection. *Expert Syst. Appl*. 217:119579. doi: 10.1016/j.eswa.2023.119579

Turrado García, F., García Villalba, L. J., Sandoval Orozco, A. L., Aranda Ruiz, F. D., Aguirre Juárez, A., and Kim, T.-H. (2018). Locating similar names through locality sensitive hashing and graph theory. *Multimed. Tools Appl*. 78, 19965–19985. doi: 10.1007/s11042-018-6375-9

Vani, K., and Gupta, D. (2017a). Detection of idea plagiarism using syntax-semantic concept extractions with genetic algorithm. *Expert Syst. Appl*. 71, 106–122. doi: 10.1016/j.eswa.2016.12.022

Vani, K., and Gupta, D. (2017b). Text plagiarism classification using syntax based linguistic features. *Expert Syst. Appl*. 94, 29–43. doi: 10.1016/j.eswa.2017.07.006

Vani, K., and Gupta, D. (2018). Integrating syntax-semantic-based text analysis with structural and citation information for scientific plagiarism detection. *J. Assoc. Inf. Sci. Technol*. 69, 1186–1203. doi: 10.1002/asi.24027

Velásquez, J. D., Covacevich, Y., Molina, F., Marrese-Taylor, E., Rodríguez, C., and Bravo-Marquez, F. (2016). DOCODE 3.0 (DOcument COpy DEtector): a system for plagiarism detection by applying an information fusion process from multiple documental data sources. *Inf. Fusion* 27, 64–75. doi: 10.1016/j.inffus.2015.05.006