Journal of COMPUTER AND FORENSIC SCIENCES

e-ISSN 2956-0799

Volume 4 • Issue 2 • 2025

UNIVERSITY OF CRIMINAL INVESTIGATION AND POLICE STUDIES
Belgrade

UNIVERSITY OF CRIMINAL INVESTIGATION AND POLICE STUDIES, BELGRADE – THE REPUBLIC OF SERBIA
KRIMINALISTIČKO-POLICIJSKI UNIVERZITET, BEOGRAD – REPUBLIKA SRBIJA

CFS

CFS. JOURNAL OF COMPUTER AND FORENSIC SCIENCES

Belgrade, 2025

PUBLISHER

University of Criminal Investigation and Police Studies, Cara Dušana 196, 11080 Belgrade, Serbia

TABLE OF CONTENTS

Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2025, Vol. 4, Issue 2, p. 1
Editorial

EDITORIAL

CFS: vol. 4, no. 2

Milan Gnjatović
Editor-in-Chief
University of Criminal Investigation and Police Studies, Belgrade;
milan.gnjatovic@kpu.edu.rs
Published: May 14, 2026

Discussing the impact of the computer on society, Weizenbaum wrote that "we must realize that man's commitment to science has always had a masochistic component" [4]. Academic journal editors understand this all too well. However, there are brief moments when it seems that all the trouble pays off – when you get to present a new issue.

This issue of the Journal of Computer and Forensic Sciences brings three interesting research articles in the field of machine and deep learning [1-3]. I thank the authors and reviewers for their valuable support in preparing this contribution and hope you will enjoy reading it.

REFERENCES

[1] R. Birwadkar, "A Hybrid Plagiarism Detection Framework Using Lexical and Semantic Similarity with Lightweight Sentence Transformers", *Journal of Computer and Forensic Sciences*, vol. 4, no. 2, pp. 3–16, 2025.

[2] R. Birwadkar, "Semantic Paraphrase Generation Using Transformer Architectures: A Comparative Study of Pre-Trained and Fine-Tuned Models", *Journal of Computer and Forensic Sciences*, vol. 4, no. 2, pp. 17–29, 2025.

[3] S. Raičević, "Comparative Analysis of Clustering Textual and Numerical Data Using the K-Means Algorithm", *Journal of Computer and Forensic Sciences*, vol. 4, no. 2, pp. 30–42, 2025.

[4] W. Joseph, "On the Impact of the Computer on Society: How does one insult a machine?", *Science*, vol. 176, no. 4035, pp. 609–14, 1972.

ORIGINAL RESEARCH PAPERS

Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2025, Vol. 4, Issue 2, pp. 3–16
https://doi.org/10.5937/jcfs4-64419
Original research paper

A Hybrid Plagiarism Detection Framework Using Lexical and Semantic Similarity with Lightweight Sentence Transformers

Rahul Birwadkar[1*]

[1] Student, SRH University, Heidelberg, Germany

*Corresponding author: rbirwadkar6@gmail.com

Received: January 27, 2026 • Accepted: April 6, 2026 • Published: May 14, 2026

Abstract: Plagiarism detection has become increasingly challenging due to the widespread availability of paraphrasing tools and generative artificial intelligence systems. Traditional plagiarism detection techniques based on lexical similarity, such as TF-IDF and n-gram matching, often fail to identify semantically similar but lexically modified text. This paper presents a hybrid plagiarism detection framework that combines lexical similarity measures with semantic similarity derived from sentence transformer models. The proposed approach integrates TF-IDF–based cosine similarity with lightweight sentence embeddings generated using MiniLM and SBERT models. To enhance semantic detection performance, a MiniLM-based sentence transformer is fine-tuned on the PAN 2011 plagiarism detection corpus. Experimental evaluation demonstrates that the hybrid similarity approach significantly improves detection accuracy compared to purely lexical methods, particularly for paraphrased plagiarism cases. The framework is further validated using threshold-based analysis and real-world web content retrieved through automated scraping. The proposed system provides an efficient and scalable solution for plagiarism detection, balancing computational efficiency with semantic understanding, and is suitable for academic and real-world forensic applications.

Keywords: plagiarism detection, semantic similarity, sentence transformers, MiniLM, natural language processing.

1. INTRODUCTION

Plagiarism detection is a critical component of academic integrity, digital forensics, and content verification systems [1]. The rapid growth of digital content and the widespread availability of paraphrasing and generative text tools have significantly increased the complexity of identifying plagiarized material [1]. While traditional plagiarism detection techniques are effective in identifying exact text reuse, they often fail when content is modified through paraphrasing or structural rewriting [1], [2]. Conventional plagiarism detection systems primarily rely on lexical similarity measures such as n-gram matching, term frequency analysis, and string-based comparisons [2]. These methods are limited to surface-level text analysis and are highly sensitive to lexical changes, causing semantically equivalent but lexically altered text to remain undetected [1], [2]. Recent advances in natural language processing have introduced semantic representations capable of capturing

Citation: R. Birwadkar, "A Hybrid Plagiarism Detection Framework Using Lexical and Semantic Similarity with Lightweight Sentence Transformers," Journal of *Computer and Forensic Sciences*, vol. 4, no. 2, pp. 3–16, 2025, https://doi.org/10.5937/jcfs4-64419.

contextual meaning beyond exact word overlap [3]. Transformer-based language models enable the comparison of textual content at the semantic level, making it possible to identify paraphrased or meaning-preserving text reuse [3], [4]. However, purely semantic approaches may overlook exact textual matches and can be computationally expensive for large-scale applications [4], [5]. To address these limitations, this paper proposes a hybrid plagiarism detection framework that integrates lexical and semantic similarity measures. By combining TF-IDF–based similarity with sentence-level semantic embeddings generated by lightweight transformer models, the proposed approach aims to improve detection accuracy while maintaining computational efficiency. The main contributions of this work are as follows:

1) A hybrid similarity framework combining lexical and semantic plagiarism detection techniques.
2) A fine-tuned MiniLM-based sentence transformer optimized for plagiarism detection.
3) A comprehensive experimental evaluation on the PAN 2011 plagiarism detection dataset.
4) Validation of the proposed framework on real-world web content.

2. RELATED WORK

Plagiarism detection has been an active research area for several decades, particularly in academic integrity, digital forensics, and content verification systems. Existing approaches can broadly be classified into lexical-based methods, traditional machine learning approaches, and semantic or deep learning–based techniques. Each category exhibits distinct strengths and limitations, especially when applied to paraphrased or obfuscated plagiarism.

2.1. Lexical-Based Plagiarism Detection Approaches

Early plagiarism detection systems primarily relied on lexical similarity measures that compare surface-level textual features [1], [2]. Common techniques include string matching, n-gram overlap, term frequency analysis, and cosine similarity using vector space models. Among these, TF-IDF has been widely adopted due to its simplicity, interpretability, and effectiveness in detecting exact or near-exact text reuse [2]. Lexical approaches perform well when plagiarized content is copied verbatim or with minimal modifications. N-gram–based methods are effective at identifying localized overlap, while TF-IDF captures broader document-level similarity patterns, making these approaches computationally efficient and scalable [2]. However, lexical similarity methods are highly sensitive to word-level changes. Simple paraphrasing techniques such as synonym substitution, sentence restructuring, or word reordering can significantly reduce lexical overlap while preserving semantic meaning. As a result, purely lexical approaches often fail to detect paraphrased plagiarism, leading to high false-negative rates [1].

2.2. Traditional Machine Learning–Based Methods

To overcome the limitations of surface-level similarity measures, traditional machine learning approaches introduced classification-based plagiarism detection systems [1]. These methods typically rely on handcrafted features such as lexical overlap statistics, syntactic patterns, stylometric features, and similarity scores computed at different granularity levels. Although such approaches improved detection accuracy compared to purely lexical methods, they depend heavily on feature engineering and domain-specific heuristics. Designing effective features requires expert knowledge and often lacks generalization across datasets and languages, limiting robustness against advanced paraphrasing techniques [1].

2.3. Semantic Similarity and Embedding-Based Approaches

The introduction of distributed word representations marked a significant shift in plagiarism detection research [2]. Word embedding models enabled semantic comparison of text by capturing contextual relationships between words, allowing similarity computation beyond exact word overlap. Building upon word embeddings, deep learning architectures such as recurrent neural networks and encoder–decoder models enabled sentence- and document-level semantic representation [6], [7]. These approaches demonstrated improved performance in detecting meaning-preserving text transformations but often suffered from scalability limitations. Transformer-based models further advanced semantic similarity modeling by leveraging self-attention mechanisms to capture contextual dependencies across entire text sequences [3], [7]. Pre-trained transformer models provide strong general-purpose language representations that can be adapted to downstream tasks, including plagiarism detection.

2.4. Sentence-Level Embeddings for Plagiarism Detection

Sentence-level embedding models offer an efficient alternative to full sequence-to-sequence architectures by generating fixed-length vector representations for sentences or documents. These embeddings enable direct similarity computation using distance metrics such as cosine similarity, making them suitable for large-scale plagiarism detection systems [4]. Sentence transformer architectures explicitly optimize embeddings for similarity tasks using Siamese or triplet network structures. Lightweight variants, such as distilled transformer models, reduce computational cost while retaining strong semantic representation capabilities [5]. Despite these advantages, purely semantic approaches may overlook exact textual reuse or produce high similarity scores for conceptually related but non-plagiarized content.

2.5 Hybrid Plagiarism Detection Approaches

Hybrid plagiarism detection methods aim to integrate lexical and semantic similarity measures to leverage the strengths of both approaches [1], [4]. Lexical similarity contributes

precision by identifying exact text reuse, while semantic similarity improves recall by detecting paraphrased or meaning-preserving content. The proposed framework builds upon this line of research by introducing a lightweight hybrid approach that combines TF-IDF–based lexical similarity with sentence-level semantic embeddings generated by a fine-tuned MiniLM model. This design balances detection performance and computational efficiency, making it suitable for scalable plagiarism detection in academic and forensic contexts.

Earlier neural approaches based on recurrent architectures, including RNNs and encoder–decoder models, have been widely explored for sequence modeling tasks [6], [8], [9], [10]. More recently, transformer-based architectures such as BERT and its variants have significantly improved contextual text representation capabilities [3]. Generative transformer models, including BART and T5, have further advanced text generation and transformation tasks [11], [12]. In addition, knowledge distillation techniques have enabled the development of lightweight transformer models for efficient inference [13], supporting the use of compact architectures such as MiniLM. Earlier work on paraphrase generation and semantic similarity also includes statistical and rule-based approaches [14], [15].

3. PROPOSED HYBRID PLAGIARISM DETECTION FRAMEWORK

This section presents the proposed hybrid plagiarism detection framework, which integrates lexical and semantic similarity measures to improve the detection of plagiarized and paraphrased text. The framework is designed to address the limitations of purely lexical approaches while maintaining computational efficiency suitable for real-world deployment [16].

3.1. System Overview

The proposed framework follows a modular pipeline consisting of text preprocessing, lexical similarity computation, semantic similarity computation, hybrid similarity aggregation, and threshold-based classification. Given a suspicious document and a reference document, similarity scores are computed using both lexical and semantic approaches and combined into a unified hybrid similarity score used for plagiarism detection.

Figure 1 illustrates the workflow of the proposed hybrid plagiarism detection framework. Input document pairs (suspicious and reference documents) are first preprocessed using standard natural language processing techniques. The system then computes similarity through two parallel components.

In the lexical similarity branch, TF-IDF vectorization is applied, followed by cosine similarity to obtain the lexical similarity score. In the semantic similarity branch, sentence-level embeddings are generated using a MiniLM-based transformer model, and cosine similarity is applied to compute the semantic similarity score. The final hybrid similarity score is calculated as the average of the maximum lexical and semantic similarity scores for each document pair. This score is then used in a threshold-based classification mechanism to determine whether the document pair is plagiarized or non-plagiarized. This modular design enables flexibility by allowing individual components to be adapted or extended based on application requirements.

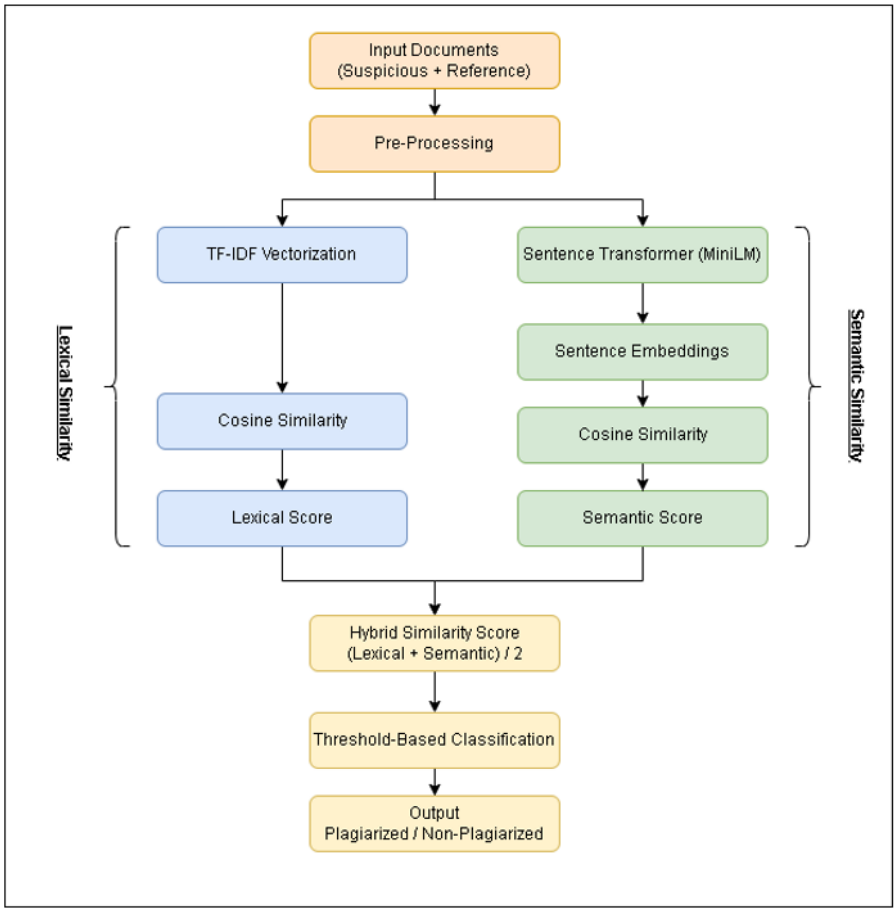


Figure 1: *Overview of the proposed hybrid plagiarism detection framework.*

3.2. Lexical Similarity Computation

Lexical similarity is computed using the Term Frequency–Inverse Document Frequency (TF-IDF) representation combined with cosine similarity. TF-IDF assigns higher weights to terms that are frequent within a document but rare across the corpus, making it effective for identifying exact or near-exact text reuse [2]. Each document is transformed into a TF-IDF vector, and cosine similarity is used to measure similarity between document pairs. To ensure fair comparison across documents of varying length, L2 normalization is applied to the TF-IDF vectors. This lexical similarity component serves as a strong baseline for detecting verbatim plagiarism but is limited in handling paraphrased or semantically altered text.

3.3. Semantic Similarity Using Sentence Transformers

To capture semantic similarity beyond surface-level word overlap, the framework employs sentence-level embeddings generated using a transformer-based language model.

Sentence transformer architectures are specifically designed to produce embeddings optimized for similarity comparison tasks [4]. In this work, a lightweight MiniLM-based sentence transformer is used due to its balance between semantic representation quality and computational efficiency [5]. Each document is encoded into a fixed-length dense vector representing its semantic content, and cosine similarity is computed between embedding pairs. To improve task-specific performance, the MiniLM model is fine-tuned using labelled document pairs derived from the PAN 2011 plagiarism detection dataset. Fine-tuning adapts the embedding space to better distinguish between plagiarized and non-plagiarized content while preserving the general linguistic knowledge of the pre-trained model.

3.4. Hybrid Similarity Score

The central contribution of this work is the integration of lexical and semantic similarity measures into a unified hybrid similarity score. Instead of relying on a single similarity metric, the proposed approach combines both perspectives to achieve more robust plagiarism detection.

The hybrid similarity score is defined as:

$$Hybrid\ Similarity = \frac{Lexical\ Similarity + \ Semantic\ Similarity}{2}$$

In the proposed implementation, the hybrid similarity score is computed as the average of the maximum lexical and semantic similarity scores obtained for each document pair. This formulation ensures that both exact lexical overlap and semantic equivalence contribute equally to the final similarity assessment. Lexical similarity provides precision for detecting copied text, while semantic similarity improves recall for paraphrased or meaning-preserving text reuse. The use of an arithmetic mean provides a simple and interpretable aggregation strategy while avoiding the need for additional hyperparameter tuning. While equal weighting is effective for the current study, future work may explore weighted combinations of lexical and semantic similarity, where optimal weighting factors can be learned or empirically determined to further improve detection performance.

3.5. Threshold-Based Classification

A threshold-based decision mechanism is applied to the hybrid similarity score to classify document pairs. If the computed hybrid similarity exceeds a predefined threshold, the document pair is classified as plagiarized; otherwise, it is classified as non-plagiarized. Different threshold values are evaluated experimentally to analyze their effect on detection performance. Lower thresholds favor recall by identifying a broader range of plagiarism cases, while higher thresholds improve precision by reducing false positives. This flexibility allows the framework to be adapted to different application scenarios, such as academic evaluation or forensic analysis.

3.6. Computational Complexity and Efficiency

Computational efficiency is an important consideration for plagiarism detection systems intended for large-scale or real-world deployment. The proposed hybrid framework is designed to balance detection accuracy with computational cost by combining efficient lexical similarity computation with a lightweight semantic model. TF-IDF vectorization and cosine similarity are computationally inexpensive and scale well with corpus size once document vectors are constructed. The MiniLM-based sentence transformer significantly reduces inference time and model size compared to larger transformer architectures while maintaining strong semantic representation capabilities. Fine-tuning is performed only once during training. During inference, embeddings for reference documents can be precomputed and stored, allowing similarity computation to be performed efficiently using vector operations. The hybrid similarity formulation avoids complex feature fusion or multi-stage classification pipelines, further reducing computational overhead. Overall, the proposed framework provides a practical and scalable solution for plagiarism detection that balances performance and efficiency.

4. EXPERIMENTAL SETUP

This section describes the dataset, preprocessing pipeline, model configuration, training strategy, evaluation protocol, and implementation details used to evaluate the proposed hybrid plagiarism detection framework. The goal is to ensure reproducibility and provide sufficient experimental detail for fair comparison with existing approaches.

4.1. Dataset Description

The experimental evaluation is conducted using the PAN 2011 plagiarism detection dataset, a widely used benchmark for external plagiarism detection research. The dataset consists of suspicious documents, corresponding source documents, and XML-based annotations identifying plagiarized text segments. The dataset includes various forms of plagiarism, ranging from exact copying to heavily paraphrased content. Each suspicious document may contain zero or more plagiarized segments originating from one or multiple source documents. This diversity makes the dataset suitable for evaluating both lexical and semantic plagiarism detection methods. The XML annotation files provide detailed metadata, including character offsets of plagiarized segments and plagiarism type information. These annotations are parsed to generate labelled document pairs for supervised training and evaluation.

4.2 Data Preprocessing

All documents undergo a standardized preprocessing pipeline prior to similarity computation. The preprocessing steps include tokenization, lowercasing, stop-word removal, and

normalization of punctuation and special characters. For lexical similarity computation, lemmatization is applied to reduce inflected word forms to their base representations. For semantic similarity computation, raw text is preserved as much as possible to retain contextual information required by transformer-based models. Documents that exceed the maximum token length supported by the model are truncated to ensure consistent input representation. This preprocessing strategy balances semantic fidelity with computational efficiency.

4.3. Lexical Similarity Configuration

Lexical similarity is computed using TF-IDF vectorization combined with cosine similarity. Unigrams and bigrams are used to capture both individual term usage and short contextual patterns. L2 normalization is applied to TF-IDF vectors to mitigate the influence of document length on similarity scores. The TF-IDF model is trained on the combined corpus of suspicious and source documents to ensure consistent term weighting across all document pairs. This configuration provides a reliable baseline for detecting exact and near-exact text reuse.

4.4. Semantic Similarity Model and Fine-Tuning Strategy

Semantic similarity is computed using a lightweight sentence transformer based on the MiniLM architecture. The model generates fixed-length dense embeddings that represent the semantic content of documents, enabling efficient similarity computation using cosine similarity. To adapt the model to the plagiarism detection task, fine-tuning is performed using labeled document pairs derived from the PAN dataset. Each training sample consists of a suspicious document, a corresponding source document, and a binary label indicating the presence of plagiarism. Fine-tuning optimizes the embedding space to increase similarity between plagiarized document pairs while reducing similarity for non-plagiarized pairs. A low learning rate is used to preserve general language understanding while enabling task-specific adaptation.

4.5. Training Configuration

The fine-tuning process is conducted using mini-batch training with a fixed batch size. The model is trained for a limited number of epochs to balance convergence and generalization. Dropout regularization is applied to reduce overfitting and improve robustness. The AdamW optimizer is employed due to its effectiveness and stability when fine-tuning transformer-based models. After training, the fine-tuned model is used exclusively in inference mode for embedding generation, allowing document embeddings to be precomputed and reused during similarity evaluation.

4.6. Evaluation Metrics

The framework is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. These metrics provide complementary perspectives on detection performance, particularly in cases of class imbalance. Confusion matrices are additionally used to analyze false positive and false negative cases, providing insight into error characteristics across different similarity approaches.

4.7. Evaluation Protocol

The evaluation follows a document-pair comparison protocol. Lexical, semantic, and hybrid similarity scores are computed independently for each document pair. A threshold-based decision mechanism is then applied to classify each pair as plagiarized or non-plagiarized. Multiple similarity thresholds are evaluated to analyze robustness. Lower thresholds emphasize recall, while higher thresholds favor precision. This analysis supports adaptation of the framework to different application requirements.

4.8. Implementation Details

The experimental framework is implemented in Python using widely adopted natural language processing and deep learning libraries. Lexical similarity is computed using scikit-learn, including TF-IDF vectorization and cosine similarity. Semantic similarity is computed using Sentence-Transformers and Hugging Face Transformers, while model training and inference are carried out using PyTorch. Data handling and analysis are performed using pandas and NumPy, text preprocessing is performed using standard Python and NLTK utilities, and result visualization is performed using Matplotlib and Seaborn. Additional utilities such as tqdm are used for progress tracking.

5. RESULTS AND ANALYSIS

This section presents the experimental results of the proposed hybrid plagiarism detection framework. The performance of lexical, semantic, and hybrid similarity approaches is evaluated and compared using standard classification metrics. In addition, a threshold-based analysis is conducted to examine the robustness of the framework under different operating conditions.

5.1. Lexical Similarity Results

Lexical similarity based on TF-IDF and cosine similarity is evaluated as a baseline approach. The results show that lexical similarity performs reliably in cases of exact or near-exact text reuse. High precision is achieved when plagiarized content exhibits substantial lexical overlap with source documents. However, the performance of lexical simi-

larity decreases significantly for paraphrased plagiarism cases. Modifications such as synonym replacement, sentence restructuring, and word reordering reduce lexical overlap, leading to lower similarity scores and increased false negatives. These findings confirm the limitations of purely lexical plagiarism detection methods.

5.2. Semantic Similarity Results

Semantic similarity is evaluated using sentence embeddings generated by MiniLM-based sentence transformer models. Compared to lexical similarity, semantic approaches demonstrate improved performance in detecting paraphrased and meaning-preserving text reuse. The fine-tuned MiniLM model shows better recall and overall detection capability than the pre-trained variant. Fine-tuning enables the model to adapt to the characteristics of plagiarism detection, resulting in more reliable similarity scores for semantically equivalent document pairs. However, semantic similarity alone may produce higher similarity scores for conceptually related but non-plagiarized text, which can affect precision.

Table 1: *Performance comparison of pre-trained and fine-tuned MiniLM sentence embeddings under different similarity thresholds.*

Model	Approach	Accuracy	Precision	Recall	F1 Score	False Positive	False Negative
Pre-Trained (T=0.4)	Pre-trained Embeddings + Cosine Similarity	0.8825	0.9961	0.7678	0.8672	3	232
Pre-Trained (T=0.8)	Pre-trained Embeddings + Cosine Similarity	0.501	1	0.001	0.002	0	998
Sentence Transformer based (T=0.4)	Fine-tuned Embeddings + Cosine Similarity	0.494	0.493	0.4605	0.4762	473	539
Sentence Transformer based (T=0.8)	Fine-tuned Embeddings + Cosine Similarity	0.761	1	0	0	0	999
Direct Fine-Tune Model (T=0.4)	Fine-tuned MiniLM Model Cosine Similarity	0.761	0.6776	0.995	0.8062	473	5
Direct Fine-Tune Model (T=0.8)	Fine-tuned MiniLM Model Cosine Similarity	0.6355	0.9355	0.2903	0.4431	20	709

Table 1 compares the performance of pre-trained and fine-tuned MiniLM sentence embeddings under different similarity thresholds. The results indicate that fine-tuning significantly improves recall and overall F1-score, particularly at lower threshold values, enabling more effective detection of paraphrased plagiarism. In contrast, higher threshold values result in overly conservative behavior, leading to a sharp decline in recall for both pre-trained and fine-tuned models.

5.3. Hybrid Similarity Results

The proposed hybrid similarity framework combines lexical and semantic similarity scores into a unified metric. Experimental results indicate that the hybrid approach consistently outperforms individual similarity methods across evaluation metrics. By inte-

grating lexical precision with semantic robustness, the hybrid framework achieves a more balanced trade-off between precision and recall. This balance is particularly important in plagiarism detection scenarios, where both false positives and false negatives can have significant consequences. The hybrid approach effectively mitigates the weaknesses of individual similarity methods while preserving their strengths.

5.4. Threshold-Based Performance Analysis

To analyze the sensitivity of the framework, multiple similarity thresholds are evaluated. Lower threshold values increase recall by identifying a larger number of plagiarism cases, including heavily paraphrased content, but may introduce additional false positives. Conversely, higher thresholds improve precision by reducing false positives at the cost of lower recall. The results indicate that moderate threshold values provide the most effective balance between precision and recall. This flexibility allows the framework to be configured according to application requirements, such as strict academic evaluation or broader forensic screening.

5.5. Comparative Summary

A comparative analysis across lexical, semantic, and hybrid approaches highlights the advantages of the proposed framework. Lexical similarity performs well for detecting direct copying but struggles with paraphrased text. Semantic similarity improves detection of meaning-preserving modifications but may affect precision when used alone. The hybrid similarity framework leverages the complementary strengths of both approaches, resulting in improved overall performance. These results demonstrate that combining lexical and semantic similarity provides a more robust and reliable solution for plagiarism detection than using either method independently.

6. DISCUSSION

The experimental results demonstrate that combining lexical and semantic similarity measures provides a more robust approach to plagiarism detection than relying on either method independently. Lexical similarity techniques, such as TF-IDF, are effective for identifying direct text reuse but are limited when confronted with paraphrased or structurally modified content. Semantic similarity methods address this limitation by capturing contextual meaning, enabling the detection of meaning-preserving text reuse.

The proposed hybrid framework benefits from the complementary strengths of both approaches. Lexical similarity contributes precision by accurately identifying exact or near-exact overlap, while semantic similarity improves recall by detecting paraphrased plagiarism. The aggregation of these similarity signals results in a balanced detection strategy that reduces false negatives without introducing an excessive number of false positives.

The use of a lightweight MiniLM-based sentence transformer further enhances the practicality of the framework. Compared to larger transformer architectures, MiniLM offers reduced computational cost while maintaining strong semantic representation capability. Fine-tuning the model on a plagiarism-specific dataset enables better adaptation to the characteristics of plagiarism detection, improving task-specific performance without sacrificing efficiency. Threshold-based analysis highlights the adaptability of the framework. By adjusting similarity thresholds, the system can be configured for different application scenarios. Lower thresholds are suitable for exploratory or forensic analysis where recall is prioritized, while higher thresholds are appropriate for academic evaluation scenarios that require higher precision.

Despite these strengths, the framework has certain limitations. Similarity alone does not necessarily indicate plagiarism, as proper citation and source attribution play an important role in distinguishing legitimate referencing from unethical text reuse. It operates primarily at the document level and does not explicitly distinguish between cited and uncited text reuse. As a result, properly referenced quotations may still be flagged as similar. In addition, the performance of semantic similarity methods depends on the representativeness of the training data, which may affect generalization to domains with substantially different writing styles or vocabulary. Overall, the discussion confirms that the hybrid similarity strategy provides a practical and effective solution for plagiarism detection, balancing detection accuracy, interpretability, and computational efficiency.

7. CONCLUSION AND FUTURE WORK

This paper presented a hybrid plagiarism detection framework that integrates lexical and semantic similarity measures to improve the detection of plagiarized and paraphrased text. By combining TF-IDF–based lexical similarity with sentence-level semantic embeddings generated using a lightweight transformer model, the proposed approach addresses the limitations of traditional plagiarism detection techniques.

Experimental evaluation on the PAN 2011 plagiarism detection dataset demonstrates that the hybrid framework outperforms purely lexical and purely semantic approaches across multiple evaluation metrics. Fine-tuning a MiniLM-based sentence transformer further enhances detection performance while maintaining computational efficiency. Additional validation using real-world web content highlights the applicability of the framework in practical plagiarism detection scenarios.

Despite its effectiveness, the proposed framework focuses on textual similarity and does not explicitly distinguish between cited and uncited text reuse. Future work may incorporate citation-aware analysis to better differentiate legitimate, properly referenced reuse from plagiarism. Furthermore, extending the framework to support sentence- and paragraph-level analysis may improve fine-grained detection accuracy. Additional future research directions include cross-lingual plagiarism detection and the integration of contextual metadata to enhance robustness. These extensions could further improve the effectiveness of hybrid similarity-based plagiarism detection systems in diverse academic and forensic contexts.

FUNDING:
This research received no external funding.

INSTITUTIONAL REVIEW BOARD STATEMENT:
Not applicable.

INFORMED CONSENT STATEMENT:
Not applicable.

CONFLICTS OF INTEREST:
The author declares no conflict of interest.

REFERENCES

[1] K. T. Kalleberg, "Plagiarism detection using machine learning techniques," International Journal of Computer Applications, vol. 119, no. 13, pp. 1–6, 2015.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proceedings of the International Conference on Learning Representations (ICLR), 2013.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019, pp. 3982–3992.

[5] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 3104–3112.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proceedings of the International Conference on Learning Representations (ICLR), 2015.

[8] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157–166, 1994, doi: 10.1109/72.279181.

[9] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[10] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in Proceedings of EMNLP, 2015, pp. 1412–1421.

[11] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.

[12] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.

[13] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," International Journal of Computer Vision, vol. 129, pp. 1789–1819, 2021.

[14] C. Quirk, C. Brockett, and W. Dolan, "Monolingual machine translation for paraphrase generation," in Proceedings of EMNLP, 2004, pp. 142–149.

[15] D. Lin and P. Pantel, "Discovery of inference rules for question answering," Natural Language Engineering, vol. 7, no. 4, pp. 343–360, 2001.

[16] R. V. Birwadkar, "Plagiarism Detection and Paraphrasing based on Generative Artificial Intelligence," Master's thesis, Dept. of Information and Technology, SRH Hochschule Heidelberg, Heidelberg, Germany, 2025.

Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2025, Vol. 4, Issue 2, pp.17–29
https://doi.org/10.5937/jcfs4-64420
Original research paper

Semantic Paraphrase Generation Using Transformer Architectures: A Comparative Study of Pre-Trained and Fine-Tuned Models

Rahul Birwadkar[1*]

[1] Student, SRH University, Heidelberg, Germany

*Corresponding author: rbirwadkar6@gmail.com

Received: January 27, 2026 • Accepted: April 6, 2026 • Published: May 14, 2026

Abstract: Semantic paraphrase generation plays a crucial role in academic and technical writing by enabling authors to restate content while preserving its original meaning. Traditional paraphrasing approaches, such as rule-based rewriting and statistical methods, often struggle to maintain semantic consistency and linguistic fluency, especially for complex or longer text segments. Recent advances in transformer-based architectures have significantly improved text generation capabilities by leveraging contextual representations and self-attention mechanisms. This paper presents a comparative study of pre-trained and fine-tuned transformer models for semantic paraphrase generation. The study builds upon prior work presented in [1], extending the analysis of transformer-based approaches for paraphrase generation. We evaluate encoder–decoder transformer architectures, with a primary focus on the BART model in both pre-trained and fine-tuned settings, alongside a large generative language model used for paraphrase generation. The fine-tuning process adapts pre-trained models to paraphrasing tasks using task-specific data, enabling improved control over semantic preservation and output consistency. The evaluation is conducted using both quantitative and qualitative analyses, including training and validation loss trends and comparative examination of generated paraphrases. Experimental results demonstrate that fine-tuned transformer models produce paraphrases with higher semantic fidelity and structural coherence compared to their pre-trained counterparts, while large generative models offer fluent but less deterministic outputs. The findings highlight the importance of task-specific fine-tuning for controlled and semantically accurate paraphrase generation. This study contributes practical insights into the selection and adaptation of transformer architectures for paraphrasing applications, particularly in academic and research-oriented writing contexts.

Keywords: semantic paraphrase generation; transformer models; BART; fine-tuning; natural language processing.

1. INTRODUCTION

Paraphrasing is a fundamental practice in academic and technical writing, enabling authors to express existing ideas in new linguistic forms while preserving the original semantic meaning. Effective paraphrase generation supports clarity, originality, and ethical content creation, particularly in research communication where similar concepts are often discussed across multiple works. However, producing high-quality paraphrases that maintain semantic fidelity and grammatical correctness remains a challenging task for automated systems. Early approaches to paraphrase generation relied on rule-based methods and statistical tech-

niques, such as synonym substitution and phrase-based machine translation. While these methods offered limited rewriting capabilities, they frequently failed to preserve contextual meaning, resulting in paraphrases that were either semantically distorted or linguistically unnatural. Neural network–based sequence-to-sequence models, including recurrent neural networks and long short-term memory architectures, improved fluency but faced scalability issues and difficulty in capturing long-range dependencies within text.

The introduction of transformer-based architectures has significantly advanced natural language generation tasks, including paraphrase generation. By employing self-attention mechanisms and parallelized processing, transformer models enable richer contextual understanding and improved text generation performance. Encoder–decoder architectures, in particular, have demonstrated strong capabilities in text rewriting tasks by learning to map input sentences to semantically equivalent outputs. Pre-trained transformer models further enhance this capability by leveraging large-scale language knowledge acquired during pre-training.

Despite these advancements, a key practical question remains: to what extent does task-specific fine-tuning improve the quality of paraphrase generation compared to using pre-trained models directly? While large pre-trained and generative models can produce fluent paraphrases, their outputs may lack consistency and control when applied to structured academic rewriting tasks. Fine-tuning transformer models on paraphrasing data offers a potential solution by adapting general language representations to task-specific objectives. This paper presents a comparative study of pre-trained and fine-tuned transformer models for semantic paraphrase generation. The study focuses on evaluating encoder–decoder transformer architectures, with particular emphasis on the BART model in both pre-trained and fine-tuned configurations, alongside a large generative language model used for paraphrasing. The evaluation combines quantitative analysis of training and validation behaviour with qualitative examination of generated paraphrases, emphasizing semantic preservation and structural coherence.

The contributions of this work are threefold:

1) This study provides an empirical comparison between pre-trained and fine-tuned transformer models for paraphrase generation,

2) It demonstrates the impact of task-specific fine-tuning on semantic consistency and output control, and

3) It offers practical insights for selecting transformer architectures in academic paraphrasing applications.

2. RELATED WORK

Paraphrase generation has been extensively studied within the field of natural language processing, evolving through multiple methodological stages ranging from rule-based rewriting to modern transformer-based architectures. The primary objective of paraphrase generation is to reformulate text while preserving its semantic meaning, a task that requires both linguistic fluency and contextual understanding. Early paraphrase generation techniques were predominantly rule-based and relied on manually crafted linguistic

rules, lexical resources, and synonym dictionaries. These approaches typically performed word- or phrase-level substitutions using predefined rules and thesauri. While rule-based systems were straightforward to implement, they lacked robustness and often produced grammatically incorrect or semantically altered outputs due to their inability to capture contextual dependencies and deeper semantic relationships [2]. As a result, such methods were limited in their applicability to real-world writing tasks.

To overcome these limitations, statistical approaches were introduced, most notably phrase-based statistical machine translation (SMT). In this paradigm, paraphrase generation was treated as a monolingual translation problem, where sentences were probabilistically mapped to alternative surface forms using aligned phrase pairs [3]. Statistical methods improved linguistic fluency and introduced data-driven learning; however, their effectiveness was highly dependent on the availability of high-quality parallel corpora. Moreover, SMT-based paraphrasing struggled with long sentences and complex syntactic structures, limiting its scalability and generalization. The emergence of neural network–based sequence-to-sequence models marked a significant advancement in paraphrase generation. Recurrent neural networks (RNNs) and long short-term memory (LSTM) architectures enabled models to learn continuous latent representations of sentences, allowing for more flexible paraphrase generation [4]. Attention mechanisms further enhanced these models by allowing dynamic alignment between input and output sequences, improving semantic coherence and grammatical consistency [5]. Despite these improvements, recurrent architectures suffered from inherent drawbacks, including limited parallelization, slow training times, and difficulty in modelling long-range dependencies.

Transformer-based architectures have since become the dominant framework for paraphrase generation and other natural language generation tasks. By replacing recurrence with self-attention mechanisms, transformers enable parallel processing and more effective modelling of global contextual relationships within text [6]. Encoder–decoder transformer architectures, in particular, are well suited for paraphrase generation, as they explicitly learn mappings between semantically equivalent sentence representations. These models have demonstrated strong performance in text rewriting tasks by capturing both syntactic structure and semantic meaning.

The introduction of large-scale pre-trained transformer models further advanced paraphrase generation by leveraging linguistic knowledge learned from massive corpora during pre-training [7]. Models such as BART employ denoising autoencoder objectives that make them particularly effective for text reconstruction and rewriting tasks [8]. Fine-tuning these pre-trained models on paraphrase-specific datasets allows them to adapt to controlled rewriting objectives, improving semantic fidelity and output stability. Recent research has also explored the use of large generative language models for paraphrase generation. While these models are capable of producing fluent and diverse paraphrases, their generative flexibility often results in less deterministic behaviour and occasional semantic drift. This characteristic can be advantageous for creative text generation but poses challenges for applications that require precise semantic preservation, such as academic writing.

Despite the substantial progress achieved through transformer-based models, limited work has systematically compared pre-trained and fine-tuned transformer architectures for semantic paraphrase generation under controlled evaluation settings. This study addresses this gap by providing a focused comparative analysis that emphasizes semantic

preservation, output consistency, and practical applicability in academic and research-oriented writing contexts. Recent work such as the T5 model [9] further unified text-to-text learning approaches, demonstrating strong performance across multiple natural language processing tasks, including paraphrase generation.

3. METHODOLOGY

This section describes the methodology adopted for semantic paraphrase generation using transformer-based architectures. The overall approach focuses on evaluating and comparing pre-trained and fine-tuned transformer models with respect to their ability to generate semantically consistent paraphrases while preserving grammatical structure, contextual meaning, and linguistic fluency.

3.1. Dataset Preparation and Preprocessing

The paraphrase generation task is formulated using paired text samples consisting of original sentences and their corresponding paraphrased versions. Each data instance is structured as an input–output pair, where the input represents the source sentence and the output represents a semantically equivalent paraphrase. This supervised formulation enables effective training and fine-tuning of encoder–decoder transformer models for controlled rewriting tasks. Prior to model training and evaluation, the textual data undergoes standard preprocessing steps to ensure consistency and compatibility with transformer-based architectures. These steps include text normalization, lowercasing, removal of unnecessary special characters, and sentence-level segmentation. The preprocessing pipeline is designed to preserve semantic content while eliminating noise that may negatively impact model learning and generation quality.

3.2. Transformer Architecture Overview

Transformer models are built upon self-attention mechanisms that enable the modelling of contextual relationships between all tokens in a sequence simultaneously. Unlike recurrent architectures, transformers process entire sequences in parallel, allowing efficient learning of long-range dependencies and improving scalability for large datasets [6]. The self-attention mechanism computes weighted interactions between tokens, enabling the model to capture both local and global contextual information.

In encoder–decoder transformer architectures, the encoder maps the input sentence into a sequence of contextualized representations, while the decoder generates the paraphrased output in an auto-regressive manner based on the encoded context. This architecture is particularly suitable for paraphrase generation, as it explicitly learns mappings between semantically equivalent textual representations rather than performing direct word-level substitutions. Figure 1 illustrates the transformer-based encoder–decoder architecture used for semantic paraphrase generation.

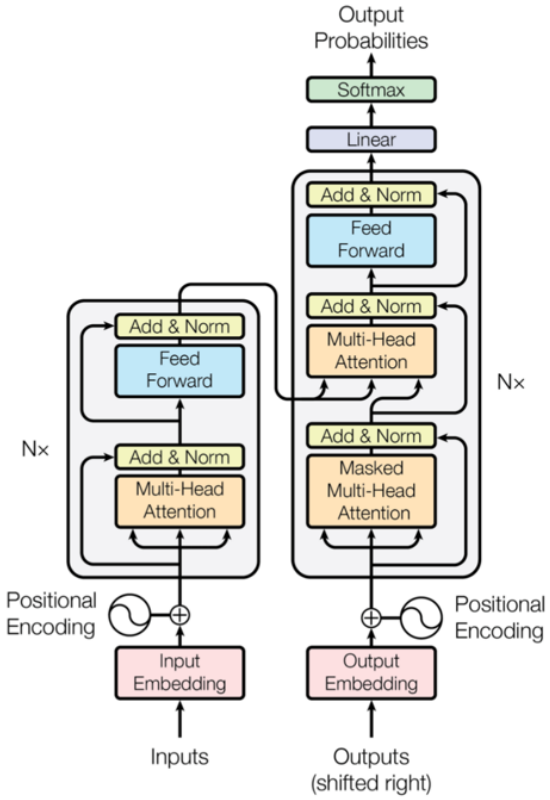


Figure 1. *Transformer-based encoder–decoder architecture used for semantic paraphrase generation. [6]*

3.3. Model Selection

To analyse the impact of pre-training and task-specific adaptation, multiple transformer-based models are considered in this study:

- **Pre-trained Encoder–Decoder Transformer Model:**A transformer model used directly for paraphrase generation without additional task-specific training. This configuration leverages general linguistic knowledge acquired during large-scale pre-training [7].
- **Fine-Tuned Encoder–Decoder Transformer Model:**The same pre-trained architecture further trained on paraphrase-specific data to improve semantic alignment, structural consistency, and output stability.
- **Large Generative Language Model:**Included as a comparative baseline to assess the trade-off between generative fluency and semantic control in paraphrase generation.

The primary focus is placed on the BART architecture due to its encoder–decoder design and its demonstrated effectiveness in text rewriting and generation tasks [8]. Figure 2 presents the BART encoder–decoder architecture employed in this study.

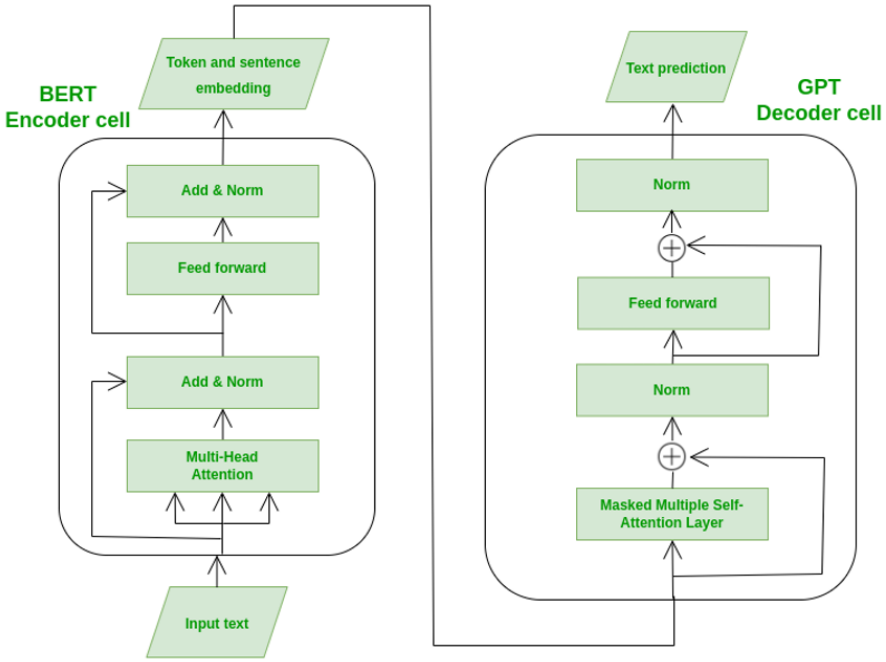


Figure 2. *BART encoder–decoder architecture employed for semantic paraphrase generation. [10]*

3.4. Fine-Tuning Strategy

Fine-tuning adapts a pre-trained transformer model to the paraphrase generation task by optimizing it on paired paraphrase data. During fine-tuning, the encoder processes the source sentence while the decoder learns to generate the corresponding paraphrased output. The training objective minimizes the discrepancy between the generated paraphrase and the reference paraphrase, allowing the model to internalize task-specific rewriting patterns. A low learning rate and a controlled number of training epochs are employed to prevent overfitting and preserve the general language representations learned during pre-training. This strategy enables the model to balance semantic preservation with lexical and syntactic variation, which is essential for producing high-quality paraphrases suitable for academic writing contexts. Fine-tuning is particularly important for encoder–decoder transformer models, as it aligns the model's generative behaviour with paraphrase-specific objectives rather than general text reconstruction or generation tasks.

3.5. Inference Process

During inference, the trained models generate paraphrases for previously unseen input sentences. Decoding strategies are configured to promote fluent and semantically consistent outputs while avoiding excessive repetition or unnecessary deviation from the original meaning. The generated paraphrases are then evaluated using both quantitative and qual-

itative criteria to assess semantic fidelity, grammatical correctness, and structural variation. This inference setup allows for a fair and consistent comparison between pre-trained, fine-tuned, and large generative models under identical generation conditions.

4. EXPERIMENTAL SETUP AND EVALUATION METRICS

This section describes the experimental configuration and evaluation strategy adopted to assess the performance of pre-trained and fine-tuned transformer models for semantic paraphrase generation. The experiments are designed to analyse both model learning behaviour during training and the qualitative characteristics of generated paraphrases, with particular emphasis on semantic preservation, grammatical correctness, and output consistency.

4.1. Experimental Setup

All experiments are conducted using transformer-based encoder–decoder architectures configured specifically for paraphrase generation tasks. The primary model evaluated in this study is the BART architecture, examined in both pre-trained and fine-tuned settings due to its suitability for text rewriting and sequence-to-sequence generation tasks [8]. In addition, a large generative language model is included as a comparative baseline to analyse differences in fluency, diversity, and controllability of generated paraphrases. For fine-tuning, the pre-trained transformer model is trained on paired paraphrase data using a supervised learning framework. The encoder receives the original sentence as input, while the decoder generates the corresponding paraphrased output. Training is performed for a fixed number of epochs to ensure sufficient convergence without excessive overfitting. A low learning rate is employed to preserve the linguistic representations acquired during pre-training and to enable stable optimization, a common practice in transfer learning for transformer-based models [7]. Batch size and maximum sequence length are selected based on computational feasibility and model constraints, ensuring a balance between training efficiency and representation capacity. During inference, identical decoding configurations are applied across all evaluated models to ensure fair comparison of generated paraphrases.

4.2. Training and Validation Monitoring

Model performance during training is monitored using both training and validation loss. Tracking loss trends provides insight into model convergence, stability, and generalization behaviour. A consistent reduction in training loss accompanied by stable validation loss indicates effective learning, whereas divergence between the two may signal overfitting. For fine-tuned models, validation loss serves as a key indicator of successful task adaptation. Comparing training and validation loss trajectories between pre-trained and fine-tuned configurations allows assessment of the impact of task-specific fine-tuning on learning dynamics. This analysis provides empirical evidence of how fine-tuning improves alignment between generated paraphrases and reference outputs. Monitoring loss behav-

iour is particularly important for paraphrase generation tasks, where semantic alignment is not always fully captured by surface-level lexical overlap metrics.

4.3. Evaluation Metrics

Given the focus on semantic paraphrase generation, evaluation is conducted using a combination of quantitative and qualitative measures. Rather than relying exclusively on automatic n-gram–based metrics, the evaluation strategy emphasizes learning behaviour and semantic consistency, which are more indicative of paraphrase quality in controlled rewriting tasks.

4.3.1. Quantitative Evaluation

Quantitative evaluation is primarily based on training and validation loss observed during model optimization. These metrics provide an objective measure of how effectively the model learns to map input sentences to semantically equivalent paraphrases. Lower validation loss is interpreted as improved semantic alignment and generalization capability. The use of loss-based evaluation is particularly suitable for encoder–decoder transformer models trained with sequence-level objectives, as it reflects the model's ability to generate outputs that closely match reference paraphrases in meaning and structure.

4.3.2. Qualitative Evaluation

Qualitative evaluation complements quantitative analysis by examining the generated paraphrases produced by different models. Generated outputs are assessed based on:

- Semantic fidelity to the original sentence
- Grammatical correctness and fluency
- Structural variation and lexical diversity

This qualitative assessment enables human-interpretable comparison of paraphrasing behaviour across pre-trained, fine-tuned, and large generative models. It also allows identification of semantic drift, unnecessary content alteration, or excessive variability, which are critical considerations for academic and technical writing applications.

5. RESULTS AND ANALYSIS

This section presents and analyses the experimental results obtained from evaluating pre-trained and fine-tuned transformer models for semantic paraphrase generation. The analysis focuses on model learning behaviour during training and the qualitative characteristics of generated paraphrases, with particular attention to semantic preservation, output consistency, and linguistic fluency.

5.1. Training and Validation Performance

The training and validation loss trends provide insight into how effectively the evaluated models learn the paraphrase generation task. For the pre-trained transformer model used without task-specific adaptation, the reduction in training loss is limited, and validation loss shows comparatively weaker convergence. This behaviour indicates that although the pre-trained model possesses strong general language generation capabilities, it is not optimally aligned with the controlled paraphrasing objective. In contrast, the fine-tuned transformer model demonstrates a more consistent and stable decrease in both training and validation loss across epochs. The convergence pattern suggests that task-specific fine-tuning enables the model to better internalize paraphrase-oriented rewriting patterns, resulting in improved alignment between generated outputs and reference paraphrases. The relatively stable validation loss further indicates enhanced generalization and reduced overfitting, confirming the effectiveness of fine-tuning for this task.

Table 1. *Training and validation loss for the pre-trained BART model.*

Metric	Value	Metric	Value
Mean Similarity Score	0.5685	Standard Deviation of Similarity Scores	0.1059
Mean BLEU Score	0.1522	Standard Deviation	0.0380
Mean ROUGE-1 Score	0.2965	Standard Deviation	0.0470
Mean ROUGE-2 Score	0.2714	Standard Deviation	0.0476
Mean ROUGE-L Score	0.2965	Standard Deviation	0.0470

Table 2. *Training and validation loss for the fine-tuned BART model.*

Metric	Value	Metric	Value
Mean Similarity Score	0.8558	Standard Deviation of Similarity Scores	0.1291
Mean BLEU Score	0.4098	Standard Deviation	0.2258
Mean ROUGE-1 Score	0.8972	Standard Deviation	0.0903
Mean ROUGE-2 Score	0.7482	Standard Deviation	0.1999
Mean ROUGE-L Score	0.8349	Standard Deviation	0.1565

Comparative analysis of loss trajectories between the pre-trained and fine-tuned configurations highlights the impact of task adaptation on learning dynamics. Fine-tuning allows the model to shift from general-purpose text generation toward more controlled semantic rewriting, which is essential for producing reliable paraphrases in structured writing contexts.

Table 1 presents the quantitative performance metrics for the pre-trained BART model, while Table 2 summarizes the corresponding results for the fine-tuned model. The fine-tuned model consistently outperforms the pre-trained model across all metrics, including similarity, BLEU, and ROUGE scores. This improvement demonstrates the effectiveness of

task-specific fine-tuning in enhancing semantic preservation and structural consistency in paraphrase generation.

5.2. Qualitative Analysis of Generated Paraphrases

Qualitative evaluation reveals clear differences in paraphrasing behaviour across the evaluated models. The pre-trained transformer model is capable of producing fluent paraphrases with noticeable lexical variation; however, its outputs occasionally exhibit semantic drift. In such cases, the generated paraphrase partially alters the original meaning or introduces additional contextual information that was not present in the source sentence. While these variations may be acceptable in creative rewriting scenarios, they reduce suitability for academic or technical writing tasks where semantic precision is critical.

The fine-tuned transformer model produces paraphrases that more consistently preserve the semantic intent of the original input while maintaining grammatical correctness and appropriate structural variation. Sentence restructuring is more controlled, and the generated outputs remain closer to the source meaning without unnecessary elaboration. This behaviour reflects the benefits of task-specific fine-tuning, which guides the model toward producing paraphrases that balance linguistic diversity with semantic fidelity.

The large generative language model included for comparison generates paraphrases that are generally fluent and stylistically diverse. These outputs often demonstrate strong natural language fluency; however, they exhibit higher variability and less deterministic behaviour. In some instances, this flexibility leads to paraphrases that deviate from the original meaning or introduce stylistic changes that may not be desirable for structured rewriting tasks. While such models are valuable for creative text generation, their reduced semantic control limits their reliability for academic paraphrasing applications.

Table 3 presents representative paraphrases generated by the evaluated model, highlighting how semantic meaning is preserved while introducing controlled lexical and structural variation. The examples demonstrate that the generated paraphrases maintain the original intent while applying syntactic restructuring and synonym substitution, confirming the model's ability to produce semantically consistent and fluent outputs.

Table 3. *Example paraphrases generated by a large generative language model.*

Original Sentence	DeepSeek-R1 Paraphrase
"AI is transforming healthcare services."	"Healthcare is being revolutionized by artificial intelligence."
"The economy is facing inflation challenges."	"Rising inflation is impacting the economy."

5.3 Comparative Summary

The comparative analysis of results highlights three key observations:

- **Pre-trained transformer models** provide a strong baseline for paraphrase generation in terms of fluency but lack sufficient control over semantic consistency when used without task-specific adaptation.

- **Fine-tuned transformer models** achieve superior semantic fidelity, improved output stability, and more consistent paraphrasing behaviour, making them better suited for controlled paraphrase generation in academic and technical writing contexts.
- **Large generative language models** offer enhanced linguistic diversity and expressive flexibility but exhibit less predictable semantic behaviour, limiting their applicability for tasks that require precise meaning preservation.

Overall, the results demonstrate that task-specific fine-tuning plays a critical role in improving the quality and reliability of semantic paraphrase generation using transformer architectures. These findings reinforce the importance of model adaptation when deploying paraphrase generation systems in contexts where semantic accuracy and consistency are essential.

6. DISCUSSION

The results presented in this study provide important insights into the effectiveness of transformer-based architectures for semantic paraphrase generation, particularly with respect to the role of task-specific fine-tuning. The comparative analysis between pre-trained and fine-tuned models highlights that while modern transformer models possess strong general language generation capabilities, their performance in controlled paraphrasing tasks depends heavily on alignment with task-specific objectives. One of the key observations is that pre-trained transformer models, when used without additional adaptation, tend to prioritize fluency and surface-level variation over precise semantic preservation. Although such models generate grammatically correct and diverse paraphrases, the absence of paraphrase-specific fine-tuning can lead to semantic drift, especially in structured or information-dense sentences. This behaviour reflects the general-purpose nature of pre-trained models, which are optimized for broad language understanding rather than controlled rewriting.

In contrast, fine-tuned transformer models demonstrate improved semantic consistency and output stability. Fine-tuning enables the model to internalize rewriting patterns that emphasize meaning preservation while still allowing appropriate lexical and syntactic variation. This finding underscores the importance of transfer learning strategies that adapt pre-trained representations to downstream paraphrasing tasks. The improved convergence behaviour and stable validation performance observed during training further support the conclusion that fine-tuning enhances generalization for paraphrase generation. The comparison with a large generative language model reveals an important trade-off between generative flexibility and semantic control. Large generative models produce fluent and expressive paraphrases, often exhibiting greater stylistic diversity than encoder–decoder transformers. However, this flexibility comes at the cost of reduced determinism, which can result in paraphrases that deviate from the original meaning. For applications such as academic and technical writing, where semantic accuracy is critical, this unpredictability limits the practical usefulness of purely generative approaches. Another significant implication of the findings is the limitation of relying solely on surface-level lexical variation as an indicator of paraphrasing quality. The qualitative analysis demonstrates that paraphrases with substantial structural or lexical differences may still fail to preserve

semantic intent. This reinforces the need for evaluation strategies that prioritize semantic fidelity over simple n-gram overlap, particularly for controlled rewriting tasks.

Overall, the discussion highlights that fine-tuned encoder–decoder transformer models offer a balanced solution for semantic paraphrase generation by combining fluency with reliable meaning preservation. These characteristics make such models particularly well suited for academic and research-oriented writing contexts, where controlled paraphrasing is essential for maintaining clarity, accuracy, and ethical standards.

7. CONCLUSION AND FUTURE WORK

This paper presented a comparative study of pre-trained and fine-tuned transformer models for semantic paraphrase generation, with a focus on evaluating their ability to preserve meaning while producing fluent and structurally varied paraphrases. By examining encoder–decoder transformer architectures under different training configurations, the study investigated the impact of task-specific adaptation on paraphrase quality and output consistency. The experimental results demonstrate that pre-trained transformer models provide a strong baseline for paraphrase generation in terms of linguistic fluency but are not optimally aligned with controlled rewriting objectives when used without further adaptation. In contrast, fine-tuned transformer models exhibit improved semantic fidelity, greater output stability, and more consistent paraphrasing behaviour. These improvements highlight the importance of fine-tuning in aligning general-purpose language models with task-specific paraphrasing requirements, particularly in academic and technical writing contexts where semantic accuracy is critical.

The comparison with a large generative language model further emphasizes the trade-off between generative flexibility and semantic control. While large generative models are capable of producing expressive and diverse paraphrases, their reduced determinism can lead to semantic drift, limiting their reliability for structured paraphrasing applications. This observation reinforces the need to carefully select and adapt models based on the intended use case.

Overall, the findings of this study indicate that fine-tuned encoder–decoder transformer models offer a practical and effective solution for semantic paraphrase generation when accuracy, consistency, and control are prioritized. The results provide valuable insights for the design and deployment of paraphrasing systems intended to assist academic authors and researchers in ethical and reliable text rewriting.

Future work may explore extending this approach to multilingual paraphrase generation, enabling broader applicability across different languages and writing contexts. Additional research could investigate the integration of semantic similarity metrics for automatic evaluation of paraphrase quality, as well as domain-specific fine-tuning strategies to further enhance performance in specialized technical and scientific fields. Furthermore, hybrid frameworks that combine the controlled behaviour of fine-tuned models with the expressive capabilities of large generative models represent a promising direction for advancing semantic paraphrase generation.

FUNDING:
This research received no external funding.

INSTITUTIONAL REVIEW BOARD STATEMENT:
Not applicable.

INFORMED CONSENT STATEMENT:
Not applicable.

CONFLICTS OF INTEREST:
The author declares no conflict of interest.

REFERENCES

[1] R. V. Birwadkar, "Plagiarism Detection and Paraphrasing based on Generative Artificial Intelligence," Master's thesis, Dept. of Information and Technology, SRH Hochschule Heidelberg, Germany, 2025.

[2] I. Androutsopoulos and P. Malakasiotis, "A Survey of Paraphrasing and Textual Entailment Methods," Journal of Artificial Intelligence Research, vol. 38, pp. 135–187, 2010.

[3] C. Quirk, C. Brockett, and W. Dolan, "Monolingual Machine Translation for Paraphrase Generation," in Proc. 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain, 2004, pp. 142–149.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 2014, pp. 3104–3112.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in Proc. 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015.

[6] A. Vaswani et al., "Attention Is All You Need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 5998–6008.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Minneapolis, MN, USA, 2019, pp. 4171–4186.

[8] M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 2020, pp. 7871–7880.

[9] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.

[10] GeeksforGeeks. (n.d.). Website Summarizer using BART. Retrieved from https://www.geeksforgeeks.org/website-summarizer-using-bart/

Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2025, Vol. 4, Issue 2, pp. 30–42
https://doi.org/10.5937/jcfs4-64411
Original research paper

Comparative Analysis of Clustering Textual and Numerical Data Using the K-Means Algorithm

Sanja Raičević[1*]

[1] Ministry of Interior of the Republic of Serbia

*Corresponding author: sanjaraicevic09@gmail.com

Received: January 27, 2026 • Accepted: April 25, 2026 • Published: May 14, 2026

Abstract: This paper presents a comparative analysis of the application of the K-Means clustering algorithm on two different types of data – textual and numerical. The aim of the research was to examine the reliability, stability, and interpretability of the results when the same algorithm is applied to semantically diverse datasets. The textual data were taken from the articles of the Criminal Code of the Republic of Serbia, where clustering was performed after preprocessing and TF-IDF vectorization. The numerical data refer to traffic accident statistics from 2015 to 2021, analyzing parameters such as the number of property-damage-only accidents, the number of injured persons, and the number of fatalities.

The results showed that clustering on textual data produced a relatively clear separation of thematic groups of articles, but with a moderate silhouette coefficient value due to a high degree of semantic similarity among documents. On the other hand, clustering on numerical data demonstrated a more stable structure, where the optimal number of clusters was two, indicating the possibility of distinguishing periods with different intensity and severity of traffic accidents.

It was concluded that the K-Means algorithm provides more reliable and interpretable results for numerical data, while in the case of textual data, it requires more precise vector space modeling and possibly the application of semantic models such as Word2Vec or BERT. The paper serves as a basis for further research in the field of integrating machine learning techniques for analyzing heterogeneous data sources.

Keywords: clustering, K-Means, textual data, numerical data, TF-IDF, PCA, silhouette analysis.

1. INTRODUCTION

The modern era is characterized by an exponential growth in the volume of data generated daily across various domains – from textual documents and administrative records to numerical statistical reports and sensor data. In such an environment, the ability to efficiently discover structures and patterns within data has become one of the key tasks of modern analytics and machine learning. One of the fundamental approaches in this context is *clustering*, which refers to the grouping of data into clusters based on their internal similarity.

Among numerous clustering algorithms, K-Means occupies a central position due to its simplicity, speed, and broad applicability. It enables the identification of groups of objects that share similar characteristics, without prior knowledge of the underlying data structures. However, the effectiveness of the K-Means algorithm largely depends on the nature

of the processed data, their distribution, and the applied preprocessing methods. For this reason, it is important to examine how this algorithm performs when applied to different types of data.

The subject of this research is a comparative analysis of clustering results for textual and numerical data using the same algorithm – K-Means. The textual data were taken from the corpus of articles of the Criminal Code of the Republic of Serbia, while the numerical data were based on official traffic accident statistics from 2015 to 2021. This combination of two semantically distinct datasets allows for a deeper analysis of the K-Means algorithm's ability to recognize internal structures within data of different natures.

The aim of the paper is to determine the extent to which the K-Means algorithm produces accurate and interpretable results in both cases, as well as to highlight its limitations and potential applications in various analytical contexts. The paper presents the steps of data preparation, clustering implementation, visualization of results, and interpretation of the obtained clusters.

2. RELATED WORK

Research on data clustering has a long history and represents one of the key fields in data analysis and machine learning. The foundations of this approach were established by MacQueen [1], who defined the K-Means algorithm as a method for grouping multivariate data based on a measure of similarity.

A review of existing studies shows that clustering is a dynamic field that unifies classical statistical methods with modern deep learning approaches. In this context, the present paper builds upon previous studies that applied K-Means and PCA techniques to the analysis of traffic accident data [2], with the possibility of future expansion through the application of advanced vector models and hybrid clustering methods.

The classification and analysis of textual data pose a particular challenge within this domain, as the text must be transformed into a numerical form suitable for further processing. Havrlant and Kreinovich [3] provided a formal explanation of the TF-IDF heuristic, one of the most commonly used approaches for converting textual content into vector representations, while Wu and Xu [4] emphasized the importance of various machine learning techniques for text classification. The TF-IDF method often represents the initial step in the process of clustering textual documents, which was also the case in this study.

Verification and evaluation of the quality of the formed clusters represent an essential aspect of any research in this area. Rousseeuw [5] proposed the Silhouette Score metric as both a visual and quantitative means for assessing the trade-off between cluster compactness and separation, which later became a standard criterion for evaluating clustering results.

The application of Principal Component Analysis (PCA) has a long tradition in data analysis and dimensionality reduction. Recent studies on PCA have demonstrated its effectiveness as a natural method for visual and statistical data exploration, such as the work by Gewers et al. [6], while Dunteman [7] and Jolliffe [8] systematized the mathematical foundations of the PCA approach. In the context of these works, PCA was applied for graphical

representation of relationships among parameters affecting the number and severity of traffic accidents, thereby enabling a better understanding of variability within the dataset.

Manning, Raghavan, and Schütze [9], in their seminal work *Introduction to Information Retrieval*, defined the theoretical and practical aspects of information processing, text classification, and retrieval, establishing the framework within which modern machine learning algorithms and text analysis methods have evolved. Subsequent research inspired by MacQueen, such as the work of Xu and Tian [10], expanded this concept through a comprehensive overview of contemporary clustering methods, highlighting their advantages and limitations depending on data structure and dimensionality.

The processing of textual data has been further improved through the application of word vectorization methods based on deep learning. Mikolov et al. [11] developed the Word2Vec model, which enables the representation of words in vector space while preserving semantic proximity, whereas Pennington, Socher, and Manning [12] proposed the GloVe model as a more advanced variant that combines local and global statistical information. These models now form the foundation of modern natural language processing techniques and have the potential to significantly enhance clustering results compared to the traditional TF-IDF approach.

3. THEORETICAL FRAMEWORK AND RESEARCH METHODOLOGY

3.1. Theoretical Framework

Clustering methods represent one of the fundamental approaches within unsupervised machine learning, with the goal of identifying natural groups of objects based on their similarities or differences. Unlike supervised learning methods, where categories are predefined, clustering operates without prior knowledge of data structures, making it particularly useful for uncovering hidden patterns.

One of the most commonly used clustering algorithms is K-Means, proposed by James MacQueen (1967) [1]. This algorithm seeks to minimize the total distance of data points from their respective cluster centroids, thereby producing groups that are as homogeneous as possible. The basic principle of operation is based on defining the number of clusters K, selecting initial centroids, assigning each object to the nearest centroid, and iteratively updating centroid positions until a stable cluster structure is achieved.

The K-Means algorithm can be applied to different types of data – numerical data, where values have directly measurable relationships, and textual data, where prior vectorization is necessary to represent textual content in numerical form. For textual data, one of the most important techniques is TF-IDF (Term Frequency – Inverse Document Frequency), which quantifies the significance of words relative to their occurrence in the entire corpus. The resulting values form vectors that serve as input data for the clustering algorithm.

To reduce data complexity and improve visualization, the Principal Component Analysis (PCA) technique is often applied. PCA transforms the original set of correlated variables into a smaller number of uncorrelated principal components, retaining most of the total data variance. This approach enables better interpretation of results and clearer visualization in two- or three-dimensional space [2].

The application of clustering to the analysis of textual and numerical data allows researchers to gain deeper insights into data structures and underlying patterns. While clusters in textual datasets are formed based on semantic similarity, in numerical data they are created based on quantitative and statistical relationships among variables. This research aims to demonstrate the differences in processing methods, result quality, and cluster interpretability between these two data types using the K-Means algorithm [3].

3.2. Research Methodology

The methodological approach in this paper is based on the experimental application of the K-Means algorithm to two datasets of different natures:

1) Textual data – articles of the Criminal Code in *.txt* format,

2) Numerical data – datasets containing numerical values related to the occurrence and consequences of traffic accidents in *.csv* format.

3.2.1. Data Preparation and Preprocessing

For the textual corpus, several preprocessing stages were carried out:

– text cleaning (removal of punctuation, numbers, and stop words),

– tokenization and transformation of words into numerical vectors using TF-IDF vectorization,

– dimensionality reduction using PCA to improve interpretability and optimize computation,

data normalization to ensure that all values are within the same scale [4].

For the numerical data, standard preparation phases were conducted, including:

– identification and removal of missing values,

– data scaling and normalization,

– selection of relevant variables representing the intensity, frequency, and consequences of the observed phenomena.

3.2.2. Application of the Algorithm and Determination of the Optimal Number of Clusters

The K-Means algorithm was applied to both datasets, with systematic testing of different values of the parameter *K*. To determine the optimal number of clusters, two standard methods were used:

– The **Elbow method**, which observes the relationship between inertia and the number of clusters,

The **Silhouette Score**, which measures the quality of clustering by considering both the distance between clustered objects and the compactness within each group [5].

3.2.3. Visualization and Analysis of Results

The clustering results were visualized in a two-dimensional space using PCA, which enabled a clearer representation of the boundaries between the formed clusters. In the case of textual data, the visualization reveals semantic groupings of legal articles, while the numerical data display trends in the frequency and severity of occurrences across different time intervals [6].

3.2.4. Objective and Expected Contribution

The main objective of the methodology is to determine the extent to which the type of data (textual or numerical) influences the clustering results, as well as to evaluate the effectiveness of the K-Means algorithm in different contexts. The obtained results may contribute to the development of systematic approaches for data analysis in legal, social, and technical domains, where it is necessary to identify natural groupings and patterns without predefined labels.

4. ANALYSIS OF RESULTS

The research was conducted on two datasets of different nature – textual and numerical types. The goal was to apply the same clustering algorithm (K-Means) and compare the obtained results in order to observe differences in the way this algorithm groups data of different structures.

The textual data consisted of segments of articles from the Criminal Code of the Republic of Serbia. Each article was treated as a separate document. Before clustering, a preprocessing procedure was carried out, which included the following steps:

– removal of punctuation marks and stop words,

– conversion of text to lowercase for consistency,

– lemmatization to reduce words to their base form,

– transformation into a numerical format using TF-IDF vectorization.

The resulting TF-IDF weight matrix for the textual data served as the basis for applying the K-Means algorithm. The value of the K parameter was selected experimentally, based on the Elbow method, which determines the number of clusters at which the internal inertia stabilizes. After that, PCA analysis was performed to reduce dimensionality to two components, enabling visualization of the results. The results showed that the legal articles were grouped according to thematic similarities, meaning that sections of the law related to similar areas (e.g., criminal acts against life and body, property-related crimes, etc.) appeared within the same clusters. This confirms the applicability of the K-Means algorithm for semantic clustering of textual content.

The second part of the research was based on the analysis of numerical data related to traffic accidents. The dataset contained information on the number of accidents, fatalities, and injuries per year, as well as other relevant parameters.

Before applying the K-Means algorithm, data standardization was performed using the StandardScaler technique to ensure that all attributes had an equal impact on cluster formation. Then, the data were transformed into a two-dimensional space using PCA analysis, which allowed for visual representation of the clusters and easier interpretation of the results.

The clustering results showed that the data were grouped according to years and the severity of traffic accident consequences. It was observed that more recent years exhibited higher concentrations within clusters characterized by greater accident intensity and a higher number of severe outcomes, which may indicate changes in traffic intensity, infrastructure, or participants' behavior.

On the PCA diagram, the first component (PCA1) represents the main direction of variance in the data and includes parameters with the greatest influence on the number and severity of accidents, while the second component (PCA2) represents the remaining variance related to secondary characteristics such as the number of injured persons or local conditions. The visual analysis clearly shows a separation between periods with fewer and more accidents, allowing for a better understanding of temporal trends. [7]

4.1. Comparative Analysis of Clustering Results for Textual and Numerical Data

By comparing the clustering results of textual data from the Criminal Code and numerical data on traffic accidents, both common methodological features and differences arising from the nature of the data can be observed.

In both cases, the K-Means algorithm was applied, along with the Elbow method and Silhouette Score, to determine the optimal number of clusters. For the textual data, the optimal number of clusters was three, whereas for the numerical data, the best results were achieved with two clusters. This difference arises from the complexity of textual documents, which contain multiple semantic nuances, compared to the simpler and more directly measurable structure of numerical data.

From an interpretative perspective, clustering the textual data enabled grouping legal articles according to thematic similarity, allowing for the automatic identification of areas related, for example, to criminal acts against life, property, or public order. Clustering the numerical data, on the other hand, resulted in the identification of periods with higher or lower frequency of traffic accidents, which has direct applications in monitoring safety trends and planning preventive measures.

Both analyses highlighted the importance of applying PCA for visualizing results. Dimensionality reduction allowed for clear graphical representation of clusters in 2D space and improved understanding of the relationships between the observed objects. While clusters in textual data were more dispersed and scattered, numerical data formed more compact and precisely defined groups [8].

It can be concluded that clustering represents a universal analysis technique, applicable to both unstructured (textual) and structured (numerical) data. However, the success of the analysis largely depends on preprocessing, choice of metrics, and understanding of the data context. By combining these approaches, it is possible to build integrated systems that

simultaneously analyze legal regulations and statistical data, thereby enabling a deeper understanding of the relationship between the legislative framework and the social reality described by the data.

4.2. Quantitative Analysis and Model Stability

The results presented in Table 1 show that the K-Means algorithm demonstrates greater stability with numerical data than with textual data. For textual data, there is a higher sensitivity to the choice of parameters (number of clusters, number of components in PCA, method of vectorization), which leads to fluctuating results. In contrast, with numerical data, the algorithm consistently identifies stable boundaries between groups, allowing for more reliable interpretation.

Table 1. Quantitative analysis of clustering results.

Data Type	Optimal Number of Clusters (K)	Silhouette Score	Visual Separation	Model S tability
Textual Data	Difficult to determine (gradual decline)	0.02 – 0.045	Weak, clusters overlap	Low
Numeri-cal Data	K = 2	≈ 0.557	Clearly separated clusters	Medium–High

The main difference lies in the nature of the data itself. Numerical attributes are quantitatively defined and suitable for the distance metrics used by K-Means, whereas textual data represent complex semantic structures that cannot be easily projected into Euclidean space without loss of meaning. Even after TF-IDF transformation and PCA reduction, textual data retain semantic ambiguity. Similarity between legal articles often depends on context rather than just word frequency, leading to less clear boundaries between classes [9]. On the other hand, numerical data (such as the number of accidents, injuries, and fatalities) have a clear statistical structure, allowing the algorithm to effectively detect natural groups and identify patterns, such as differences between days with increased and decreased numbers of traffic incidents.

Table 2 presents the clustering results for the two datasets—textual and numerical—using the K-Means algorithm. The display is divided into three segments for each data type:

1) Visualization of clusters after dimensionality reduction using PCA (Principal Component Analysis),

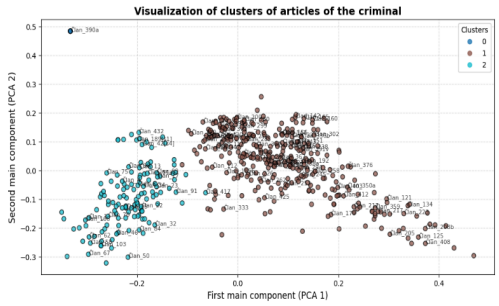
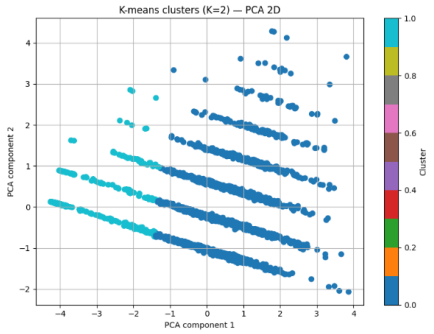
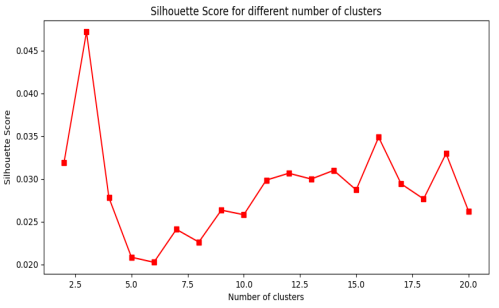
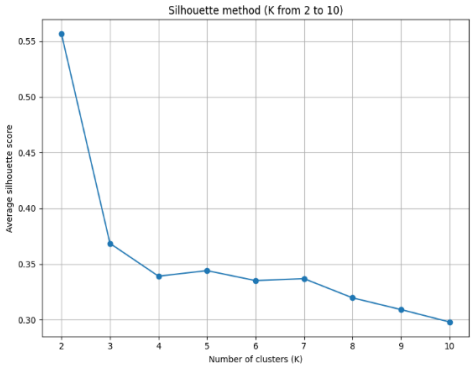
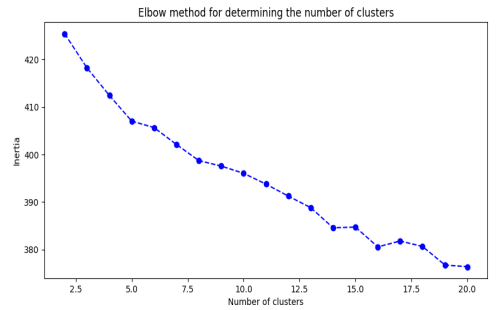
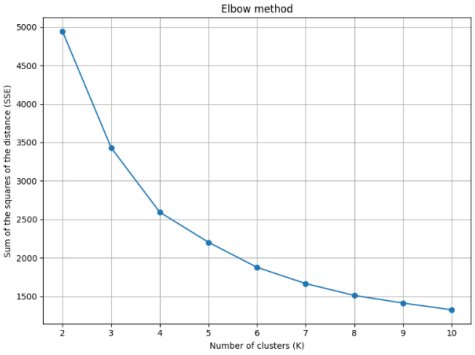
2) Analysis of clustering quality via the Silhouette Score metric,

3) Determination of the optimal number of clusters using the Elbow method.

These three levels of analysis together provide a comprehensive view of the data structure and algorithm performance in different contexts.

4.3. Visual Analysis of the Obtained Results

Table 2. *Comparative analysis of visual representations of clustering.*

TEXTUAL DATA	NUMERICAL DATA
	
	
	

In the left column of Table 2, the visual clustering results for the textual data are shown. The first image reveals that the points, representing individual legal articles, are distributed across multiple areas without clear boundaries between them. The feature space, obtained from the transformation of TF-IDF vectors, shows a high degree of overlap among terms, which is typical for legal terminology where the same words appear in different contexts. Silhouette analysis indicated an average value of only 0.035, pointing to weak internal cohesion of clusters and significant overlap between them. The graphical representation of Silhouette results confirms values in the range of 0.02–0.045, without pronounced local maxima. The Elbow method did not show a clear "elbow point" but a gradual decline

of inertia from 1200 to 850 as the number of clusters increased from K=2 to K=10. This suggests that there is no natural cluster structure in the data and that the data are semantically continuous. These results indicate that the K-Means algorithm, which assumes linear separation, is not suitable for clustering legal texts without additional semantic modeling (e.g., using Word2Vec, BERT, or topic clustering via LDA).

The right column of Table 2 shows the clustering results for the numerical dataset related to traffic accidents (attributes such as number of vehicles, road type, severity of consequences, and weather conditions). For this dataset, the PCA visualization clearly shows the formation of two compact and separated groups, which is the first indication of natural segmentation. Silhouette analysis showed a maximum value of 0.557 for K=2, which is an excellent result in clustering. Values then sharply decrease for larger K, further confirming the existence of two natural groups in the data. This is consistent with the Elbow method, which shows a characteristic "break" at K=2, where inertia sharply drops from 1040 to 650 and then stabilizes.

These results clearly demonstrate that the K-Means algorithm is highly effective in this case, as the data have clearly defined numerical relationships and low dimensionality, enabling precise separation based on Euclidean distance.

4.3. Concluding Observations of the Results

The obtained results provide the following key insights:

1) **Textual Data:** Low Silhouette metric values and the absence of a clear "elbow" confirm that K-Means fails to detect semantic similarities in legal texts. This suggests the need for more advanced techniques (e.g., BERT embeddings or topic-based clustering).

2) **Numerical Data:** High Silhouette scores and the stable drop in inertia at K=2 indicate that K-Means can be very effective in clustering well-structured data.

3) **Methodological Conclusion:** The nature of the data (textual or numerical) determines the applicability of the algorithm. Different types of data require tailored preprocessing and model selection approaches.

Overall, the presented results and analysis show that the K-Means algorithm can deliver high-quality outcomes only when input data are well-defined and linearly separable. In contrast, textual and semantically rich datasets require methodological extensions and deeper context-aware models.

5. DISCUSSION OF CLUSTERING METHODS

The analysis and clustering results clearly show that each algorithm provides a different perspective on the structure of the Criminal Code text. K-Means demonstrated its strength in quickly segmenting large datasets and clearly dividing them into thematic groups, but careful selection of the number of clusters is required – small variations in the n_clusters parameter can significantly alter the results. DBSCAN revealed natural "groups" of documents and identified several articles that do not fit into standard themes, indicat-

ing potential anomalies or articles with specific subject matter. Mean-Shift confirmed the existence of density centers in the texts, but it was slower on larger sets of TF-IDF vectors, suggesting it is better suited for detailed analysis of smaller subsets. Hierarchical clustering proved to be the most visually informative, allowing the tracking of how Criminal Code articles group together at different levels, which is useful for legal analysis and the creation of thematic maps. [10]

From the code and experiments, a practical insight can also be drawn: the key to good results lies in high-quality preprocessing and TF-IDF transformation. Without this step, even the most sophisticated algorithms could not differentiate between semantically distinct but superficially similar articles. Additionally, combining the results of multiple methods allows for the identification of both "major" themes (K-Means, hierarchical) and "unusual" or anomalous groups (DBSCAN, Mean-Shift), which has practical applications in legal analysis: it makes it easier to spot articles that deviate or may be relevant to specific cases.

This insight shows that, in practice, the choice of algorithm should not be rigid – it should be adapted to the type of data, the goal of the analysis, and the need to identify both core groups and unusual exceptions within the text.

6. RECOMMENDATIONS FOR FUTURE RESEARCH AND DIRECTIONS FOR IMPROVEMENT

The comparative analysis shows that:

– The K-Means algorithm has a clear advantage with numerical, well-structured data, providing stable and interpretable results.

– For textual data, the traditional TF-IDF + PCA approach provides only limited insight, and the results are not sufficiently stable, indicating the need to integrate semantic and deep learning models.

– Future research is recommended to experiment with different text vectorization models, as well as to combine textual and numerical data for multimodal analysis, in order to gain deeper insight into complex social or legal phenomena.

Although the study confirmed that K-Means can successfully process both types of data, the results indicate significant potential for improvement. Key directions for enhancing future research can be summarized in several areas, as outlined in Table 3:

1) Application of Advanced Techniques for Textual Data

   – Instead of the classical TF-IDF representation, it is preferable to apply deep learning models for natural language processing, such as Word2Vec, GloVe, or modern BERT models; [11]

   – These models allow the creation of vectors containing richer semantic information, which can result in clearer and more meaningful clusters, as well as better metric values, such as the Silhouette Score.

2) Combining Different Clustering Algorithms

– In addition to the K-Means algorithm, which requires a predefined number of clusters and assumes spherical group shapes, it is recommended to explore alternatives such as DBSCAN and Hierarchical Clustering;

– These algorithms often provide better results for unstructured and semantically rich data, as they can identify irregular and overlapping structures within the dataset.

3) Expansion of Datasets and Dimensionality Reduction

– For numerical data, it is recommended to include additional attributes such as weather conditions, road type, time of day, driver age, and other contextual factors. This would enable better understanding of causal relationships between variables and increase cluster accuracy.

– For textual data, expanding the corpus to include a larger number of legal articles or other normative documents can lead to more stable and statistically reliable results.

– The use of dimensionality reduction techniques such as PCA, t-SNE, or UMAP can improve visualization and the identification of patterns in high-dimensional data. [12]

Table 3. *Recommendations for improving clustering results.*

Data Type	Current Result	Recommendations for Improvement	Expected Effect
Numerical (traffic accidents)	Higher Silhouette Score (≈ 0.35–0.56), clearly formed clusters	- Expand the dataset (weather conditions, road type, time of day, driver age) - Apply DBSCAN and Hierarchical Clustering - Remove "noisy" data	More precise and richer clusters, better understanding of risk factors
Textual (legal articles)	Low Silhouette Score (≈ 0.02–0.045), overlapping clusters	- Use NLP models (Word2Vec, GloVe, BERT) - Combine multiple clustering methods - Apply dimensionality reduction (PCA, t-SNE, UMAP)	Clearer semantic relationships and more structured classification of legal concepts

7. CONCLUSION

This study conducted a comparative analysis of the application of the K-Means clustering algorithm on textual and numerical data. The results demonstrated that the nature of the data significantly affects the quality and stability of the clustering.

For textual data, clustering enabled thematic grouping of the articles of the Criminal Code, but with low Silhouette scores (0.02–0.045) and noticeable cluster overlap. This indicates the need for more precise text vector modeling and potential application of semantic models to achieve a clearer cluster structure. On the other hand, numerical data on traffic accidents exhibited stable and interpretable clustering, with an optimal number of clusters K=2 and a good Silhouette score (≈0.557). These results allow practical applications in trend monitoring, planning preventive measures, and analyzing traffic safety.

Based on the comparative analysis, it can be concluded that the K-Means algorithm provides more reliable and easily interpretable results for numerical data, whereas textual data require additional preprocessing techniques and advanced vectorization methods. Combining both approaches offers potential for integrated analyses, encompassing both the legislative framework and statistical indicators of social reality.

This work provides a foundation for further research on the application of machine learning algorithms to heterogeneous datasets and highlights the importance of method selection depending on the nature of the data and the objectives of the analysis.

FUNDING:
This research received no external funding.

INSTITUTIONAL REVIEW BOARD STATEMENT:
Not applicable.

INFORMED CONSENT STATEMENT:
Not applicable.

CONFLICTS OF INTEREST:
The author declares no conflict of interest.

REFERENCES

1] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* vol. 1, pp. 281–297, 1967.

[2] X. &. Z. H. Cheng, "Clustering Analysis for High-Dimensional Data Using K-Means and PCA," *Data Science and Engineering,* vol. 5, pp. 107–118, 2020.

[3] L. K. V. Havrlant, "A Simple Probabilistic Explanation of Term Frequency-Inverse Document Frequency (TF-IDF) Heuristic," *International Journal of General Systems,* vol. 46, pp. 27–36, 2017.

[4] X. &. X. S. Wu, "Machine Learning for Text Classification: A Survey," *International Journal of Computational Intelligence,* vol. 4, pp. 143–154, 2008.

[5] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Graphical Statistics,* vol. 1, pp. 53–65, 1987.

[6] F. L. Gewers, G. R. Ferreira, H. F. de Arruda, F. N. Silva, C. H. Comin, D. R. Amancio and L. da Costa, "Principal Component Analysis: A Natural Approach to Data Exploration," 2018. Available: https://arxiv.org/abs/1804.02502. [Accessed: 02 October 2025].

[7] G. H. Dunteman, Principal Components Analysis, SAGE Publications, 1989.

[8] I. T. Jolliffe, Principal Component Analysis, Springer, 2002.

[9] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge, United Kingdom: Cambridge University Press, 2008.

[10] D. &. T. Y. Xu, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, pp. 165–193, 2015.

[11] T. C. K. C. G. &. D. J. Mikolov, "Efficient Estimation of Word Representations in Vector Space," Cornell University (arXiv), 2013.

[12] J. S. R. M. C. D. Pennington, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Journal of COMPUTER AND FORENSIC SCIENCES

Instructions and Information for Authors

Thank you for considering to submit a manuscript to the Journal of Computer and Forensic Sciences. The points below provide general instructions and information for authors. If you have any questions, please contact us at **comput.forensic.sci@kpu.edu.rs**.

Submission Checklist

- Ensure that your manuscript fits the **Aims and Scope** of the Journal.
- Use the **Microsoft Word template** or the **LibreOffice template** to prepare your manuscript.
- Ensure that your manuscript *complies with our* **research** *and* **publishing ethics** guidelines.
- Ensure that all authors have signed the **Author Statement**.

Aims and Scope

Journal of Computer and Forensic Sciences covers advanced and innovative research across the fields of computer and forensic sciences. More information about the Aims and Scope is available **here**.

Open Access

The Journal of Computer and Forensic Sciences is an open access, peer-reviewed scientific journal. All accepted manuscripts are made freely and permanently available online immediately upon publication, without subscription charges.

No Article Processing Charges (APC)

The journal does not have *submission charges or article processing charges.*

Manuscript Types

The journal publishes:

- Original research papers – recent research results in computer and forensic sciences,
- Review articles (solicited reviews) – comprehensive and up-to-date systematic review of a specific area,
- Case reports – describing interesting and exceptional cases, providing new information to the readership.

Research Ethics

Manuscripts reporting on research involving human subjects, animals, cell lines, and plants will be scrutinized by the editorial office. Editors may ask the authors for documentary evidence or reject any submission that does not meet the research ethics requirements.

Please note the following research ethics guidelines:

- Research involving human subjects must be carried out following the rules of **the Declaration of Helsinki**[1] of 1975, revised in 2013.
- For research on animals, authors should ensure that their research complies with the principles of the 3Rs (i.e., **Replacement, Reduction and Refinement**[2]) which provide a framework for ethical decision making in the use of animals in research and teaching.
- For research involving cell lines, the origin of any cell line should be stated.

Publication Ethics

We adhere to the Core Practices and Guidelines of the **Committee on Publication Ethics**[3], and expect authors to comply with its best ethical publication practices. Plagiarism, data fabrication, and image manipulation are not acceptable. If evidence of misconduct is found, appropriate action will be taken to correct or retract the publication.

Manuscript Submission

The authors may submit a manuscript that has not been published before, that is not under consideration for publication elsewhere, and that has been approved by all co-authors. All manuscripts should be submitted through **the online submission system.**

Templates

Please use the **Microsoft Word template** or the **LibreOffice template** to prepare your manuscript.

Language

All manuscripts should be written in English. Please note the following:

- Our reviewers are advised to distinguish between the quality of writing and the quality of ideas. However, authors are strongly encouraged to carefully edit and proofread their manuscripts.
- All accepted manuscripts undergo professional English editing (free of charge), and proofreading by the authors.
- Authors are also strongly urged to avoid using language or examples that may be perceived as discriminatory.

Manuscript Length

Different types of manuscripts require more or less space. Therefore, we imply no restrictions on the length of manuscripts, provided that the text is concise and comprehensive.

Manuscript Structure

All manuscripts should consist of three main parts: the front matter, the main body, and the back matter.

The front matter should include:

- **Title**: The manuscript title should be specific and relevant.
- **Author list**, **affiliations,** and **email addresses**: At least one author should be named as the corresponding author.
- **Abstract**: The a*bstract should be a single paragraph and must not exceed* 200 words. It should express the purpose of the study, indicate the main methods applied, and summarize the main findings.
- **Keywords**: Three to five specific keywords should be added.

The main body structure depends on the type of manuscript.

- In original research articles, it should include the following sections: **Introduction**, **Materials and Methods**, **Results**, **Discussion**, and **Conclusions** (authors can make appropriate minor modifications to this section structure).
- In review articles, it should consist of literature review sections.
- In case studies, it should consist of sections describing and discussing the case study.

The back matter should include the following sections:

- **Funding: Authors should disclose a**ll sources of funding for their research. For research that did not receive external funding, please add "This research received no external funding".
- **Acknowledgments (optional):** The authors may acknowledge any support that contributed to their manuscript, which is not included in the funding section.
- **Author Contributions (optional):** For manuscripts with several authors, their individual contributions can be specified.
- **Institutional Review Board Statement: For studies involving humans or animals, p**lease add "The study was conducted according to the guidelines of the Declaration of Helsinki" and a**dd the Institutional Review Board Statement and approval number. If ethical review and approval were waived, the authors are required to provide a detailed justification. For studies not involving humans or animals, please add "Not applicable".**
- **Informed Consent Statement: For studies involving humans, p**lease add "Informed consent was obtained from all subjects involved in the study"**. If informed consent was waived, the authors are required to provide a detailed justification. For studies not involving humans, please add "Not applicable".**
- **Conflicts of Interest:** Authors must disclose all conflicts of interest that may directly or potentially influence or impart bias on the work. Examples of potential conflicts of interest include but are not limited to: research grants, honoraria, financial support, employment, consultancies, affiliations, intellectual property rights, financial relationships, personal or professional relationships, and personal beliefs. If there is no conflict of interest, please add "The authors declare no conflict of interest."
- **References:** The Reference section must provide a numbered list of references, as recommended by the **IEEE Citation Guidelines**[4]. The list is comprised of the sequential enumerated citations. A number enclosed in square brackets, placed in the text of the report, indicates the specific reference. Citations are numbered in the order in which they appear.

Figures, Tables, and Equations

- All figures and tables should be inserted into the main body of the manuscript (preferably close to their first citation) and numbered following their number of appearance.
- All figures and tables should have an explanatory caption.
- All figures should be at a sufficiently high resolution (i.e., 300 dpi or higher) and provided in a single zip archive. Preferable formats are TIFF, JPEG, and EPS.
- All equations should be numbered following their number of appearance.
- All equations should be editable by the editorial office (i.e., not provided in a picture format).

Citation Policy

If a manuscript includes material (e.g., figures, tables, text passages, etc.) taken from other sources, its source must be clearly cited. When appropriate, the authors should obtain permission from the copyright owner(s) and include evidence that such permission has been granted when submitting their manuscripts.

References cited in the text must appear in the References list, and vice versa. Personal communications and classical works are cited in text only and are not included in the References list.

Authors should not engage in citation manipulation, including but not limiting to excessive self-citation and "honorary" citations. Authors should not cite advertisements or advertorial material.

Editorial Procedures and Peer-Review

- **Initial checks:** All submitted manuscripts are first checked whether they fit the aims and scope of the Journal and meet its standards. At this stage, your manuscript may be rejected before peer-review or returned to the authors for revision and resubmission.
- **Peer review:** Once a manuscript passes the initial checks, it is assigned to at least two independent experts for peer-review. A double-blind review is applied. The guidelines for reviewers are available **here**.

Editorial decision and revision: The decision on a manuscript is one of the following:

- Accept in present form,
- Minor revision,
- Major revision,
- Reject.

Author appeals: In a response to the reviewers, the authors should address all reviewers' comments. The response should be organized by presenting reviewers' comments one by one, followed by the authors' response. Authors may appeal a rejection by sending an e-mail including a detailed justification to the Editorial Office.

Guidelines for Reviewers

Thank you for considering reviewing a manuscript for the Journal of Computer and Forensic Sciences. We rely upon the knowledge and commitment of our peer reviewers to ensure the academic integrity of our Journal. The points below provide general reviewing guidelines. If you have any questions, please contact us at **comput.forensic.sci@kpu.edu.rs**.

Before Reviewing

Before you accept our invitation to review a manuscript, please consider the following:

- **Timeliness:** Please try to submit your reviews on time. If you cannot meet a given deadline, please let the editor know.
- **Reviewer qualifications:** You have been invited to review the manuscript because the editor believes that your expertise covers the topic of the manuscript. However, if the manuscript is outside your expertise, you should decline to review it. If the manuscript is *generally within your expertise but you do not feel confident assessing certain parts of it, please notify the editor.*
- **Conflicts of interest:** You should disclose potential conflicts of interest. If you recognize the author's work, have a financial or commercial conflict of interest related to the reported results, or have strong feelings about a controversial question considered in the manuscript, you should disqualify yourself. If you are unsure whether you have a conflict of interest, discuss your concerns with the editor.
- **Confidentiality:** You should keep the content of the manuscript confidential. If you want to involve your students or postdocs in your review, you must obtain permission from the editor. If permitted, your assistants must be informed of the confidentiality requirement.
- **Anonymity:** Our journal operates double-blind peer review, which means that the reviewers and authors are unaware of each other's identities. You must not reveal your identity to the authors (e.g., in your comments, in metadata of submitted files, etc.).
- **Interactions:** There is no open interaction between reviewers and no public commenting during formal peer review. After you have submitted your report, you will have access to other reviewers' reports. We will also inform you on the final editorial decision of the paper. The reviewer reports, the author responses to reviews, and the editor decision letters are not published.
- **Language:** All review reports must be written in English.
- **Reviewer acknowledgment:** Once a year, we recognize our reviewers with annual listings in the journal. If you do not wish to have your name included in this list, please let us know.
- **Ethical guidelines:** Please note that all reviewers for our Journal are expected to follow **the COPE Ethical Guidelines for Peer Reviewers**[5].

Evaluating Manuscripts

Regarding your comments for authors

Start your review report by writing a paragraph or two in which you summarize the manuscript, emphasize its main contributions, and list its strengths and weaknesses. Then continue with the assessment of the individual sections of the manuscript. You should consider questions such as:

- Is the manuscript relevant for the field and suitable for the Journal?
- Is the research question original and well-defined?
- Is the manuscript clear and well-structured? Does it contain all of the sections you would expect?
- Are the cited references relevant and complete?
- Is the methodology clearly explained? Are the methods appropriately selected?
- Are the data underlying the research representative and balanced?
- Is the manuscript scientifically and technically sound? Is the experimental design appropriate? Are the reported results reproducible?
- Are the results analyzed and interpreted correctly? Are the conclusions supported by evidence? Is the manuscript statistically sound?
- Does the theory fit the data?
- Are the figures, tables, source codes, etc. appropriate?
- Is the manuscript of interest to the scientific community and the Journal's audience?
- Do you think that the reported results may advance the field?
- Is the English language of sufficient quality?

When preparing your review report, you should:

- **Ensure that your identity is not disclosed;**
- Be objective and constructive;
- Be detailed; your feedback should help the authors improve their manuscript;
- **Number each comment;**
- **Cite page numbers when referring to specific parts of the manuscript;**
- Scrutinize the manuscript, not the authors; avoid any derogatory personal comments or unfounded accusations;
- Make sure to distinguish between the quality of writing and the quality of ideas, especially for authors whose first language may not be English;
- Immediately report any suspected breaches of ethics, including scientific misconduct, fraud, and plagiarism.

Regarding your comments for editors

Your comments to the editors will not be revealed to the authors or other reviewers. These comments are optional. However, if provided, they should be consistent with your comments to the authors.

Regarding your final recommendation

To make a final recommendation on a manuscript, please choose one of the following options:

- **Accept in present form:** The manuscript fulfills all of the requirements described above, although some small fixes may be required (e.g., typos or grammatical corrections, etc.). No additional action by the review is required.
- **Minor revision:** The manuscript requires a small number of easily correctable errors or minor content correction or clarification. The article is in principle accepted after revision based on the reviewer's comments.
- **Major revision:** The manuscript offers relevance or value but contains significant deficiencies and requires a major rework. The acceptance of the manuscript depends on the revisions.
- **Reject:** The manuscript has serious flaws or does not offer relevance or value.

Your final recommendation should match your comments for the authors. Please note that the final recommendation will be visible to editors and other reviewers, but not to the authors.

Acknowledgment to Reviewers

The editorial team of the Journal of Computer and Forensic Sciences wishes to thank the following reviewers, who in 2025 have performed an essential role in ensuring the academic integrity of this publication:

- Dušan Joksimović, PhD, Full Professor, Faculty of Computer Science and Information Technology, University of Criminal Investigation and Police Studies, Belgrade, Serbia;
- Vojkan Nikolić, PhD, Associate Professor, Faculty of Computer Science and Information Technology, University of Criminal Investigation and Police Studies, Belgrade, Serbia;
- Nenad Korolija, PhD, Assistant Professor, Faculty of Computer Science and Information Technology, University of Criminal Investigation and Police Studies, Belgrade, Serbia;
- Ivan Tot, PhD, Associate Professor, Department of Telecommunications and Informatics, Military Academy, University of Defence, Belgrade, Serbia;
- Ivan Košanin, PhD, Ministry of Interior of the Republic of Serbia;
- Nemanja Maček, PhD, Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, and SECIT Security Consulting, Pančevo, Serbia;
- Branimir Trenkić, PhD, Professor of Vocational Studies, School of Electrical and Computer Engineering, Academy of Art and Technical Applied Studies, Belgrade, Serbia;
- Negovan Stamenković, PhD, Full Professor, Faculty of Natural Sciences and Mathematics, University of Priština, Kosovska Mitrovica, Serbia.