Indonesian Journal of Electrical Engineering and Computer Science
Vol. 38, No. 3, June 2025, pp. 1722~1734
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v38.i3.pp1722-1734

Plagiarism detection using text-representing centroids techniques

Sureeporn Nualnim[1], Maleerat Maliyaem[1], Herwig Unger[2]
[1]Department of Science and Technology, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
[2]Department of Communication Networks, University of Hagen, Hagen, Germany

Article Info

Article history:

Received Jul 3, 2024
Revised Nov 19, 2024
Accepted Nov 24, 2024

Keywords:

Co-occurrence graph
Plagiarism detection
Text representing centroid
Text similarity
Text-based representation

ABSTRACT

This study addresses the limitations of traditional plagiarism detection methods by introducing the text-representing centroid (TRC) technique. TRC is designed to improve the accuracy of detecting semantic similarities and sophisticated forms of plagiarism. It utilizes a co-occurrence graph to identify centroid terms that represent the core meaning of text documents, effectively capturing the contextual associations between terms. Extensive experiments were conducted on a dataset of academic papers to assess TRC's performance against traditional techniques across various categories of plagiarism, including near-copy, modified-copy, and paraphrasing. The results demonstrate the effectiveness of the TRC technique, achieving an average precision of 0.96 and a recall of 0.71. This performance surpasses methods such as Jaccard and Cosine similarity in accurately detecting more, complex forms of plagiarism. These findings highlight TRC's potential as a robust tool for both academic and industry applications, helping to ensure integrity in textual content through precise and comprehensive plagiarism detection.

Corresponding Author:

Sureeporn Nualnim
Department of Science and Technology, Faculty of Information Technology and Digital Innovation
King Mongkut's University of Technology North Bangkok
1518 Pracharat 1 Road, Wongsawang, Bangsue, Bangkok
Email: Sureeporn.n@mail.rmutk.ac.th

1. INTRODUCTION

In today's digital age, technology and the internet have significantly transformed how information is created, shared, and consumed [1]. The rise of online publications has made plagiarism a prevalent issue in academics and other fields. Plagiarism involves using someone else's ideas, results, or words without giving them credit. This includes copying text without proper citation and claiming it as one's work [2]-[4]. This unethical practice is widespread in academia, where the integrity of scholarly work is highly valued. Plagiarism can take several forms, it can occur as verbatim copying, where content is directly reproduced without giving credit to the original source. Another form is paraphrasing without citation, where ideas are rephrased without acknowledgment. Additionally, there is mosaic plagiarism, which involves blending elements from multiple sources, and self-plagiarism, where an individual reuses their own previous work. Finally, citation errors occur when references are incomplete or misleading [1]. Furthermore, plagiarism also be classified as literal plagiarism and intelligent plagiarism [5]-[7]. Literal plagiarism involves directly copying text with minimal changes, whereas intelligent plagiarism includes techniques like paraphrasing, summarizing, translating, or adapting ideas to retain the original meaning while altering the form [5], [7], [8]. The issue of plagiarism has worsened due to the increasing number of online publications in recent decades [9]. There is now a vast amount of unstructured text available on the web and in large-scale repositories, much of

which is repetitive. This repetition makes it easier for people to engage in plagiarism and more challenging for original work to be recognized. Therefore, plagiarism detection has become crucial in various fields, including publishing, journalism, patent verification, and academia, to ensure the uniqueness of texts, materials, and resources [10]. Plagiarism has become a significant issue, raising concerns about academic integrity and the quality of educational content and research [11]. It is essential to have effective plagiarism detection to uphold standards of intellectual honesty and ensure proper credit is given.

Traditional approaches to plagiarism detection, such as the vector space model (VSM) and bag-of-words (BOW), represent documents as numerical vectors where each component reflects the weight of individual words, assuming each word is treated independently [12]. These methods typically calculate similarity using Euclidean or cosine distance within this vector space [7]. However, they have significant limitations. Firstly, they cannot effectively handle synonym substitution, as they lack semantic understanding. For instance, replacing "happy" with "joyful" in a plagiarized text would go undetected. Secondly, VSM and BOW methods struggle to capture the meaning and structure of the text, ignoring word order and relationships, leading to difficulties in comparing paraphrased documents [13]. Studies by Chang *et al.* [12] and Huynh *et al.* [14] demonstrated the inadequacy of VSM and BOW in handling word semantics, often resulting in missed detections when synonyms or semantically similar words are used. These methods fail to capture semantic relationships and structural information within the text. Given these limitations, Kubek and Unger [15] introduced text-representing centroids (TRC), a technique for classifying and grouping texts based on semantic content. TRC identifies core terms, or "centroids," representing the main themes of a document. Inspired by the concept of a center of mass, these centroids serve as focal points for understanding and comparing text content. Unlike traditional BOW models, which rely solely on word frequency, TRC leverages co-occurrence graphs to capture semantic relationships. This enables TRC to effectively handle short texts and provide a deeper understanding of document content. Moreover, TRC is language-independent, offering an advantage over BOW methods that often require language-specific preprocessing. Despite advancements in natural language processing (NLP), traditional plagiarism detection methods face challenges in accurately identifying sophisticated plagiarism techniques. The inability to effectively handle semantic variations, paraphrasing, and structural modifications hinders the prevention of academic dishonesty.

To address these challenges, this paper explores the application of TRC in plagiarism detection. The study aims to assess the effectiveness of TRC in detecting plagiarism, particularly its ability to differentiate between near-verbatim copying and more sophisticated forms of plagiarism, such as modified copying and paraphrasing. Applying the TRC technique, this research enhances traditional similarity measures by capturing document semantic relationships and identifying sentence centrality conditions. The hypothesis is that the TRC technique will demonstrate significantly higher accuracy and precision in identifying nuanced cases of plagiarism, particularly in scenarios involving paraphrasing and modified copying, compared to traditional methods such as Jaccard and Cosine similarity.

The main contributions of this work are as follows:

- We present the TRC technique, which enhances plagiarism detection by more accurately capturing semantic relationships within texts.
- We demonstrate that TRC effectively identifies various forms of plagiarism, such as near-copies, modified copies, and paraphrases, thereby increasing the accuracy of plagiarism detection.
- We provide empirical evidence through comprehensive evaluation using standard performance metrics, supporting the effectiveness of the TRC method.

The structure of the paper is as follows: section 2 describes the proposed methodology, followed by the experiments and results in section 3. Section 4 concludes the paper and offers suggestions for future work.

2. METHOD

The TRC technique [15] is a method for determining the centroid terms of text documents, which effectively represent those documents. Centroid terms can help measure the semantic distance and similarity between different documents, even when they use different vocabulary to describe similar topics. These centroid terms are obtained from a co-occurrence graph created from a collection of text documents. Co-occurrence graphs capture the relationships between terms in a text corpus by identifying terms that frequently appear together, providing insights into semantic relationships.

For plagiarism detection, this technique follows a structured, six-step process as illustrated in Figure 1, which includes:

- Document corpus: collecting the set of documents to analyze.
- Document preprocessing: preparing the text by removing noise and standardizing format.
- Co-occurrence graph construction: building a graph based on term co-occurrence relationships.
- Centroid term identification: identifying key terms that represent the document.

- Centroid-based distance calculation: computing distances between centroids to quantify semantic similarity.
- Converting distance to similarity score: translating the distance into a similarity score, which informs the plagiarism detection decision.

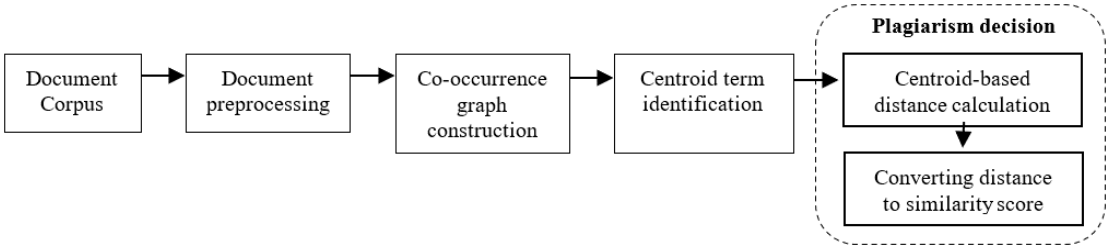

Figure 1. The process overview of the proposed method

2.1. Document preprocessing

Text pre-processing is a foundational step in NLP tasks, significantly impacting performance, particularly in areas such as plagiarism detection [16]. The process begins with tokenizing the text, dividing the document into sentences, and then further into words or tokens. This transformation of raw text into smaller, manageable pieces is essential for downstream analysis. Before removing stopwords, several normalization steps are applied to ensure consistency and improve the analyzability of the text. First, all text is converted to lowercase to avoid case sensitivity issues, treating "The" and "the" as the same word. The next step is lemmatization, which removes suffixes and reduces words to their root forms. For example, "running" is transformed into "run." Moreover, lemmatization can be applied more accurately by considering the grammatical context; for instance, "better" can be changed to "good." Once standardized, common stopwords (e.g., "and," "the," "is") that provide little meaningful context are removed to enhance data quality. Finally, part-of-speech (POS) tagging is performed to assign grammatical categories (e.g., nouns, verbs) to words. This step is critical for filtering out less relevant terms and retaining the most informative ones [17]. In this study, we retained nouns and proper nouns, which are essential for constructing meaningful co-occurrence relationships. The NLTK library is used for all pre-processing tasks, as it offers a comprehensive set of functions and modules suitable for our analysis [18].

2.2. Co-occurrence graph construction

Constructing the co-occurrence graph forms the basis of the TRC technique in plagiarism detection, as it captures semantic relationships between words [15], [19], [20]. Unlike traditional BOW models, which rely on simple character matching or topic similarity assessments and often ignore words' syntactic and semantic contexts [7], co-occurrence graphs capture the nuanced connections between words based on their proximity within a specified window size. This approach enables a more sophisticated understanding of the underlying structure of the text [21].

Following the preprocessing step, processed sentences are used to create the co-occurrence graph. We first initialize an empty co-occurrence matrix to store co-occurrence statistics based on the frequency of term pairs in each sentence. For each sentence, we increment the co-occurrence count for each word pair to capture their semantic and syntactic relationships [22]. The co-occurrence graph is then constructed by mapping words as nodes and adding edges weighted by the frequency of term pairs, resulting is an undirected weighted graph that captured these relationships.

In this co-occurrence graph $G = (V, E)$, V represent the unique terms $T = \{t_1, t_2, \ldots, t_n\}$ in the documents, and E is the set of edges $E = \{e_{12}, e_{13,} \ldots, e_{ij}\}$ connecting terms t_i and t_j that co-occur within a sentence. Each edge e_{ij} has a weight function $g(t_\mathrm{i}, t_\mathrm{j})$, representing the co-occurrence frequency between terms [23]:

$$g(t_\mathrm{i}, t_\mathrm{j}) = \frac{2 \times count(t_i, t_j)}{count(t_i) + count(t_j)} \quad (1)$$

Where $count(t_i)$ and $count(t_\mathrm{j})$ represent the frequency of term t_i and t_j appearing individually, and $count(t_\mathrm{i}, t_\mathrm{j})$ is the frequency of their co-occurrence. This weighting ensures that $g(t_\mathrm{i}, t_\mathrm{j})$ ranges between

0 and 1, with values above 1 adjusted to 1. The weight function $g(t_i, t_j)$ is used to calculate the distance $d(t_i, t_j)$ between terms (t_i, t_j) in the graph, providing a measure of their semantic closeness. The distance metric used to calculate this distance is provided in (2).

$$d(t_i,t_j)=\frac{1}{g(t_i,t_j)+\text{smoothing } factor} \quad (2)$$

Where $d(t_i, t_j)$ is the distance between two terms, and the smoothing factor default value of 0.01 prevents division by zero [24]. Following this process, we obtain an undirected co-occurrence graph representing term relationships, forming the foundation for centroid term identification.

2.3. Centroid term identification

After constructing the co-occurrence graph, centroids are identified for each sentence. A centroid represents the word that best represents and encapsulates the meaning of the sentence, selected for its average proximity to all the other words in that sentence [25]. This approach ensures that the centroid captures the main idea of the sentence, which is essential for meaning-based plagiarism detection. By focusing on centroid terms rather than more word matches, the system can detect sentences sharing essential similarities, which might suggest plagiarism based on deeper textual meaning.

A centroid term t is considered the most semantically representative term in a sentence S. It is chosen by minimizing the average distance $d(S,t)$ between the term t and all other words w_i in the sentence S, defined mathematically as [15].

$$d(S,t) = \frac{1}{N}\sum_{i=1}^{N} d(w_i,t) \quad (3)$$

Where $d(S,t)$ is the average distance between the term t and the sentence S. N is the number of words in the sentence S that were reachable from the term t in the co-occurrence graph. $d(w_i,t)$ is the shortest path distance between the term t and the word w_i in the co-occurrence graph. This process aims to find a centroid term t that has the smallest average distance to all other words, indicating that t is central to the meaning of the sentence.

This process identifies the centroid terms t that are closest to all other words, making t central to the meaning of the sentence. The centroid term identification process involves several steps, outlined as follows:

Step 1: generate word pairs

All possible pairs of unique centroid candidate words within the sentence are generated according to the procedure outlined in Algorithm 1. These word pairs are critical for calculating the distances between terms, which contributes to identifying the most central word in the sentence.

Algorithm 1. Create unique pair words

```
Input: A list of unique centroid words (unique_centroid_word)
Output: A list pairword, which will contain all possible pairs of unique words without
duplicates

1: Initialize an empty list pairword
2: for index i from 0 to length(unique_centroid_word) - 2 do
3:     for index j from i+1 to length(unique_centroid_word) - 1 do
4:                 Append the pair (unique_centroid_word[i], unique_centroid_word[j]) to
pairword
5:     end for
6: end for
7: return pairword
```

Step 2: calculate distances

Using Algorithm 2, distances between the word pairs are computed based on the co-occurrence data. The sum of distances to other words for each candidate centroid word is calculated to evaluate its centrality in the sentence.

Algorithm 2. Finding the centroid terms

```
Input: pairword: List of word pairs (from Algorithm 1), cooccurr: Co-occurrence data
(containing distances between word pairs), z: Set of unique centroid candidate words.
Output: result_centroid: The centroid term that has the minimum average distance to all
other words.
1: Initialize an empty list result.2: Set amt to the length of the set z.3: For each pair
of words i in pairword:4:     Initialize sum_dist to 0.5:     If i consists of identical
words, set dist to 0.6:     Else:
```

```
7:         For each pair in cooccurr:8:             If the pair matches i, set dist to the
corresponding co-occurrence value.9:             Break.10:         End For.11:     Add dist
to sum_dist.12:     Append the pair (i,sum_dist) to result.13: Initialize an empty list
centroid_list.14: For each word k in z:15:     Set sum_dist to 0.
16:     For each (word,distance) in result:17:         If word=k, add distance to
sum_dist.18:     Calculate the average distance avg_dist = sum_dist / amt.19:     Append
(k,avg_dist) to centroid_list.20: Set min_dist to a large value.21: Set result_centroid to
None.22: For each (word,avg_dist) in centroid_list:23:     If avg_dist <
min_dist:24:         Set min_dist to avg_dist.
25:         Set result_centroid to word.26: Return result_centroid.
```

Step 3: select the centroid

The word with the smallest average distance to all other words in the sentence is chosen as the centroid. This centroid term reflects the core meaning of the sentence, making it useful for identifying semantic similarities between sentences in plagiarism detection.

2.4. Plagiarism detection decision

The plagiarism detection process involves two main steps: calculating the centroid-based distance between sentences and converting this distance into a similarity score to determine whether plagiarism has occurred. This approach lets the system detect complex plagiarism cases, such as paraphrasing or sentence modification, by capturing semantic relationships between words. Unlike traditional methods such as Jaccard and Cosine similarity, which primarily focus on word-level overlaps, the TRC technique uses centroid-based distances to detect plagiarism even when sentences have undergone significant rewording or restructuring.

To detect potential plagiarism, the centroid of an original sentence is compared to that of a potentially plagiarized sentence. The distance between the two centroids is calculated using Dijkstra's algorithm, which finds the shortest path between the centroids in the co-occurrence graph. This distance represents the degree of semantic similarity between the sentences. The calculated distance is then converted into a similarity score to assess the likelihood of plagiarism, as discussed in the following section.

2.4.1. Centroid-based distance calculation

The first step in plagiarism detection is to calculate the distance between the centroids of the original sentence (S_1) and the potentially plagiarized sentences (S_2). Centroids represent the most semantically significant terms in a sentence, capturing the core meaning of the text. The centroid-based distance between two sentences quantifies how similar their content is based on the central terms that encapsulate their meanings.

To calculate the centroid-based distance between sentences S_1 and S_2, the centroid term $t_1$is selected from sentence S_1 as the term with the minimum average distance to all other words in the sentence. This distance is denoted as $d(S_1, t_1)$. Similarly, t_2 is the centroid term of sentence S_2 and the centroid-based distance between the two sentences is defined as the distance between these two centroids, calculated using the shortest path in the co-occurrence graph. The centroid-based distance is mathematically expressed as:

$$\zeta(S_1, S_2) = d(t_1, t_2) \tag{4}$$

Where t_1 and t_2 are the centroids of sentences S_1 and S_2, respectively, $d(t_1, t_2)$ is the distance between these centroids, calculated using the shortest path metric in the co-occurrence graph. In this study, we used Dijkstra's algorithm [26] to calculate the shortest path between the centroids on a graph representing the semantic relationships between terms. This shortest path distance serves as a measure of semantic similarity between the sentences. By comparing the distances between their centroids, we can effectively assess how closely related the sentences are in content, even if they employ different word choices or phrasing.

2.4.2. Converting distance to similarity score

After calculating the centroid distance, the system converts this distance into a similarity score. The similarity score $\zeta_{sim}(S_1, S_2)$ is calculated using the following formula:

$$\zeta_{sim}(S_1, S_2) = \frac{1}{1+\zeta(S_1, S_2)} \tag{5}$$

This formula ensures that shorter centroid distances result in higher similarity scores. The similarity score ranges between 0 and 1 [27], with a score closer to 1 indicating higher similarity between the sentences. Once calculated, the similarity score is then compared against a predefined threshold (typically set at around 0.8) to determine whether plagiarism has occurred is present. If the similarity score $\zeta_{sim}(S_1, S_2)$ exceeds the

threshold, the sentence is flagged as plagiarized; otherwise, it is considered original. This approach effectively assesses text similarity while distinguishing between original and potentially plagiarized content based on a clear, quantifiable metric.

2.4.3. Example: TRC techniques in action

To demonstrate the TRC technique in action, consider the following example. We have two sentences: one from the original text (S_1) and one potentially plagiarized sentence (S_2).

- S_1 (Original sentence): "The algorithm used information retrieval and keyword sequence matching techniques to detect plagiarized sentences."
- S_2 (Potentially plagiarized sentence): "The algorithm detects plagiarized sentences using information retrieval and keyword sequence matching techniques."

The first step in plagiarism detection using TRC involves constructing a co-occurrence graph from a set of text documents. This graph represents the relationships between words and phrases within the documents. We can identify similarities and potential plagiarism by analyzing the co-occurrence patterns in the graph.

As shown in Figure 2, both sentences undergo preprocessing, which involves text cleaning and tokenization. During this process, stop words like "the," "and," and "to" are removed, and only the important terms are extracted, leaving the significant words from each sentence. After preprocessing, the terms from the original sentence S_1 and S_2 are:

- S_1: algorithm, information, retrieval, keyword, sequence, match, technique, sentence.
- S_2: algorithm, sentence, information, retrieval, keyword, sequence, match, technique.

After preprocessing, the next step is to find the centroid (the most representative word) for each sentence. The centroid is calculated by measuring the average distance between each word in the sentence and all other words in the co-occurrence graph. In this case, the centroid term for both sentences (S_1 and S_2) is "match". This word acts as a central reference point that best captures the essence of the sentence.

Once the centroids are identified, the centroid-based distance between the two sentences is calculated. Since both sentences share the same centroid ("match"), the distance is 0. This distance is converted into a similarity score, where a distance of 0 corresponds to a similarity score of 1, indicating identical centroid-based representations. This study compares the similarity score to a predefined threshold of 0.8. If the score exceeds 0.8, the system flags the sentences as plagiarized. In this case, with a similarity score of 1, which surpasses the threshold, the sentences are flagged as plagiarized.

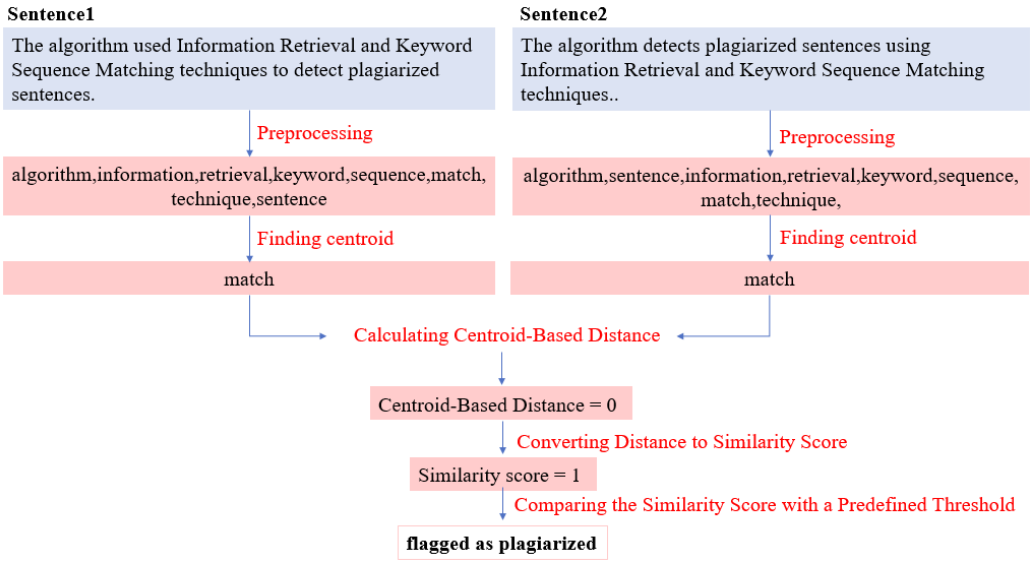

Figure 2. Example of plagiarism detection using TRC

3. EXPERIMENT RESULTS

This section outlines the experimental process and presents the results obtained while evaluating the TRC technique for plagiarism detection. The experiments are designed to assess the performance of TRC in comparison with traditional text similarity methods, such as Jaccard and Cosine similarity, across different plagiarism scenarios, including near-copy, modified copies, and paraphrased.

3.1. Dataset preparation

The dataset for evaluating the effectiveness of the TRC technique in plagiarism detection consisted of two primary components: a dataset for constructing the co-occurrence graph and a dataset for testing plagiarism detection. The first dataset included a corpus of 100 academic documents selected from publicly accessible academic archives and repositories. These documents were chosen to represent diverse topics and writing styles. Ensuring the generalizability of the co-occurrence graph. The co-occurrence graphs generated from this corpus provided a reference for identifying centroid terms in text documents, which is essential for applying the TRC technique.

The second dataset was specifically prepared to assess the plagiarism detection capabilities of the TRC technique. It comprised original and plagiarized sentences, with the original sentences sourced from the same academic archives as the first dataset. This dataset contained 300 cases of plagiarism, categorized into three types: near copy, modified copy, and paraphrase. To create consistent examples for each category, ChatGPT was used to simulate the cases, following predefined definitions of each plagiarism type to ensure uniformity. Sample messages were crafted for each case, producting controlled and consistent instances of plagiarism across all trials. Table 1 presents a detailed breakdown of the instances of plagiarism categorized by type. The even distribution among the three categories allowed for a comprehensive evaluation of the effectiveness of the TRC technique in identifying various forms of plagiarism.

Table 1. Number of plagiarized cases by type

Type of plagiarism	Number of plagiarized cases
Near-copy	100
Modified copy	100
Paraphrase	100

3.2. Experimental setup

3.2.1. Baseline comparison

In this study, we compare the effectiveness of the TRC technique with two well-established baseline methods for measuring text similarity: Jaccard Similarity and Cosine similarity. These methods are widely used in plagiarism detection and text analysis due to their simplicity and effectiveness in detecting various forms of content overlap. Cosine similarity: measures the similarity between two text vectors based on their cosine angle.

- Jaccard similarity

The Jaccard similarity measures the degree of overlap between two sets by dividing the size of their intersection by the size of their union. When applied to text analysis, each word or token in a text segment is considered an element of the set [28]. The Jaccard coefficient, used to calculate this similarity, is defined as follows:

$$J(A,B) = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{|A|+|B|-|A \cap B|} \quad (6)$$

Where A and B are the sets of tokens from two text segments, this method is particularly effective in detecting near-copy plagiarism, where there is a substantial overlap in the words used between the original and the suspected copy. However, it can be sensitive to minor modifications, as small word-choice changes can significantly reduce the similarity score.

- Cosine similarity

Cosine similarity is a metric that quantifies the similarity between two vectors by measuring the cosine of the angle between them in a multi-dimensional space. In text analysis, each document is represented as a vector within a term space, where each dimension corresponds to a unique word or token. This approach enables the comparison of documents based on the direction of their vectors rather than their magnitude, making cosine similarity particularly effective in identifying semantic similarity between texts [28]. The cosine similarity score is calculated as:

$$similarity(A,B) = \frac{AB}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{m} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \sum_{i=1}^{n} B_i^2}} \quad (7)$$

Where A and B are the vector representations of the two text segments, cosine similarity is widely used for comparing the overall semantic content of two texts, making it effective for identifying modified copies where some words may have been replaced, but the overall meaning remains similar.

For the similarity between Jaccard and Cosine, we preprocess the text data by tokenizing the documents and converting them to lowercase. Stopwords are removed to ensure similarity measurements focus on meaningful content rather than common function words. For Jaccard similarity, text segments are treated as sets of tokens. In contrast, for Cosine similarity, term frequency-inverse document frequency (TF-IDF) weighting represents each document as a vector. The comparison is performed using the scikit-learn library in Python.

3.2.2. Evaluation metrics

To assess the effectiveness of the plagiarism detection approach, four standard metrics are employed: accuracy, precision, recall, and F-measure [29]. These metrics are calculated based on the entries in the confusion matrix, shown in Table 2, which summarizes the model's classification performance.

Table 2. Confusion matrix

	Predicted plagiarized	Predicted non-plagiarized
Actual plagiarized	True positive (TP)	False negative (FN)
Actual non-plagiarized	False positive (FP)	True negative (TN)

TP: cases where the model correctly identifies plagiarized content, flagging it as plagiarism.
TN: instances where non-plagiarized content is correctly classified as original [30].
FP: occurrences where non-plagiarized content is mistakenly flagged as plagiarized, potentially including correctly cited content [30].
FN: instances where actual plagiarism is undetected, leading to its incorrect classification as original content.

These categories form the basis for calculating performance metrics as follows:

- Accuracy: proportion of all correctly classified instances (both TP and TN) out of the total instances.

$$Accuracy\ (A) = \frac{TP+TN}{TP+FP+FN+TN} \quad (8)$$

- Precision: proportion of correctly identified plagiarized cases among all instances flagged as plagiarized.

$$Precision\ (P) = \frac{TP}{TP+FP} \quad (9)$$

- Recall: ability of the model to correctly identify all actual instances of plagiarism.

$$Recall\ (R) = \frac{TP}{TP+FN} \quad (10)$$

- F-measure: a balanced measure that combines precision and recall, providing an overall effectiveness score.

$$F-measure\ (F) = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (11)$$

As proposed in previous research [29], [30], these metrics allow a comprehensive evaluation of the model's ability to detect plagiarism accurately and effectively.

3.3. Results and discussion

In this section, we present and discuss the results of the experiments conducted to evaluate the performance of the TRC technique in comparison to other established methods like Jaccard and Cosine similarity. The evaluation metrics include accuracy, precision, recall, and F-measure. The dataset used for this analysis consists of 100 academic papers split into 5,406 sentences, forming an undirected co-occurrence graph with 3,172 nodes and 97,216 edges. The performance of the TRC technique was evaluated in three key plagiarism detection cases: near copy, modified copy, and paraphrase. Tables 3 to 5 provide the detailed results for each case, and the combined results are presented in Table 6.

As shown in Table 3, Cosine similarity achieved the highest accuracy for near-copy plagiarism at 85%. The TRC technique strikes a better balance between precision (0.89) and recall (0.64) compared to Jaccard, which had poor recall (0.38). The results show that cosine similarity performs better overall, but TRC is more adaptable in detecting minor near-copy plagiarism, which often requires a balance between

precision and recall. This implies that TRC may be more useful in real-world situations where near-copy plagiarism is less frequent, but instances of textual similarity can still be identified.

Table 3. Result of performance measurement for near copy plagiarism case

	Jaccard	Cosine	Proposed method
Accuracy	69%	85%	78%
Precision	1.00	0.86	0.89
Recall	0.38	0.84	0.64
F-measure	0.55	0.85	0.74

In Table 4, it was found that the TRC technique once again performed better than the Jaccard and Cosine similarity methods in cases involving modified copies, achieving an accuracy of 86%. The TRC technique also demonstrated a higher F-measure (0.85), highlighting its strength in handling modified text effectively. These results indicate that the TRC technique is well-suited for detecting exact copying and content that has been restructured or reworded, making it adaptable and robust in such scenarios.

Table 4. Result of performance measurement for modified copy cases

	Jaccard	Cosine	Proposed method
Accuracy	71%	84%	86%
Precision	1.00	0.85	0.91
Recall	0.42	0.82	0.80
F-measure	0.59	0.84	0.85

In Table 5, the TRC method showed 80% accuracy for paraphrase detection, slightly lower than the 83% achieved by Cosine. However, TRC exhibited better recall (0.68) than both Cosine and Jaccard, making it more effective in identifying paraphrased content. Since paraphrasing often involves not only replacing synonyms but also restructuring syntax, the ability of the TRC method to balance precision (0.89) and recall makes it particularly effective in this context. Its relatively higher F-measure (0.77) compared to other methods also highlights its usefulness in detecting more sophisticated forms of plagiarism.

Table 5. Result of performance measurement for paraphrase cases

	Jaccard	Cosine	Proposed method
Accuracy	62%	83%	80%
Precision	1.00	0.85	0.89
Recall	0.24	0.80	0.68
F-measure	0.39	0.82	0.77

The overall findings from all cases, presented in Table 6, show that the TRC technique consistently performs well in all tested plagiarism scenarios. While Cosine similarity generally delivers high accuracy, the TRC technique stands out for its ability to maintain a strong balance between precision and recall in various cases, making it a viable option for detecting plagiarism, especially when dealing with modified or paraphrased content. These results emphasize the adaptability of the TRC technique, which can detect a broader range of plagiarism while minimizing the loss of precision and recall.

Table 6. Result of performance measurement for all cases

	Jaccard	Cosine	Proposed method
Accuracy	51%	83%	76%
Precision	1.00	0.95	0.96
Recall	0.35	0.82	0.71
F-measure	0.51	0.88	0.82

To fully understand the strengths and limitations of the proposed method, we thoroughly analyzed instances where false positive and false negative results occurred. By examining specific examples, we aimed to pinpoint potential issues and areas that could be improved. We carried out a comprehensive examination

of false positive instances, detailed in Table 7, and false negative instances, outlined in Table 8, to recognize the limitations of the suggested method and to ascertain regions that require improvement.

Table 7. Example of the source sentence and suspicious in false positive case

Source sentence	Suspicious sentence	Similarity value		
		Cosine	Jaccard	TRC
The primary heuristic retrieval step can lead to decreasing the search space for subsequent text alignment step.	The initial heuristic retrieval phase helps reduce the scope of the search for the following text alignment process.	63.5%	46.5%	100%

The high similarity score from TRC indicates a possible false positive due to its sensitivity to semantic relationships. TRC can interpret slight rephrasing's and synonymous expressions as identical, which may cause it to incorrectly flag a sentence as plagiarized, even if there are some lexical differences from the original source. This highlights a potential limitation of TRC in distinguishing between genuine plagiarism and acceptable rewording.

Table 8 presents a case of near-copy plagiarism where the TRC method yields a relatively low similarity score of 32%. This score suggests a potential false negative. The low score may arise because redundant phrases were removed from the suspicious sentence, which disrupts the alignment of centroid terms. As a result, the centroid terms in the source and suspicious sentence may not correspond closely, leading to a reduced similarity score. Furthermore, the TRC method's sensitivity to minor structural changes, such as rephrasing or reordering terms, can also cause a decrease in the similarity score, even though the sentences still convey similar meanings and contexts.

Table 8. Example of the source sentence and suspicious in the false negative case

Source sentence	Suspicious sentence	Similarity value		
		Cosine	Jaccard	TRC
Factors contributing to plagiarism include lack of awareness, lack of understanding, lack of competence, and personal attitudes.	Factors contributing to plagiarism include lack of awareness, understanding, competence, and personal attitudes.	93.7%	87.9%	32%

This study examined the limitations of traditional plagiarism detection methods, such as the VSM and BOW, in accurately identifying nuanced forms of plagiarism. While previous research has assessed the effectiveness of these methods in detecting exact copies or highly similar content, it has not specifically addressed their limitations in handling complex semantic variations, such as synonym substitution and paraphrasing. This gap underscores the need for advanced techniques that capture semantic and structural nuances. Our findings indicate that the TRC technique is well-suited for detecting nuanced plagiarism, particularly in modified and paraphrased content cases. The proposed method achieved an accuracy rate of 86% and a precision score 0.91 in identifying modified copies, outperforming traditional methods like Cosine and Jaccard similarity. TRC demonstrated balanced performance across different types of plagiarism, including near-copy, modified copy, and paraphrase, excelling in cases of paraphrased content where traditional methods typically fail to capture semantic relationships.

The study also suggests that the higher sensitivity of the TRC technique to semantic relationships does not compromise its effectiveness in detecting modified and paraphrased text. Compared to other studies, such as those by Chang *et al.* [12] and Huynh *et al.* [14], our results show that the centroid-based approach of TRC is better equipped to handle synonym substitution and structural rephrasing, providing an advantage over methods that rely solely on lexical matching. The comprehensive evaluation presented here underscores the potential of TRC in managing complex forms of plagiarism; however, further research is recommended to confirm its robustness in real-world applications, particularly in contexts where structural variations may lead to occasional false positives or negatives.

Despite the strengths of TRC, this method faces scalability challenges due to the computational demands of co-occurrence graph construction. Future studies should explore optimization strategies to enhance the feasibility of TRC for larger datasets and diverse linguistic contexts. Additionally, the integration of syntactic parsing and embedding-based models could reduce the sensitivity of TRC to structural variations, potentially improving precision in complex cases. Feasible approaches, such as hybrid methods or neural network approximations, may improve the computational efficiency of the co-occurrence graph, enhancing the applicability of TRC in large-scale datasets. Expanding the TRC approach to address cross-language plagiarism detection, potentially through multilingual embeddings, is also a promising direction for

extending its utility across diverse linguistic contexts. Recent observations affirm that the TRC method is particularly effective in detecting complex forms of plagiarism, including modified and paraphrased content. These results emphasize the ability of TRC to capture semantic relationships beyond traditional methods like Cosine and Jaccard similarity, especially in cases where lexical matching proves insufficient. Nonetheless, limitations persist, particularly in cases involving significant structural variations that may cause misalignments in centroid terms, resulting in occasional false positives or negatives. Addressing these limitations is essential for further refining the applicability of the TRC method.

4. CONCLUSION

This paper introduces the TRC technique as an innovative approach for plagiarism detection, aiming to evaluate its effectiveness across various types of plagiarism, including near-copy, modified copy, and paraphrasing. The TRC technique leverages centroid terms that capture the core meaning of text documents, addressing limitations found in traditional methods like Jaccard and Cosine similarity. By utilizing co-occurrence graphs to represent semantic relationships, TRC can detect contextual and semantic similarities often missed by conventional methods. This study also compares TRC with these traditional methods, demonstrating its superior accuracy, precision, recall, and effectiveness in detecting sophisticated forms of plagiarism.

The findings reveal that the TRC technique is particularly effective in identifying modified and rephrased content, showing a notable improvement over traditional methods in balancing accuracy and completeness. The TRC approach successfully captures nuanced semantic differences through centroid-based similarity measures, making it well-suited for cases where reworded text retains the same meaning. While TRC exhibits strong accuracy, certain rephrased content occasionally remains undetected, indicating areas for further refinement. Additionally, the complexity of constructing co-occurrence graphs presents scalability challenges, particularly for large datasets.

In the future, research should focus on overcoming the current limitations to enhance the performance and scalability of the TRC method. Incorporating deep learning techniques, such as word or sentence embeddings, could improve its ability to recognize and handle complex paraphrasing. Reducing the computational overhead of constructing co-occurrence graphs will also be crucial for scalability. Further experimentation with larger and more diverse datasets is also necessary to assess the generalizability of the method. Addressing these challenges, the TRC method can be refined into a powerful tool for tackling plagiarism across various domains.

ACKNOWLEDGMENTS

The authors would like to thank King Mongkut's University of Technology North Bangkok (KMUTNB) for supporting the development of this research. The authors also appreciate the support from the Rajamangala University of Technology Krungthep (RMUTK) for providing the opportunity and resources to pursue this study.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sureeporn Nualnim	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Maleerat Maliyaem	✓	✓				✓				✓	✓		✓	
Herwig Unger	✓	✓		✓			✓	✓		✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, S.N., upon reasonable request.

REFERENCES

[1] H. Veisi, M. Golchinpour, M. Salehi, and E. Gharavi, "Multi-level text document similarity estimation and its application for plagiarism detection," *Iran Journal of Computer Science*, vol. 5, no. 2, pp. 143–155, Jun. 2022, doi: 10.1007/s42044-022-00098-6.

[2] Z. F. Alfikri and A. Purwarianti, "Detailed analysis of extrinsic plagiarism detection system using machine learning approach (Naive Bayes and SVM)," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, no. 11, Nov. 2014, doi: 10.11591/telkomnika.v12i11.6652.

[3] H. Ezzikouri, M. Erritali, and M. Oukessou, "Semantic similarity/relatedness for cross language plagiarism detection," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 1, no. 2, p. 371, Feb. 2016, doi: 10.11591/ijeecs.v1.i2.pp371-374.

[4] S. Jain and Renu, "Plagiarism and ethics in research," *Proceedings of DHE approved One Day National Seminar on Role of Digitization during COVID-19*, pp. 241-245, 2021.

[5] A. Ali and A. Taqa, "Analytical study of traditional and intelligent textual plagiarism detection approaches," *Journal of Education and Science*, vol. 31, no. 1, pp. 8–25, Mar. 2022, doi: 10.33899/edusj.2021.131895.1192.

[6] D. Gupta, K. Vani, and C. K. Singh, "Using natural language processing techniques and fuzzy-semantic similarity for automatic external plagiarism detection," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2014, pp. 2694–2699, doi: 10.1109/ICACCI.2014.6968314.

[7] T. Zhang, B. Lee, and Q. Zhu, "Semantic measure of plagiarism using a hierarchical graph model," *Scientometrics*, vol. 121, no. 1, pp. 209–239, Oct. 2019, doi: 10.1007/s11192-019-03204-x.

[8] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 133–149, Mar. 2012, doi: 10.1109/TSMCC.2011.2134847.

[9] M. S. Pera and Y.-K. Ng, "SimPaD: a word-similarity sentence-based plagiarism detection tool on web documents," *Web Intelligence and Agent Systems: An International Journal*, vol. 9, no. 1, pp. 27–41, 2011, doi: 10.3233/WIA-2011-0203.

[10] A. Chitra and A. Rajkumar, "Paraphrase extraction using fuzzy hierarchical clustering," *Applied Soft Computing*, vol. 34, pp. 426–437, Sep. 2015, doi: 10.1016/j.asoc.2015.05.017.

[11] V. K and D. Gupta, "Detection of idea plagiarism using syntax–semantic concept extractions with genetic algorithm," *Expert Systems with Applications*, vol. 73, pp. 11–26, May 2017, doi: 10.1016/j.eswa.2016.12.022.

[12] C. Y. Chang, S. J. Lee, C. H. Wu, C. F. Liu, and C. K. Liu, "Using word semantic concepts for plagiarism detection in text documents," *Information Retrieval Journal*, vol. 24, no. 4–5, pp. 298–321, 2021, doi: 10.1007/s10791-021-09394-4.

[13] A. H. Osman and O. M. Barukub, "Graph-based text representation and matching: a review of the state of the art and future challenges," *IEEE Access*, vol. 8, pp. 87562–87583, 2020, doi: 10.1109/ACCESS.2020.2993191.

[14] T. T. Huynh, T. Phamnguyen, and N. V. Do, "A keyphrase graph-based method for document similarity measurement," *Engineering Letters*, vol. 30, no. 2, pp. 692–710, 2022.

[15] M. M. Kubek and H. Unger, "Centroid terms as text representatives," in *Theory and Application of Text-representing Centroids*, VDI Verlag, 2019, pp. 7–26.

[16] Z. Ceska and C. Fox, "The influence of text pre-processing on plagiarism detection," *International Conference Recent Advances in Natural Language Processing, RANLP*, pp. 55–59, 2009.

[17] X. Liu, "The application of NLTK toolkit based on python in corpus research," *Journal of Kunming Metallurgy College*, vol. 31, no. 5, pp. 65–69, 2015.

[18] M. Wang and F. Hu, "The application of NLTK library for python natural language processing in corpus research," *Theory and Practice in Language Studies*, vol. 11, no. 9, pp. 1041–1049, Sep. 2021, doi: 10.17507/tpls.1109.09.

[19] D. Q. Nguyen, V. Tong, D. Phung, and D. Q. Nguyen, "Node co-occurrence based graph neural networks for knowledge graph link prediction," *WSDM 2022 - Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pp. 1589–1592, 2022, doi: 10.1145/3488560.3502183.

[20] D. Jiang *et al.*, "Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models," *Journal of Cheminformatics*, vol. 13, no. 1, 2021, doi: 10.1186/s13321-020-00479-8.

[21] K. Bijari, H. Zare, E. Kebriaei, and H. Veisi, "Leveraging deep graph-based text representation for sentiment polarity applications," *Expert Systems with Applications*, vol. 144, 2020, doi: 10.1016/j.eswa.2019.113090.

[22] G. Zhang, S. Zhang, and Z. Peng, "Community education knowledge graph based on Co-occurrence analysis and log-likelihood ratio algorithm," *ACM International Conference Proceeding Series*, pp. 53–61, 2022, doi: 10.1145/3571513.3571523.

[23] Y. Chen, J. Wang, P. Li, and P. Guo, "Single document keyword extraction via quantifying higher-order structural features of word co-occurrence graph," *Computer Speech and Language*, vol. 57, pp. 98–107, 2019, doi: 10.1016/j.csl.2019.01.007.

[24] Y. Ruamsuk, W. Tirasopitlert, A. Mingkhwan, and H. Unger, "Medical recommendation system using co-occurrence graphs," *NU. International Journal of Science*, vol. 17, no. 1, pp. 111–119, 2020, doi: 10.14456/nujst.2020.9.

[25] H. Unger and M. Kubek, "On evolving text centroids," in *Advances in Intelligent Systems and Computing*, vol. 769, 2019, pp. 75–82.

[26] M. A. Javaid, "Understanding dijkstra algorithm," *SSRN Electronic Journal*, 2013, doi: 10.2139/ssrn.2340905.

[27] W. Shafqat and Y.-C. Byun, "Incorporating similarity measures to optimize graph convolutional neural networks for product recommendation," *Applied Sciences*, vol. 11, no. 4, p. 1366, Feb. 2021, doi: 10.3390/app11041366.

[28] R. Singh and S. Singh, "Text similarity measures in news articles by vector space model using NLP," *Journal of The Institution of Engineers (India): Series B*, vol. 102, no. 2, pp. 329–338, 2021, doi: 10.1007/s40031-020-00501-5.
[29] S. Chotirat and P. Meesad, "Part-of-Speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning," *Heliyon*, vol. 7, no. 10, 2021, doi: 10.1016/j.heliyon.2021.e08216.
[30] L. Ahuja, V. Gupta, and R. Kumar, "A new hybrid technique for detection of plagiarism from text documents," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, pp. 9939–9952, 2020, doi: 10.1007/s13369-020-04565-9.

BIOGRAPHIES OF AUTHORS

Sureeporn Nualnim received the B.Eng. (computer engineering) degree from the Rajamangala Institute of Technology North Bangkok and the M.Ed.C. (computer technology from King Mongut's Institute of Technology North Bangkok, Thailand. She held several administrative posts, such as the position of the head of the Information Technology Department at the Office of Academic Resources and Information Technology from 2007 to 2011, Rajamangala University of Technology Krungthep. She also served as the deputy director at the Office of Academic Resources and Information Technology from 2011 to 2018, Rajamangala University of Technology Krungthep. She is currently a teacher in the computer science program with the Department of Mathematical Computer Science, Faculty of Science and Technology. Her research interests include information retrieval and natural language processing. She can be contacted at email: sureeporn.n@mail.rmutk.ac.th.

Asst. Prof. Dr. Maleerat Maliyaem is a prominent academic and researcher affiliated with the Faculty of Information Technology and Digital Innovation at King Mongkut's University of Technology North Bangkok (KMUTNB). Her educational background includes a Ph.D. in information technology (international program) from KMUTNB, a Master's degree in computer and information technology from King Mongkut's University of Technology Thonburi, and a Bachelor's degree in computer technology from King Mongkut's Institute of Technology North Bangkok. Her teaching portfolio includes information retrieval systems, decision support systems, natural language processing, machine learning, and data mining courses. Her research interests and contributions are significant in these areas, and she is involved in various academic and professional activities, including her role as Treasurer for the IEEE Computational Intelligence Society (CIS) Thailand Chapter. She can be contacted at email: maleerat.m@itd.kmutnb.ac.th.

Prof. Dr.-Ing. habil. Dr. h.c. Herwig Unger is a distinguished professor at the Faculty of Mathematics and Computer Science, University of Hagen, Germany. He holds a Doctor of Engineering degree (Dr.-Ing.) with habilitation. He has been honored with Doctor honoris causa (Dr. h.c.) for his significant contributions to computer science and engineering. He research areas span distributed systems, network architectures, and information systems. He has made substantial contributions to the development of developing decentralized algorithms, distributed control systems, and the optimization of communication networks. As the head of the Distributed Systems and Computer Networks Research Group at the University of Hagen, he leads several innovative projects to improve network performance, reliability, and security. His research has been widely published in leading journals and conferences. He is also a prolific author and has contributed to several books and numerous scientific papers. His innovative research has led to several patents in network technology and distributed systems. He is an active member of various professional organizations, including the IEEE and ACM, and serves on the editorial boards of several prestigious journals. His research interests include distributed systems, network architectures, information systems, and decentralized algorithms. He can be contacted at email: herwig.unger@fernuni-hagen.de.