applied sciences

MDPI

Article

Evaluation of Different Plagiarism Detection Methods: A Fuzzy MCDM Perspective

Kamal Mansour Jambi [1,*], Imtiaz Hussain Khan [1] and Muazzam Ahmed Siddiqui [2]

[1] Department of Computer Science, King Abdulaziz University, Jeddah 80200, Saudi Arabia; ihkhan@kau.edu.sa

[2] Department of Information System, King Abdulaziz University, Jeddah 80200, Saudi Arabia; maasiddiqui@kau.edu.sa

* Correspondence: kjambi@kau.edu.sa

Abstract: Due to the overall widespread accessibility of electronic materials available on the internet, the availability and usage of computers in education have resulted in a growth in the incidence of plagiarism among students. A growing number of individuals at colleges around the globe appear to be presenting plagiarised papers to their professors for credit, while no specific details are collected of how much was plagiarised previously or how much is plagiarised currently. Supervisors, who are overburdened with huge responsibility, desire a simple way—similar to a litmus test—to rapidly reform plagiarized papers so that they may focus their work on the remaining students. Plagiarism-checking software programs are useful for detecting plagiarism in examinations, projects, publications, and academic research. A number of the latest research findings dedicated to evaluating and comparing plagiarism-checking methods have demonstrated that these have restrictions in identifying the complicated structures of plagiarism, such as extensive paraphrasing as well as the utilization of technical manipulations, such as substituting original text with similar text from foreign alphanumeric characters. Selecting the best reliable and efficient plagiarism-detection method is a challenging task with so many options available nowadays. This paper evaluates the different academic plagiarism-detection methods using the fuzzy MCDM (multi-criteria decision-making) method and provides recommendations for the development of efficient plagiarism-detection systems. A hierarchy of evaluation is discussed, as well as an examination of the most promising plagiarism-detection methods that have the opportunity to resolve the constraints of current state-of-the-art tools. As a result, the study serves as a "blueprint" for constructing the next generation of plagiarism-checking tools.

Keywords: plagiarism detection; semantic analysis; machine learning; fuzzy TOPSIS; text-matching software

check for updates

Citation: Jambi, K.M.; Khan, I.H.; Siddiqui, M.A. Evaluation of Different Plagiarism Detection Methods: A Fuzzy MCDM Perspective. *Appl. Sci.* **2022**, *12*, 4580. https://doi.org/10.3390/app12094580

Academic Editor: Valentino Santucci

Received: 31 March 2022
Accepted: 27 April 2022
Published: 30 April 2022

1. Introduction

The ease with which information may be shared via global interactive communication platforms has motivated literature to explore the internet for information in the chosen manner. This capability has had a detrimental impact in that people seek to authorship their work (written, organized, and formatted documents, presentations, and scripts) by reproducing other people's concepts or research without proper acknowledgment, which has been seen, especially in the academic world. As of now, plagiarism detection is among the highly critical tasks with a multi-focus on academic literature standards and text mining, as well as NLP, with many unresolved concerns including standards and borderline sets. Some are very simple, while others necessitate the use of complex algorithms as well as scientific principles in order to provide the numerous advantages of plagiarism detection [1–8].

Plagiarism by researchers in papers as well as by students in projects is not really a new challenge, but it has been exacerbated by the simplicity of "copying and pasting" information from publications as well as other sources of information on the Web. Utilizing content created by others can be purposeful, but it is more typically an unexpected mistake. With the software applications that are presently available, publishers, educators, examiners, and others can better easily recognize plagiarism as well as do not having to depend on their own talents to recognize similarities to already-published work [9–12]. Plagiarism-checking systems can analyze author documents as well as student papers in a matter of a few minutes, comparing what they have provided to previously published literature. This article explores technology and software and makes some tips for the editor as well as authors on how to use it correctly [13–15].

Selecting the best reliable and efficient plagiarism-detection method can be challenging with so many options available nowadays. Therefore, this paper presents a fuzzy TOPSIS-based method for selecting an efficient academic-plagiarism-detection method. A quantitative research study on various types of plagiarism, indicated over a wide range of plagiarized scripts, revealed that every year, plagiarism checkers have managed to succeed in considering extrinsic plagiarism activities, whereas intrinsic plagiarism purely relies on stylometric functionality investigated by leveraging the arrangement of the papers. An in-depth assessment of the classification of plagiarism-checking methods that centered on textual characteristics, semantic structures, organizational characteristics, and candidate-information extraction prototypes, as well as plagiarism-finding procedures, was conducted. Concept plagiarism appears in the downstream hierarchical structure of smart plagiarism types and cannot be revealed as it is deprived of the textual semantics which convey the concept as well as the localization of perspective in the format.

This study focuses on evaluating the functioning mechanism of different plagiarism-detection methods, a topic that has yet to be thoroughly investigated. The rest of the paper is organized as follows: Section 2 goes through some of the linked publications, including a summary of research projects on the assessment of plagiarism-detection methods. Section 3 outlines the evaluation approach developed for prioritizing various plagiarism-detection methods. Section 4 describes the investigational findings and results. Lastly, Section 5 covers the conclusion, including the societal benefits of the method used and research directions in this field of study.

2. Literature Review

Altheneyan and Menai [15] presented a critical review of existing approaches for paraphrase detection, as well as their use of automated plagiarism findings. It described the classifications of weird occurrences and the basic methodologies, as well as the groups of attributes that each technique utilized. All of the approaches and characteristics employed were reviewed as well as being listed in a table for comparative purposes. The performance of available plagiarism-detection methods capable of detecting paraphrases in benchmark corpora was evaluated and reviewed. Their significant observation was that word overlapping and structural interpretations, as well as MT procedures, are feature subcategories that contribute to the greatest presentation outcomes for a support vector machine (SVM) in both paraphrase recognition as well as plagiarism detection in corpora. A study of the effectiveness of deep learning methods demonstrated that they are the most interesting study direction in this discipline.

Gipp et al. [16] introduced a novel technique named citation-based plagiarism finding, that was tested with the help of a Ph.D. thesis [17], and in which a volunteering crowd-sourcing effort termed GuttenPlag [18] found significant levels of plagiarism through meticulous manual review. They demonstrated that citation-based plagiarism detection outperforms text-based techniques in detecting robust paraphrasing and conversion, as well as certain-idea plagiarism. They showed that merging citation-based, as well as text-based plagiarism identification, can enhance finding capability.

Modiba et al. [19] investigated and compared the differences among several systems, as well as how their effectiveness compared to manual testing. They looked at the various strategies students used to attempt plagiarism. Afterwards they looked more deeply at the technologies that help with plagiarism detection, from their features to how they operate.

During the process, they evaluated how these methods compared to their own, as well as their potential for assisting in the detection of entries that were plagiarized in the beginning C++ course.

Barrón-Cedeo et al. [20] presented a freshly produced large-scale corpus of artificial plagiarism that may be used to evaluate both intrinsic and external plagiarism detection. Furthermore, novel detection performance metrics for evaluating plagiarism-detection algorithms were provided.

Kakkonen and Mozgovoy [21] developed a hierarchical classification of the most popular types of plagiarism found in student papers. Their purpose-built trial set comprised texts that contain examples of various regularly used plagiaristic strategies. Although Sherlock was definitely the greatest hermetic detection tool overall, SafeAssignment was the strongest at identifying web-based plagiarism. Turnitin was discovered to be a highly effective method for identifying semi-automatic types of plagiarism, as the replacement of Cyrillic counterparts for particular letterings or the addition of bogus whitespaces. According to their research, none of the solutions is able to successfully identify plagiarism from both local as well as web sources while also identifying the technical methods that plagiarizers employ to disguise plagiarism.

Jurii et al. [22] introduced a detection approach for source code-based plagiarism depending on the transitional representation as well as demonstrating its application in e-learning. The technique was validated on a sufficient amount of test cases representing the most common code-change strategies. The findings and effectiveness of the method were matched to the contemporary source code-based plagiarism-detection techniques used in a few of the most well-known plagiarism finding tools and systems.

Acampora and Cosma [23] suggested a unique fuzzy-based method to source-code-based plagiarism findings that takes advantage of the popular Fuzzy C-Means (FCM) as well as adaptive neuro-fuzzy inference system methods. Furthermore, the quality of the suggested approach was evaluated by the modern plagiarism identification Running Karp–Rabin Greedy-String-Tiling (RKR-GST) method. Fuzzy C-Means as well as the Adaptive-Neuro Fuzzy Inference System (ANFIS) were the foundations of their suggested algorithm. The suggested technique had the advantage of being programming-language-free, therefore there is no requirement to construct any permitted parsers or translators for the fuzzy-based prediction to recognize in multiple programming languages. Their findings showed that the suggested fuzzy-based technique outperforms all existing methods on popular source code datasets, revealing impressive outcomes as an effective and convenient solution to source-code plagiarism identification.

Ali et al. [24] provided a summary of efficient plagiarism-detection systems that have been utilized for natural-language textual plagiarism-finding, external plagiarism-finding, and clustering-based plagiarism-finding, as well as a few techniques utilized in code-source plagiarism identification. They correspondingly conducted a comparative analysis of five applications utilized for text-plagiarism identification.

Hage et al. [25] examined some plagiarism-detection methods in source-code documents. The tools were evaluated in terms of functionality and effectiveness. They conducted two tests for the performance analysis. To assess the sensitivity of the techniques for different plagiarism tactics, they applied the tools to a sample of purposefully plagiarised programs. To gain a sense of the tools' accuracy, they executed them through numerous incarnations of a student project and evaluated the top ten findings.

According to our literature review, a wide variety of techniques and platforms have been created over the recent decades to encourage robust and efficient plagiarism identification. The most well-known techniques have been capable of addressing the important challenges concerning the retrieval of important syntactic as well as semantic features, managing both monolingual as well as cross-lingual plagiarism identification, and the identifying plagiarism in both textual documents and program-source code even without the use of references.

Moreover, despite the accelerated advancement of technologies to ensure its reproduction, backup, as well as distribution of these techniques and platforms, some major concerns and research directions remain unaddressed. In this segment, we will highlight some of the challenges and opportunities.

- Creating advanced plagiarism-checking techniques that can function without external references while maintaining good precision is a complex job.
- There is still a lack of a detection approach for both textual data and source code which helps to ensure both evidence of accuracy and reliability and proof of thoroughness.
- Another difficult task is the creation of a correct and comprehensive catalogue of references depending on the article writer traces.

According to plagiarism-detection groups, until recent years, many studies were concentrated on verbatim plagiarism, such as word-for-word plagiarism as well as paraphrase plagiarising. Efficient academic plagiarism-detection method selection using the MCDM method was employed in this research for detailed comparison, co-occurrence similarity, language-model likelihood, and dependency-connection matching.

3. Materials and Methods

3.1. Hierarchy for the Evaluation

The initial step in this research was to develop a hierarchy structure. Thorough literature analyses, as well as expert recommendations, was used to determine the most important criteria for the evaluation of different plagiarism-detection methods. The hierarchy structure is made up of five criterion groups: ghostwriting, syntax-preserving, character-preserving, semantics-preserving, and idea-preserving, indicated by P1, P2, P3, P4 and P5, respectively, to select efficient academic plagiarism-detection approaches among different alternatives. Vector space models, stylometry, non-textual feature analysis, n-gram comparisons, LSA, ESA, semantic graph analysis, and machine learning are the six alternatives evaluated for assessment, indicated by A1, A2, A3, A4, A5 and A6, respectively. Table 1 discusses the different criteria used for the evaluation of plagiarism-detection methods. Figure 1 illustrates the hierarchical arrangement for the assessment of different plagiarism-detection methods.

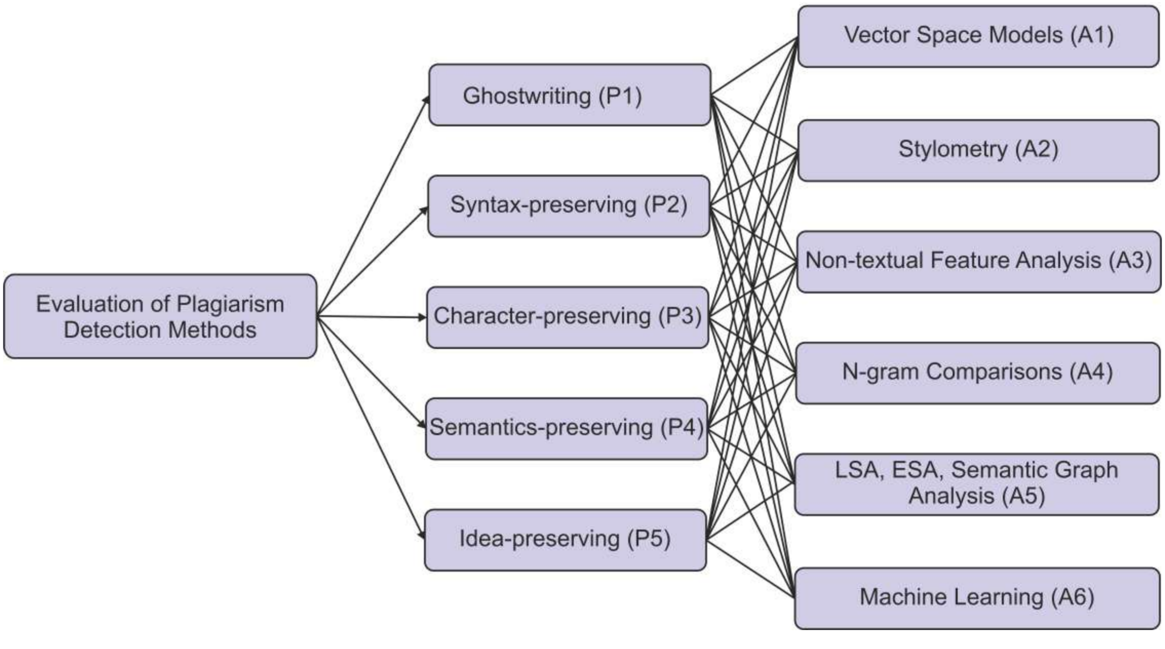
Evaluation of Plagiarism Detection Methods
Ghostwriting (P1)
Syntax-preserving (P2)
Character-preserving (P3)
Semantics-preserving (P4)
Idea-preserving (P5)
Vector Space Models (A1)
Stylometry (A2)
Non-textual Feature Analysis (A3)
N-gram Comparisons (A4)
LSA, ESA, Semantic Graph Analysis (A5)
Machine Learning (A6)

Figure 1. Hierarchy for the evaluation of different plagiarism-detection methods.

After the hierarchical structure has been formed, the factors must be examined in pairs to assess their comparative significance and weight in relation to the overall objective. The research was prepared and carried out in order to compare the various parameters that experts and domain specialists consider when picking an adequate plagiarism-detection system. A questionnaire survey was created to identify the order of importance of the characteristics that should be considered when selecting effective academic plagiarism-detection methods.

The following sub-sections briefly discuss the different academic plagiarism-detection methods.

3.1.1. Vector Space Models

The vector space model (VSM) [26] is a well-known method for classical information retrieval (IR). It has the capacity to resolve the constraints of the string-comparison method. A given document is represented here by one or more vectors. The distance between the respective vectors is determined to assess pair-wise similarity among documents. To evaluate similarity between two provided vectors, a conventional cosine similarity measurement or more advanced similarity algorithms can be utilized. The model's results are highly dependent on evaluation of the individual plagiarism occurrence as well as the parameter setup [27]. The global similarity evaluation of VSMs might occasionally impede system performance.

3.1.2. Stylometry

Stylometry is the assessment of authorial style as well as writing features, is one method of analysis that has the ability to be used in the discovery of contract fraud. In theory, this technique of investigation can distinguish between writers; consequently, if there is a change among real student contributions as well as those accused of being ghostwritten, education institutions may have actionable proof [28]. Since its inception, stylometry has been employed in a variety of applications. In scientific authorship analysis, as in criminal as well as civil lawsuits, stylometry has been used. The approach has also been used by intelligence services to identify the writers of threats and Internet action, as well as other potentially dangerous publications. Stylometry objectives can differ based on the specific consequence desired by an investigator. The goal of authorial identification is to "establish the likelihood that a work was produced by a certain author primarily on stylistic qualities rather than content" [29]. The authorial authentication process comprises a "binary classification issue that determines if two texts were created by the same author".

3.1.3. Non-Textual Feature Analysis

Non-textual information such as citations, photographs, and mathematical equations are taken into account by idea-based detection algorithms. This covers preprocessing techniques as well as similarity metrics that are pertinent to all finding approaches before addressing each class of detection systems. Idea-based finding approaches examine non-textual material in addition to lexical, syntactic, and semantic text similarities. The concept for this family of detection algorithms was first proposed by Gipp and Beel [30]. They advocated looking for comparable patterns in the series of in-textual citations in academic papers. These arrangements can imply that the information of the papers is highly semantically comparable, irrespective of whether the language has been paraphrased or converted.

3.1.4. N-Gram Comparisons

Typically, the main strategy for detecting text similarity is to generate a collection of words from complete text samples. The frequencies matrix is then generated, which means that texts are translated to numerical statements. The findings of such a technique are determined by the filters used when the bag of words is formed. As a consequence, it is critical to choose the appropriate filters in order to obtain correct outcomes. We can use this technique to evaluate entire texts or simply sections of them, such as phrases, paragraphs, sections, or n-grams. Based on the problem, n-grams can be produced at the character as well as word level [31,32]. Similarity findings can be analyzed using many approaches, such as analytical or numerical value estimates, or several clustering algorithms, for example, k-means, artificial neural networks, or Bayesian. [33].

3.1.5. LSA, ESA, Semantic Graph Analysis

Semantics-based approaches work on the assumption that the semantic relationship of two passages is determined by the presence of comparable semantic parts in both sections. The semantic closeness of the two parts is determined by their appearance in similar situations. Thesauri are used in several semantics-based methodologies (e.g., WordNet or EuroVoc7). Incorporating semantic variables such as alternative words or hypernyms, as well as hyponyms into the exploration enhances paraphrase detection [34]. The use of a canonical alternative for every word aids in the detection of synonym-replacement obfuscation as well as minimizing the vector space dimension [35]. All semantics-based identification methods rely on sentence segmentation as well as text tokenization. Tokenization removes the analysis's atomic elements, which are often words or sentences. For extrinsic plagiarism identification, a common, as well as a successful technique, is to use proven semantic text analysis tools such as latent semantic analysis (LSA) or explicit semantic analysis (ESA), as well as word embeddings.

This series of approaches is centered on the concept of "distributional semantics", which shows that terms that occur in comparable circumstances tend to communicate a comparable meaning. In the opposite direction, distributional semantics implies that comparable term patterns reflect semantically related texts. The approaches differ in the breadth of co-occurring terms that they examine. Word embeddings examine only the terms that are directly surrounding them, whereas LSA analyzes the full document and ESA makes use of an external corpus. Knowledge graph analysis (KGA) visualizes a document for example a weighted directed graph, with nodes representing the semantic concepts provided by the text's words as well as edges representing the relationships among these ideas [36].

3.1.6. Machine Learning

Many common machine-learning methods, for example, support vector machines (SVM), nearest neighbor, as well as artificial neural networks (ANN), necessitate a large number of 'labeled' or 'annotated' data with the intention of developing accurate models. Labels are essential for a pair of programs, particularly in our plagiarism-detection assignment. Pairwise labels are the names given to these labels. Enumerating as well as annotating all possible combinations from individual data points is both costly and complex. Active learning can considerably lessen the load of annotating information needed for model training. In the actual world, unbalanced datasets are frequent, with the number of instances of one class considerably outnumbering the other. In such circumstances, rather than considering the issue as a classification problem, it might be treated as an anomaly detection problem that can be solved [37].

Table 1. Different criteria for the evaluation.

Criteria	Description
Ghostwriting	When it comes to plagiarism as well as attribution, ghostwriting is among the most contentious and polarizing ethical concerns. It is an action that is commonly accepted in some circles while being viewed as a major ethical blunder in others. At the moment, the sole technical method for detecting potential ghostwriting is to match stylometric aspects of a maybe ghost-written paper with textual documents clearly written by the putative writer [38–40].
Syntax-preserving	Syntax-preserving plagiarism is frequently the consequence of applying basic substitution methods, such as pattern-matching. Basic synonym replacement procedures work in the same way; but, using more advanced substitution techniques has become more common [41].
Character-preserving	Characters-preserving plagiarism comprises, in addition to verbatim copying, plagiarism methods that indicate sources, such as "pawn sacrifice" and "cut and slide" [42].

Table 1. *Cont.*

Criteria	Description
Semantics-preserving	Semantics-preserving plagiarism relates to advanced kinds of obfuscation in which the words, as well as sentence structure, are changed while the sense of passages is preserved [43].
Idea-preserving	Idea-preserving plagiarism refers to instances wherein plagiarists utilize a source's concept or structure as well as express it wholly in their own terms. This type of plagiarism is challenging to detect and even more difficult to demonstrate [44].

3.2. Fuzzy-TOPSIS Method

Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) is one of many popular MCDM methods. It is a practicable and helpful method for rating and choosing a figure of viable options by calculating Euclidean distances. The TOPSIS method was created by Hwang and Yoon [45]. It is centered on the notion that the optimal result must be the closest to the positive ideal solution (PIS), which is the solution that maximizes the advantage criteria while minimizing the cost criteria, as well as the furthest away from the negative ideal solution (NIS), which is the solution that maximizes the cost criteria while minimizing the advantage criteria.

Weights of qualities in standard multiple criteria decision-making (MCDM) systems show comparative relevance in the decision-making procedure. We cannot suppose that every assessment criterion is of equivalent relevance because the assessment of criteria involves varied viewpoints and interpretations [46]. Weighting techniques are divided into two types: subjective approaches as well as objective approaches. Subjective approaches are used to generate weights primarily based on decision makers' preferences or judgments. Furthermore, using mathematical approaches, for example, the eigenvector technique or the weighted least square technique, as well as computational programming approaches, compute the total evaluation of every decision-maker. The probability technique, multiple objective programming, and other objective methods calculate weights by performing mathematical models systematically without regard for the decision maker's opinions. To demonstrate that the weighting techniques have an effect on the assessment outcome, both subjective and objective weighting techniques were used in the comparison. Subjective weighting is focused on the knowledge and judgment of the decision-maker, whereas objective weighing is founded on mathematical computations. The objective weighting approach is especially useful in cases when valid subjective weights cannot be achieved [47–50].

As shown in Figure 2, the step-by-step sequential approach for weighting calculation, as well as significance-rating with the support of fuzzy TOPSIS, is as follows:

Step 1: Construct a decision matrix

In this research, five criteria and six alternatives are evaluated using the fuzzy TOPSIS approach. The category represents the types of different criteria. Suppose the decision-making team has K participants. If the fuzzy score as well as priority weight of the k-th assessment expert, about the i-th alternative on j-th criterion, are:

$$\check{x}_{ij}^{k} = \left(a_{ij}^{k}, b_{ij}^{k}, c_{ij}^{k}\right) \text{ and } \check{w}_{j}^{k} = \left(w_{j1}^{k}, w_{j2}^{k}, w_{j3}^{k}\right) \text{ correspondingly}$$

where if i = 1, 2, . . . , m, and j = 1, 2, . . . , n, then the aggregated fuzzy ratings $\check{x}_{ij}$ of alternatives (i) with regard to each criterion (j) are specified by $\check{x}_{ij} = (a_{ij}, b_{ij}, c_{ij})$. Table 2 displays the category of criterion as well as the weight applied to every criterion.

A triangular fuzzy number (TFN) is represented as L, M, or U. The parameters L, M, and U depict the least likely, most likely, and highest possible values, respectively. Table 3 demonstrates the fuzzy scale utilized in the model.

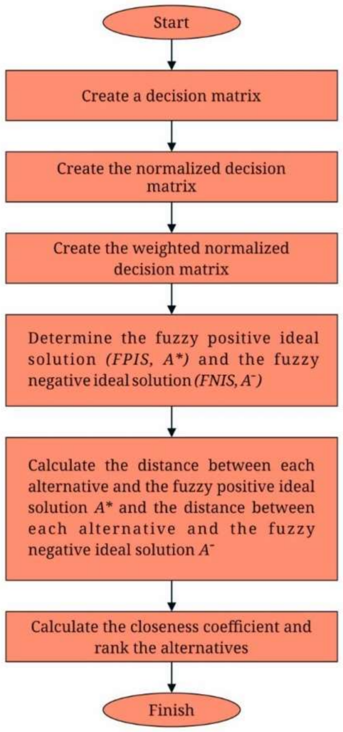
Start
Create a decision matrix
Create the normalized decision matrix
Create the weighted normalized decision matrix
Determine the fuzzy positive ideal solution *(FPIS, A*)* and the fuzzy negative ideal solution *(FNIS, A⁻)*
Calculate the distance between each alternative and the fuzzy positive ideal solution *A** and the distance between each alternative and the fuzzy negative ideal solution *A⁻*
Calculate the closeness coefficient and rank the alternatives
Finish

Figure 2. Flow diagram of fuzzy TOPSIS method.

Table 2. Characteristics of different criteria.

	Criteria	Category	Weight
1	P1	+	(0.200, 0.200, 0.200)
2	P2	+	(0.200, 0.200, 0.200)
3	P3	+	(0.200, 0.200, 0.200)
4	P4	+	(0.200, 0.200, 0.200)
5	P5	+	(0.200, 0.200, 0.200)

Table 3. Fuzzy scale.

Code	Linguistic Terms	L	M	U
1	Very low	1	1	3
2	Low	1	3	5
3	Medium	3	5	7
4	High	5	7	9
5	Very high	7	9	9

Step 2: Construct the normalized decision matrix

Further, a normalized decision matrix may be determined by using the subsequent relation depending on the positive as well as negative ideal options:

$$\widetilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*}\right);\ c_j^* = \max_i c_{ij};\ \text{Positive ideal solution} \tag{1}$$

$$\widetilde{r}_{ij} = \left(\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{a_{ij}}\right);\ a_j^- = \min_i a_{ij};\ \text{Negative ideal solution} \tag{2}$$

Step 3: Create the weighted normalized decision matrix

The weighted normalised decision matrix may be generated by multiplying the weight of every criterion in the normalised fuzzy decision matrix using the equation as follows, taking into account the varying weights of every criterion.

$$\tilde{v}_{ij} = \tilde{r}_{ij} \cdot \tilde{w}_{ij} \quad (3)$$

where $\tilde{w}_{ij}$ denotes weight of criterion c_j.

Step 4: Calculate the fuzzy positive ideal solution (FPIS, A*) as well as the fuzzy negative ideal solution (FNIS, A^-)

The FPIS and FNIS of the alternatives may be well-defined as Equations (4) and (5), respectively:

$$A^* = \{\tilde{v}_1^*, \tilde{v}_2^*, \ldots, \tilde{v}_n^*\} = \left\{\left(\max_j v_{ij} | i \in B\right), \left(\min_j v_{ij} | i \in C\right)\right\} \quad (4)$$

$$A^- = \{\tilde{v}_1^-, \tilde{v}_2^-, \ldots, \tilde{v}_n^-\} = \left\{\left(\min_j v_{ij} | i \in B\right), \left(\max_j v_{ij} | i \in C\right)\right\} \quad (5)$$

where $\tilde{v}_i^*$ is the maximum amount of i for all the alternatives and $\tilde{v}_1^-$ is the minimum amount of i for all the alternatives. B and C symbolize the corresponding positive and negative ideal solutions, respectively.

Step 5: Determine the gap between every alternative as well as the fuzzy positive ideal solution A*, as well as the distance among every interim solution and the fuzzy negative ideal solution A^-.

The gap among each alternative as well as FPIS and among every alternative as well as FNIS is computed using Equations (6) and (7), respectively:

$$S_i^* = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}_j^*) \; i = 1, 2, \ldots, m \quad (6)$$

$$S_i^- = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}_j^-) \; i = 1, 2, \ldots, m \quad (7)$$

where d is the distance among two fuzzy numerals, when two triangular fuzzy numbers (a_1, b_1, c_1) and (a_2, b_2, c_2) are assumed, e distance among the two can be estimated as follows:

$$d_v\left(\tilde{M}_1, \tilde{M}_2\right) = \sqrt{\frac{1}{3}\left[(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2\right]} \quad (8)$$

Note that $d\left(\tilde{v}_{ij}, \tilde{v}_j^*\right)$ and $d\left(\tilde{v}_{ij}, \tilde{v}_j^-\right)$ are crisp numbers.

Step 6: Determine the proximity coefficient as well as rank the options.

Every alternative's proximity coefficient could be calculated by the following equation:

$$CC_i = \frac{S_i^-}{S_i^+ + S_i^-} \quad (9)$$

There are numerous approaches available for evaluating as well as ranking alternates with a different set of criteria. Every approach has benefits and drawbacks over the others. The fuzzy TOPSIS approach has the merits of being simple in its mathematical formulation and convenient for demonstrating human priorities, as well as allowing clear and direct trade among multiple criteria [50,51]. Furthermore, the tactic is categorized as a compromising concept, with the concept that while no optimal situation persists, a solution with optimized values on all criteria is achievable. As a result, fuzzy TOPSIS with a triangular membership value is utilized in this research study to evaluate different plagiarism methods.

4. Results

The researchers calculated the data using the regular fuzzy scale (shown in Table 3) and Equations (1)–(9). The solutions are examined in terms of numerous criteria, as well as the decision matrix findings are presented below. The choice matrix presented in Table 4 provides the arithmetic mean of all 50 expert opinions. Table 5 shows the normalized decision matrix. Table 6 shows the weighted normalized decision matrix. Further, Table 7 shows the positive and negative ideal solutions. Furthermore, Table 8 shows distance from positive and negative ideal solutions.

Table 4. Decision matrix.

	P1	P2	P3	P4	P5
A1	(4.440, 6.440, 8.440)	(4.440, 6.440, 7.720)	(4.800, 6.800, 8.000)	(4.080, 6.080, 7.480)	(3.840, 5.800, 7.360)
A2	(4.040, 6.000, 7.560)	(4.320, 6.320, 7.720)	(4.000, 6.000, 7.440)	(4.160, 6.160, 7.600)	(3.920, 5.920, 7.520)
A3	(4.040, 6.000, 7.760)	(4.560, 6.560, 7.920)	(4.360, 6.280, 7.720)	(4.480, 6.480, 7.760)	(4.400, 6.360, 7.760)
A4	(3.880, 5.800, 7.520)	(4.160, 6.160, 7.800)	(4.120, 6.120, 7.720)	(4.320, 6.320, 7.800)	(4.320, 6.320, 7.960)
A5	(4.320, 6.320, 7.840)	(4.200, 6.200, 7.560)	(4.320, 6.320, 7.880)	(4.000, 6.000, 7.480)	(4.440, 6.440, 8.120)
A6	(4.560, 6.520, 8.000)	(4.560, 6.560, 7.920)	(4.760, 6.760, 7.880)	(4.840, 6.840, 8.000)	(4.640, 6.640, 7.880)

Table 5. A normalized decision matrix.

	P1	P2	P3	P4	P5
A1	(0.526, 0.763, 1.000)	(0.561, 0.813, 0.975)	(0.600, 0.850, 1.000)	(0.510, 0.760, 0.935)	(0.473, 0.714, 0.906)
A2	(0.479, 0.711, 0.896)	(0.545, 0.798, 0.975)	(0.500, 0.750, 0.930)	(0.520, 0.770, 0.950)	(0.483, 0.729, 0.926)
A3	(0.479, 0.711, 0.919)	(0.576, 0.828, 1.000)	(0.545, 0.785, 0.965)	(0.560, 0.810, 0.970)	(0.542, 0.783, 0.956)
A4	(0.460, 0.687, 0.891)	(0.525, 0.778, 0.985)	(0.515, 0.765, 0.965)	(0.540, 0.790, 0.975)	(0.532, 0.778, 0.980)
A5	(0.512, 0.749, 0.929)	(0.530, 0.783, 0.955)	(0.540, 0.790, 0.985)	(0.500, 0.750, 0.935)	(0.547, 0.793, 1.000)
A6	(0.540, 0.773, 0.948)	(0.576, 0.828, 1.000)	(0.595, 0.845, 0.985)	(0.605, 0.855, 1.000)	(0.571, 0.818, 0.970)

Table 6. The weighted normalized decision matrix.

	P1	P2	P3	P4	P5
A1	(0.105, 0.153, 0.200)	(0.112, 0.163, 0.195)	(0.120, 0.170, 0.200)	(0.102, 0.152, 0.187)	(0.095, 0.143, 0.181)
A2	(0.096, 0.142, 0.179)	(0.109, 0.160, 0.195)	(0.100, 0.150, 0.186)	(0.104, 0.154, 0.190)	(0.097, 0.146, 0.185)
A3	(0.096, 0.142, 0.184)	(0.115, 0.166, 0.200)	(0.109, 0.157, 0.193)	(0.112, 0.162, 0.194)	(0.108, 0.157, 0.191)
A4	(0.092, 0.137, 0.178)	(0.105, 0.156, 0.197)	(0.103, 0.153, 0.193)	(0.108, 0.158, 0.195)	(0.106, 0.156, 0.196)
A5	(0.102, 0.150, 0.186)	(0.106, 0.157, 0.191)	(0.108, 0.158, 0.197)	(0.100, 0.150, 0.187)	(0.109, 0.159, 0.200)
A6	(0.108, 0.155, 0.190)	(0.115, 0.166, 0.200)	(0.119, 0.169, 0.197)	(0.121, 0.171, 0.200)	(0.114, 0.164, 0.194)

Table 7. The positive and negative ideal solutions.

	Positive Ideal	Negative Ideal
P1	(0.108, 0.155, 0.200)	(0.092, 0.137, 0.178)
P2	(0.115, 0.166, 0.200)	(0.105, 0.156, 0.191)
P3	(0.120, 0.170, 0.200)	(0.100, 0.150, 0.186)
P4	(0.121, 0.171, 0.200)	(0.100, 0.150, 0.187)
P5	(0.114, 0.164, 0.200)	(0.095, 0.143, 0.181)

Table 8. Distance from positive and negative ideal solutions.

	Distance from the Positive Ideal	Distance from the Negative Ideal
A1	0.043	0.043
A2	0.071	0.014
A3	0.04	0.046
A4	0.059	0.029
A5	0.051	0.037
A6	0.011	0.078

Based upon the significance value of Ci presented in Table 9 and Figure 3, it is determined that the optimal ranking of the efficient academic plagiarism-detection methods using the fuzzy TOPSIS approach is A6 > A3 > A1 > A5 > A4 > A2 (\>" means \superior to"). So, A6 (machine learning) is considered the preferable academic plagiarism-detection method.

Table 9. Closeness coefficient.

	Ci	Rank
A1	0.503	3
A2	0.167	6
A3	0.534	2
A4	0.332	5
A5	0.419	4
A6	0.874	1

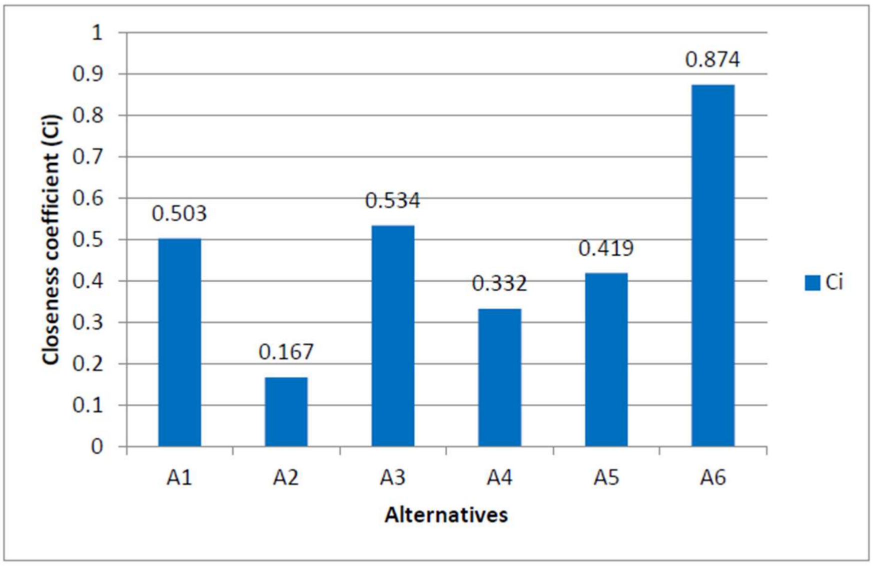

Figure 3. Closeness coefficient graph.

5. Discussion

Plagiarism has become more prevalent in the age of information technology and has become a major concern. Plagiarism is still a problem in publishing as well as research, and there are software systems available to analyze similarities to originally published literature, submitted earlier student papers, theses, and dissertations, including webpages. These free or paid software packages can assist new authors to discover cases of 'unintended' plagiarism, as well as editors and academic staff, to detect all sorts of plagiarism. When utilizing any plagiarism-checking application, the user should be careful to carefully analyze the significant findings. A similarity measure or index does not show plagiarism on its own because properly referenced sources can be reported as 'similar.' Furthermore, some plagiarism-checking sites keep ownership of the draught entries in the system and use them for later similarity assessments. While software programs can be exceedingly useful in identifying and counseling writers about plagiarism, specialists suggest that before utilizing any plagiarism identification tool, editors, teaching staff, and researchers, as well as students obtain appropriate training on the use of the tool, final preservation of reports submitted, significance of findings, and implications.

The consequences of being exposed to plagiarising are extremely serious. If a person is detected plagiarising, the immediate consequence is that the individual has shown to be dishonest and unethical. These characteristics are not only directly damaging to their academic progress, but they may also have a negative impact on their ability to obtain a high-paying job at some point. A prospective employer sees little value in hiring someone who has previously proven himself to be dishonest and manipulative before they have ever worked a shift for the firm.

This research study discusses methods for reducing plagiarism issues and also evaluates some of the most common academic plagiarism-detection methods. The study also highlights the most efficient method for plagiarism detection, which is a machine-learning-based approach. The selection of features is undoubtedly the most significant component of machine-learning mechanisms. A variety of machine-learning methods have been created and trained with the help of computational features. The artificial neural network was discovered to be the most impactful machine-learning technique for making assumptions. The advent of machine learning as well as the strong credibility that its algorithms have earned, particularly in the domain of plagiarism detection, is another motivating element for tactic integration application. Plagiarism-prevention strategies based on a shift in society's attitude toward plagiarism are beyond a doubt the most effective means of combating plagiarism, but putting these approaches into practice is a difficulty for society overall. Plagiarism-detection methods must be prioritized by academic institutions. Due to their convenience and ease of implementation in tools, distinct statistical measures are commonly used in extensively utilized plagiarism-detection methods, according to an analysis of extensively employed plagiarism-detection methods.

According to an assessment of recognized plagiarism-detection tools, even though these methods provide outstanding service in finding corresponding text among records, even enhanced plagiarism finding applications cannot identify plagiarism, as well as humans, can. They have a number of flaws, thus manual inspection and human judgment are still required. The human mind is a ubiquitous plagiarism-detection tool that can assess documents using statistical as well as semantical methods and can work with both textual and non-textual data. Such capabilities are now available in plagiarism-detection software programs using machine-learning technology.

6. Conclusions

This paper provides an outline of the tools as well as strategies used to detect text plagiarism. It makes an attempt to offer some insight into the recent advancement in this subject, including approaches used and tools. Some of the popular academic plagiarism-detection methods have been studied and analyzed, highlighting the key issues with these tools as well as the areas for further research. For identifying plagiarised writings, NLP is used in pre-processing phases such as sentence segmentation, tokenization, stop-word elimination, punctuation removal, synonym substitution, stemming, and numeral replacement. The accuracy, as well as effectiveness, of the plagiarism-detection system, is improved by these pre-processing procedures. The findings presented in this paper show that the machine-learning (ML) method for plagiarism detection can be used significantly and efficiently. There are some studies being conducted to perform this task utilizing ML and neural networks, as well as deep learning. Intelligent strategies for detecting high obfuscations are still in their early stages, and the majority of accessible online or stand-alone, as well as web-based applications, fail to identify intricate manipulations. Therefore, this study sheds light on the vast research potential in this subject for building effective smart detection systems to combat these inappropriate practices.

Author Contributions: Conceptualization, K.M.J., I.H.K. and M.A.S.; methodology, K.M.J., I.H.K. and M.A.S.; software, K.M.J., I.H.K. and M.A.S.; validation, K.M.J., I.H.K. and M.A.S.; formal analysis, K.M.J., I.H.K. and M.A.S.; investigation, K.M.J., I.H.K. and M.A.S.; resources, K.M.J., I.H.K. and M.A.S.; data curation, K.M.J., I.H.K. and M.A.S.; writing—original draft preparation, K.M.J., I.H.K. and M.A.S.; writing—review and editing, K.M.J., I.H.K. and M.A.S.; visualization, K.M.J., I.H.K. and M.A.S.; supervision, K.M.J., I.H.K. and M.A.S.; project administration, K.M.J. and I.H.K.; funding acquisition, K.M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a King Abdulaziz City of Science and Technology (KACST) funding: Grant No. 38-597(11-INF1520-03). We thank KACST for their financial support.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not available.

Acknowledgments: The authors thank Science and Technology Unit, King Abdulaziz University for the technical support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, X.; Francia, B.; Li, M.; Mckinnon, B.; Seker, A. Shared information and program plagiarism detection. *IEEE Trans. Inf. Theory* **2004**, *50*, 1545–1551. [CrossRef]
2. Lancaster, T.; Culwin, F. A comparison of source code plagiarism detection engines. *Comput. Sci. Educ.* **2004**, *14*, 101–112. [CrossRef]
3. Potthast, M.; Stein, B.; Barrón-Cedeño, A.; Rosso, P. An evaluation framework for plagiarism detection. In *Coling 2010: Posters*; Coling 2010 Organizing Committee: Beijing, China, 2010; pp. 997–1005.
4. Lukashenko, R.; Graudina, V.; Grundspenkis, J. Computer-based plagiarism detection methods and tools: An overview. In Proceedings of the 2007 International Conference on Computer Systems and Technologies, Ruse, Bulgaria, 14–15 June 2007; pp. 1–6.
5. Ansari, M.T.J.; Pandey, D.; Alenezi, M. STORE: Security threat oriented requirements engineering methodology. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 191–203. [CrossRef]
6. Gupta, D. Study on Extrinsic Text Plagiarism Detection Techniques and Tools. *J. Eng. Sci. Technol. Rev.* **2016**, *9*, 8–22.
7. Ansari, M.T.J.; Baz, A.; Alhakami, H.; Alhakami, W.; Kumar, R.; Khan, R.A. P-STORE: Extension of STORE methodology to elicit privacy requirements. *Arab. J. Sci. Eng.* **2021**, *46*, 8287–8310. [CrossRef]
8. Donaldson, J.L.; Lancaster, A.M.; Sposato, P.H. A plagiarism detection system. In Proceedings of the Twelfth SIGCSE Technical Symposium on Computer Science Education, St. Louis, MO, USA, 26–27 February 1981; pp. 21–25.
9. Parker, A.; Hamblen, J.O. Computer algorithms for plagiarism detection. *IEEE Trans. Educ.* **1989**, *32*, 94–99. [CrossRef]
10. Eissen, S.M.Z.; Stein, B. Intrinsic plagiarism detection. In *European Conference on Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 565–569.
11. Si, A.; Leong, H.V.; Lau, R.W. Check: A document plagiarism detection system. In Proceedings of the 1997 ACM Symposium on Applied Computing, San Jose, CA, USA, 28 February–1 March 1997; pp. 70–77.
12. Potthast, M.; Barrón-Cedeno, A.; Stein, B.; Rosso, P. Cross-language plagiarism detection. *Lang. Resour. Eval.* **2011**, *45*, 45–62. [CrossRef]
13. Sorokina, D.; Gehrke, J.; Warner, S.; Ginsparg, P. Plagiarism detection in arXiv. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 1070–1075.
14. Ansari, M.T.J.; Pandey, D. Risks, security, and privacy for HIV/AIDS data: Big data perspective. In *Big Data Analytics in HIV/AIDS Research*; IGI Global: Hershey, PA, USA, 2018; pp. 117–139.
15. Altheneyan, A.; Menai, M.E.B. Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2053004. [CrossRef]
16. Gipp, B.; Meuschke, N.; Beel, J. Comparative evaluation of text-and citation-based plagiarism detection approaches using guttenplag. In Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries, Ottawa, ON, Canada, 13–17 June 2011; pp. 255–258.
17. zu Guttenberg, K.T. *Verfassung und Verfassungsvertrag: Konstitutionelle Entwicklungsstufen in den USA und der EU*; Duncker & Humblot: Berlin, Germany, 2009.

18. Guttenplag, W. Online Resource. 2011. Available online: http://de.guttenplag.wikia.com (accessed on 12 March 2022).
19. Modiba, P.; Pieterse, V.; Haskins, B. Evaluating plagiarism detection software for introductory programming assignments. In Proceedings of the Computer Science Education Research Conference, Pretoria, South Africa, 4–6 July 2016; pp. 37–46.
20. Barrón-Cedeño, A.; Potthast, M.; Rosso, P.; Stein, B. Corpus and evaluation measures for automatic plagiarism detection. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 17–23 May 2010.
21. Kakkonen, T.; Mozgovoy, M. Hermetic and web plagiarism detection systems for student essays—An evaluation of the state-of-the-art. *J. Educ. Comput. Res.* **2010**, *42*, 135–159. [CrossRef]
22. Juričić, V.; Jurić, T.; Tkalec, M. Performance evaluation of plagiarism detection method based on the intermediate language. In Proceedings of the 3rd International Conference "The Future of Information Sciences: INFuture2011–Information Sciences and e-Society", Zagreb, Croatia, 9–11 November 2011.
23. Acampora, G.; Cosma, G. A Fuzzy-based approach to programming language independent source-code plagiarism detection. In Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, 2–5 August 2015; pp. 1–8.
24. Ali, A.M.E.T.; Abdulla, H.M.D.; Snasel, V. Overview and comparison of plagiarism detection tools. In *CEUR Workshop Proceedings*; VŠB-Technical University of Ostrava: Písek, Czech Republic, 2011; pp. 161–172.
25. Hage, J.; Rademaker, P.; Van Vugt, N. *A Comparison of Plagiarism Detection Tools*; Utrecht University: Utrecht, The Netherlands, 2010; Volume 28.
26. Kasprzak, J.; Brandejs, M. Improving the Reliability of the Plagiarism Detection System. Lab Report for PAN at CLEF 2010. pp. 359–366. Available online: http://ceur-ws.org/Vol-1176/CLEF2010wn-PAN-KasprzakEt2010.pdf (accessed on 15 March 2022).
27. Gutbrod, M.A. Nachhaltiges E-Learning Durch Sekundäre Dienste. Ph.D. Thesis, Technische Universität Braunschweig, Braunschweig, Germany, 2007.
28. Juola, P. Detecting contract cheating via stylometric methods. In Proceedings of the Conference on Plagiarism across Europe and Beyond, Brno, Czech Republic, 24–26 May 2017; pp. 187–198.
29. Neal, T.; Sundararajan, K.; Fatima, A.; Yan, Y.; Xiang, Y.; Woodard, D. Surveying stylometry techniques and applications. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–36. [CrossRef]
30. Gipp, B.; Beel, J. Citation based plagiarism detection: A new approach to identify plagiarized work language independently. In Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto, ON, Canada, 13–16 June 2010; pp. 273–274.
31. Kanaris, I.; Kanaris, K.; Houvardas, I.; Stamatatos, E. Words versus character n-grams for anti-spam filtering. *Int. J. Artif. Intell. Tools* **2007**, *16*, 1047–1067. [CrossRef]
32. Lopez-Gazpio, I.; Maritxalar, M.; Lapata, M.; Agirre, E. Word n-gram attention models for sentence similarity and inference. *Expert Syst. Appl.* **2019**, *132*, 1–11. [CrossRef]
33. Bao, J.; Lyon, C.; Lane, P.C.; Ji, W.; Malcolm, J. Comparing Different Text Similarity Methods. 2007. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.578.841&rep=rep1&type=pdf (accessed on 15 March 2022).
34. Mohammad, A.S.; Jaradat, Z.; Mahmoud, A.A.; Jararweh, Y. Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features. *Inf. Process. Manag.* **2017**, *53*, 640–652.
35. Shima, R.; Esfahani, F.; Hamid, R. A Persian fuzzy plagiarism detection approach. *J. Inf. Syst. Telecommun.* **2015**, *3*, 182–190.
36. Fersini, E.; Messina, E.; Pozzi, F.A. Expressive signals in social media languages to improve polarity detection. *Inf. Process. Manag.* **2016**, *52*, 20–35. [CrossRef]
37. Katta, J.Y.B. Machine Learning for Source-Code Plagiarism Detection. Ph.D. Thesis, International Institute of Information Technology Hyderabad, University of Science and Technology, Hyderabad, India, 2018.
38. Chong, M.Y.M. *A Study on Plagiarism Detection and Plagiarism Direction Identification Using Natural Language Processing Techniques*; University of Wolverhampton: Walsall, UK, 2013.
39. Fusch, P.I.; Ness, L.R.; Booker, J.M.; Fusch, G.E. The ethical implications of plagiarism and ghostwriting in an open society. *J. Soc. Change* **2017**, *9*, 4. [CrossRef]
40. Hourrane, O.; Benlahmar, E.H. Survey of plagiarism detection approaches and big data techniques related to plagiarism candidate retrieval. In Proceedings of the 2nd International Conference on Big Data, Cloud and Applications, Tetouan, Morocco, 29–30 March 2017; pp. 1–6.
41. Meuschke, N. Analyzing Non-Textual Content Elements to Detect Academic Plagiarism. *arXiv* **2021**, arXiv:2106.05764.
42. Weber-Wulff, D. *False Feathers: A Perspective on Academic Plagiarism*; Springer Science & Business: Berlin/Heidelberg, Germany, 2014.
43. Jhi, Y.C.; Wang, X.; Jia, X.; Zhu, S.; Liu, P.; Wu, D. Value-based program characterization and its application to software plagiarism detection. In Proceedings of the 33rd International Conference on Software Engineering, Honolulu, HI, USA, 21–28 May 2011; pp. 756–765.
44. Foltýnek, T.; Meuschke, N.; Gipp, B. Academic plagiarism detection: A systematic literature review. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–42. [CrossRef]
45. Hwang, C.L.; Yoon, K. Methods for multiple attribute decision making. In *Multiple Attribute Decision Making*; Springer: Berlin/Heidelberg, Germany, 1981; pp. 58–191.
46. Chen, M.F.; Tzeng, G.H.; Ding, C.G. Fuzzy MCDM approach to select service provider. *IEEE Int. Conf. Fuzzy Syst.* **2003**, *1*, 572–577.

47. Ansari, M.T.J.; Al-Zahrani, F.A.; Pandey, D.; Agrawal, A. A fuzzy TOPSIS based analysis toward selection of effective security requirements engineering approach for trustworthy healthcare software development. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 236. [CrossRef]
48. Deng, H.; Yeh, C.H.; Willis, R.J. Inter-company comparison using modified TOPSIS with objective weights. *Comput. Oper. Res.* **2000**, *27*, 963–973. [CrossRef]
49. Alhakami, W.; Binmahfoudh, A.; Baz, A.; Alhakami, H.; Ansari, M.T.J.; Khan, R.A. Atrocious impinging of COVID-19 pandemic on software development industries. *Comput. Syst. Sci. Eng.* **2021**, *36*, 323–338. [CrossRef]
50. Ansari, M.T.J.; Agrawal, A.; Khan, R.A. *DURASec: Durable Security Blueprints for Web-Applications Empowering Digital India Initiative*; EAI Endorsed Transactions on Scalable Information Systems: Gent, Belgium, 2022.
51. Kannan, D.; Khodaverdi, R.; Olfat, L.; Jafarian, A.; Diabat, A. Integrated fuzzy multi criteria decision making method and multi-objective programming approach for supplier selection and order allocation in a green supply chain. *J. Clean. Prod.* **2013**, *47*, 355–367. [CrossRef]