Taylor & Francis
Taylor & Francis Group

Accountability in Research
Policies and Quality Assurance

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gacr20

# Weighted semantic plagiarism detection approach based on AHP decision model

SeyyedMohammad JavadiMoghaddam, Fatemeh Roosta & Asadolla Noroozi

**To cite this article:** SeyyedMohammad JavadiMoghaddam, Fatemeh Roosta & Asadolla Noroozi (2021): Weighted semantic plagiarism detection approach based on AHP decision model, Accountability in Research, DOI: 10.1080/08989621.2021.1911654

**To link to this article:** https://doi.org/10.1080/08989621.2021.1911654

ACCOUNTABILITY IN RESEARCH, 2021
https://doi.org/10.1080/08989621.2021.1911654

Taylor & Francis
Taylor & Francis Group



# Weighted semantic plagiarism detection approach based on AHP decision model

SeyyedMohammad JavadiMoghaddam [a], Fatemeh Roosta[b], and Asadolla Noroozi[c]

[a]Department of Computer Engineering, Bozorgmehr University of Qaenat, Qaen, Iran; [b]Department of Computer Engineering, AZAD University of Birjand, Birjand, Iran; [c]Civil Engineering, CA, USA

**ABSTRACT**

The increasing rate of academic plagiarism is a social problem that engages institutions and publishers. Plagiarists try to mislead the plagiarism detection system using synonyms and inverted word order. Numerous algorithms tried to overcome these problems using structural and semantic detection. However, most of them focus on overcoming some challenges. Moreover, all of them consider the same significant degree for all terms of the documents. On the other hand, the time complexity is an essential parameter that must be considered. This paper presents an effective way to detect structural and semantic similarity degrees among two papers only using some part of the paper's content instead of all content, decreasing the time complexity. The similarity is calculated using a set of impressive terms and various combinations to augment plagiarism detection ability even if the word order is changed. Different weight is assigned to the word according to its position in various sections of the paper. Finally, an AHP (Analytical Hierarchy Process) model uses to calculate a weighted similarity. The results indicated that the proposed approach has more ability to detect semantic academic plagiarism, and the runtime is reduced compared to similar ones.

**KEYWORDS**

Text similarity; plagiarism detection; semantic plagiarism; AHP model; WordNet

## Introduction

Vast electronic texts on the web and convenient access to them have made plagiarism a significant issue. Plagiarism is the name given for using somebody else's text, idea, and thought without acknowledging the owner (Bahadori, Izadi, and Hoseinpourfard 2012). Even if the author has used sources that he/she has already published (Bruton 2014). Therefore, plagiarism detection has become a hot topic in the research area (Ohno, Yamasaki, and Tokiwa 2020). Discovering plagiarism cases based on text similarity scores seems like a straightforward and quick approach. However, the significant challenges are when the words are replaced with their synonyms while holding their paper positions (Fisher and Partin 2014).

**CONTACT** SeyyedMohammad JavadiMoghaddam smjavadim@buqaen.ac.ir. Department of Computer Engineering, Bozorgmehr University of Qaenat, Qaen, Iran

Many researchers attempt to find a way to calculate the degree of similarity between two texts. Authors (Gipp and Beel 2010; Soleman and Fujii 2018) proposed a citation analysis method. This method calculated plagiarism only based on the bibliographic coupling. Likewise, Hourrane (Hourrane et al. 2018) suggested a citation-based method using deep learning. Some authors found the words with the most frequent and their synonyms in documents of a database and a suspicious document (Ragini and Rubesh Anand 2016). Another one has used the graph model of sentences and WordNet database to find similar graphs between two sample texts(Khadilkar, Kulkarni, and Bone 2018) that computes the similarity semantically. With inspiration from the knowledge graph, this approach detected plagiarism by making the graph of each sentence and the equivalent graphs. If two graphs were similar, it was considered a plagiarism instance. The authors (Ragini and Rubesh Anand 2016) presented a semantic plagiarism method based on a synonym and specific domain. The method's main objective was to detect semantic plagiarism even when someone replaces the synonyms of words with the original ones. However, they only focused on the proposed approach section of two documents to determine the similarity. Authors (Eisa, Salim, and Abdelmaboud 2019) proposed a method for semantic plagiarism based on both synonyms and antonym words. This method focuses on finding figure similarity using image processing approaches. Consequently, the mentioned approaches cannot detect plagiarism if the words' order is changed.

Meuschke (Meuschke et al. 2017) suggested an approach based on semantic similarity and frequency indexing. This model considers semantic sequence scoring using a heuristic technique. Then, the similarity is computed regarding the frequency of the term and semantic rank. This method considers none of the time complexity and the position of the word in the document. Roman (Isele 2018; Meuschke et al. 2019) focused on mathematical content similarity and frequent mathematics use. The method presented in (Meuschke et al. 2018) described a hybrid model based on mathematical expressions, text, images, and citation. This work considers the same important degree for all terms.

What is evident in all the researches done in the domain of plagiarism is that they considered the same importance degree for all matched parts of the documents. On the other hand, most of them only focus on one or two features to detect plagiarism, such as the terms and their synonyms using WordNet. Moreover, they consider the terms individually. These approaches cannot detect plagiarism when the words' order is changed. While considering two or three sequences of the adjacent word can be more impressive than individual terms to detect plagiarism. This paper proposes a new approach that considers a weighted value for matched terms based on the section (such as title, abstract, etc.). Moreover, some key terms are applied to calculate the similarity of two documents instead of all of them, reducing the time complexity. Likewise, it considers a type

for the matched instances, including 1-, 2- and 3-terms of the core terms, which are the number of adjacent words, which causes the ability to detect plagiarism if the words' order is changed. Finally, a multivariate model for decision-making named AHP forms the foundation of this approach to find a suitable similarity.

Therefore, this work aims to assess the time complexity and the ability to detect plagiarism of the proposed algorithm. The rest of the paper is as follows: Next Section describes a flowchart of the proposed approach. Evaluation results are reported and analyzed in section 3. Finally, section 4 expresses the conclusion and future works.

## Materials and methods

Main phases of the proposed method are as follows.

- Preprocessing
- Document Representation
- Feature Selection
- Feature Extraction
- AHP-based Weighting Modeling
- Semantic Plagiarism Detection

### *Pre-processing stage*

The special actions of this step are tokenization, segmentations, removing some useless words, and so on. Figure 1 shows a brief schema of the pre-processing phase which is explained in detail in the following.

#### *Scientific paper*

The sections of a paper in the various journals are different. This paper focuses on medical papers published in the free online library, named as PubMed. These papers like the other scientific papers consist of special structures such as title, keywords, abstract, introduction, related works, proposed method, implementation, results, conclusions, and references.

#### *Paper segmentation*

In this step, the content and the references of the input document are separated. This process can be done manually or by a program.

#### *Tokenization*

Tokenization usually refers to break down the content of a document to its elements such as lines, words, or even a word that is created for a non-English language. The final aim of this step is to extract useful words

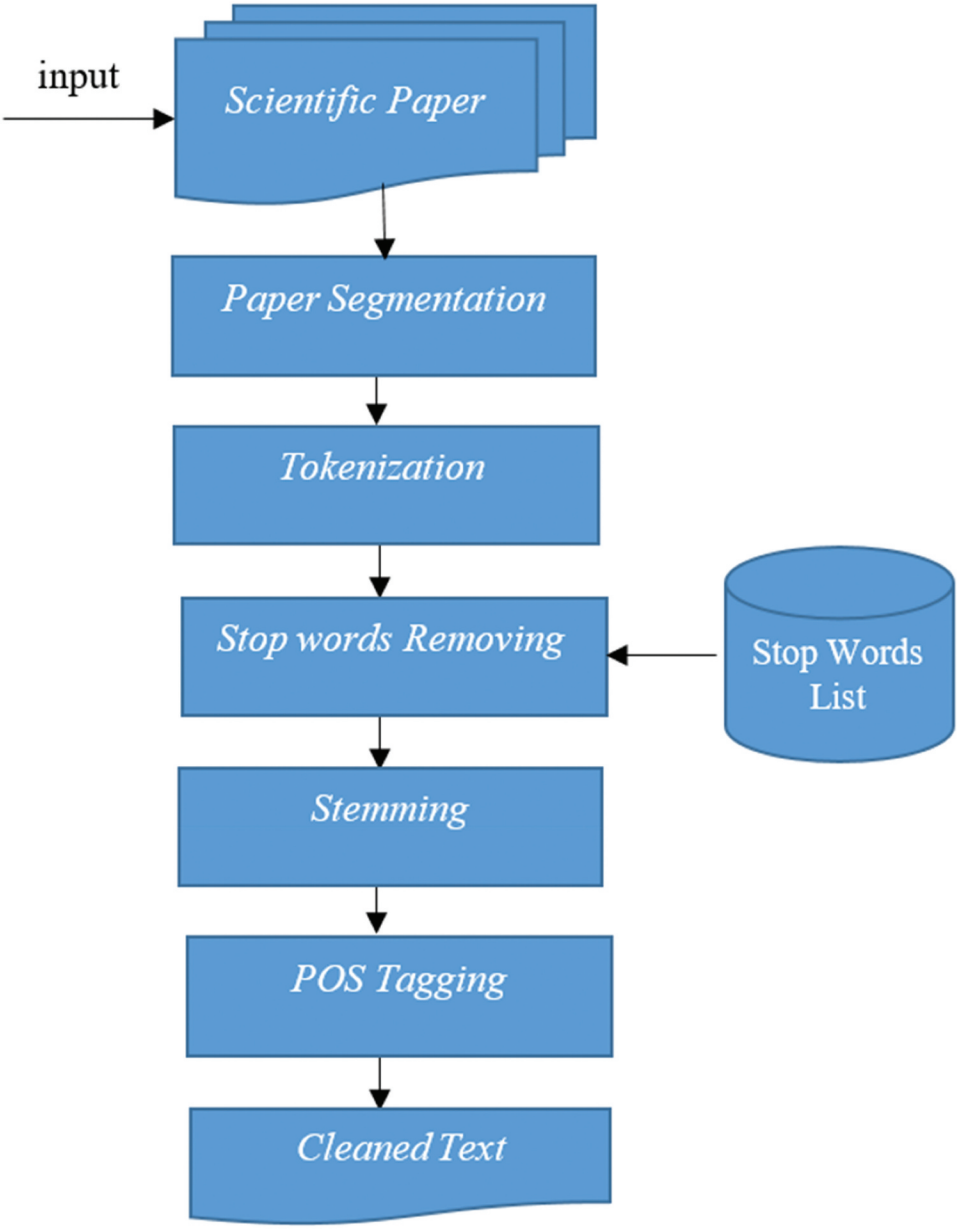


**Figure 1.** Pre-processing stage of the proposed approach.

individually and without any separators. The separators list has been explained by Singh, Tomar, and Sangaiah (2020).

This paper focuses on sentence and word tokenization by adapting tokenization functions built-in into the NLTK toolkit in the python environment.

### Stop words removing

A major pre-processing task of natural language processing is to remove useless data which refers to stop words (Ladani and Desai 2020). A stop word is a commonly used term (such as "the," "a," "an," "in") which can be easily removed from the raw text to reduce the processing operations. SMART stop word list is the most widely used set of stop words to be discarded (Buckley 1985).

### Stemming

Another common preprocessing step is to perform stemming. It is a morphological normalization to improve results in information retrieval. Stemming uses simple rules of transformation to eliminate common morphological affixes from words. Porter stemmer is the most widely used stemming algorithm for English texts (Porter 1980) which is also embedded in the NLTK toolkit named Porter Stemmer.

### POS tagging

As it is known, words take different roles depending on their position in the sentence (such as noun, verb, etc.). In this step, each word is related to an information tag named POS which is only compared for a pair of words ($W_q$, $W_k$) in which $W_q$ and $W_k$ have the same role so that it can prevent the additional attempts to compare unrelated words. Consequently, the accuracy and speed of plagiarism detection are improved (Vani and Gupta 2018).

The pre-processing stage returns a list of cleaned tokens of the paper content ready for semantic plagiarism detection.

## Document representation stage

In the text mining field, each term can be considered as a feature. After the elimination of useless words and performing the stemming phase, there is a set of candidate terms as features for constructing document vectors. The most common form of document representation in text processing is the bag of words (BOW), where the features correspond to the words which appear at least once in the training corpus (Liu et al. 2019; Fu, Feng, and Cunningham 2019). In general, the majority of these terms are not discriminative. As a matter of fact, a subset of them is employed in document representation.

The selection of the best-fitting term set is a challenging problem and numerous measures are studied for this purpose. Experiments have shown that best performance scores are achieved when a few thousand terms are employed (Mazyad, Teytaud, and Fonlupt 2018). One of the widely explored approaches to enhance the BOW representation is the use of co-occurrence of terms in addition to individual terms. In this document-representation

approach, co-occurrences of the individually valuable terms are generally considered. After computing a suitable set of co-occurrences based on the features, the BOW-based vectors are augmented. Then, the features can be categorized into three groups including syntactic phrases, statistical phrases (i.e., n-grams), and term sets. Syntactic phrases express a series of words built syntax rules such as noun phrases, verb phrases, and adjective phrases (Fiorenzo et al. 2020). There are extensive researches about statistical phrases known as n-grams. They define the features sequences of adjacent terms (n-grams) based on the co-occurrence. It generally considers the sequences of two and three words known as bigrams and trigrams, respectively.

Some researchers have shown that higher lengths are not useful (Moschitti and Basili 2004; Nagar, Bhasin, and Mathur 2019). In the terms-based approach, the order is not important. Therefore, a novel feature can be defined regarding some co-occurrences of different terms that are far from each other. In this situation, the sequence of the terms and non-syntactical sequences may be meaningless. Moreover, the computational of all combinations of terms is an NP problem. Consequently, a primary term set, a subset of available terms, is more suitable. For instance, Bokhabrine (Bokhabrine, Biskri, and Ghazzali 2019) defined a frequent item set to measure the similarity of two documents.

Given the fact that plagiarists may use the terms of the original paper in a way that they necessarily do not come adjacent together and also to reduce the processing overhead, this paper considers a document representation based on item set.

## Feature selection

Feature selection aims to remove non-relevant features for dimensional reduction of the feature space and employ a discriminative set of features for plagiarism detection. As mentioned before, in the process of detecting plagiarism, checking similarity throughout the content is time-consuming. Therefore, the proposed algorithm looking for candidate features that are more impressive than others. The simplest approach to detect such features is to discover terms with the most frequency by their occurrence number over the document. The phases of the feature selection step are as follows.

### Term frequency (TF) calculation

This stage computes the number of repetitions of each term of a document regarding the appearance number. It is considered as *Term Frequency* (TF).

### Term frequency sorting

The frequencies of the previous step are arranged in descending order to select the most frequent terms.

### K specification

The purpose of this step is to determine a cutoff point for selecting a list of terms with the most frequency. The proposed algorithm considers this index equal to 10. Although, the next section examines the effect of the different cutoff points on runtime and accuracy criteria.

### Top-k selection

At this stage, the k number of terms which represented the highest frequencies are used to create the candidate_set for checking plagiarism which is enriched with semantic features extracted in the next stage.

## Feature extraction stage

Feature extraction means extracting a new set of features which is essential in a text. In many cases of plagiarism, a plagiarist only changes the original words with their synonyms (Badawi and Altınçay 2019). Therefore, the algorithm inserts the new features in the form of synonyms of top-k features to complement the candidate set using WordNet. Figure 2 illustrates the pseudo-code of feature selection and feature extraction stages.

Moreover, WordNet only supports single words. Therefore, the method creates all possible two and three combinations of top-k features and their synonyms which are called 2- and 3-terms sets, respectively.

| Algorithm 1 | *Feature Selection and Feature Extraction* |
|---|---|

**Input:** T: Cleaned tokens of paper; k: number of most frequent terms
**Output:** candidate _set (including most frequent terms and their synonyms)

```
Begin
    foreach w ∈ T do
        tf ← calculate the frequency of w in T
    end

    tf_sorted ← sort the tf in descending order
    most_freq_terms ← get the top k elements from tf_sorted

    foreach term ∈ most_freq_terms do
        synonym ← the synonyms of term using by WordNet
        synset ← synset ∪ synonym
    end

    return candidate_set: most_freq_terms ∪ synset
end
```

Figure 2. Pseudo-code of feature selection and feature extraction.

### Semantic plagiarism detection stage

At this stage, the process of detecting plagiarized sentences among two scientific documents begins. Suppose that two documents namely $d_{suspicious}$ and $d_{original}$ are given. First, both documents are preprocessed to obtain the list of cleaned tokens of $d_{suspicious}$ and $d_{original}$ named $T_{suspicious}$ and $T_{original}$ respectively. The goal is to determine the sentences in $d_{suspicious}$ which are copied from $d_{original}$ whether syntactically and semantically. A straightforward way is to check all contents of $d_{suspicious}$with that of $d_{original}$. However, it is obvious that this approach is very time-consuming. Therefore, as mentioned before, the methods look for candidate sentences in which the probability of plagiarism occurrence is high. Therefore, it seems that the frequency of the terms is a suitable method. These terms can include the main idea of an article and also represent important sentences with a high probability of fraud. To find the terms with the most frequency, the frequency of all terms $t_1 \in T_{suspiciuos}$ and all terms $t_2 \in T_{suspiciuos}$ are calculated and stored as $tf_1$ and $tf_2$ sets, respectively. By sorting these two sets in descending order, the k numbers of terms with the most frequent ones are specified and stored as *most_freq_terms*. Since plagiarists may use the synonyms of the terms of the sentences in the original document, the synonyms of the *most_fq_terms* is obtained. Then, the method uses the union of *most_freq_terms* set and *synset* as the *candidate_set*. Suppose that the $candidate_set_1$and the $candidate\_set_2$ are the common terms and their synonyms in $d_{suspicious}$ and $d_{original}$, respectively. Now, the algorithm extracts all possible 2- and 3-terms sets of these candidate sets, separately. After that, it executes the paper segmentation to extract all sections. Then, the method calculates the similarity using the intersection among the $candidate_set_1$and the $candidate\_set_2$, $2-termset_1$ and $2-termset_2$, $3-termset_1$ and $3-termset_2$ for each section.

### AHP-based weighting modeling stage

Given that scientific documents consist of different sections (such as title, abstract, keyword, introduction, related works, proposed method, implementation, evaluation, results, and conclusion) and the other hand, the occurrence frequency of top-k features in each section can be varied, the plagiarism detection problem can be posed as a decision-making problem based on multiple criteria (Marcelino et al. 2019) and various weighted criteria can be considered.

The AHP technique is a known decision-making model based on multi-criteria. This method was developed by Saaty in 1980 (Saaty 1980). This technique is a powerful and flexible approach to multi-criteria decision-making techniques that can solve complex problems at different levels. The

AHP method combines both objective and subjective assessments into a single structure based on pairwise comparative scales and helps analysts to organize the underlying aspects of a problem in a hierarchical format. Benefits of this approach include: assessing the consistency of decision-makers' judgments, making pairwise comparisons in selecting the optimal solution and options, the ability to consider criteria and sub-criteria in evaluating options, making the best option accessible through paired comparisons. Figure 3 shows the hierarchical structure of AHP modeling in the proposed algorithm. AHP modeling consists of several steps as follows.

### *Step1: Goals, criteria, and options definition*

The first step in each AHP problem is to determine the goal, criteria, and options. The proposed method intends to measure the similarity of a suspicious document and the previous documents based on the position of matching occurrence and the type of matched terms (candidate set, 2- and 3-terms sets). Figure 3 shows that the goal is about calculating the similarity among a dubious document and a corpus of documents. As mentioned earlier, one of the most important innovations of this work is the consideration of the different importance degrees of the adaptations made in different parts of a given article. Moreover, it considers different weights for the type

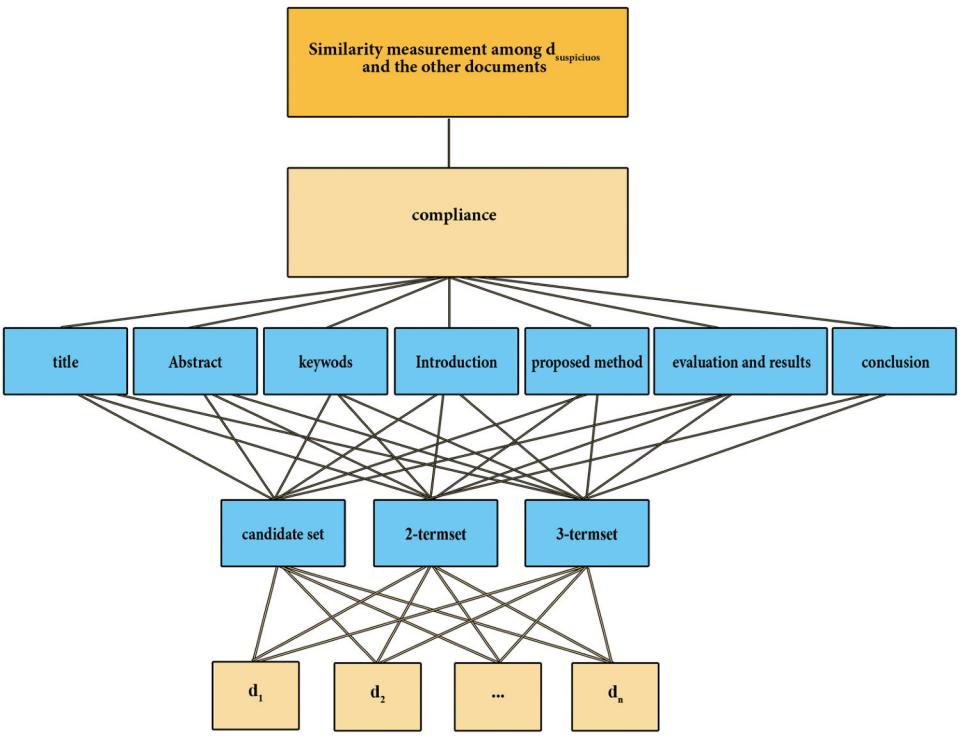


**Figure 3.** AHP modeling in our proposed plagiarism detection method.

of term sets adapted in terms of single, 2- and 3. Therefore, the criteria of the proposed model include the place of adaptation and the type of adapted terms sets. Finally, the similarity of the document and other documents is obtained as a weighted model.

### *Step2: Pairwise comparison of criteria*

According to the AHP approach, this stage must compare the criteria with respect to the goal in a pairwise manner. The numbers attributed in this comparison are numbers in the range of 1 to 9 (see Table 1).

Consequently, the pairwise comparison results based on both criteria including the position of the matching cases and the type of matched terms or term sets are presented in Tables 2 and 3 respectively. Therefore, the output of this step is a matrix in which each element as $m_{ij}$ illustrates the priority of criteria $i$ to $j$. It is trivial that the higher value of $m_{ij}$ means the higher priority of the criteria $i$ to criteria$j$.

**Table 1.** Saaty's 9-point Scale for pairwise comparison in AHP (Badawi and Altınçay 2019).

| Importance Degree | Description | Explanation |
|---|---|---|
| 1 | Same Importance | Both criterion have same favors |
| 3 | Middle Importance | One criterion has slightly favors |
| 5 | High Importance | One criterion has strongly favors |
| 7 | Too high Importance | One criterion has judgment strongly favors |
| 9 | Absolute/Extreme Importance degree | Evidence afforming shows one criterion has judgment strongly favors |
| 2,4,6,8 | Instant values between above scale amounts | There is no Absolute judgment and it needs a compromise |

**Table 2.** Expert questionnaire matrix for the site criteria.

| Criterions | Title | Abstract | Keywords | Introduction | Proposed method | Evaluation and results | Conclusion |
|---|---|---|---|---|---|---|---|
| Title | 1 | 7 | 7 | 9 | 8 | 8 | 7 |
| Abstract | 0.14 | 1 | 1 | 5 | 3 | 3 | 1 |
| Keywords | 0.14 | 1 | 1 | 5 | 3 | 3 | 1 |
| Introduction | 0.11 | 0.20 | 0.33 | 1 | 0.5 | 0.5 | 0.33 |
| Proposed method | 0.12 | 0.33 | 0.33 | 2 | 1 | 1 | 0.5 |
| Evaluation and results | 0.12 | 0.33 | 0.33 | 2 | 1 | 1 | 0.5 |
| Conclusion | 0.14 | 1 | 1 | 3 | 1 | 2 | 1 |

**Table 3.** Expert questionnaire matrix for the type of terms criteria.

| Criterions | Candidate set | 2-termset | 3-termset |
|---|---|---|---|
| Candidate set | 1 | 0.14 | 0.11 |
| 2-termset | 7 | 1 | 0.14 |
| 3-termset | 1 | 0.14 | 0.11 |

### Step3: Weights calculation

Finally, to obtain the weight of each criterion, the following steps should be done step by step.

### Action 1: Columnar summation of comparison matrix

First, the columnar summation of the comparison matrixes is obtained by Equation (1).

$$j \in \{1,2,\ldots,|criteria|\} col\_sum_j = \sum_{i=1}^{|critera|} m_{ij} \tag{1}$$

where $m_{ij}$ is an element. $i$ and $j$ are row and column of the comparison matrix. The index $i$refers to a row of that matrix and is in range of 1 to $|criteria|$ (the number of criteria which also represent the number of matrix rows which is equal to 7 and 3 for criteria related to the position and type of matching cases, respectively).

### Action 2: Normalization

Each matrix element should be divided by the calculated columnar summation of its own column. Equation (2) is a formal shape.

$$mij = \frac{m_{ij}}{col_{sum_j}} \tag{2}$$

where $m_{ij}$ is an element $(i, j)$ in comparison matrix. $col\_sum_j$ is the columnar summation of column j.

### Action 3: Weight calculation

Ultimately, the average of the resulted elements in each row is considered as a weighted value for the criteria on that row. In other words, for one criterion in row $i$, the weight is calculated as row summation of normalized values obtained in the previous step as $m'_{ij}$. The mathematical form of this step is Equation (3).

$$w_i = \sum j = 1|critera|mij \tag{3}$$

The final results of steps 1 and 2 are illustrated in Tables 4 and 5.

The final weight of each criterion is calculated as the average of values in each row in Tables 4 and 5. Table 6 shows the obtained weights. It should be noted that Tables 2 and 4 are based on the structure of the articles used in the evaluation section. Consequently, if the structure of the articles changes, these tables must change.

**Table 4.** Expert questionnaire matrix for the position criteria.

| Criterions | Title | Abstract | Keywords | Introduction | Proposed method | Evaluation and results | Conclusion |
|---|---|---|---|---|---|---|---|
| Title | 0.56 | 0.64 | 0.63 | 0.33 | 0.45 | 0.43 | 0.61 |
| Abstract | 0.07 | 0.09 | 0.09 | 0.18 | 0.17 | 0.16 | 0.08 |
| Keywords | 0.07 | 0.09 | 0.09 | 0.18 | 0.17 | 0.16 | 0.08 |
| Introduction | 0.06 | 0.01 | 0.03 | 0.03 | 0.02 | 0.02 | 0.02 |
| Proposed method | 0.06 | 0.03 | 0.03 | 0.07 | 0.05 | 0.05 | 0.04 |
| Evaluation and results | 0.06 | 0.03 | 0.03 | 0.07 | 0.05 | 0.05 | 0.04 |
| Conclusion | 0.07 | 0.09 | 0.09 | 0.11 | 0.05 | 0.1 | 0.08 |

**Table 5.** Expert questionnaire matrix for the type of terms criteria.

| Criterions | Candidate set | 2-termset | 3-termset |
|---|---|---|---|
| Candidate set | 0.05 | 0.01 | 0.08 |
| 2-termset | 0.41 | 0.12 | 0.11 |
| 3-termset | 0.52 | 0.85 | 0.8 |

### *Similarity score calculation stage*

After obtaining the AHP-based model, the proposed algorithm calculates the similarity between a given document and the others. It calculates the similarity between $d_{suspicious}$ and $d_{original}$ regarding the weight coefficients, the count and type of matched cases obtained in the previous step. Equation (4) clearly describes how to calculate that.

$$Similarity\_Score = \frac{\sum_{i=1}^{7}\sum_{j=1}^{3} w_{s_i} w_{t_j} C_{ij}}{\left(\sum_{k=1}^{10} w_k\right) \times length\left(d_{suspicious}\right)} \tag{4}$$

Where $C_{ij}$ shows the matched cases in section $i$ of paper and $j$ represent the type of terms set. For example, $C_{12}$ refers to matched 2-terms sets which have been found in the title part of two papers. $w_{s_i}$, $1 < i < 7$ refers to the calculated weight for each paper section as order shown in Table 6. For example, $w_{s_2}$ refers to$w_{abstrat}$. $w_{t_j}$, $1 < j < 3$ also determines the weight specified for each type of matched terms (single terms, 2-terms, or 3-terms set). Therefore, $w_{t_2}$ is equal to $w_{2-termset}$. To normalize the similarity score, the method divides the result by the sum of the total weights as well as the length of the document. Consequently, the resulted score falls into the range of 0 up to 1.

The steps of the proposed algorithm are summarized as follows:

(1) Build a list of stem terms (tokens) of the paper
(2) Sort the list according to the frequency of the terms
(3) Select K terms of the first of the list

**Table 6.** Final criteria weight.

| Criteria | Weight | Criteria | Weight |
|---|---|---|---|
| $w_{title}$ | $\frac{0.56+0.64+0.63+0.33+0.45+0.43+0.61}{7} = 0.52$ | $w_{evaluation/results}$ | $\frac{0.06+0.03+0.03+0.07+0.05+0.05+0.04}{7} = 0.04$ |
| $w_{abstrat}$ | $\frac{0.07+0.09+0.09+0.18+0.17+0.16+0.08}{7} = 0.12$ | $w_{conclusion}$ | $\frac{0.07+0.09+0.09+0.11+0.05+0.1+0.08}{7} = 0.08$ |
| $w_{keywords}$ | $\frac{0.07+0.09+0.09+0.18+0.17+0.16+0.08}{7} = 0.12$ | $w_{candidate\_set}$ | $\frac{0.05+0.01+0.08}{3} = 0.04$ |
| $w_{introduction}$ | $\frac{0.06+0.01+0.03+0.03+0.02+0.02+0.02}{7} = 0.02$ | $w_{2-termset}$ | $\frac{0.41+0.12+0.11}{3} = 0.21$ |
| $w_{proposedmethod}$ | $\frac{0.06+0.03+0.03+0.07+0.05+0.05+0.04}{7} = 0.04$ | $w_{3-termset}$ | $\frac{0.52+0.85+0.80}{3} = 0.72$ |

(4) Create all possible two and three combinations of the terms and their synonyms which are called 2- and 3-terms sets, respectively
(5) Calculate the weights of the terms
(6) Compute the similarity using the AHP model

### Dataset

The evaluation method has been executed using 100 scientific papers extracted from PubMed standard datasets ("PubMed standard datasets"), which are extracted from the U.S. National Library of Medicine. PubMed includes a lot of open-access articles in the region of biomedical and medicine. About 20 million PubMed files are listed with abstracts, and 21.5 million files have links to full-text versions (of which 7.5 million articles are available for free). Over the past 10 years (ending 31 December 2019), an average of approximately 1 million new records are added each year.

### Test environment

The proposed approach is implemented in Python 3.8. and Jupyter Notebook (Anna- conda3). Jupyter Notebook is an open-source web application that provides useful libraries for text mining purposes. The specifications of the testing environment are as follows: the processor is Intel(R) Core(TM) i7-7500 U, 2.70 GHz, 8 GB RAM, windows 10.

## Results

This section evaluates the proposed approach regarding the ability to find the matched cases and runtime. More details are in the following.

### Comparison with similar algorithms

This section presents the comparison of the performance of the proposed method and the approach proposed by Al-Shamery et al. (Shamery, Salih, and Gheni 2016). The words in the same context of the source document were searched in the document of the initial database. If the word or one of its synonyms are found then the words after that word was checked. Finally, if any case of matching was found, the source document was considered as semantic plagiarism.

The evaluation method focused on the ability of both algorithms to discover the matched cases alongside the time complexity. The proposed approach was executed with setting $k = 10$ on 30 documents of the corpus.

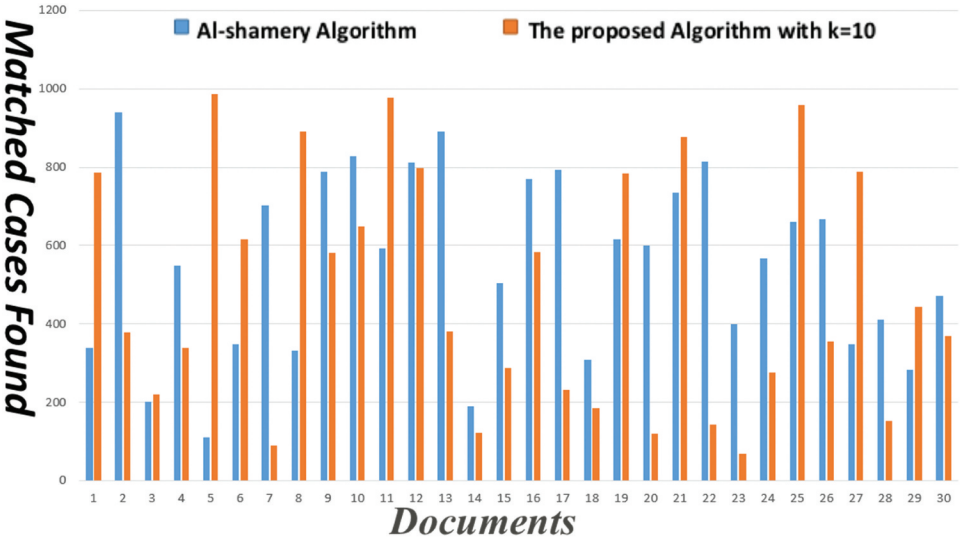


**Figure 4.** The comparison results against the ability of plagiarized cases retrieval for k = 10.

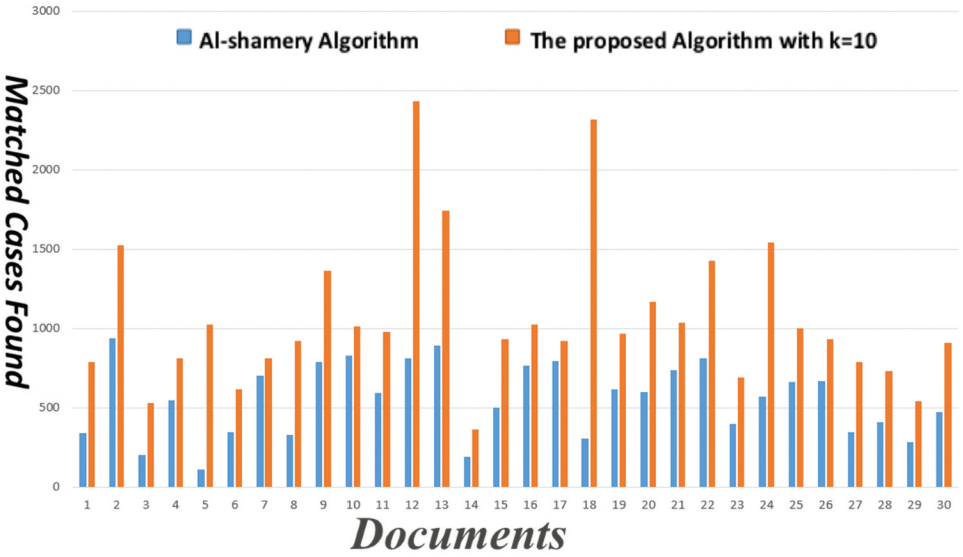


**Figure 5.** The comparison results against the ability of plagiarized cases retrieval for k = 30.

Figures 4 and 5 illustrate the results of retrieval ability. Figure 6 depicts the runtime of the proposed algorithm and compared method.

### *Parameter assessment*

As illustrated before, the proposed method focuses on the top-k terms which are the most frequent ones in a given paper. To better understand the effect of

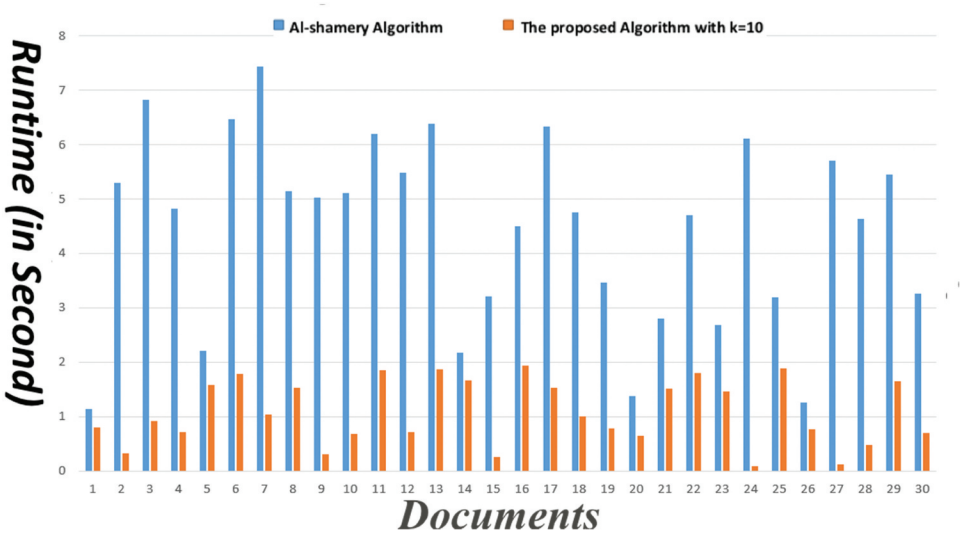


**Figure 6.** The runtime assessment of algorithms (Al-shamery's algorithm and our proposed approach).

variation of this value on the performance, the evaluation method calculated the effect of different values of k parameter on the number of matched cases found and runtime of the proposed algorithm. Therefore, the proposed algorithm was run with k∈{10,20,30,40,50}. Then, the number of single terms, 2-terms, and 3-terms sets which are common among the two selected documents were obtained. Figures 7 and 8 show the statistical results about the effect of k on the number of matched cases found and the runtime, respectively.

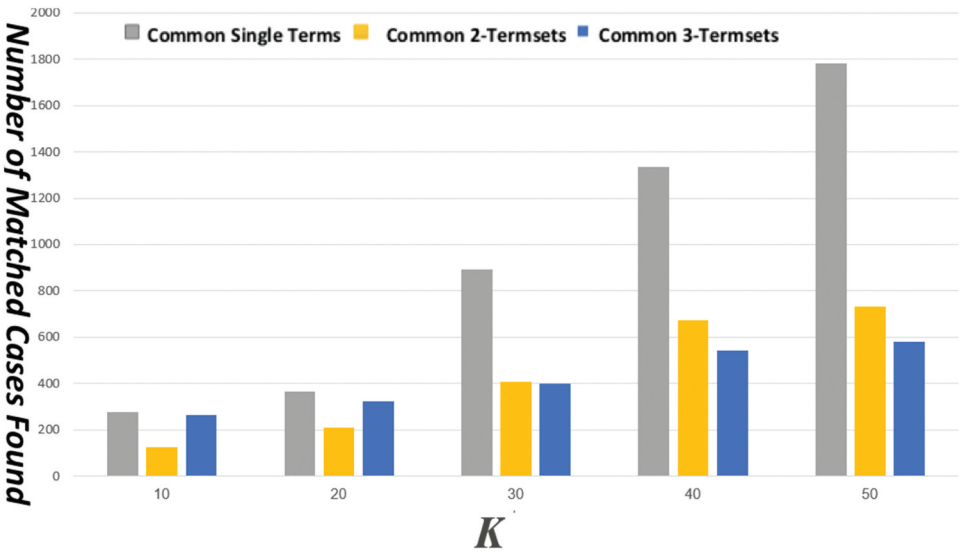


**Figure 7.** The assessment of retrieval ability with respective values of k.

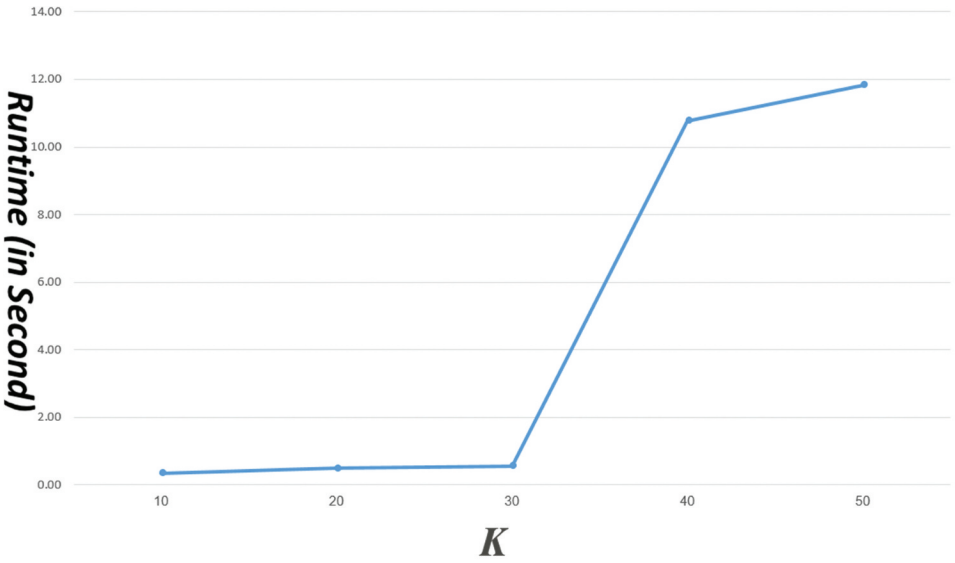


**Figure 8.** The effect of k on the runtime of the proposed algorithm.

### *Similarity scores assessment on dataset*

The proposed steps to semantic plagiarism detection were applied to 100 randomly chosen documents of PubMed online publications. In this research, the input is two papers and the output is a report about the semantic similarity among them. Table 7 indicates the results of similarity scores. Based on the similarity score, one paper is reported as a plagiarized case, if it has the similarity>0.30. The reported values show there is a little similarity (<0.09) among the test dataset (including 100 random papers).

## Discussion

This paper proposes a new approach that defines more accurate plagiarism detection models for semantic plagiarism detection with less time complexity. The reason for the decrease in time complexity is that only the article's impressive terms are considered, which are considered the top-K terms. Figures 3 and 4 show that although the similarity calculation is based on only some of the terms, it performs better in discovering plagiarized cases than the comparison method (only in about 33% of document comparisons when $k = 10$).

The results of time complexity are also impressive. According to Figure 6, the runtime of the proposed algorithm has better output than the other one. This result seems to be due to the limited number of comparisons and words that need to be searched.

**Table 7.** Similarity scores for 100 randomly chosen documents.

| #docs | Similarity Score | #docs | Similarity Score | #docs | Similarity Score | #docs | Similarity Score |
|---|---|---|---|---|---|---|---|
| 1 | 0.027801 | 26 | 0.023837 | 51 | 0.081895 | 76 | 0.071403 |
| 2 | 0.082987 | 27 | 0.009223 | 52 | 0.000659 | 77 | 0.056261 |
| 3 | 0.098157 | 28 | 0.087899 | 53 | 0.031075 | 78 | 0.012519 |
| 4 | 0.65298 | 29 | 0 | 54 | 0.0946 | 79 | 0.003563 |
| 5 | 0.066367 | 30 | 0.055592 | 55 | 0.090074 | 80 | 0 |
| 6 | 0.035088 | 31 | 0.039613 | 56 | 0.068388 | 81 | 0.095481 |
| 7 | 0.045009 | 32 | 0.051659 | 57 | 0.089782 | 82 | 0.061943 |
| 8 | 0.088957 | 33 | 0.077701 | 58 | 0 | 83 | 0.085699 |
| 9 | 0.015082 | 34 | 0.012856 | 59 | 0.066954 | 84 | 0.027795 |
| 10 | 0 | 35 | 0.036842 | 60 | 0.041125 | 85 | 0.047596 |
| 11 | 0.029565 | 36 | 0.027144 | 61 | 0.069124 | 86 | 0.055479 |
| 12 | 0.0712 | 37 | 0.028887 | 62 | 0.020278 | 87 | 0.022746 |
| 13 | 0.044834 | 38 | 0.098455 | 63 | 0.003395 | 88 | 0.071794 |
| 14 | 0.088487 | 39 | 0.002326 | 64 | 0.057011 | 89 | 0.00198 |
| 15 | 0.007448 | 40 | 0.079445 | 65 | 0.017306 | 90 | 0.085547 |
| 16 | 0.07408 | 41 | 0 | 66 | 0.081595 | 91 | 0 |
| 17 | 0.043768 | 42 | 0.047896 | 67 | 0.040849 | 92 | 0.095263 |
| 18 | 0.063053 | 43 | 0.045127 | 68 | 0.04111 | 93 | 0.001729 |
| 19 | 0.015728 | 44 | 0.01427 | 69 | 0.066513 | 94 | 0.02632 |
| 20 | 0 | 45 | 0.093346 | 70 | 0.046609 | 95 | 0.088906 |
| 21 | 0.039865 | 46 | 0.008852 | 71 | 0.056589 | 96 | 0.012939 |
| 22 | 0.019105 | 47 | 0.072525 | 72 | 0.06618 | 97 | 0.095273 |
| 23 | 0.025183 | 48 | 0.060706 | 73 | 0.047727 | 98 | 0.090091 |
| 24 | 0.058025 | 49 | 0.27431 | 74 | 0.025064 | 99 | 0.06322 |
| 25 | 0.061574 | 50 | 0.039426 | 75 | 0.028451 | 100 | 0.086716 |

Since the K parameter plays an essential role in the proposed method for checking this parameter's value, different values have been studied. According to Figures 7 and 8, increasing the value of k positively affects matched case retrieval. However, when k increases from 30 to 40, the runtime has a significant mutation. Therefore, it seems that it is the best domain of values for k. In other words, this range of values is a trade-off between increasing the number of matched cases found and the runtime of the algorithm.

## Conclusion and future works

Scientific plagiarism is one of the most important concerns of many scientific academia and journalists. One of the challenging tasks in assessing a new prepared paper is to find the similarity to the previous publications from both structurally and semantically. A plagiarist's standard plagiarism method changes the words with their synonyms or alters one sentence's structure by changing the order of words. On the other hand, the time complexity of the methods is an essential factor. One of the parameters that has not been considered in previous works is the degree of importance of the term in

different parts of the article. The degree of importance of the plagiarized detected in each section of a paper is different. This paper proposes a new approach that takes into account both structural and semantic similarity.

In other words, it comes over the significant degree problem by considering the AHP-induced weighted similarity for each section of the paper. The proposed method defines a more accurate plagiarism detection model for semantic plagiarism detection. The evaluation section implemented and tested the proposed approach using PubMed medicine publications. The results stand for a significantly better performance of the proposed algorithm regarding the runtime criterion. Moreover, the results expressed that the proposed approach has a high ability to find plagiarized terms.

To sum it up, the proposed model is more accurate due to calculation of the similarity according to 1-, 2- and 3-terms of the core terms, which are the number of adjacent words, which causes the ability to detect plagiarism if the words' order is changed. Moreover, the time complexity is better because of using a part of documents instead of all of them without decreasing detection's ability. Finally, it considers a different significant degree for various sections of the documents.

Probing deeper, this paper's results also provide a strong foundation for future work in plagiarism detection. One area of future work is in combining the deep learning techniques to calculate similarity.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This work was supported by the Bozorgmehr University of Qaenat [39175].

## ORCID

SeyyedMohammad JavadiMoghaddam http://orcid.org/0000-0003-1569-1994

## References

Badawi, D., and H. Altınçay. 2019. "Effective Use of 2-termsets by Discarding Redundant Member Terms in Bag-of-words Representation." *Neural Computing and Applications* 31 (9): 5401–5418. doi:10.1007/s00521-018-3371-y.

Bahadori, M., M. Izadi, and M. Hoseinpourfard. 2012. "Plagiarism: Concepts, Factors and Solutions." *Journal of Military Medicine* 14 (3): 168–177.

Bokhabrine, A., I. Biskri, and N. Ghazzali. 2019. "Frequent Itemsets as Descriptors of Textual Records." Paper presented at the International Conference on Computational Collective Intelligence,Hendaye, France.

Bruton, S. V. 2014. "Self-plagiarism and Textual Recycling: Legitimate Forms of Research Misconduct." *Accountability in Research* 21 (3): 176–197. doi:10.1080/08989621.2014.848071.

Buckley, C. 1985. *Implementation of the SMART Information Retrieval System*. Cornell University, NewYork, United States.

Eisa, T. A. E., N. Salim, and A. Abdelmaboud. 2019. "Content-Based Scientific Figure Plagiarism Detection Using Semantic Mapping." Paper presented at the International Conference of Reliable Information and Communication Technology, Johor, Malaysia.

Fiorenzo, A., C. Eleonora, C. Francesca, B. Franco, P. Veronica, S. Ivana, G. L. Russo, S. F. Cappa, M. Silvestro, and A. Moro. 2020. "High Gamma Response Tracks Different Syntactic Structures in Homophonous Phrases." *Scientific Reports (Nature Publisher Group) 10* (1):7537.

Fisher, E. R., and K. M. Partin. 2014. "The Challenges for Scientists in Avoiding Plagiarism." *Accountability in Research* 21 (6): 353–365. doi:10.1080/08989621.2013.877348.

Fu, Y., Y. Feng, and J. P. Cunningham. 2019. "Paraphrase Generation with Latent Bag of Words." Paper presented at the Advances in Neural Information Processing Systems, Vancouver Convention Center.

Gipp, B., and J. Beel. 2010. "Citation Based Plagiarism Detection: A New Approach to Identify Plagiarized Work Language Independently." Paper presented at the Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto Ontario Canada.

Hourrane, O., S. Mifrah, N. Bouhriz, and M. Rachdi. 2018. "Using Deep Learning Word Embeddings for Citations Similarity in Academic Papers." Paper presented at the International Conference on Big Data, Cloud and Applications, Kenitra, Morocco.

Isele, M.-R. 2018. *Analyzing Similarity in Mathematical Content to Enhance the Detection of Academic Plagiarism,Arxivpreprint arxiv:1801.08439*.

Khadilkar, K., S. Kulkarni, and P. Bone. 2018. "Plagiarism Detection Using Semantic Knowledge Graphs." Paper presented at the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India.

Ladani, D. J., and N. P. Desai. 2020. "Stopword Identification and Removal Techniques on TC and IR Applications: A Survey." Paper presented at the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Sri Eshwar College of Engineering, India.

Liu, L., J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen. 2019. "From BoW to CNN: Two Decades of Texture Representation for Texture Classification." *International Journal of Computer Vision* 127 (1): 74–109. doi:10.1007/s11263-018-1125-z.

Marcelino, P., M. De Lurdes Antunes, E. Fortunato, and M. C. Gomes. 2019. "Development of a Multi Criteria Decision Analysis Model for Pavement Maintenance at the Network Level: Application of the MACBETH Approach." *Frontiers in Built Environment* 5: 6. doi:10.3389/fbuil.2019.00006.

Mazyad, A., F. Teytaud, and C. Fonlupt. 2018. "Information Gain Based Term Weighting Method for Multi-label Text Classification Task." Paper presented at the Proceedings of SAI Intelligent Systems Conference, London, United Kingdom.

Meuschke, N., N. Siebeck, M. Schubotz, and B. Gipp. 2017. "Analyzing Semantic Concept Patterns to Detect Academic Plagiarism." Paper presented at the Proceedings of the 6th international workshop on mining scientific publications, Toronto, Canada.

Meuschke, N., V. Stange, M. Schubotz, and B. Gipp. 2018. "HyPlag: A Hybrid Approach to Academic Plagiarism Detection." Paper presented at the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, MI, USA.

Meuschke, N., V. Stange, M. Schubotz, M. Kramer, and B. Gipp. 2019. "Improving Academic Plagiarism Detection for STEM Documents by Analyzing Mathematical Content and

Citations." Paper presented at the 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL), Illinois, USA.

Moschitti, A., and R. Basili. 2004. "Complex Linguistic Features for Text Classification: A Comprehensive Study." Paper presented at the European Conference on Information Retrieval, Sunderland, UK.

Nagar, A., A. Bhasin, and G. Mathur. 2019. "Text Classification Using Gated Fusion of N-gram Features and Semantic Features." *Computación y Sistemas* 23 (3). doi:10.13053/cys-23-3-3278.

Ohno, A., T. Yamasaki, and K.-I. Tokiwa. 2020. "Similarity Measurement Based on Author's Writing Styles for Academic Report Plagiarism Detection." *IEEJ Transactions on Electronics, Information and Systems* 140 (2): 235–241. doi:10.1541/ieejeiss.140.235.

Porter, M. F. 1980. "An Algorithm for Suffix Stripping." *Program* 14 (3): 130–137. doi:10.1108/eb046814.

"PubMed Standard Datasets." https://www.ncbi.nlm.nih.gov/pubmed/?term=pmc+cc+license%5Bfilter%5D

Ragini, J. R., and P. M. Rubesh Anand. 2016. "Sentiment Analysis: A Comprehensive Overview and the State of Art Research Challenges." *Indian Journal of Science and Technology* 9 (1): 47. doi:10.17485/ijst/2016/v9i47/108465.

Saaty, T. L. 1980. *The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation*. McGraw-Hill, New York, United States.

Shamery, A., E. Salih, and H. Q. Gheni 2016. "Plagiarism Detection Using Semantic Analysis." *Indian Journal of Science and Technology* 9 (1): 1–20

Singh, N. K., D. S. Tomar, and A. K. Sangaiah. 2020. "Sentiment Analysis: A Review and Comparative Analysis over Social Media." Journal of Ambient Intelligence and Humanized Computing: 1–2111 (1): 97–117

Soleman, S., and A. Fujii. 2018. "A Method for Plagiarism Detection over Academic Citation Networks." Paper presented at the KDIR, Seville, Spain.

Vani, K., and D. Gupta. 2018. "Integrating Syntax-semantic-based Text Analysis with Structural and Citation Information for Scientific Plagiarism Detection." *Journal of the Association for Information Science and Technology* 69 (11): 1330–1345. doi:10.1002/asi.24027.