HiLL PUBLISHING

Advances in Computer and Communications, 2022, 3(1), 34-52
https://www.hillpublisher.com/journals/acc/
ISSN Online: 2767-2875

Detection of Summary Obfuscation Plagiarism Using an Aggregation Approach

Mohsen Safari, Elham Ghanbari*

Department of Computer Engineering, Yadegar-e-Imam Khomeini (RAH) Shahr-e-Rey Branch, Islamic Azad University, Tehran, Iran.

How to cite this paper: Mohsen Safari, Elham Ghanbari. (2022) Detection of Summary Obfuscation Plagiarism Using an Aggregation Approach. *Advances in Computer and Communication*, 3(1), 34-52.
DOI: 10.26855/acc.2022.06.004

Received: April 28, 2022
Accepted: May 25, 2022
Published: June 28, 2022

***Corresponding author:** Elham Ghanbari, Department of Computer Engineering, Yadegar-e-Imam Khomeini (RAH) Shahr-e-Rey Branch, Islamic Azad University, Tehran, Iran.
Email: el.ghanbari@iau.ac.ir

Abstract

Plagiarism is considered a field of text analysis, and it refers to the copying of text from an original source without referencing it. Plagiarism, which appears at various scientific and academic levels, computer programming, etc., is not only considered a fraudulent act, but it also destroys the sense of creativity and ingenuity that may otherwise develop. Creating obfuscation through text summarization and compression is a type of plagiarism in which the perpetrator replaces the words used in the sentences of an original manuscript with a synonymous word. Accordingly, the detection of this type of plagiarism is a complex task, and it is made even more challenging by having a plagiarized phrase that is shorter in length than the original one. The proposed system in this work comprises 3 main steps: preprocessing, phrase selection, and filtering. In this approach, by customizing the Okapi Best Matching (BM25) technique and detecting semantic similarities by WordNet, the levels of two sentences in a dubious document and source document are equalized to a large extent and the scores of several similarity measures are combined by using the proposed aggregation approach; and then based on the outcome of this approach, it can be decided whether the examined text is plagiarized or not. By testing the proposed model on the PAN data set, 78% of the documents plagiarized through summarization were detected correctly.

Keywords

Plagiarism Detection, Text Analytics, Summary Obfuscation, External Plagiarism Detection

1. Introduction

PLAGARISM has various definitions. In one of these, plagiarism is defined as copying the ideas, works, and words of another person, without making any reference to the original source, and ultimately crediting them as one's own ideas, works, and words [1]. Plagiarism occurs in various fields such as scientific and academic research, music composition, computer programming, etc., and its occurrence has been accelerated with the proliferation of mass media and the increased access to information, relative to the past. Plagiarism adversely affects the sense of innovation and creativity, especially in academic centers and institutions involved in scientific research. In the past, plagiarism used to be detected manually and only in the scientific field related to the plagiarized text. Since the level of knowledge in a scientific field differs from person to person, making errors is plausible in this method of manual plagiarism detection. Also, with the proliferation in the number of texts to be checked for content similarity, a human being cannot be as fast or efficient as a computer.

Since every plagiarist uses a different method of text copying, a different detection technique should be employed for each type of plagiarized text. One person may copy a text word-by-word and another may paraphrase it or use a

combination of the two methods. In paraphrasing, the words in a source document are replaced by synonymous and equivalent words, so that at the end, a text of similar meaning to the original manuscript is reproduced. Sometimes, plagiarism may occur inadvertently; nevertheless, using others' ideas without referencing them and claiming work to be one's own is obviously considered a fraudulent and law-breaking act. Considering the availability of numerous information sources, manual detection of plagiarism will be cumbersome and inefficient and will be limited by the knowledge of the person conducting the detection procedure. Therefore, different computational methods are used for this purpose to automatically detect plagiarized texts in the least amount of time possible. The various types of plagiarism detection techniques are generally divided into two groups: intrinsic and extrinsic.

Intrinsic plagiarism detection [2]: In this type of plagiarism analysis, it is attempted to find a plagiarized text within a document without any external knowledge and to detect the suspicion document through the difference in the writing styles; in other words, in the intrinsic plagiarism detection technique, without having access to a set of source documents, we try to detect those segments of a suspicion document are potentially written by another author. This method only tests the suspicion documents [3].

Extrinsic plagiarism detection [4]: In this type of plagiarism detection, a query document is compared against a set of source documents; and if plagiarism is found within it, it will be designated as a suspicious document. There are different techniques for comparing suspicious documents against source documents, including comparisons based on semantic, clustering, and character. Extrinsic plagiarism detection techniques require a text corpus including the documents that are prone to plagiarism. In this approach, a suspicious document is compared with all the documents in a text corpus to find the recurrent or nearly-recurrent passages in the source documents [5].

In this research, the extrinsic plagiarism detection method is employed to detect summary plagiarism cases. In this approach, the likely plagiarized documents (suspicious documents) and the source documents that have been copied are both available; and as the inputs of the plagiarism detection system, a suspicious document and a source document are compared with each other in order to find the plagiarized sections.

The proposed detection framework comprises three major steps: Preprocessing, phrase selection, and filtering. In the first step (preprocessing), with the intention of omitting the sections not involved in plagiarism, the texts of source and suspicious documents (system inputs) are preprocessed and readied for the next step. In the first step, by applying different techniques such as Natural Language Processing (NLP), the texts of original and suspicious documents are generalized to a greater extent and the levels of two input documents are standardized. Also in this step, by eliminating some unnecessary words that do not provide useful information in plagiarism detection the data processing time required by the main algorithms is reduced. An important undertaking in this step is the conversion of text to vector. In this process, inspired by the method normally used for information retrieval and ranking, the frequencies of the repeated words likely associated with plagiarism in each sentence are computed and reported to the next step in which their semantic similarities are measured by relevant criteria. In the second step (phrase selection), to detect plagiarized pieces such as a sentence, paragraph, etc., the preprocessed sections of source and suspicious documents are selected and the outcomes (plagiarized cases) are sent to the filtering step. In the second step, after applying the semantic similarity detection methods to sentences, the degree of similarity of these sentences is measured by several criteria, and the results are sent to the third step (filtering). In the filtering step, the goal is to decide whether a suspicious text is plagiarized or not. This is a highly sensitive step; and if a plagiarized sentence or phrase is not detected properly as such, the used system may be deemed inefficient. So an approach has been proposed here, by accumulating the scores achieved by the measures in the preceding step, which can help one make the right decision on a likely plagiarized text. By using specific coefficients, this approach changes the degrees of the effect of similarity measures and ultimately determines the plagiarized cases according to a defined threshold.

The proposed system was evaluated on PAN-13 text corpora and the effectiveness of the system was measured by 4 different criteria. In the test section, our proposed cases including the 'WordNet' and customized 'BM25' were individually entered and analyzed. Then by using the proposed aggregation approach, the proposed system was evaluated. And finally, by changing some default parameters associated with the number of selected sentences, a Precision of 0.98, Recall of 0.65, and 'PlgDet' of 0.78 were obtained. The advantages of the proposed system include a better conversion of text to vector by using our customized method instead of more common techniques such as Term Frequency–Inverse Document Frequency (tf-idf), better understanding of sentence structures, and thus better detection of semantic similarities between words, and more efficient summary plagiarism detection by using the proposed aggregation approach and summing the scores of similarity measuring criteria.

The rest of the present paper has been organized as follows: In Section 1, the research works conducted on the

subject of extrinsic plagiarism detection will be described and in Section 2, the general framework of the proposed method for the detection of summary plagiarism will be presented. Section 3 describes the data set and the evaluation criteria used; and in Section 4, the proposed system and the obtained results are analyzed. Finally, in Section 5, the paper is concluded and suggestions for future works are presented.

2. Related works

Plagiarism detection is performed either manually or automatically. In the more traditional method of manual plagiarism detection, a human expert attempts to inspect suspicious documents and detect the plagiarized cases by relying on his/her knowledge and expertise. This is a time-consuming undertaking and since all the sources might not be comprehensively searched, it is prone to error. The automatic method of plagiarism detection, which is aimed at increasing the speed and accuracy of the process, uses natural language processing, machine learning, data mining, and other techniques in order to detect plagiarized cases [6]. Generally, in plagiarism detection processes, it is attempted to find the degree of similarity of a suspicious document to one or several source documents; thus, the plagiarism detection methods are also known as similarity search techniques [7]. Moreover, concerning the availability of an original source document, the plagiarism detection processes can be divided into two classes: Intrinsic plagiarism detection and extrinsic plagiarism detection. In the intrinsic detection approach, only a suspicious document is available. In these cases, it is usually assumed that every author has a different writing style and uses a different grammatical structure; which is considered a type of fingerprint for that writer. In the intrinsic plagiarism detection method, the goal is to detect the alterations of a writer's style of writing in different segments of a suspicious document.

In this approach, an unknown text passage claimed to belong to a particular author is given and we are asked to detect plagiarism in that document based on other sample works of that writer or; according to the problem definition, to find out to which author the examined document belongs. The main difference between this method and the extrinsic technique is that, contrary to the latter, there are no source documents and writings available in the intrinsic approach, and plagiarism detection has to be carried out without comparing the dubious document with source documents. One of the most common schemes in this method is to divide the text of a suspicious document into several segments. In the next step, a set of features is extracted from these segments, and using the author style function, each of these text segments is analyzed to eventually determine if it is plagiarized or not [8].

In the extrinsic plagiarism detection technique, in addition to a suspicious document, the source documents are also available and the goal is to detect those parts of the suspicious document which have been copied from the source documents. This technique relies on a collection of text documents from which a phrase, several words, a sentence, etc. have been probably copied and plagiarized in the suspicious document. In this approach, a suspicious document is analyzed to find the plagiarized segments by searching the phrases which have been copied exactly or roughly from a collection of source writings. The simple way of doing this is to compare every expression in a suspicious document with that in every source document from a collection of original writings. Obviously, this is not economical in terms of processing time and cost; because for finding several plagiarized phrases, an extensive set of source texts has to be searched [9]. In addition, if used synonym words and changed sentence structure, the mentioned approach will be even less efficient whether the corpus of including source documents is large or not. In the present research, we have tried to find an appropriate solution to this problem.

A review paper is written by Kohler and Weber-Wulf [10] in 2010 reports that the number of online plagiarism detection systems has increased from 5 systems in the year 2000 to 47 systems in 2010. In 2004, Weber-Wulf tested a plagiarism detection system on a small manually-created text corpus and examined its effectiveness in detecting the translated and directly-copied plagiarized samples. These tests were performed on a set of samples plagiarized by real students and on manually-created cases. Since then, this testing method has been repeated every 6 years and in the most recent testing, 26 systems have been examined (42 Japanese, British and German cases). Each of these systems has been graded and ranked according to its level of effectiveness, usefulness, and professionalism.

The automatic plagiarism detection system of 'Encoplot' was designed by Grozea and Popescu [11]. In a test performed on an intelligently-plagiarized document, and considering a large volume of written texts provided, this method achieved a detection accuracy of 75%. It should be mentioned that this technique has not been tested on natural language.

Another automatic plagiarism detection method based on passage similarity was developed by Vania and Adriani [12]. This technique was used in external plagiarism detection and in the preprocessing step to detect non-English

documents, translate them to the English language, and finally index them and recover the documents likely to have been plagiarized. This method divides the recovered documents into **s** passages each consisting of 20 smaller passages; and ultimately, plagiarism is detected by using the number of source words in the corpus that overlap the suspicious phrases.

An algorithm for extrinsic plagiarism detection was presented by Lalitha et al. [13] in the PAN-10 competition. This algorithm consists of two steps. In the first step, similar documents are identified and the passages in a suspicious document likely plagiarized from a set of given source documents are found by using the vector space model and the 'cosine similarity' criterion. And in the second step, the extent of plagiarism in a suspicious document is determined by using the chunk ratio. This method does not perform any preprocessing on the documents.

Extrinsic plagiarism detection can also be performed as the closest neighbor in vector space. Zechner et al. [9] implemented such a scheme by using a standard model of information retrieval to select a candidate document. In this procedure, the source documents are indexed for every sentence level, and the sentences in a suspicious document are used as a query. Also, the similarity is calculated through the cosine similarity measure.

In 2012, Ekbal et al. [14] presented a 4-step technique based on vector space model which was efficient in detecting plagiarized documents in English. The text corpora used included documents written in German, English, and Spanish languages (multiLingual plagiarism cases). However, due to a lack of translation of non-English documents, their method had little success in detecting plagiarized non-English documents; and the amount of recall in these documents was reduced considerably. The authors hoped to deal with the drawbacks of the designed method in detecting non-English plagiarized texts by employing Google's translation service and using fuzzy methods to compute the degree of semantic similarity.

In 2013, Naseem et al. [15] proposed a method that combined vector space model with fuzzy semantic similarity scheme. In their 3-step algorithm, the candidate documents are retrieved according to the vector space model used and the candidate documents are validated based on fuzzy semantic similarity measures. They maintained that by using fuzzy similarity criteria, intelligently plagiarized cases such as rephrasing (rewriting) can be detected. Based on measuring criteria such as precision, recall, and F-measure, the proposed system displayed a satisfactory performance.

Sanchez et al. [16] were the winners of the next round of CLEF competition (i.e., PAN-14). Their presented system was based on 4 general steps: Preprocessing, Seeding, Extension, and Filtering. In the preprocessing phase, by using their proposed semantic similarity measure called 'tf-isf' and a recursive algorithm, they were able to improve the recall rate; although accuracy was reduced slightly. Also, in the seeding step, by employing a recursive algorithm that adjusted various parameters such as sentence depth, they were able to extract small samples of plagiarized text. The goal of the extension phase was to process the set of seeds in the preceding step (seeding) in order to find the maximum number of proximate passages in a suspicious document and a source document. By designing a recursive algorithm, they succeeded in finding the maximum number of plagiarized texts. In the filtering phase, they attempted to solve the problem of the overlapping of each text pair extended in the previous step and also to find very small likely plagiarized passages based on 'src_len' and 'susp_len' values. Eventually, they were able to improve the detection of summary-type plagiarized cases by 35%.

In 2015, Suryani and Hussain [17] presented a method for improving the detection rate of intelligently plagiarized documents. Instead of using a foreign dictionary as a resource, they attempted to find semantic similarities by considering the statistical occurrences of words in a text. They also used support vector machines and the k-nearest neighbor scheme in their algorithm. The dataset used for testing was collected manually on subjects of different levels. The test results revealed a good improvement in the rate of plagiarism detection, relative to plagiarism detection software programs such as 'Turntin'.

Abdi et al. [18] presented an external plagiarism detection scheme in 4 steps. In the first step (preprocessing), natural language processing techniques are used and in the second (candidate retrieval), the source documents that are very similar to the suspicious document are found. In the third step (detailed comparison), these documents are fragmented into sentences and compared in more detail. And in the last step (post-processing), the detected plagiarized segments are filtered according to a defined threshold.

The distinct features and the novelty of the present research lie in the new system it presents for detecting extrinsic summary plagiarism cases. In this model, for finding the occurrences of highly repeated words associated with plagiarism in source and suspicious documents, a technique that normally uses a ranking function is customized and employed to analyze the sentences in a document; in this way, we convert the words in each sentence to vectors more

intelligently. In the design of this system, we have paid particular attention to the semantic structure of words as well. In other words, since in the summary type of plagiarism the words in a source document are changed and replaced by synonymous words, the proposed system, instead of superficially searching for such words, examines the word structures semantically by employing a semantic network and extracts a set of word meanings. Also in this system, for improving the rate of detecting semantic similarities in source and suspicious text documents, an aggregation approach is used to sum up the scores and to apply several important semantic similarity measures.

3. The proposed method

In this paper, a system has been proposed for extrinsic plagiarism detection in texts written in English language. This method compares the available source and suspicious documents in order to find the maximum degree of similarity in these texts. To avoid a superficial analysis of the examined texts, the proposed system applies 3 different steps to input documents and attempts to discover the true meanings of words and the relationship between the compared sentences.

The objective of the proposed system is to explore the summary plagiarism cases at the sentence level and to improve the detection of semantic similarity between the words in each sentence. For determining the similarity of words in two sentences within the source and suspicious documents, after finding and calculating the frequently repeated words by the customized BM25 method, the structure of each word and its corresponding synonyms are extracted through WordNet. The increase in the rate of summary plagiarism detection largely depends on the type of similarity criterion selected. Hence, in the presented system, the similarity is measured by different criteria based on a defined threshold, and eventually, the passages that have been plagiarized are specified and their locations in the source and the suspicious documents are marked for subsequent inspection by humans or for measuring the performance of the system by standard criteria. 'Cosine', 'Dice', 'wup' and 'Lin' similarity measures are used in the proposed model. Each of these measures assigns a different score to the words within a sentence, and then the final decision is made by using an approach that adjusts the degree of participation of each score.

The general structure of the proposed system and its steps has been depicted in Figure 1. The inputs of the system at any moment consist of a suspicious document and a source document which enter the first step (preprocessing). The presented model includes the following three main steps:

1) Preprocessing: In this step, natural language processing techniques are used to eliminate the unnecessary and superfluous words from source and suspicious documents.

2) Phrase selection: In this step, text is converted to vector and then similarity measuring functions are applied to the sentences in source and suspicious documents.

3) Filtering: In this step, by using an aggregation approach it is decided whether a sentence is plagiarized or not.

3.1 Preprocessing

At the onset of the extrinsic plagiarism detection process, the examined texts must be processed in order to increase the chance of detecting similar sentences by eliminating the cases not involved in plagiarism detection, attain a more generalized framework of existing words in documents, and also to reduce the number of comparisons between the sentences of source and suspicious documents. The following tasks are implemented in the preprocessing step:

Case conversion: The most common task in the preprocessing step is to convert all the characters in a text to lower case. One reason for performing this task is that if a word at the beginning of a sentence (whose first letter is capitalized) is compared with the same word in the middle of a sentence (whose characters are all in the lower case), the degree of similarity of that word will not be reduced. In fact, by taking this important step, the text frameworks in source and suspicious documents are standardized and a detection algorithm can be implemented more properly; although some refrain from this undertaking [19].

Tokenization: One of the best tools for tokenization is the 'Toktok Tokenizer' tool, which has also been used here in this paper.

Eliminating the stop-words: In this step, by omitting the stop-words, which are very common in a text, the likelihood of increasing false similarity is reduced. In this work, the stop-words have been removed according to the list of such words in the 'nltk' tool [20].

Lemmatization: This task is implemented as needed in the preprocessing step. The 'WordNet' database has been used in this paper to extract the word lemma. In this procedure, first we try to retrieve a word lemma by searching in the 'WordNet' database and return the result; but if the algorithm does not succeed in finding a lemma, the input word

itself appears as the output. It should be mentioned that we have used the 'WordNet' lemmatizer, because it is a comprehensive database in terms of the number of English words, and the most exact word lemma can be extracted from it.

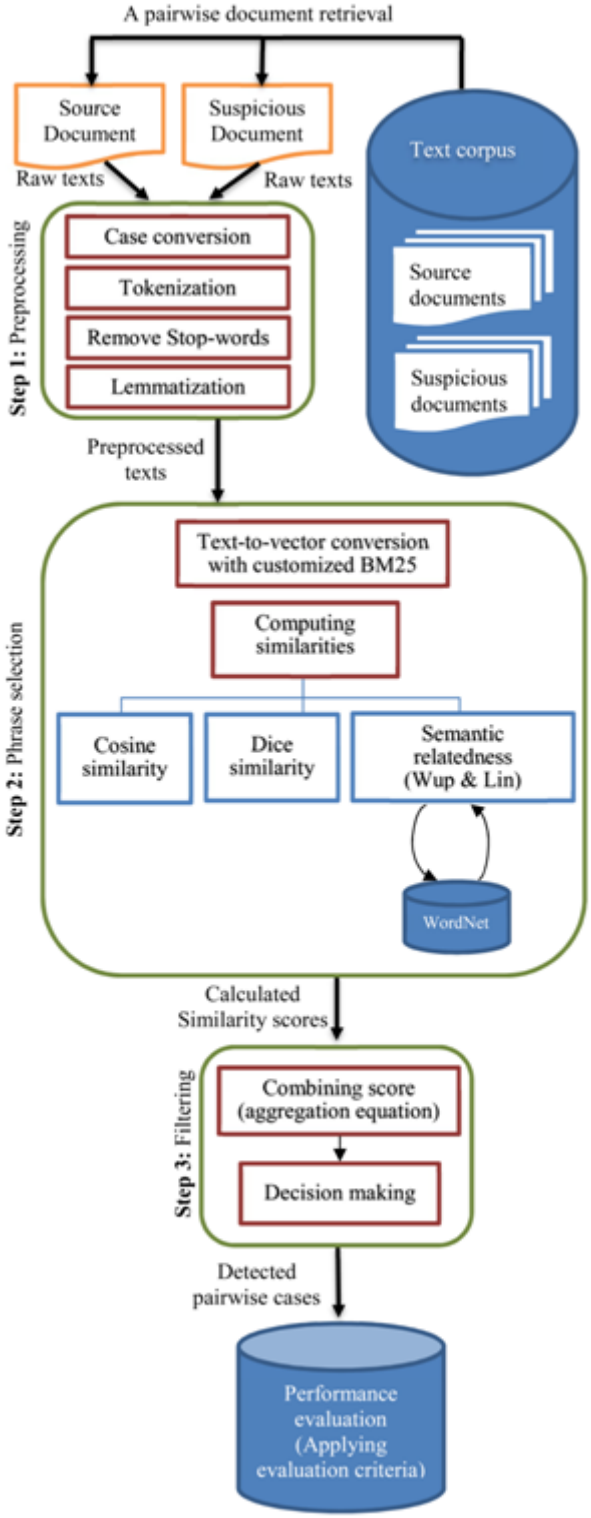

Figure 1. The general framework of the proposed system for summary plagiarism detection.

3.2 Phrase selection

The phrase selection step plays a very important role in the success of plagiarism detection, especially in summarization form; because most of the work of plagiarism detection takes place in this phase. In the phrase selection process, by using a proposed vector space model known as customized BM25, the preprocessed sentences of source and suspicious documents are converted to vectors so that the similarity measures could be applied to them and the plagiarized cases in a suspicious document could be detected. This step comprises two general phases. In the first phase, the text is converted to vector according to the following procedures:

1- Receiving the preprocessed sentences of source and suspicious documents as inputs.

2- Computing the values needed for use in the customized BM25 scheme (e.g. numbers of words and sentences in source and suspicious documents).

3- Converting text to vector by using Eqs. 1 and 2.

In the second phase, each sentence from a suspicious document is compared against all the sentences from a source document; and the likely plagiarized segments of the suspicious document are selected based on a specified threshold. Since using just one similarity criterion cannot guarantee the detection of the most complicated type of plagiarism (i.e. summarization), in this phase we have used several similarity criteria in addition to our proposed similarity measure. The procedures in the second phase include:

1- Computing the semantic similarities by using the 'cosine' criterion.

2- Computing the semantic similarity by using the 'dice' criterion.

3- Extracting the synset of source and suspicion sentences by using 'WordNet'.

4- Applying the 'Wup' criterion on synsets and extracting the semantically similar sentences.

5- Computing the semantic similarities by using the 'Lin' criterion based on the sentences extracted by the 'Wup' measure.

And now the details of the first and second phases in the phrase selection step are presented.

3.2.1 The first phase (text-to-vector conversion)

To apply the similarity measures on the words and sentences of a source document and a suspicious document, a vector space model is used to display each document as a vector. Each dimension indicates the frequency of a distinct word and includes the coordinates on an obtained vector. Ultimately, the similarity measures getting these coordinates as input, and the similarity between a suspicious document and a source document is computed. There are different ways of doing this (e.g. the binary technique); and of course, the most common approach is the 'tf-idf' weighing method. Generally, in this phase, we present our proposed vector space for converting the sentences of source and suspicious documents. The procedure for customizing the structure of this vector space which is based on BM25 scheme (a type of ranking function) is explained below.

1) The proposed customized BM25 scheme

Instead of using the more common 'tf-idf' method for text-to-vector conversion, in the present research, we made some changes to the Okapi BM25 function (where BM stands for 'best matching') and used the modified version to convert text to vector. The weighted BM25 scheme is in fact the extended form of the inverse document frequency method (created by Jones et al.); albeit, with the consideration of term frequency and document length. This scheme is a ranking function that is used in search engines for ranking the found documents according to, and associated with, an input query. In general, for a query in the form of $Q = \{q_1, \ldots., q_m\}$, the score of document *D* is calculated by Eqs. (1) and (2) [21], as follows:

$$score\ (D,Q) = \sum_{i=1}^{m} idf(q_i).\frac{tf.(k+1)}{tf+k.(1-b+b.\frac{|D|}{avgdl})} \tag{1}$$

$$idf(q_i) = \log\left(\frac{N-n(q_i)+0.5}{n(q_i)+0.5} + 1\right) \tag{2}$$

$$tf = f(q_i, D) \tag{3}$$

In the above equations, *m* denotes the number of keywords in a query, *N* is the total number of documents, and $n(q_i)$ is the number of documents containing q_i. Also, $|D|$ represents the document length (number of words) and 'Avgdl' indicates the average document length in the text set. In the above equations, the values of parameters k_1 and *b* are chosen arbitrarily, and the range of *k* is [1.2, 2.0]; which is usually taken as 1.2. Also, parameter *b* has a range of [0, 1]; which is normally considered as 0.75. It should be mentioned that there are different equations for calculating

the inverse document frequency (idf), and that this parameter will not be necessarily obtained as described.

In examining the different methods of text-to-vector conversion, numerous reasons prompted us to eventually implement the BM25 scheme instead of 'tf-idf' (which is commonly used in plagiarism detection systems) in our summary plagiarism detection model. The mentioned problem is described below.

Evaluation of 'idf': By examining Eq. (2) of the BM25 scheme, it is realized that this equation is not much different from the 'idf' equation associated with the classic 'idf', and the reason for using the BM25 method cannot be solely due to its 'idf' equation. According to Figure 2, the variations of the two curves related to BM25 'idf' and classic 'idf' equations are very similar; thus, the increase in the total number of documents in text corpus (N) and in (q_i) versus the number of documents that include q_i will not have an immediate effect on the output of 'idf' function.

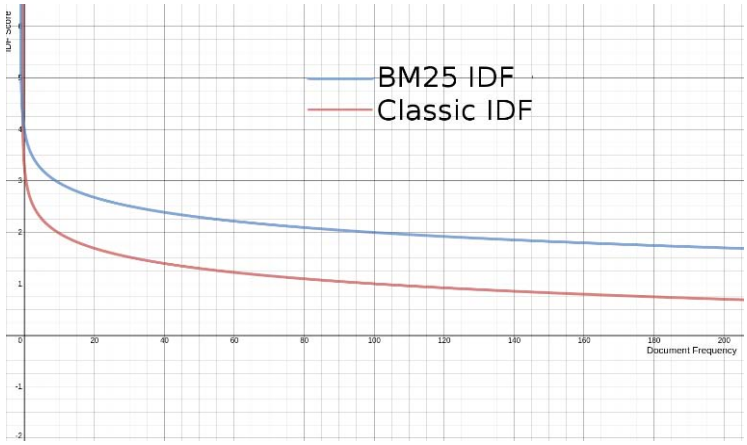

Figure 2. Comparing the 'idf' functions associated with the classic and the BM25 schemes [22].

Evaluation of 'tf': In view of Eq. 3 and the analysis of 'tf' associated with the BM25 scheme, it is realized that in addition to the 'tf' of the BM25 scheme in Eq. 1, there is another parameter 'k' whose value can be altered in order to increase or decrease the degree of influence of 'tf'. The effect of 'tf' is always increasing and it alternatingly approaches a certain value. Without the consideration of document length, the comparison between the 'tf' functions associated with the classic and the BM25 methods can be observed in Figure 3.

As it is observed, the classic 'tf' increases continually and never approaches a specific and constant point. Through the facilities provided by BM25, and using the 'k' parameter, the frequency range of each word (tf) within the sentences of source and suspicious documents can be increased or decreased, and by performing tests (through trial and error), this value can be properly tuned to achieve the right output; while this important capability, necessary for summary plagiarism detection, is lacking in the classic 'tf'.

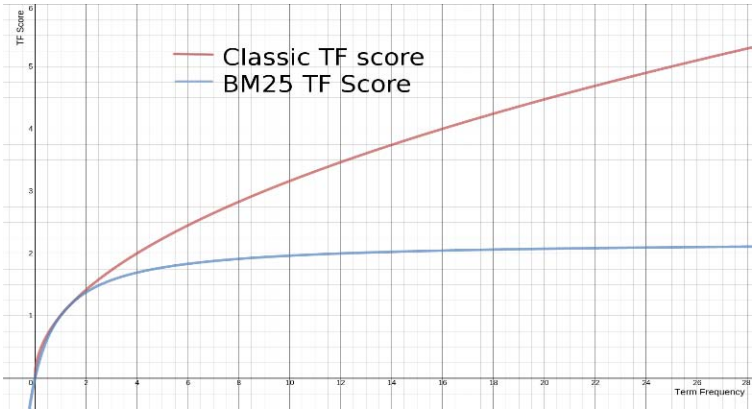

Figure 3. Comparing the 'tf' functions associated with the classic and the BM25 schemes [22].

Examining the document length: Since plagiarism involves summarization, the length of a plagiarized sentence in a suspicious document is usually shorter than that in a source document, we are looking for a text-to-vector conversion equation that can be used to adjust the degree of effect of document length to the average document length in

the text corpus $\frac{|D|}{avgdl}$.

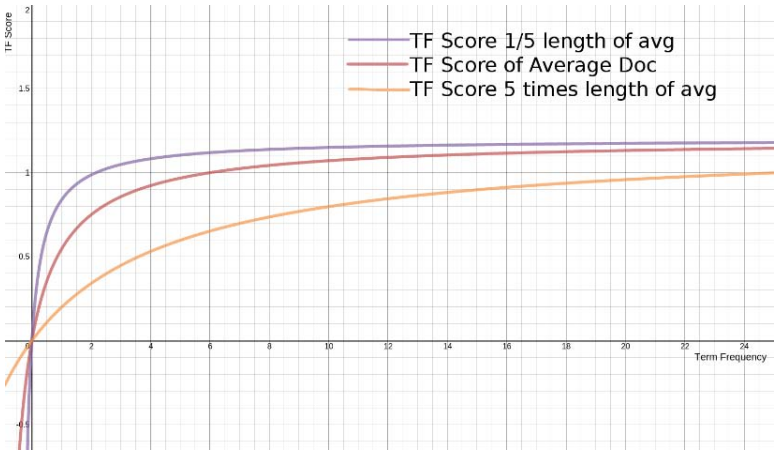

Figure 4. The relationship between document length and the average document length in text corpus in BM25 [22].

In view of Figure 4, the shorter a document length is relative to the average length of all documents, the faster the curve becomes asymptotic. In other words, it seems that the documents of shorter length that contain a larger number of a particular word would be more relevant; and conversely, if a document is long, it should contain a larger number of a particular word to be considered relevant to a query word. In short, through Eq. (1) of BM25 scheme, and with the increase in the value of 'b', the influence of document length could be increased; this would help improve the summary plagiarism detection rate, considering the fact that each sentence in a source document and suspicious document has a different length.

Table 1. The customized parameters of BM25 model

Parameters	unit changes description
N	Sum of the suspicions & source documents sentences
$Avgdl$	Total word of suspicions & source documents/2
$n(q_i)$	The number of repeats q_i in the total word of suspicions & source documents

Considering what was said above, some changes are needed in implementing the BM25 scheme for the detection of summary plagiarism and the conversion of text to vector. Since at any moment, only two documents (one suspicious and one source) are compared with each other, we won't necessarily get the right answer if we consider a document as a computation unit. Therefore, we changed the computation unit from 'document' to 'the sentences of each document' according to Table 1; and in this way, we were able to implement the BM25 scheme in the summary plagiarism detection system.

3.2.2 The second phase (applying the semantic similarity criteria)

In this phase, by using several similarity measures, in addition to finding the semantic similarities of words, the efficiency of our proposed summary plagiarism detection system is also improved. The distinctiveness of this step relative to other systems designed for this purpose is the use of 'wup' and 'Lin' criteria for calculating the degree of similarity in various texts; which enhances the detection of semantic similarities and improves the rate of summary plagiarism detection. The criteria used in this phase along with their functions are described in the following subsections.

1) Cosine similarity

Cosine similarity is a criterion that can be used for document comparison or ranking based on a vector of query words. If x and y are two vectors for comparison, by using the cosine similarity measure, which is determined according to Eq. 4, we have

$$sim(x,y) = \frac{x.y}{\|x\|\|y\|} \quad (4)$$

Where $\|x\|$ denotes the Euclidean norm of vector $x = (x_1, x_2, \dots, x_p)$ and is defined as $\sqrt{x_1^2 + x_2^2 + \cdots x_p^2}$. Similarly, $\|y\|$ is the Euclidean norm of vector y. The mentioned measure calculates the cosine of the angle between the two vectors x and y. A cosine of 0 means that the angle between the two vectors is 90º (they are perpendicular to each other) and indicates that there is no matching between the vectors. As the cosine value approaches one (1), the angle gets smaller and more matching is created between the two vectors [23].

2) The dice coefficient

In statistics, the Sorensen–Dice index is used to determine the similarity of two samples (texts in the case of plagiarism). This measure has other names like "Dice Similarity Coefficient (DSC)" [24]. Based on Eq. 5, the mentioned index is expressed as follows:

$$Dice_{coefficient} = \frac{2|X \cap Y|}{|X|+|Y|} \equiv \frac{2|X.Y|}{|X|+|Y|} \quad (5)$$

The result of the above equation is a number between 0 and 1, which indicates the degree of similarity between X and Y.

3) Computing the similarities in the WordNet

An important issue in plagiarism is the fact that a plagiarist, through his/her fraudulent act, tries to deceive the readers by rewriting, rephrasing, and summarizing a text and by replacing the words with their synonyms. So we should employ a method that can extract the original essence (meaning) of words in both a source document and a suspicious document, even if these words have been altered and replaced by other words. In other words, the expressions in source and suspicious documents should be standardized in terms of their original root meaning before they can be compared for semantic similarity. For this purpose, we have used the 'WordNet' database in our proposed system.

WordNet is a database of English words. This huge database has classified English words in terms of their synonyms into groups, each one called a 'synset' [24]. Every synset is a set of synonymous words with common features. In order to compute the semantic similarities between the synsets of WordNet and source and suspicious text pairs, all the synsets for every word are selected as comparison keys. The synsets of suspicious texts are compared with those of source texts. Then the degrees of similarity are computed and normalized by using the 'Jaccard' or another coefficient and the total number of synsets in suspicious and source texts. When n = 1, every synset is considered as one 'n-gram'. To have a match between a source text and a suspicious text, at least one of their synsets must be the same. This notion has been expressed in Eq. 6.

$$Sim_{WordNet}(A, B) = \frac{|S(A,n) \cap S(B,n)|}{|S(A,n) \cup S(B,n)|} \quad (6)$$

In the above equation, S(A, n) and S(B, n) respectively represent the unique synsets in source and suspicious texts; which in general, the intersection of the two sets is divided by their union.

Note that for more precise detection of plagiarized texts in the form of rephrasing (rewriting) or other schemes, in which the semantic and context similarities of words and phrases are important, similarity criteria should be selected more meticulously. In order to increase the chances of detection in this type of plagiarism, in addition to calculating the similarities through the Cosine and the Dice coefficient measures, the two criteria of 'Lin' and 'wup' were also used after extracting the word synonyms from source and suspicious documents. Using the 'wup' criterion, the sentences with maximum similarities in terms of information content were extracted and sent to the 'Lin' criterion for reassessing the similarities.

In this research, after extracting the synsets, the two criteria of 'wup' and 'Dekang Lin' were used to obtain the similarities. 'wup' is a common criterion for computing the similarities through the WordNet database, and it was used here to improve the rate of plagiarism detection. The equation associated with the 'wup' criterion also involves the depth of the constructed LCS tree. It should be mentioned that the similarity value computed for this measure will always be in the form of $0 < Sim_{wup}(c_1, c_2) \leq 1$. This similarity value will never be zero because at least one node (word) always exists as the tree root which causes the similarity to be non-zero. The similarity computing procedure is shown in Eq. 7.

$$Sim_{wup}(c_1, c_2) = \frac{2 \times dept\,h\,(LCS(c_1,c_2))}{dept\,h(c_1)+IC(c_2)} \quad (7)$$

The 'Dekang Lin' criterion, which was presented in 1998 [25], is better correlated with human judgment and real-world conditions. In general, this criterion provides a measuring scale for two objects based on the information

content (IC) of their Least Common Subsumer (LCS) divided by the sum of the information contents of their individual concepts. In other words, the 'Lin' criterion defines similarity as the rate of common information between two concepts (e.g., IC (LCS)), and it uses Eq. 8 to express the information needed to completely describe these concepts (e.g., the separate IC of each concept).

$$Sim_{lin}(c_1, c_2) = \frac{2 \times IC\ (LCS(c_1,c_2))}{IC(c_1)+IC(c_2)} \tag{8}$$

In the above equation, information content (IC) refers to "the amount of information provided by the term when appearing in a context" [26]. Information content constitutes an important aspect of word knowledge when trying to evaluate the similarity between two words or phrases for their substance and meaning. The usual procedure for measuring the tangible information content of two words is to combine the knowledge of their hierarchical structure from the standpoint of ontology (through a database such as WordNet) with the statistics related to their actual usage in a text adapted from a large text corpus [27]. Also, the LCS characterizes the closest meaning between two concepts c_1 and c_2 as an ancestor in the is-a relation tree.

In the proposed system, first, the synonyms of words in the sentences of suspicious and source documents are obtained by using the WordNet database, and the following two sets (expressed by Eqs. 9 and 10) are established.

$$All_{synset\ susp} = \{w_1 | w_1 \cap synset_{WordNet}\ \ and\ w_1 \cap sentence_{susp}\ \} \tag{9}$$

$$All_{synset\ src} = \{w_2 | w_2 \cap synset_{WordNet}\ \ and\ w_2 \cap sentence_{src}\ \} \tag{10}$$

Next, for each existing component in the All_synset_{susp}, the component with the highest degree of similarity to one of the components in All_synset_{src} is selected. This maximum similarity will be computed by the 'wup' measure. Then in the next step, the similarity of the selected component is determined through the 'Lin' measure and if it is greater than the considered threshold, that sentence is added to set S (phrase selection) as a plagiarized sentence. The general procedures mentioned above are implemented according to Fig. 5.

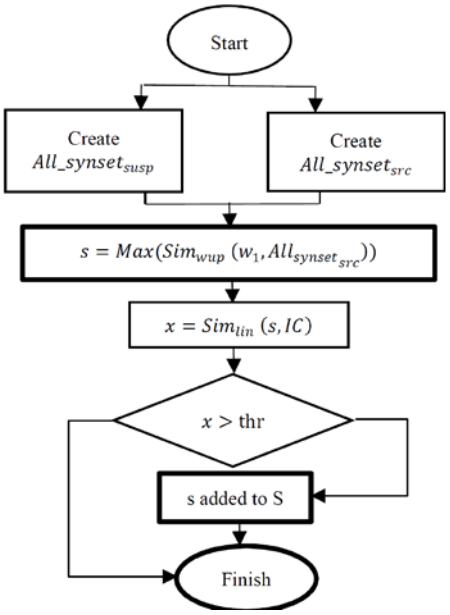

Figure 5. Computing the semantic similarities by using the WordNet database.

3.3 Filtering

We cannot ignore common similarity measures such as Cosine similarity and Dice coefficient; however, these two criteria are not sufficient by themselves for comparing the conceptual similarity between words, which plays an effective role in summary plagiarism detection. Besides these two measures, we employed our customized BM25 text-to-vector conversion scheme; and for obtaining the semantic and structural similarities of words, we implemented similarity measures of 'wup' and 'Lin', which were applied after extracting the 'synsets' through 'WordNet' in our proposed system. Now in the filtering step, for combining the previously achieved results and computing the

final output (i.e. degree of similarity) Eq. 11 is proposed. Using the outcome of this equation and a defined threshold, it can be decided whether a particular sentence is plagiarized or not.

$$Score\_cbn = \frac{\alpha.WordNet_{score}}{\beta.(Cosine_{score} + Dice_{score})} > thr \tag{11}$$

In the above equation, $WordNet_{score}$ is the final score obtained by the proposed method based on 'wup' and 'Lin' criteria, αis the coefficient of impact in the proposed scheme, $Cosine_{score}$ is the score of Cosine measure, $Dice_{score}$ is the score of Dice measure, β is the influence factor of the combined scores of Cosine and Dice measures, and 'thr' is the threshold defined for detecting summary plagiarism cases. Since in summary plagiarism, normally, the words are replaced by their synonyms, and the length of a plagiarized sentence in a suspicious document is shorter than that in the source document, in most of these two cases, the Cosine and Dice scores underestimate the similarity value and therefore, a less reduced value of β should be applied. This is the reason the Cosine and Dice scores were summed up and put in the denominator of the proposed approach in order to lessen the degree of involvement of these two criteria in the final score; thus, separate coefficients were not defined for these two criteria. Conversely, since the score obtained from the extracted similarities in WordNet has a high value, the effect of the semantic and structural similarities of sentences is enhanced by using α. An important point to consider is that the increase and the reduction of both factors α and β should be done carefully. In other words, there should be a tradeoff between the sentences whose semantic similarities need to be detected and whose structures have to be analyzed and the sentences in which more words have been plagiarized. This has been explained more thoroughly in the Experimental Results section.

4. Experimental settings

In this section, the settings needed for evaluating the results are introduced. For this purpose, first, the data set used is presented and then the evaluation criteria are introduced.

4.1 Data sets

To evaluate the proposed summary plagiarism detection system, we used PAN-13 text corpus which is constructed based on webis-TRC-13 [28], and whose structure has been described by Potthast et al. [29]. This dataset includes simulated and intelligently plagiarized samples such as paraphrased items, and part of it contains summary obfuscation cases. This section contains 1185 pairs of suspicious and source documents as training data and 121 pairs of suspicious and source documents as test data in English. The sentences in these datasets were compared and the similarities between them were found by using the proposed system. The training data[1] were used to tune the system parameters and ultimately the proposed system was tested by the test dataset[2]. The results of this test have been presented in the Experimental Results section.

4.2 Evaluation measures

The performance of the proposed system was evaluated using 4 standard measures: Precision, Recall, Granularity, and PlagDet; which are used in plagiarism detection competitions [29]. The details of the mentioned criteria are as follows:

If d_{plg} is a plagiarism (real) case with 4 parameters $s =< s_{plg}, d_{plg}, s_{src}, d_{src} >$, where s_{plg} is the plagiarized phrase in document d_{plg} and s_{src} is phrase matching as original in source document d_{src}, and also if there is a detected plagiarism in document d_{plg} (what our program has detected) in the form of $r =< r_{plg}, d_{plg}, r_{src}, d'_{src} >$ and r claims that the phrase r_{plg} in document d_{plg} is a plagiarized phrase from r_{src} in d'_{src}, we can say that r has detected s if the conditions of Eq. 12 are true.

$$\begin{cases} r_{plg} \cap s_{plg} \neq \emptyset \\ r_{src} \cap s_{src} \neq \emptyset \\ d'_{src} = d_{src} \end{cases} \tag{12}$$

[1] http://www.uni-weimar.de/medien/webis/corpora/corpus-pan-labs-09-today/pan-13/pan13-data/pan13-text-alignment-training-corpus-2013-01-21.zip
[2] http://www.uni-weimar.de/medien/webis/corpora/corpus-pan-labs-09-today/pan-13/pan13-data/pan13-text-alignment-test-corpus1-2013-03-08.zip

Ultimately, sets of *S* and *R* are established, which respectively indicate the "set of plagiarized cases" (i.e., what exists in the real world) and "set of detected plagiarisms" (i.e., plagiarism cases detected by the designed program). Considering the abovementioned, and based on Eqs. 13 and 14, the values of Precision and Recall are defined as

$$prec_{macro}\ (S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(s\sqcap r)|}{|r|} \tag{13}$$

$$rec_{macro}\ (S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R}(s\sqcap r)|}{|s|} \tag{14}$$

where $s\sqcap r$ is expressed as Eq. 15.

$$s\sqcap r = \begin{cases} s\cap r & if\ r\ detects\ s \\ \emptyset & otherwise \end{cases} \tag{15}$$

After defining the Precision and Recall measures, the question would be whether, for a true plagiarism case $s \in S$, the designed program has been able to detect all the *s* or just part of it? Ideally, an algorithm should establish a one-to-one correspondence from Set *R* to the set of true cases *S*. In other words, the set of plagiarized cases detected by the program should include the whole set of true cases *S*. In order to consider this capability, we incorporate a criterion called 'Granularity', which is expressed by Eq. 16 as:

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_s| \tag{16}$$

$S_R \subseteq S$, which includes the detected plagiarism cases in Set *R*; and also $R_s \subseteq R$, which includes the detections of a given *s*. In other words, based on Eqs. 17 and 18, we will have:

$$S_R = \{s|s \in S \wedge \exists r \in R: r\ detects\ s\} \tag{17}$$

$$R_S = \{r|r \in R \wedge r\ detects\ s\} \tag{18}$$

The range of $Gran(S,R)$ is $[1,|R|]$, where 1 indicates the best-case scenario of one-to-one correspondence; and $|R|$ shows the worst case, in which $s \in S$ has been detected repeatedly.

Through the previously defined criteria (Precision, Recall, and Granularity), the value of the main criterion (i.e., Plagiarism Detection) is computed as follows:

$$plagdet = \frac{F_\alpha}{\log_2(1+gran\ (S,R))} \tag{19}$$

In Eq. 19, the value of F_α equals the weighted harmonic average of the values of Precision and Recall variables; which is expressed by Eq. 20:

$$F_\alpha = \frac{2\times rec\times prec}{rec+prec} \tag{20}$$

5. Experimental results

In this section, we focus on the results obtained from the system we have designed for improving the detection rate of summary plagiarism. For better understanding and easier analysis of the obtained results, we have tried to examine the separate effect of each proposed technique incorporated into the system besides the common Cosine and Dice similarity measures (without applying the filtering step). Finally, by integrating all the similarity measures and applying the filtering procedure, the ultimate system is evaluated.

For comparing the results of the proposed scheme, we have tried to use two plagiarism detection systems as the basis for comparison: 1) the system designed by Sanchez et al.[3] and 2) the Baseline system[4] presented in PAN-12, PAN-13 and PAN-14 competitions.

5.1 Evaluating the effect of BM25

Normally, plagiarism detection systems directly or indirectly use 'tf-idf' as their vector space model. For example, Nawab and Stevenson [30] used 'tf-idf' in their designed system in the Recall step. Also, inspired by the mentioned approach, Sanchez et al. implemented a scheme called 'tf-isf' as their vector space model. Instead of applying this common method, we implemented our proposed system using a customized BM25 scheme, and eventually, we were

[3] https://www.gelbukh.com/plagiarism-detection/PAN-2014
[4] http://pan.webis.de/clef12/pan12-code/pan12-text-alignment-baseline.py

able to achieve a considerable improvement in Recall and PlagDet scores (while the Precision value remained almost constant).

The improvement of the mentioned values is indicative of the fact that BM25, besides its effective role in ranking, has a good performance in plagiarism detection as well. This has been well demonstrated in Fig. 6 by comparing 3 systems (Baseline, Sanchez, and customized BM25). It should be mentioned that the constant parameters of BM25 equation, k_1 and b, have been considered as 1.2 and 0.9, respectively.

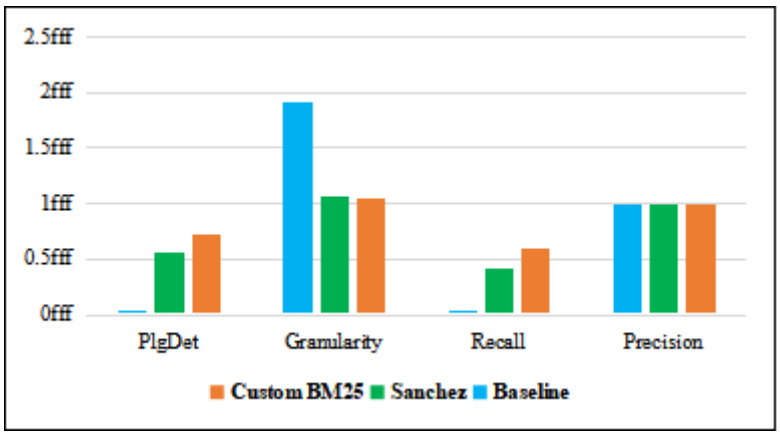

Figure 6. The effect of incorporating the BM25 scheme.

Based on what was discussed in Section (3-2-1) and also based on the implementation results, since the BM25 scheme is a more correct approach than the 'tf-idf' method in terms of limiting the frequency of highly-repeated words and it increases mildly, using the customized BM25 scheme considerably increased the Recall and PlagDet values. Overall, by implementing the mentioned scheme, it was concluded that by converting text to vector by the BM25 method, more plagiarized words will be found and semantic similarities will be measured more efficiently based on the results obtained from this scheme.

5.2 Evaluating the effect of WordNet

Based on the results and statistics presented in Fig. 7, as was expected, by implementing the WordNet and applying the 'wup' and 'Lin' similarity measures, the Recall, Granularity and PlagDet values increased substantially; which indicates the successful implementation of the mentioned scheme. It should be mentioned that the threshold value of 0.30 used in this work was determined through numerous trials on the training and testing text corpora and that the 'tf-idf' vector space was used in individual tests.

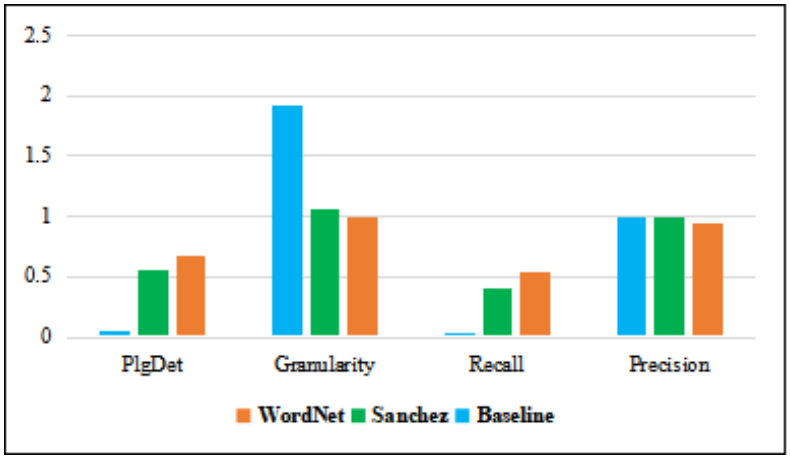

Figure 7. The effect of implementing the WordNet.

5.3 Analyzing the incorporated parameters

As was observed in the previous sections, the techniques used in our approach for improving the rate of summary

plagiarism detection were incorporated individually, and the effect of each one was analyzed and discussed separately.

In view of Fig. 8, by examining the Precision, Recall, Granularity, and PlagDet criteria for the customized BM25, WordNet, Sanchez, and Baseline methods, it is concluded that the customized BM25 scheme performs better than the other 2 methods in all the considered measures of similarity, followed by the use of WordNet, which has a good effect on Precision and ultimately on the overall summary plagiarism detection criterion.

The important conclusion deducted from these experiments is that designing and using only one plagiarism detection scheme is not sufficient and that a combination of various methods should be employed. This issue is even more crucial in summary plagiarism detection; because in this type of plagiarism, in addition to the shortened length of text in a suspicious document (which reduces the scores of the Precision and especially the Recall measures), the words might be replaced by their synonyms (which requires semantic analysis of sentences). Therefore, using just one technique is not enough. Based on the findings so far, we will describe our proposed summary plagiarism detection technique in the next section, hoping to achieve more success through this method.

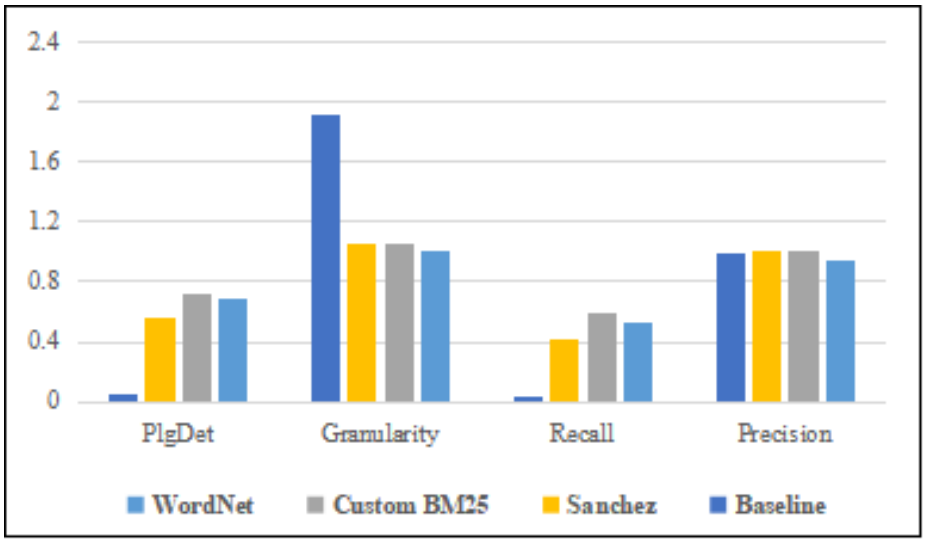

Figure 8. Comparing the individual criteria entering the system.

5.4 The filtering step (the proposed aggregation approach)

As was previously mentioned, our proposed method includes a combination of several techniques; which have been modified and implemented according to our approach. This combination is achieved according to Eq. 11, by combining the Cosine and Dice similarity measures and the computed scores of WordNet based on 'wup' and 'Lin' criteria. In implementing our scheme, the system was executed by using the default parameters of Table 2 (which are related to the phrase selection step), and finally, the filtering step was also applied to the system. It should be mentioned that the depth of the sentence means the number of sentences selected from a source or suspicious document for similarity measurement purposes.

Table 2. The list of default parameters used

Parameters	Values	Descriptions
th1	0.33	Threshold of cosine similarity measure
th2	0.33	Threshold of dice similarity measure
src_gap	4	Depth of selective sentences (source document)
src_gap_least	2	Minimum depth of selective sentences (source document)
susp_gap	4	Depth of selective sentences (suspicious document)
susp _gap_least	2	Minimum depth of selective sentences (suspicious document)

During implementation, it was realized that excessively increasing the effect of WordNet measure reduces the power of plagiarism detection. Conversely, the excessive increase of the sum of Cosine and Dice similarity measures has a similar effect. Thus, experientially, the coefficients of α and β were set as 4 and 2, respectively; and the threshold value was considered as 1.5.

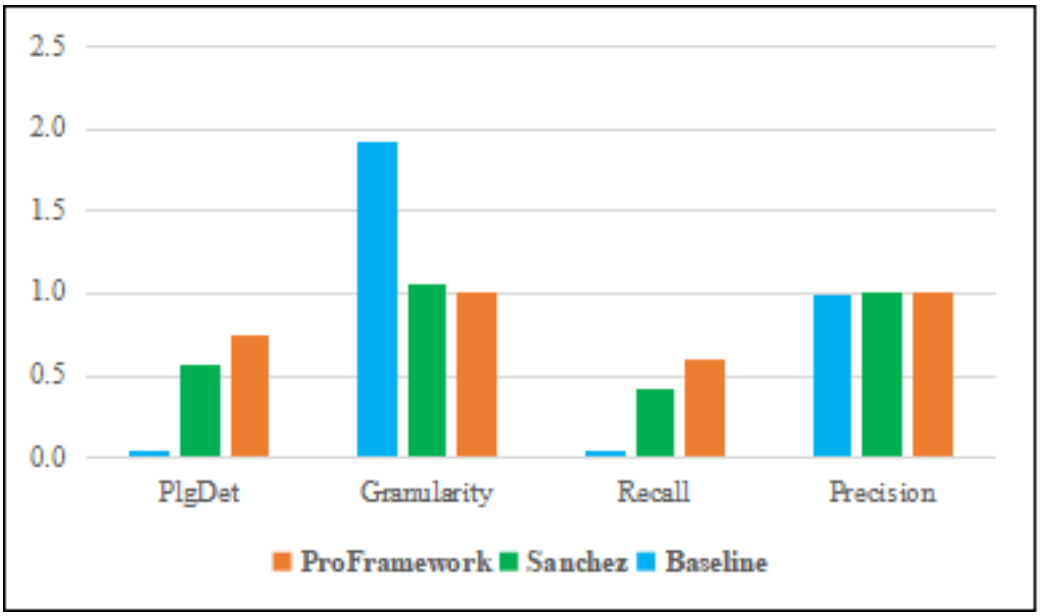

Figure 9. The results of implementing the proposed method with default parameters.

Based on the results displayed in Fig. 9, by keeping the Precision almost constant, the value of Recall increases considerably. Also, the Granularity measure, which depends on the correct detection of the location of plagiarized phrases, has a satisfactory score (in this criterion, a reduction of score means better performance). In total, the abovementioned improvements significantly increase the efficiency of the proposed summary plagiarism detection system.

In the next step of implementation, by changing some of the mentioned default parameters, the program is executed using the parameters in Table 3. Through numerous experiments, it was found that by changing the values of src_gap, src_gap_least, susp_gap, and susp_gap_least and Cosine similarity, the percentage of summary plagiarism detection cases increases.

Table 3. The list of modified parameters

Parameters	Values	Descriptions
th1	0.33	Threshold of cosine similarity measure
th2	0.33	Threshold of dice similarity measure
src_gap	9	Depth of selective sentences (source document)
src_gap_least	4	Minimum depth of selective sentences (source document)
susp_gap	9	Depth of selective sentences (suspicious document)
susp _gap_least	4	Minimum depth of selective sentences (suspicious document)

In view of Fig. 10, this research has concluded that in the summary plagiarism detection process, if sentences are selected with more depth in the phrase selection step and their similarities are computed more accurately, the Recall and PlagDet scores will improve. In this work, the values of Precision, Recall, and 'PlagDet' were obtained as 0.98, 0.65, and 0.78, respectively. The improved proposed method was also compared with the Sanchez and Baseline systems, and the results of these comparisons have been presented in Table 4.

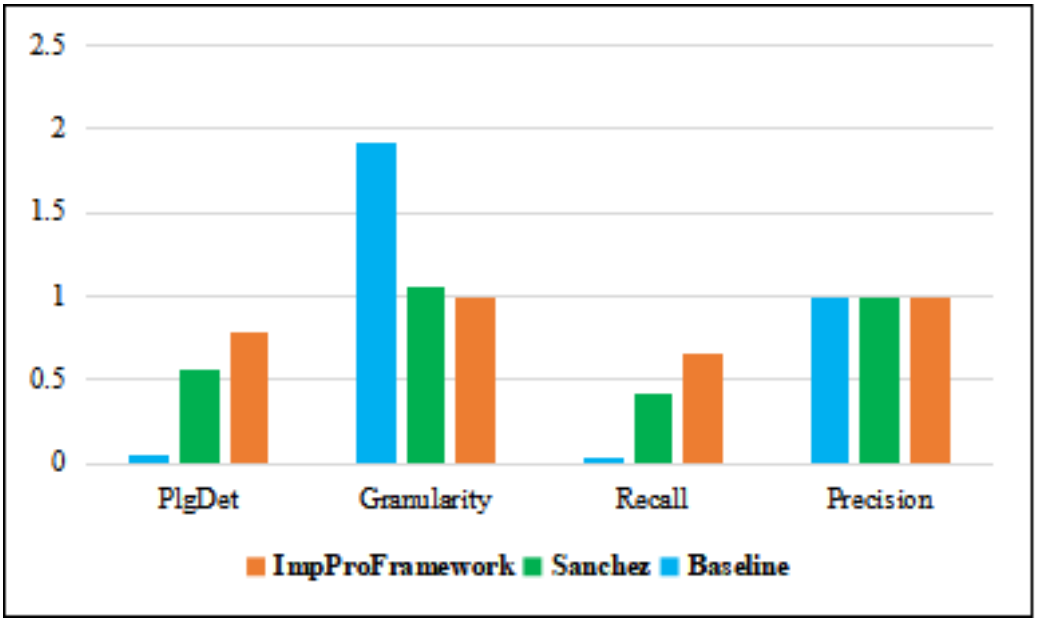

Figure 10. The results of implementing the proposed method by changing some of the default parameters.

Table 4. Comparing the improved proposed method with the Sanchez and Baseline systems

	Baseline (%)	Sanchez (%)
Precision	---	---
Recall	1730	59
Granularity	48	6
PlagDet	1658	41

6. Conclusion and future works

In this research, a summary plagiarism detection system composed of 3 main steps was proposed, which uses semantic similarity techniques to compare and measure the similarities in the texts of source and suspicious documents and, by accumulating the scores of several similarity detection measures, finds intelligently plagiarized phrases. Since in most summary plagiarized cases, the words are replaced by their synonyms, the goal was to structurally analyze the examined sentences to better detect the semantic similarities of words. For this purpose, different procedures were undertaken such as implementing a customized vector space model, finding word synonyms through a foreign language dictionary and comparing the structural similarities of sentences. The proposed system was also tested on PAN-13 text corpus.

In the implementation step, the values of Recall and PlagDet were improved by using the proposed customized BM25 vector space model; and by employing the WordNet and applying the 'wup' and 'Lin' similarity measures, the Recall, Granularity and PlagDet scores were increased substantially. Finally, through a complete implementation of the proposed system and making some changes in the depth of selected sentences, 78% of the summary plagiarized texts were detected correctly.

Important advances have been made in the automation of the plagiarism detection process, but still, there is no comprehensive system that can precisely detect plagiarized documents. To deal with this problem, and as future work, this research can be expanded to improve the plagiarism detection accuracy and to bring it closer to the precision of human judgment; so that ultimately, an intelligent automatic system is achieved that can replace manual methods of plagiarism detection. To realize this goal, we propose the use of fuzzy similarity measures and also the fuzzification of the existing criteria to substantially improve the efficiency of detection systems. Also, considering the huge volume of reference documents that have to compare with a suspicious document, the use of parallel techniques is recommended to speed up the computations.

References

[1] M. S. Anderson and N. H. Stenec. (2011). “The problem of plagiarism,” in Urologic oncology: Seminars and original investigations, 2011, vol. 29, no. 1, pp. 90-94.

[2] M. Muhr, M. Zechner, R. Kern, and M. Granitzer. (2009). “External and Intrinsic Plagiarism Detection using Vector Space Models,” Sepln 2009, pp. 47-55, 2009.

[3] G. Oberreuter and J. D. Vel squez. (2013). “Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style,” Expert Syst. Appl., vol. 40, no. 9, pp. 3756-3763, 2013.

[4] J. D. Velásquez, Y. Covacevich, F. Molina, E. Marrese-Taylor, C. Rodríguez, and F. Bravo-Marquez. (2016). “DOCODE 3.0 (DOcument COpy DEtector): A system for plagiarism detection by applying an information fusion process from multiple documental data sources,” Inf. Fusion, vol. 27, pp. 64-75, 2016.

[5] A. M. El Tahir Ali, H. M. Dahwa Abdulla, and V. Snášel. (2011). “Overview and comparison of plagiarism detection tools,” in CEUR Workshop Proceedings, 2011, vol. 706, pp. 161-172.

[6] U. N. Dulhare, K. Ahmad, and K. A. Ahmad. (2020). “Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications”. John Wiley & Sons, 2020.

[7] N. Shenoy and M. A. Potey. “Semantic Similarity Search Model for Obfuscated Plagiarism Detection in Marathi Language using Fuzzy and Naïve Bayes Approaches,” IOSR J. Comput. Eng. e-ISSN, pp. 661-2278.

[8] G. Oberreuter, G. L’Huillier, S. A. Rıos, and J. D. Velásquez. (2011). “Approaches for intrinsic and external plagiarism detection,” Proc. PAN, 2011.

[9] M. Zechner, M. Muhr, R. Kern, and M. Granitzer. (2009). “External and Intrinsic Plagiarism Detection Using Vector Space Models,” Proc. 3rd Work. Uncovering Plagiarism, Authorsh. Soc. Softw. Misuse 1st Int. Compet. Plagiarism Detect., pp. 47-55, 2009.

[10] K. Kohler and D. Weber-Wul. (2010). “Plagiarism detection test 2010,” Technical report, HTW Berlin, 2010.

[11] C. Grozea and M. Popescu. (2010). “Who’s the thief? Automatic detection of the direction of plagiarism,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010, vol. 6008 LNCS, pp. 700-710.

[12] C. Vania and M. Adriani. (2010). “Automatic external plagiarism detection using passage similarities,” in CEUR Workshop Proceedings, 2010, vol. 1176.

[13] S. L. Devi, P. R. K. Rao, V. S. Ram, and A. Akil. (2010). “External Plagiarism Detection Lab Report for PAN at CLEF 2010,” 2010.

[14] A. Ekbal, S. Saha, and G. Choudhary. (2012). “Plagiarism detection in text using Vector Space Model,” in Hybrid Intelligent Systems (HIS), 2012 12th International Conference on, 2012, pp. 366-371.

[15] R. Naseem and S. Kurian. (2013). “Extrinsic plagiarism detection in text combining vector space model and fuzzy semantic similarity scheme,” Int. J. Adv. Comput. Eng. Appl. (IJACEA), ISSN, 2013.

[16] M. A. Sanchez-Perez, G. Sidorov, and A. F. Gelbukh. (2014). “A Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014.,” in CLEF (Working Notes), 2014, pp. 1004-1011.

[17] S. F. Hussain and A. Suryani. (2015). “On retrieving intelligently plagiarized documents using semantic similarity,” Eng. Appl. Artif. Intell., vol. 45, pp. 246-258, 2015.

[18] A. Abdi, S. M. Shamsuddin, N. Idris, R. M. Alguliyev, and R. M. Aliguliyev. (2017). “A linguistic treatment for automatic external plagiarism detection,” Knowledge-Based Syst., vol. 135, pp. 135-146, 2017.

[19] Y. Palkovskii and A. Belov. (2014). “Developing High-Resolution Universal Multi-Type N-Gram Plagiarism Detector,” Work. Notes Pap. CLEF 2014 Eval. Labs, pp. 984-989, 2014.

[20] “2. Accessing Text Corpora and Lexical Resources.” [Online]. Available: http://www.nltk.org/book/ch02.html. [Accessed: 23-Nov-2017].

[21] O. Vechtomova. (2009). “Introduction to Information Retrieval Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (Stanford University, Yahoo! Research, and University of Stuttgart) Cambridge: Cambridge University Press, 2008, xxi+ 482 pp; hardbound, ISBN 978-0-521-8.” MIT Press, 2009.

[22] Doug Turnbull. (2015). “BM25 The Next Generation of Lucene Relevance,” 2015. [Online]. Available: http://opensourceconnections.com/blog/2015/10/16/bm25-the-next-generation-of-lucene-relevation/. [Accessed:

24-Nov-2017].

[23] H. A. N. Jiawei, K. Micheline, and M. Data. (2007). “Concepts and Techniques.” Morgan Kaufmann, 2007.

[24] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. (1990). “Introduction to WordNet: An on-line lexical database,” Int. J. Lexicogr., vol. 3, no. 4, pp. 235-244, 1990.

[25] D. Lin. (1998). “An information-theoretic definition of similarity.,” in Icml, 1998, vol. 98, no. 1998, pp. 296-304.

[26] Y. Jiang, W. Bai, X. Zhang, and J. Hu. (2017). “Wikipedia-based information content and semantic similarity computation,” Inf. Process. Manag., vol. 53, no. 1, pp. 248-265, 2017.

[27] N. Seco, T. Veale, and J. Hayes. (2004). “An intrinsic information content metric for semantic similarity in WordNet,” in Proceedings of the 16th European conference on artificial intelligence, 2004, pp. 1089-1090.

[28] M. Potthast, M. Hagen, M. Völske, and B. Stein. (2013). “Crowdsourcing Interaction Logs to Understand Text Reuse from the Web.,” in ACL (1), 2013, pp. 1212-1221.

[29] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso. (2010). “An evaluation framework for plagiarism detection,” in Proceedings of the 23rd international conference on computational linguistics: Posters, 2010, pp. 997-1005.

[30] R. M. A. Nawab, M. Stevenson, and P. Clough. (2017). “An IR-Based Approach Utilizing Query Expansion for Plagiarism Detection in Medline,” IEEE/ACM Trans. Comput. Biol. Bioinforma., vol. 14, no. 4, pp. 796-804, 2017.