Designing Method and Algorithm of Semantic Comparison for Detecting Similarities in the Case of Fuzzy Information using Software Engineering and Medical Ontologies*

Abdualmajed Ahmed Al-Khulaidi, PhD
Assistant Professor of Software Engineering, Sana’a University
Yemen

Adel A. Nasser, PhD
Assistant Professor of Information System, Sa’adah University
Sa’adah –Yemen

Fahd Nasser Al-Wesabi, PhD
Assistant Professor, College of Computer and IT, Sana'a University
Sana'a -Yemen

ABSTRACT

In this article we will discuss the work of a semantic comparison method, through which the detection of plagiarism is revealed in the fuzzy information, which we have designed an algorithm with a semantic dimension to detect plagiarism in the fuzzy information and detect impersonations such as changing the structure of speech or replacing words with synonyms and limiting technical spelling errors such as not completely writing the end of the word or unofficial and unknown abbreviations, and analyze shows the degree of similarity of the original text, and analyzes the overall evaluation of the degree of similarity of texts from the apparent structures of the text. Experiments have shown that the proposed method with a semantic dimension in the case of fuzzy information is better than sherlock method in terms of file size criterion of 6% if using word synonyms in the file and 1% in case of rewriting the file. As for the standard time taken to examine the files through the acceleration calculation, it is noted that the proposed method for the semantic dimension in the case of fuzzy information is faster in performance than the Sherlock method in the case of the use of synonyms 1.02 times and in the case of rewriting words with a value of 1.01 times in the case of file size 382 words. The results of the experiments show that the average execution time of the proposed algorithm, for finding plagiarism, is less by 3.47% compared with the Sherlock algorithm in the case of the use of synonyms and less by 1.83% compared with the Sherlock algorithm in the case of rewriting words. The algorithm works effectively as the file size increases, the gain ratio is obtained up to 2.73% in the synonym of words and 2.69% in the case of rewriting words. From the results presented in the tables, we conclude that the average error rate of the proposed algorithm is 2% lower than the error rate sherlock algorithm. The complexity of the proposed algorithm is O(m*n).

General Terms

plagiarism detection, medical ontologies, software engineering ontology, Semantic network .

Keywords

Plagiarism detection, medical ontologies, software engineering ontology, Semantic network.

1. INTRODUCTION

Recently, a bad phenomenon emerged from some people, namely the phenomenon of scientific plagiarism of what scientists and researchers wrote of modern ideas and things, and then attributed these writings and research to them. To solve this problem, some researchers and scientists have tackled ways to eliminate the phenomenon of scientific plagiarism. Many tricks and techniques are taken and developed by some people to deceive in order to practice plagiarism without being detected. Therefore, in this research, a highly efficient method was compared to the previous methods which will be discussed in this research[2]. Define scientific plagiarism as stealing ideas or writings of others and self-sufficiency without mentioning the scientific sources in the research. Types of plagiarism can be summarized as follows[7]:

Reproduction, in which the work of others is fully served as an act of the individual.Copies, in which large parts of a specific source are copied without mentioning the source.

Replacement, in which a piece of text is copied after changing some keywords while preserving the basic information of the source and not pointing to it.
Blending, blending parts of many sources without mentioning them.Repetition, copying of the previous individual's writings without mentioning them.Mix, merge text clips that are correctly exported with other sections not mentioned. In order to prevent plagiarism, must come up with original ideas, especially in our scientific research and referencing to the work that others have done through the references you add to your research. The plagiarism has been classified into two categories according to the location of the electronic documents, namely an internal group and an external group. The internal group is in the case of both the original documents and the plagiarized work that are within the same group as the student articles collection or an electronic library. The external group is in the case of the original document and the plagiarized works are on different groups or sources, where the original documents may be a book or web documents. In order to prove the type of plagiarism the source of the original documents must be specified[11]. Impersonation of texts was also detected using syntactic technique where the similarity in characters, words, or sentences are measured using similarity measures. For example, if X is the word "abdualmajed" and Y is the word "abdul", then X = {a, b, d, u, a, l, m, a, j, e, d} and Y = {a, b, d, u, l}, so there are five common letters between the words X and Y. X∩Y={a,b,d,u,l}. Note that these common characters represent 7/11 = 63.6% of the letter X and represent 5/5 = 100% of the letter Y.The issue of plagiarism in medical research and related research in the field of computer, especially software engineering, is a complex one and concerns the exact area of research. In this research, will address the design of a method and algorithm based on the use of two types of ontologies, namely, the world's most famous medical ontologies , the most famous of which are the ontology of diseases that contain disease characterization, its characteristics, its definition and the derivation of diseases. The second type is the related ontologies in the field of software engineering.

2. EXISTING DETECTION PLAGIARISM ALGORITHMS AND METHODS

This research includes a study of existing detection algorithms and their types. In general, we can classify plagiarism detection algorithms into two main categories:Finger Printing algorithms that generate fingerprints for files to compare, where the fingerprint may be a letter-level fingerprint, a word-level fingerprint or a sentence-level fingerprint. This algorithm generates a fingerprint for files and then compares these fingerprints algorithms just like the Winnowing algorithm and the Longest Shared String algorithm. The fingerprint-based impersonation detection algorithm is a weak algorithm and the reason for this is that it is heavily affected by the rearrangement of words in the text, and it cannot replace words with its synonyms, so it must rely on algorithms that take synonyms into account.Comparative algorithms based on comparison of file content : Algorithms for comparing text strings and algorithms for comparison of tree structures of files(see Figure 1).

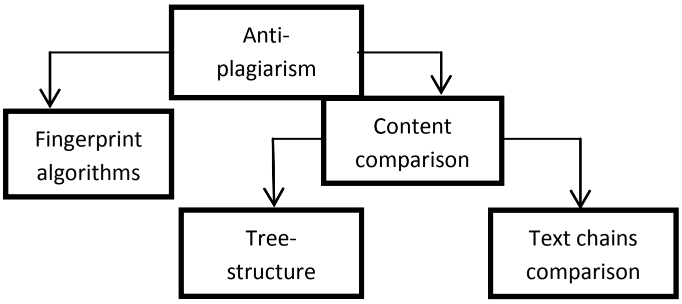


Fig 1: Explains phishing detection algorithm classes.

In general, there are flaws in the detection in the algorithms mentioned, where fingerprint algorithms and text-chains comparison algorithms are affected by word reordering hence they fail to detect plagiarism when reordering words, also text-chain comparison algorithms suffer from a variable factors, so if different variables are passed the result chain length which is used for comparison between the files , so the value of this parameter must be accurately determined by the nature of the files that compare them, it also can't detect cases where synonyms are exchanged[10],[13]. The tree structures comparison algorithms are also affected by the ambiguity in natural languages. This uncertainty leads to the generation of more than one tree representing the same tree. Because of these weaknesses in the algorithms and methods of detecting plagiarism, the researchers began to develop algorithms and methods of detecting plagiarism with a semantic dimension, which relies primarily on Web technologies. The most famous of these algorithms is the Citation Patterns algorithm, which detect the plagiarism by analyzing the models of citation and data quotes in the files and compares these data. Some data may also be vague (vague and incomprehensible) during data analysis which cannot be easily detected, such as extra characters, abbreviated words, misspelled words, spaces, etc, thus in this research new methods have been created to detect the impersonation of this inaccurate fuzzy information by making a comparison dictionary which can be used for detecting plagiarism and correcting erroneous and incomprehensible words[3]. Solving this problem in the research will be classified into the category of artificial intelligence and expert systems. The design of the general chart shown for this method is an input to detect the level of the significance of the match, in this step matching is done between the files using semantic comparison hypotheses. The general scheme for comparison of files in terms of convergence is illustrated in Figure (2).

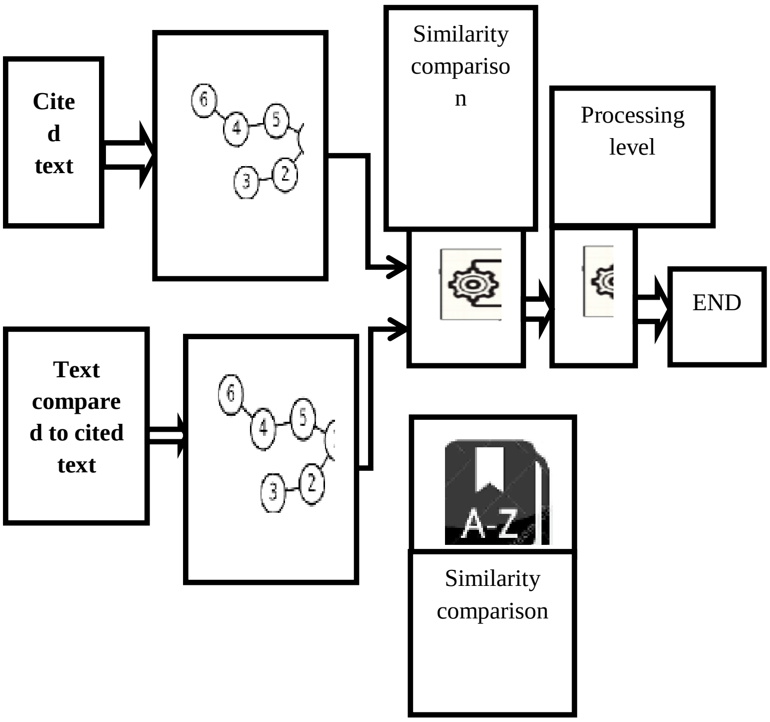


Fig 2:Shows the general schema for comparing files in terms of matching

The aim of this research is to work out a method and algorithm for detecting impersonation with a semantic dimension, relying on general anthropologies, especially in the field of medicine and field software engineering as well as the evidence base through which we conduct the local testing process. This algorithm is used to detect plagiarism in research papers in the medical field and in software engineering because it has a code because it relies on semantic networks. The main objective of this research is to design a method and algorithm to detect plagiarism using a semantic dimension by overcoming difficulties and eliminating weaknesses Semantic Ontology, which suffers from traditional detection algorithms. The reason is that semantic etiology contains valuable and rich information for concepts that include concepts and concepts that carries the same meaning.

2.1 Analysis of methods for determining fuzzy duplicates for documents:

2.1.1 Sherlock Algorithm

Sherlock algorithm is an algorithm for detecting plagiarism by comparing similarity between one sentence with other sentence. Sherlock algorithm indicates that if there are two sentences which have different sets of keywords then these two sentences have different content. The opposite is if two sentences have same sets of keywords then these two sentences have same content. Detection process is done by comparing each sentence in one document with each other sentence in another document [5].

2.1.2 The algorithm of Levenshtein

The algorithm of Levenshtein uses operations "replacement", "insert", "delete". They allow to search the distance between strings, different by length. But time of calculation of distance between strings is disproportionately increases with increases of strings size[1]. Therefore, the use of this algorithm is only suitable for comparing multiple pages of documents[21].

2.1.3 The algorithm of Wagner and Fisher

This method is based on calculating the Levenshtein distance between the strings prefixes (substrings).The matrix of editorial prescription is made. It contains a summary value of Levenshtein distance(minimum weight operations to change characters)[22]. The size of editorial prescription matrix is(p+1)•(b+1), where p and b – compared strings prefixes.

The number of string comparisons is k•p•b, where kcoefficient (for natural language k=0,2). Thecomplexity of algorithm is O(p•b) [23]. This method is the easiest way to create of editorial prescription The algorithm of Wagner and Fisher [22].This method is based on calculating the Levenshtein distance between the strings prefixes (substrings).The matrix of editorial prescription is made. It contains a summary value of Levenshtein distance (minimum weight operations to change characters). The size of editorial prescription matrix is (p+1)•(b+1), where p and b – compared strings prefixes. The number of string comparisons is k•p•b, where k – coefficient (for natural language k=0,2). The complexity of algorithm is O(p•b) [23]. This method is the easiest way to create of editorial prescription.

2.1.4 Linear search algorithm

The algorithm uses the distance metric and applies it to the text words. Efficiency of the method depends on the number of errors and mismatches of texts. More numerous they are the more increases the comparison. The algorithm complexity is O(s•p), where s – the number of errors made when checking, p – the text length[24].

2.1.5 The Bitap algorithm

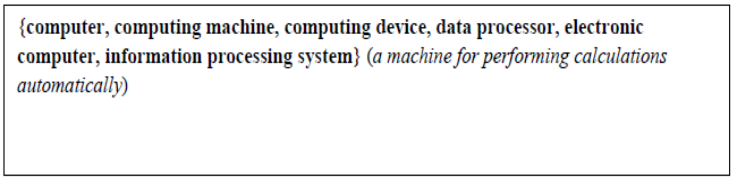
This algorithm is applied more than the linear search. In view modifications it calculates the Levenshtein distance between words. At normal conditions, speeds of these two algorithms are the same. The algorithm complexity is O(s•p), but the speed of this algorithm significantly higher on long words than a linear method[25].

3. RESEARCH METHODOLOGY

This research is based on general ontology wordnet, medical ontologies and related computer ontologies in software engineering. Among the most famous computer ontologies are[32] ,[33]:

1. Artificial Intelligence Ontology.
2. Web Semantic Ontology.
3. Systems Engineering Ontology.
4. Software Engineering Ontology.
5. Biomedical Informatics Ontology .

Here is a simple overview of the WordNet Ontology:
WordNet is a large English language database in the Internet where names, actions, attributes and circumstances are grouped into sets of Synsets. These concepts are intertwined with rookie relations. This database is used in computational linguistics and natural language processing [14]. Where the relationship between keywords in WordNet is synonymy. Figure (3) shows how the group is represented within this ontology.



{computer, computing machine, computing device, data processor, electronic computer, information processing system} (*a machine for performing calculations automatically*)

Fig 3: An example of a synset in WordNet

Among the most famous medical ontologies are[6],[27],[28] ,[29],[30],[31]:

1. Anatomy ontology.
2. Diseases ontology.
3. Gene ontology.
4. Mesh ontology.
5. Ontology for General medical science.
6. EDAM ontology .

3.1 The proposed method for detection of plagiarism with a semantic dimension

Assume that the semantic schemas $s_t, s_q$for the text portions q, t. Then the criterion of approximation α for the semantic data diagram is to be found by:

$$\left\{\begin{matrix} \propto \left(s_q, s_t\right) = \left(s_{q\approx} s_t\right) \\ \propto \left(s_q, s_t\right) \in \Delta \\ \Delta = [0..1] \end{matrix}\right\} \quad (1)$$

Where the symbol $\approx$ indicates the process of convergence. The symbol Δ indicates the set of values of the convergence criterion. In the case of $\propto \left(s_q, s_t\right) = 1$, that means there is an integrative convergence. In the case of $\propto \left(s_q, s_t\right) = 0$, this means that there is no convergence. Where there are criteria through which the comparison process is the basic standards and semantic standards. Basic criteria for comparing convergence by calculating similar words in text during comparison for query.

$$\emptyset_{Base} = \frac{p}{q} \qquad (2)$$

where p is the number of similar words in the text portions during the query.-q The number of words in the query. Two words are identical if their primary forms coincide.
The semantic standards are compared to sentences and not only the calculation of words in the text, only in relation to the criterion, as well as the relationship between the words during the comparison. For example, the semantic comparison criterion for convergence.

$$\emptyset_{Semantic} = \frac{m}{n} \qquad (3)$$

Where m - the number of matched items in the meaning of the query and the text portions.n- The total number of meaning elements in the query.We will build on the light of equation (1) in order to calculate the semantic convergence with the help of semantic classes of the wordNet, as well as special algorithms to implement the semantic comparison of the semantic schemas of the texts.The idea of the method is that the text fragments are textual sections with semantic content and not any parts.

The difference between the proposed criterion of semantic proximity of textual sections of the standard and the compared text is the calculation of the proportion of similar elements of meaning, in accordance with the semantic class of words involved in the comparison.

$$\emptyset_{Semantic\ /class} = \frac{\sum_i^k \frac{\sum p_j}{l}}{n} \qquad (4)$$

where :

-p is the coincidence factor between the words involved in the comparison for each element of the meaning,according to the semantic class in the range [0,1].

p = 1 - if the word is identical.

p = 0 - if the word is outside the semantic class and p = (0,1) depending on the degree of synonymy;

l -is the number of words of each element of the meaning.

k - the number of elements of meaning in the text section of the compared text.

n -is the total number of elements of meaning in text section of the standard.

It is necessary for the expert to pre-determine the degree of synonymy of each semantic class.

At the level of representation of textual sections in semantic schemes, we get the number of n-schemes of the reference sections and the number of m-schemes of the sections of the compared text, which will be compared in the ratio of **n** to **m**, but matches will be counted in the total number n, regardless of the number of schemes of the text being compared, so that if one scheme has matches with more than one scheme of another text, this will be considered the main factor of coincidence. For each iteration of the comparison, the following prerequisites are considered:Each semantic scheme of textual sections , both from the sample text and the compared text, has characteristics and they are obtained in accordance with the process.

An example of these schemas is shown in Figure (4).For all schematic schemes in Figure (4), their class components possess the following:Zero layer level is therefore symbolized {s (a), s (b), s (c), s (d), s (e), s (f), s (g), s (h)}.The level of the first layer is represented by the following rings {r1, r2, r3}.The level of the second layer is represented by the following rings {r4, r5, r6}.The level of the third layer is represented by the following rings {r7}.The level of the fourth layer is represented by the following rings {r8}.The level of the fifth layer is represented by the following rings {r9}.The level of the sixth layer is represented by the following rings {r10}.

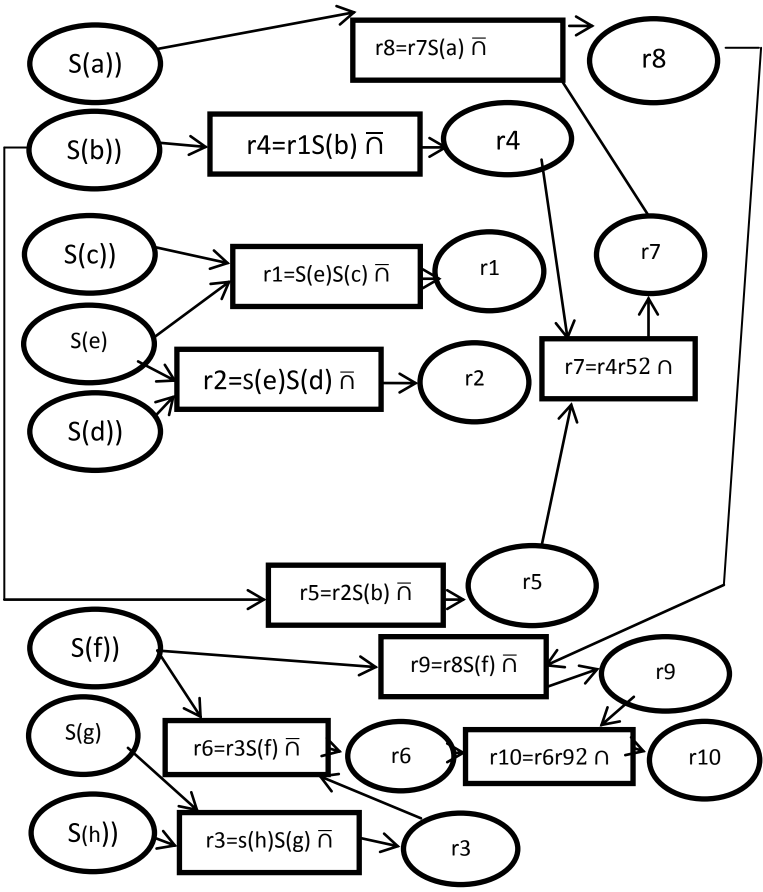
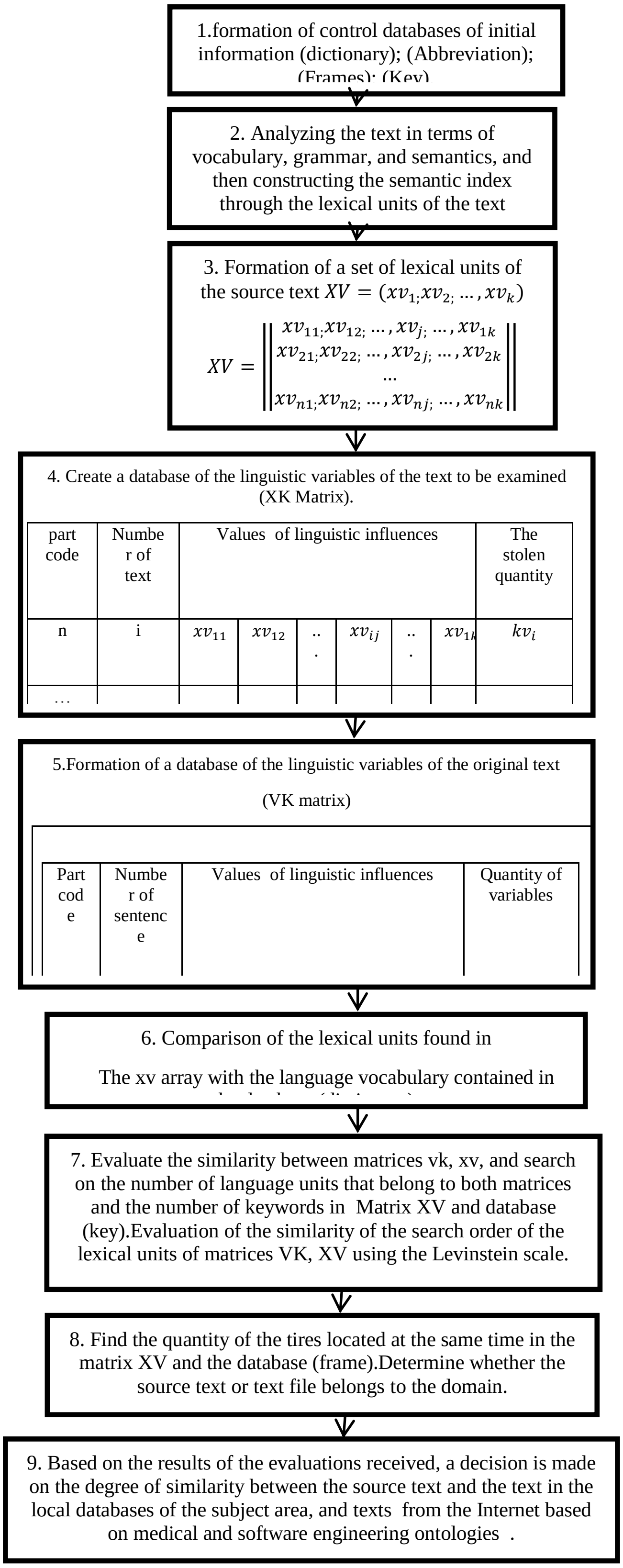


Fig 4: The example shows the semantic schemes in section texts

Every word in the text of an interdependent model text section in the list of words that belong to semantic layers and compatible with WordNet and the ideologies that are related to software engineering field and medical field.For the value of each element in the semantic schemes section texts participate in the comparison, the coefficient of comparison is p , obtained from the average values of the matching words in the semantic schemes comparative text for the words in the text of the Model Law.The degree of similarity between the sections texts will be obtained through the equation (1).

3.2 The proposed algorithm to detect plagiarism of the semantic dimension



1.formation of control databases of initial information (dictionary); (Abbreviation); (Frames); (Key).

2. Analyzing the text in terms of vocabulary, grammar, and semantics, and then constructing the semantic index through the lexical units of the text

3. Formation of a set of lexical units of the source text $XV = (xv_{1;} xv_{2;} \dots, xv_k)$

$$XV = \left\| \begin{matrix} xv_{11;} xv_{12;} \dots, xv_{j;} \dots, xv_{1k} \\ xv_{21;} xv_{22;} \dots, xv_{2j;} \dots, xv_{2k} \\ \dots \\ xv_{n1;} xv_{n2;} \dots, xv_{nj;} \dots, xv_{nk} \end{matrix} \right\|$$

4. Create a database of the linguistic variables of the text to be examined (XK Matrix).

part code	Number of text	Values of linguistic influences						The stolen quantity
n	i	xv_{11}	xv_{12}	...	xv_{ij}	...	xv_{1k}	kv_i
…								

5.Formation of a database of the linguistic variables of the original text

(VK matrix)

Part code	Number of sentence	Values of linguistic influences	Quantity of variables

6. Comparison of the lexical units found in

The xv array with the language vocabulary contained in local database (dictionary)

7. Evaluate the similarity between matrices vk, xv, and search on the number of language units that belong to both matrices and the number of keywords in Matrix XV and database (key).Evaluation of the similarity of the search order of the lexical units of matrices VK, XV using the Levinstein scale.

8. Find the quantity of the tires located at the same time in the matrix XV and the database (frame).Determine whether the source text or text file belongs to the domain.

9. Based on the results of the evaluations received, a decision is made on the degree of similarity between the source text and the text in the local databases of the subject area, and texts from the Internet based on medical and software engineering ontologies .

Fig 5:The proposed plagiarism detection algorithm

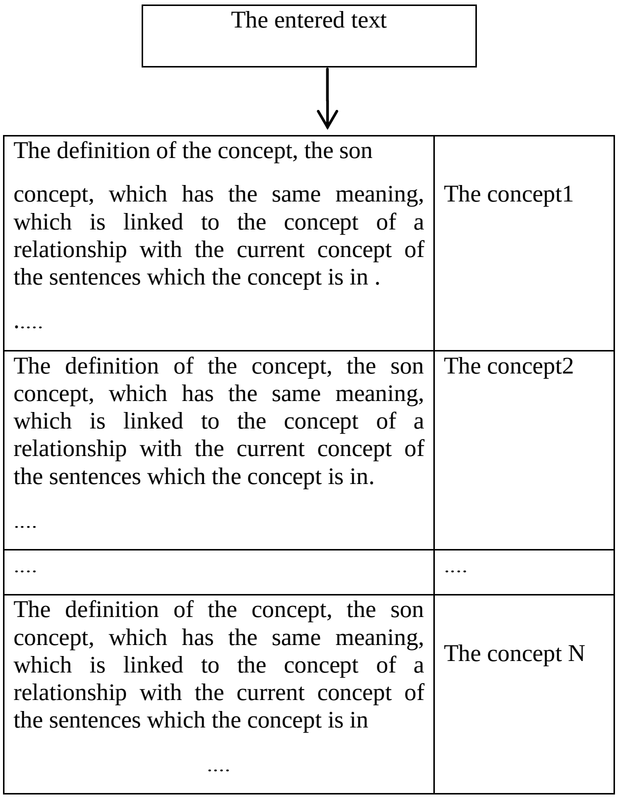
We will explain here the second paragraph of the proposed algorithm based on the local database as well as related ontologies in the field of software engineering and medical field. The algorithm proposed complexity is O(m*n) , where : m- Number of index keys in the first file. n-Number of index keys in the second file. A comparison of the semantic indexes resulting from the analysis process is carried out. All the concepts in the original text are compared to the concepts in the semantic indexes of the suspect text, whether the comparison is based on the local database or the Internet ontologies related to the medical field and software engineering field.

3.3 Analysis of the text in terms of vocabulary, grammar, and semantics

In this stage, the analysis of the extensive analysis of the vocabulary of the text where the algorithm first remove useful words of text (stop-words) using the list of Unhelpful words in the English language. Then the algorithm to determine the names and actions in the text and identifies some relations with names such as described utilize the Stanford, to address natural languages. After that obtaining the names and acts and relations between the names) Stanford tools are used to address natural languages to define relations between the names), where the algorithm determines the names of the vehicle, which reflect the concept or concepts consisting of more than one word in the text. After obtaining the names of the concepts, consisting of one word or composed of several words of the search for the meanings of these concepts in the local database or the Internet ontologies related to the medical field and software engineering field. After determining the individual and the compound concepts , The construction of indexes of the text get built by :

1- A special catalog of the concepts (concept index) : This catalog contains the keys and values, the key element of this contents is the concept and the value of this element is information associated with this concept, as it was building a semantic layer.

2- A special catalog of the ontologies (ontology index) : Where the keys are representing these types of ontological contents and the concepts that belong to that ontologies values of this contents of the index . After the completion of this phase of the algorithm, each of the original versions and suspected by the representative of the two indexes and contains all of these two indexes on the lexical units and semantic concepts and values related to it in addition of the text, including the semantic concepts .The form shows how the representation of the text after treatment and building its own indexes.



The entered text

The definition of the concept, the son concept, which has the same meaning, which is linked to the concept of a relationship with the current concept of the sentences which the concept is in	The concept1
The definition of the concept, the son concept, which has the same meaning, which is linked to the concept of a relationship with the current concept of the sentences which the concept is in.	The concept2
....	....
The definition of the concept, the son concept, which has the same meaning, which is linked to the concept of a relationship with the current concept of the sentences which the concept is in	The concept N

Fig 6: Explains how the representation of the text in the concepts index after the completion of the analysis phase of the semantic

the semantic indexes are compared, while passing on each of the concepts in the original text and comparing the existing concepts in the semantic indexes of the suspected text.

And then show the results of plagiarism by showing the phrases that contain similar semantic concepts , Where the marking of the phrases that contain a convergent concepts and shared indicative of a different color to allow the user to note a plagiarism easily and without accurate tracking and entirely read the suspected files

4. THE EXPERIMENTS USING THE PROPOSED PLAGIARISM DETECTION ALGORITHM

The detection plagiarism algorithm is applied in a system and it measures the similarity between the texts through the process of the restoration of the original texts from its own database or from the Internet using public ontology WordNet or the Internet ontologies related to the medical field and software engineering field.

That is why a database is made and linked to the system to include the original texts and photographs taken from several digital libraries, Where the database includes three fields, the original texts, the name of the author and the date of the writing of the text. Where this system retrieves the original texts through keeping a copy of the entered text in a temporary store to test the similarities, At the same time, it makes a second copy of the text for comparison. In the case of discovering the percentage of similarity between the entered texts and the existing texts in the database or between the two texts then the two texts get back to the temporary store . And the usefulness of this store in the case of the comparisons and the loss of a part of the entered text You can get back to it and make a second copy and thus not losing of the data we entered. After the discovery of a substantial similarity with another text, it gets back to the temporary store and exiting the entered text again and then presented with the quoted part from it. The algorithm was also tested on 3 scientific files within the field of computer engineering in the field of software engineering and medical field. We changed the scientific content of the files by changing the word synonyms and replacing the concepts with other similar concepts. Determining the effectiveness of the proposed algorithm that is based on the sensitivity factor. which is calculated from the following relationship:

$$\text{Sensitivity factor} = \frac{\text{The total number of cases of plagiarism}}{\text{Total number of suspicious files}} \quad (5).$$

Search results include the results of the retrieval of the percentage of similarity between the original versions and the plagiarized , instead of retrieving the number of words that are similar to the number of names , acts and circumstances of similar writing or on meaning .We have three files plagiarized manually by replacing the words in the text, they are as follows :

The first file is text1 and contains 209 words.The second file is Text2 and contains 308 words. The third file is Text3 and contains 382 words.We also have 3 files manually copied by re-written as follows: The first file is text1 and contains 209 words.The second file is Text2 and contains 308 words.The third file is Text3 and contains 382 words.

Stolen texts are compared with original texts through two criteria: file size and time.

Table (1) shows the results of the comparison of the stolen texts with the original texts based on the size of the file and the time in the case of plagiarism in the synonym of words.

Table 1. The results of the comparison of the stolen texts with the original texts in the synonym of words.

Method	**Text1 (209 words)**		**Text2 (308 words)**		**Text3 (382 words)**	
	Percentage of similarity	**Time of examination**	**Percentage of similarity**	**Time of examination**	**Percentage of similarity**	**Time of examination**
The algorithm of semantic Fuzzy	0.59046	2.11	0.63714	3.12	0.671816	4.27
Sherlock Algorithm	0.53241	2.22	0.57168	3.23	0.616112	4.39

The acceleration can be calculated by the following equation:

$$Y= \frac{T_{Sherlock}}{T_{Fuzzy\_Algorithm}} \quad . \quad (6)$$

Where:

$T_{Sherlock}$ - Time to implement the Sherlock algorithm.

$T_{Fuzzy\_Algorithm}$ - The implementation of the proposed algorithm with a semantic dimension.

Y=1.02 times **,** in the synonym of words in the case of file size 382 words.

The algorithm's gain rate is calculated as follows:

$$G=\frac{T_{Sherlock}-T_{Fuzzy\_Algorithm}}{T_{Sherlock}}*100\% \qquad (7)$$

The algorithm works effectively as the file size increases, , the gain ratio is obtained up to 2.73% in the synonym of words.

The calculation of the error is defined as follows:

$$\Delta=\frac{T_{\text{Experiment}}-T_{\text{Theory}}}{T_{\text{Experiment}}}*100\% \qquad (8)$$

Where:

T_T - The theoretical time of the proposed algorithm,

T_E - Experimental time of the proposed algorithm.

The average implementation time of the proposed algorithm is calculated as follows:

$$\text{Average}=\left|\frac{\sum T_{\text{The method of semantic Fuzzy}}-\sum T_{\text{Sherlock Algorithm}}}{\sum T_{\text{The method of semantic Fuzzy}}}\right|$$

$$*100\% \qquad (9)$$

$$\text{Average\_ time}=\left|\frac{9{,}51-9{,}84}{9{,}51}\right|*100\ \%=3{,}47\%$$

The results of the experiments show that the average execution time of the proposed algorithm, for finding plagiarism, is less by 3.47% compared with the Sherlock algorithm in the case of the use of synonyms.

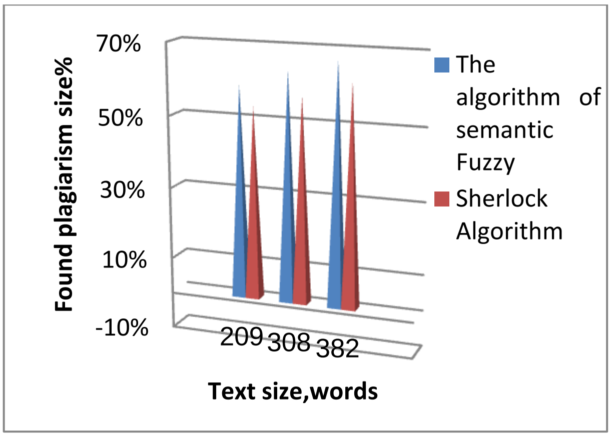


Fig 7: The graph shows the percentage of plagiarism in the file based on the criterion of the size of the text being scanned (Text synonym).

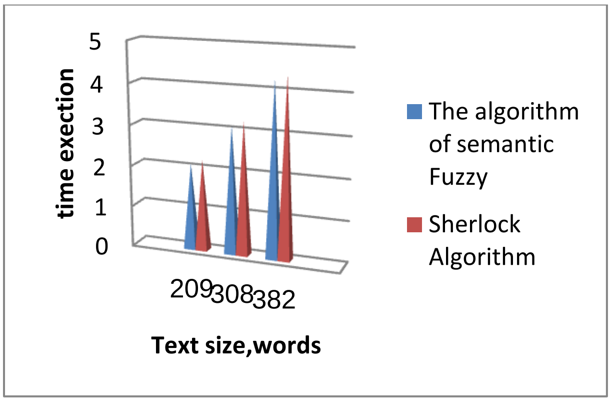


Fig 8:The graph shows the execution time based on the criterion of the size of the text being scanned (Text synonym)

Table (2) shows the results of comparison of the stolen texts with original texts based on file size and time in case of plagiarism rewrite words.

Table 2. The results of the comparison of the stolen texts with the original texts in case of plagiarism rewrite words.

Method	**Text1 (209 words)**		**Text2 (308 words)**		**Text3 (382 words)**	
	Percentage of similarity	**Time of examination**	**Percentage of similarity**	**Time of examination**	**Percentage of similarity**	**Time of examination**
The algorithm of semantic Fuzzy	0.5714	2.18	0.61814	3.23	0.6428	4.38
Sherlock Algorithm	0.5634	2.24	0.60268	3.29	0.6371	4.45

From the table we observe that the acceleration of the suggested method for fuzzy information is equal to:

Y=1.01 times in the case of rewriting words in the case of file size 382 words.

The algorithm works effectively as the file size increases, , the gain ratio is obtained up to 2.69% in the case of rewriting words. The average implementation time of the proposed algorithm is calculated as follows:

$$\text{Average\_ time}=\left|\frac{9{,}8-9{,}98}{9{,}8}\right|*100\ \%=1{,}83\%$$

The results of the experiments show that the average execution time of the proposed algorithm for finding plagiarism is 1.83% less than the Sherlock algorithm in the case of rewriting words.

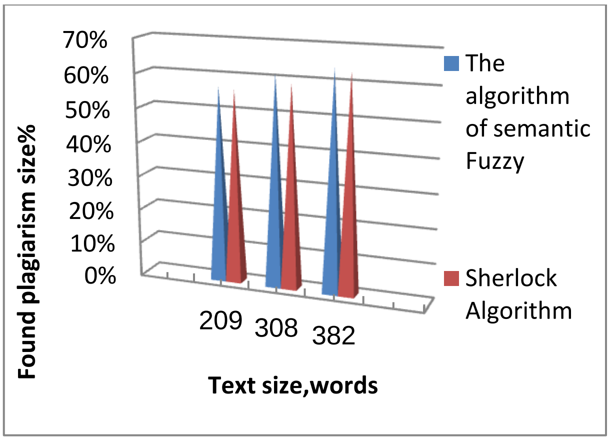


Fig 9:The graph shows how much impersonation exists based on the size of the text being scanned(rewrite).

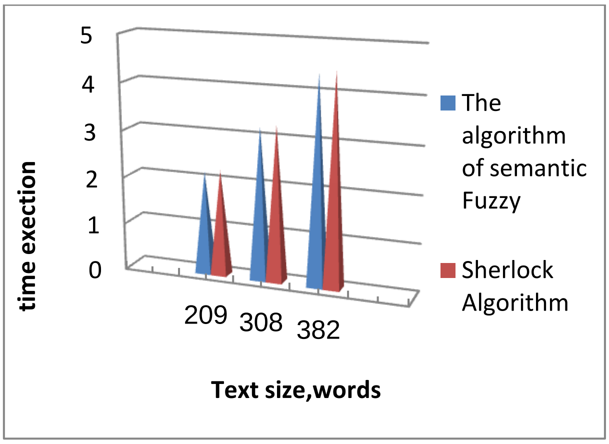


Fig 10: The execution time chart shows the size of the text being scanned (rewrite).

The results of the computational experiments on the algorithm and the proposed method are as follows:

We conclude from this that the suggested method with a semantic dimension in the case of fuzzy information is better than the Sherlock method in terms of file size standard of 6% if using word synonyms in the file and by 1% if rewriting the words in the file. As for the standard time taken to examine the files through the acceleration calculation, it is noted that the proposed method for the semantic dimension in the case of fuzzy information is faster in performance than the Sherlock method in the case of the use of synonyms 1.02 times and in the case of rewriting words with a value of 1.01 times in the case of file size 382 words . The results of the experiments show that the average execution time of the proposed algorithm, for finding plagiarism, is less by 3.47% compared with the Sherlock algorithm in the case of the use of synonyms and less by 1.83% compared with the Sherlock algorithm in the case of rewriting words. The algorithm works effectively as the file size increases, , the gain ratio is obtained up to 2.73% in the synonym of words and 2.69% in the case of rewriting words. From the results presented in the tables, we conclude that the average error rate of the proposed algorithm is 2% lower than the error rate sherlock algorithm. The complexity of the proposed algorithm is O(m*n).

5. CONCLUSION

The proposed method with a semantic dimension reveals the similarity between files in the case of fuzzy information and to detect anomalies such as changing the structure of words or replacing words with their synonyms based on two criteria of file size and execution time. It also minimizes technical spelling mistakes such as not typing the end of the word fully or abbreviations Unofficial and non-conventional, and shows the degree of similarity of the original text with the false text. The proposed method with semantic dimension in the case of fuzzy information is better than the Sherlock method in terms of file size standard of 6% in the case of plagiarism using word synonyms and 1% in the case of rewriting the words in the file. As for the standard time taken to examine the files through the acceleration calculation, it is noted that the proposed method for the semantic dimension in the case of fuzzy information is faster in performance than the Sherlock method in the case of the use of synonyms 1.02 times and in the case of rewriting words with a value of 1.01 times in the case of file size 382 words . The results of the experiments show that the average execution time of the proposed algorithm, for finding plagiarism, is less by 3.47% compared with the Sherlock algorithm in the case of the use of synonyms and less by 1.83% compared with the Sherlock algorithm in the case of rewriting words. The algorithm works effectively as the file size increases, , the gain ratio is obtained up to 2.73% in the synonym of words and 2.69% in the case of rewriting words. From the results presented in the tables, we conclude that the average error rate of the proposed algorithm is 2% lower than the error rate sherlock algorithm.

6. REFERENCES

[1] Nikhil Ghode, Shubham Jadhav, Sampada Moon, Ashmina Khan, Shrutika Bhalkar , Detecting Plagiarism In Academics Using levenshten Distance Algorithm And Semantic Similarity, International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248, Volume: 4 Issue: 3 471 – 473 ,2018.

[2] K. Vani and Deepa Gupta. 2016. Study on Extrinsic Text Plagiarism Detection Techniques and Tools. J. Engin. Sc. & Techn. Review 9, 5 (2016).

[3] Giovanni Acampora, Georgina Cosma, “A Fuzzy-based Approach to Programming Language Independent Source-Code Plagiarism Detection”, IEEE International Conference on Fuzzy Systems, 2015.

[4] S. M. Alzahrani, N. Salim, A. Abraham, Understanding plagiarism linguistic patterns, textual features, and detection methods, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (2) (2012) 133-149.

[5] R. Pike. “The Sherlock Plagiarism Detector.” Internet: http://www.cs.su.oz.au/~scilect/sherlock, 2007 [Oct. 04, 2011].

[6] Norman Meuschke, Moritz Schubotz, Felix Hamborg, Tomas Skopal, and Bela Gipp. 2017. Analyzing Mathematical Content to Detect Academic Plagiarism. In Proc. Conf. on Inform. and Knowl. Manage. (CIKM).

[7] Bela Gipp. 2014. Citation-based Plagiarism Detection - Detecting Disguised and Cross-language Plagiarism using Citation Pattern Analysis. Springer.

[8] Solange de L. Pertile, Viviane P. Moreira, and Paolo Rosso. 2016. Comparing and combining Content- and Citation-based approaches for plagiarism detection. JASIST 67, 10 (2016), 2511–2526.

[9] Salha M. Alzahrani, Naomie Salim, and Ajith Abraham. 2012. Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. In IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., Vol. 42. 133–149.

[10] S. M. Alzahrani, N. Salim, A. Abraham, Understanding plagiarism linguistic patterns, textual features, and detection methods, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (2) (2012) 133-149.

[11] Alzahrani, S. M.: iPlag: Intelligent Plagiarism Reasoner in Scientific Publications, IEEE (2011) .

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111-3119, 2013.

[13] Agarwal J, Goudar RH, Kumar P, Sharma N,Parshav V, Sharma R, Srivastava A, Rao S."Intelligent plagiarism detection mechanism using semantic technology: A different approach". International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013 Aug 22 (pp. 779-783). IEEE.

[14] Vani K., Deepak Gupta.: Study on Extrinsic Text Plagiarism Detection Techniques and Tools, Journal of Engineering Science and Technology Review 9 (4) (2016) .

[15] RDF/OWL Representation of WordNet . [online] Available at: http://www.w3.org/TR/wordnet-rdf/ [Accessed 1-may 2015].

[16] WordNet a lexical database for English . . [online] Available at: https://wordnet.princeton.edu/ [Accessed 5-may 2015].

[17] Nick Littlestone (2009). "Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm", Machine Learning 285–318(2).

[18] M. Potthast, T. Gollub, M. Hagen, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos, and B. Stein. Overview of the 5th International Competition on Plagiarism Detection. In *Working Notes Papers of the CLEF 2013 Evaluation Labs*, 2013.

[19] Ayad, L. A., Barton, C., & Pissis, S. P. (2017). A faster and more accurate heuristic for cyclic edit distance computation. Pattern Recognition Letters, 88, 81-87.

[20] Sharapova, E.V. Analysis of methods and systems for fuzzy duplicate detection / E.V. Sharapova // Proceedings of 14 International multidisciplinary scientific Geoconference SGEM2014. Informatics, Geoinformatics and Remote Sensing. – 2014. – Vol. 1. – P. 27-33.

[21] Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals / V.I. Levenshtein // Soviet Physics Doklady. – 1966. – Vol. 10(8). – P. 707-710.

[22] Wagner, R.A. The string-to-string correction problem / R.A. Wagner, M.J. Fischer // Journal of the ACM. – 1974. – Vol. 21(1). – P. 168-173.

[23] Gasfild, D. Strings, trees and sequences in the algorithms: Computer Science and Computational Biology / D. Gasfild. – 2003. – 654 p.

[24] 24. Knuth, D. The Art of Computer Programming / D. Knuth // Addison-Wesley. – 2000. – P. 396- 408.

[25] Baeza-Yates, R. A faster algorithm for approximate string matching / R. Baeza-Yates, G. Navarro // Combinatorial Pattern Matching (CPM'2004). – 2004. – P. 1-23.

[26] Foundational model of anatomy. [online] Available at:http://sig.biostr.washington.edu/projects/fm/ [Accessed 20-may 2015].

[27] Diseases Ontology . [online] Available at:

http://disease-ontology.org/ [Accessed 25-may 2015].

[28] Gene Ontology Consortium. Available at:http://geneontology.org/ [Accessed 20-may 2015].

[29] U.S. National library of Medicine. [online] Available at: http://www.nlm.nih.gov/mesh/ [Accessed 25-may 2015].

[30] The Open Biological and Biomedical Ontologies. [online] Available at: http://www.obofoundry.org/cgi-bin/detail.cgi?id=OGMS [Accessed 25-may 2015].

[31] EDAM Ontology. [online] Available at: http://edamontology.org/page [Accessed 25-may 2015].

[32] Software Engineering ontology. [online] Available at:http://dev.nemo.inf.ufes.br/seon/ [Accessed 25-sep 2017].

[33] Software Engineering ontology. [online] Available at:https://github.com/ChicoState/SoftwareEngineering [Accessed 25-may 2015].