International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2022 | Issue 53 | Volume 14 | Number 4 | Pages 17-24

PLAGIARISM DETECTION SYSTEM IN SCIENTIFIC PUBLICATION USING LSTM NETWORKS

M.N. Mansoor [1] M.S.H. Al Tamimi [2]

1. *Research and Development Department, Ministry of Higher Education and Scientific Research, Baghdad, Iraq*
mar118wa@gmail.com
2. *Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq, m_altamimi75@yahoo.com*

Abstract- Academic integrity is a sensitive topic in the academic world. Thus, it is necessary to oppose them tenaciously. However, Plagiarism is a problem throughout society, it impacts practically all industries, and it can happen by accident, it is most commonly the product of an intentional effort. The development of software detection systems has taken decades of research. Similarity detection in electronic-based documents is a procedure that is known as plagiarism detection. This procedure is essential because of the vast number of documents available on the internet and the capacity to copy and paste the text. PD was initially identified manually or by similarities to previously reviewed sources. However, it is becoming increasingly challenging due to the volume of available internet materials. As a result, developing automatic plagiarism detectors is necessary. At the same time, plagiarism is classified into two types: intelligent plagiarism and simple plagiarism (internal and external). It is indispensable to create a method for detecting each type of them. This scientific paper will address the issue of plagiarism check in practical publications based on deep learning using the LSTM algorithm to detect internal and external plagiarism types and evaluate the result by using PAN-PC 2011 data sets based on converting each document to vectors technique and TF-IDF weighting schemes. To perform natural language processing (NLP). The system results show that the accuracy measure is about 0.99%, F-measure is about 0.92%, precision is about 0.98%, and Recall is about 0.97%.

Keywords: Plagiarism Detection, Deep Learning, LSTM Algorithm, PAN-PC 2011, Doc2Vectors.

1. INTRODUCTION

Access to information has been considerably more controllable since the World Wide Web (WWW). Furthermore, rapid technological advancements allow quick access to information via numerous search engines, digital libraries, and other databases as the internet and its applications grow [1]. Plagiarism occurs when someone copies anything without the author's permission or acknowledgment.

In academic settings, plagiarism is a serious problem. It's made worse by how easily you can copy and paste from numerous materials that are available over the Internet. It is academic fraud because the offender stole and passed off someone else's work as their own. It refers to a person's honesty and integrity. Plagiarism in the educational process is a common and developing issue. It is difficult for people to identify plagiarism manually because it is incorrect and time-consuming since available data is challenging to validate [2].

Creating such digital resources and their storage and transmission is now relatively straightforward. In 1990, researchers began working on plagiarism detection in many languages to address this issue. Plagiarism detection is finding similarities in electronic-based documents [3]. Plagiarism detection systems are critical for detecting plagiarism, particularly in scientific publications. To detect plagiarism, a detailed understanding of the different types and grades of plagiarism is essential [4]. Plagiarism detection has also become a significant worry due to the availability of numerous software text editors. Plagiarism is becoming a more significant issue in academia. The problem of PD in scientific publications can be of various natures and parts, ranging from text copying to idea adoption without proper scientific attribution to achieve the goal mentioned above. Our system will solve these issues by incorporating an intelligent feature to learn and optimize detection time and result quality. Using current techniques and methodologies makes it possible to detect various plagiarism types [5]. The previous studies about detection systems (PD) focus just on one way of detection, which is (external detection).

- ➢ Examination of scientific publications to detect plagiarism for (internal and external PD methods) and better balance the two critical aspects of time and precision.
- ➢ We are solving the problem of consumer time deepened on deep learning to improve previous research results.
- ➢ Create a user-friendly program with interactive interfaces that support word file format and generate (PDF) plagiarism reports for free to the user.

1.1. Plagiarism Type

There are two types of plagiarism which are source code and textual plagiarism.

1.2. Source Code Plagiarism

This sort of plagiarism is tough to detect and done by university students. Students try or duplicate the entire or sections of source code produced by someone else as their own [6].

1.3. Textual Plagiarism

This sort of plagiarism is typically committed by students or researchers at academic institutions and involves documents that are identical or similar to the original documents, reports, essays, scientific papers, and artwork [7] and [4]. There are two types of textual plagiarism:

Intrinsic Plagiarism Detection: Without any external knowledge, this type of detection is utilized to recognize text fragments, sentences, or even a block of text copied as a whole section. This can be done by looking for modifications and inconsistencies within a document [3]. Extrinsic Plagiarism Detection using this method depends on the suspicious when knowing about suspicious references of files that the author may plagiarize and verifying if we found similarities in keywords, sentences, or even complete blocks of text [8]. Figure 1 Illustrates the categorization of plagiarism type.

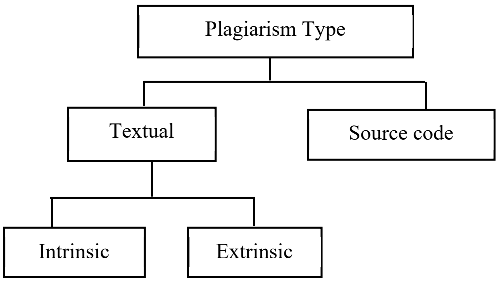

Figure 1. Plagiarism Type [9]

2. RELATED WORKS

Over time, there has been an increasing demand for any news article published by the researcher and the institution that will publish it to use this information in a working program. Some existing plagiarism-related studies have been published.

In (2011), Gupta et al [10] focused on paraphrasing included in PD from cross-lingual and monolingual points of view. The challenges of the detection process were investigated through further analysis of the performance of the (Vector Space Model) based on external PD on PAN-2011 corpus; there is a system. As candidate documents, 250 documents were used for each problematic document. In (2012), Ekbal et al [3] the system; used N-gram language model and VSM approaches. The proposed system was developed using a four-step process. The initial step, every document is processed to produce named sentence numbers - entity (NE) classes, find classes in Part of Speech (PoS), tokens, lemmas, and character offsets.

Documents were then forwarded to the second step, in which a group of documents was chosen as potential sources of plagiarism. A graph-based approach was utilized in the third step to discover the same sections in the selected source documents and suspicious documents. This N-gram-based approach was unable to recognize "plagiarized" content cases. N-grams are insufficient in situations where they are familiar. The number of the subset of the training corpus's 1,000 questionable documents was produced. The proposed Precision was (65.93), Recall was (19.04), Granularity (1.03), and Plagdet Score (28.91). In (2013), Buruiana et al [11] the proposed system that detect the external plagiarism by using Authentic Cop to identify plagiarism instances in computer science academic publications.

It tested the On PAN 2011, 1000 random preprocessing was carried out using cosine similarity and term (TF-IDF) weighting, given threshold are very similar under TF-IDF weighting with cosine similarity. The proposed system was proved by applying it, and the result was Recall (0.337), Precision (0.760), Granularity (1.265), and Pladget score (0.396). In (2015), Abdi et al [12] proposed an external plagiarism detection method. Was increased PD performance by avoiding picking source texts by combining word-word associations with the grammatical structure. This system effectively used the PAN- 10 and PAN- 11 datasets on the stop words extracted. The result of PD was used Linguistic Knowledge (PDLK)" on PAN-11 systems the PDLK Precision (0.902), Recall (0.702), F-measure (0.790) and Parameter (0.789). This result was only on 200 documents from 22000 used a subset of the datasets. In (2016), Sahi and Gupta [13] suggested a methodology for identifying plagiarized material that combined syntactic and semantic information.

The system was divided into three phases: (1) Preprocessing, then detailed analysis. A comparison was held between the source and suspected documents by implementing various weights on linguistic features as characteristic of inversion path length. The system was tested with 200 documents on the PAN- 11, indicating that it may not work with other datasets. The results of the 200 documents were Precision (0.949), Recall (0.715), and F-measure (0.815). In (2017), Abdi et al [14] Presented the proposed system as an external PD system (EPDS). It used the Semantic Role Labeling approach and (semantic, syntactic) data. The suggested technique might identify several forms of plagiarism. The proposed system was worked on the English part of the dataset and 800 suspicious and original corresponding documents. There are 450 documents in the training data and 350 documents in the testing data. The result of the evaluated method on the PAN-PC-11 dataset was Recall (0.622), Precision (0.921), F1 (0.743), Plagdet (0.737), and Granularity (1.011).

In (2020) Ahuja, et al [15]. developed a system that used an extrinsic PD technique inspired by cognition, in which semantic information was used to identify plagiarized material without the need for human intervention.

The system employed the semantic similarities between the two phrases and used the dice measure as a similarity metric. The PAN-PC-11 corpus was used for testing. The results were comparable to or somewhat better than current systems. The proposed system suspects and preprocesses the imposition of NLP features on source documents. The system was limited to the English language, and the outcome F1-measure (0.875), Precision (0.934), and Recall (0.861). One of this system's shortcomings was that it could only detect simple text plagiarism instances in (2021) F Khaled and Sabeeh [4]. The operation of verbatim plagiarism detection was stated by the researchers as a primary type of copy and paste. They've also shined a light on clever plagiarism. since it can involve altering the original content, including the thoughts of other academics, and translating to different languages, all of which might be more difficult to manage.

3. PROPOSED METHODOLOGY

This section describes the process used to accurately identify plagiarism in electronic files. Artificial intelligence refers to the development of computer programs that simulate intelligence. The suggested method is LSTM networks that compares the suspect and source to find plagiarism in publications using the PAN-PC-2011 dataset. To train deep learning models, massive amounts of labeled data and neural network topologies that automatically extract features from the data are used. And comes with an application for free. Figure 2 shows the structure of the proposed plagiarism detection system:

1. Remove any punctuation, diacritical markings, and other special characters, such as character formation in the English language, from the file for preprocessing the dataset.
2. Perform lemmatization and remove stop word operation on all texts to be ready for comparison.
3. Read text and choose the most important word based on the TF-IDF vector scheme. This will be using the help of mathematical operations to compare words.
4. Document representation represents the document's internal structure by using LSTM algorithms for each document. Determine the overall document's significant plagiarism percentage and create the final report.

3.1. Dataset

The system uses the PAN-PC-2011 under consideration. Text in corpus- based on books from Project Gutenberg (www.gutenberg.org) are called source documents, while the dataset is a corpus of publications that have been plagiarized both (manually and automatically). It's based on 22,000 English books. On the other hand, every instance of plagiarism marked in corpus [16], on the other hand, is either artificial, which means generated by a computer program, or simulated, purposefully manufactured by a human who has plagiarized. Additionally, the approach aims to identify all copied text portions in suspicious papers and their corresponding source text. The system is running on (1 1000 suspicious folders) and (11000 source folders).

3.2. Text Preprocessing

Is the process of translating content into a more palatable format, the action of cleaning and changing text can function better for NLP activities, it is an important step [4]. It is considered one of the most important basic steps for preparing files for plagiarism check. Our proposed system depends on an essential step for text preprocessing.

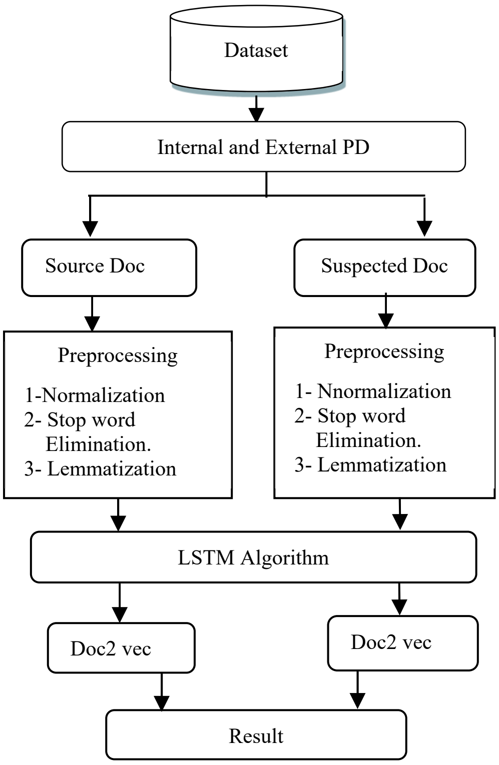

Figure 2. Structure of the proposed plagiarism detection system

3.2.1. Normalizing

Many of the text's little characters may go unnoticed. When comparing papers, it's essential to keep in mind that the more similar they are. Removing some characters from the text can increase the system's efficiency. Commas, semicolons, colons, special characters, brackets, quotes, white spaces, and other punctuation marks are unnecessary when determining the likeness [5]. Furthermore, the phrase could be an abbreviation or a misspelling that needs to be corrected.

3.2.2. Lemmatization

The lemma, which refers to a word's dictionary base form, is a technique for eliminating inflectional endings from a word and returning the base or dictionary form. By using a thesaurus, lemmatization also aids in the matching of synonyms so that while searching for "hot", the term "warm" is also found [17]. Comparing similar terms (cat vs. kitty) becomes significantly easier when using dictionary-based forms (cat or kitty). The lemmatizer is utilized to perform lemmatization on words for the system that is being suggested. An example might be, for instance: Playing, Plays, Played Common root from "Play". They have pre-processed documents that provide a set of phrases after these techniques have been applied to the papers.

3.2.3. Elimination of Stop-Words

Stop words are the most prevalent words that slow down the processing of documents. Conjunctions and articles are frequently applied to the text as stop-words. Around 50% to 60% of the words in a standard document are stopped words with no significance. By getting rid of these words, you can speed up the system and improve its accuracy and effectiveness. The suggested approach deletes all stop-words in the Stop-words list from the Natural Language Toolkit (NLTK). The list includes about 180 stop-words. Examples include the terms "is, I am, are, will, we, and me".

3.2.4. Convert Document to Vectors

A key and essential pre-processing step in many natural language processing jobs is the encoding of a lexical word into a numerical form that the computer can calculate. Vectorization, also referred to as word embedding in the NLP community, is the process of turning text input into numerical vectors. The highest level of data for any machine learning or deep learning model must be in numerical form because models do not comprehend text or visual data as well as people do. The Doc2vec is a method for efficiently creating word embeddings by leveraging a two-layer neural network to make neural-network-based embedding training more efficient. It has become the de facto standard for constructing pre-trained word embeddings.

The input of doc2vec is a text corpus, and the result is a collection of feature vectors, which stand for the words in the corpus. It is an unsupervised algorithm that learns fixed-length feature representations from text fragments of varying lengths, such as phrases, paragraphs, and documents [18]. In the suggested approach, by comparing two TF-IDF vectors, (TF-IDF - term frequency-inverse document frequency) information was used to create an overall representation of the fragment, and vector representation is a process that converts a text into a collection of vectors while maintaining the semantic and syntactic aspects provided by deep learning techniques [19]. The term frequency (TF) measures how frequently a word appears in a document. TF computing by the following Equation [20]:

$$TF\,(term) = \frac{\text{Number of times appears in a document}}{\text{Total number of items in the document}} \qquad (1)$$

When computing TF, phrases are given equal weight and their importance is determined by IDF (Inverse Document Frequency). Nevertheless, it is commonly recognized that some words, like "is", "of", and "that", may appear repeatedly but meaning little. While every one-of-a-kind word in the corpus is regarded as a feature, as a result, must scale down the common phrases while scaling up the rare ones. Computing by Equation (2) [20]:

$$IDF\,(term) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with term in it}}\right) \qquad (2)$$

So, according to the following Equation [21]:

$$TF - IDF\,(term) = F(term) * IDF\,(term) \qquad (3)$$

Inside the proposed scheme. Its advantage is that it avoids employing traditional methods, which have flaws when high execution is required. This is an intermediate stage to obtain only the crucial words in the document, that is, relying on the weight of the word, i.e., its frequency in one document and the benefit of this step was to convert the document to vectors, thus not having to enter the N-GRAAM way nor dividing sentences of all kinds, and maintaining usage of vectored representations of text data allows for easier comparison of words and sentences while also reducing the requirement for lexicons. Table 1 explains the converted document to vectors in the proposed system.

Table 1. Convert a document to vectors

	0	1	2	3	4	5	…	15469
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.041	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.098	0.0	0.099	0.0	0.0	0.0
3	0.0	0.044	0.0	0.0	0.014	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 15469 columns

3.5. Classification Using LSTM

The Long Short-Term Memory Network (LSTM) is a more advanced RNN (sequential network) that can retain data indefinitely. It can solve the vanishing gradient issue with RNNs. While LSTM is a special type of RNN that may learn long-term dependencies. The default behavior of LSTMs is for them to recall facts for a lengthy period. There are three gates in each LSTM module: a forget gate, an input gate, and an output gate. Table 2. Shows the three gates [22].

Table 2. LSTM gates [22]

No.	Name	Description
1	Forget Gate	This gate determines which facts in the cellular for that specific timestamp are to be ignored. The sigmoid function is used to calculate it [23]. And chooses whatever facts from the cells in the one-of-a-kind timestamp are to be ignored. The sigmoid function is used to calculate it.
2	Input gate	Determines the number of times in the current condition, this unit is introduced. The sigmoid function determines which values (0,1) are permissible. And The Tanh function weighs the values that can be supplied, ranking their significance from -1 to 1 [24].
3	Output Gate	a choice made by a portion of the present cell regarding the output. The Tanh characteristic lends weight to the values that can be exceeded by evaluating how relevant they are, ranging from -1 to at least one, and then enlarging it with a Sigmoid output.

In summary, the LSTM gates are; the first section determines if the information associated with the previous timestamp should be retained or ignored. The cell attempts to discover something useful from the input of the second segment. Finally, the cell delivers the most recent data in the third component from the current timestamp to the next timestamp. Moving from RNN to LSTM means more controlling knobs because of their superior ability to preserve sequence information over time. It regulates the flow and mixing of inputs according to the trained weights.

The classifier module establishes a common interface for text classification into categories [25]. As a result, there will be more control over the outputs. To achieve the necessary accuracy, the executive method learning paradigm typically employs the supervised type. So, LSTM provides the most control and better results. [26]. In the proposed system, this algorithm was adopted for four reasons:

1. An algorithm can deal with texts efficiently and effectively.
2. Maintains the relationship between words in a single document
3. It has internal memory. Through this feature, it can remember any sentence that passed through it.

It has the advantage of learning from mistakes in every training that can learn from everything you went through in halving. Figure 3. Illustrates the architecture of LSTM in the proposed system.

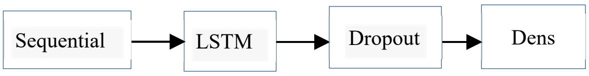

Figure 3. The architecture of LSTM in the proposed system

A set of steps are involved in the process of cross-validation in the proposed system as follows:

1. Consider the group as a holdout or test group of data.
2. The remaining groups will serve as your training data set [27].
3. Fit a model to the training data and assess it against the test data.
4. Discard the model and keep the evaluation score. Figure 4 Shows the process step of cross-validation.

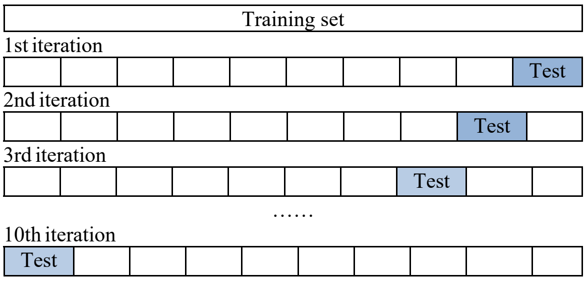

Figure 4. Process step of cross-validation

There are essential tools in systems that use language processing, and these tools were used in the proposed system; and they are as follows:

3.5.1. The Scikit-Multi Learn Library

The goal behind the proposed library is to develop a multi-label classification library added to an already implemented classification system. The scipy stack with scikit-learn is the logical choice for a basic library in Python. Scikit-learn compatible projects (scikits in short, not to be confused with the old scipy notion of scikits) have existed in various forms for some years, most notably in scikit-control. adhere to these communities' ideas and the scikit-learn API principles and licensing. Deep Learning models are used. Scikit-multi learn comes with a wrapper that lets you use any Keras-compatible backend, such as Tensorflow.

3.5.2. NLTK

The NLTK provides a simple, extendable, and standard framework for assignments, projects, and class demos. It's well-documented, simple to learn, and straightforward to use. It gives you some good beginning points: existing modules that implement the same interface, including predefined interfaces and data structures.

3.5.3. WordNet

Perl's object-oriented characteristics are used to implement similarity. It takes advantage of WordNet. To generate a WordNet object, use the Query Data Package (Rennie 2000). Numerous approaches can be used to Incorporate existing measures. WordNet Similarity offers extensive tracing that shows various diagnostic information unique to each of the many types of measures, regardless of how it is conducted [28].

4. RESULTS AND DISCUSSION

4.1. Evaluation Metrics

Matrixes of perplexity. Which will be carried out in terms of error rate and summarizes the number of events that a classification model correctly or incorrectly predicted. Table 3. Illustrates the confusion matrix of the proposed system.

Table 3. Confusion matrix of the proposed system.

		Actual	
Predicted	P	TP 10756	FP 1551
	Not P	FN 244	TN 120987449

Results of Experiments 50-dimensional hidden representations are used in our LSTM network. In our experiments, 300-dimension word embeddings were used. We assess our approach using the approved evaluation metric. Can be seen from the types of embedding experiments elaborated on later and conducted this experiment using a translation corpus, which could have resulted in translation quality issues.TF-IDF word embedding has better performance.

The following terms are routinely used to refer to the counts calculated in a confusion matrix:

- Precision: The precision equation defines precision as the proportion of the number of documents that are true positives in the dataset that the classifier has identified as positive [13]. The equation defines that:

$$Precision\,(p) = TP\,/\,TP + FP \tag{4}$$

Calculate using the mean average after applying the Equation to all documents. The Equation is [15]:

$$Average\ precision = \sum p\,/\sum N \tag{5}$$

$$Average\ precision = 10756\,/\,11000 = 0.977818 \tag{6}$$

- Recall: The fraction of positive cases correctly predicted by the classifier; the recall value is equivalent to the real positive rate. Where recall Equation is [15]:

$$Recall = TP\,/\,TP + FN \tag{7}$$

$$Recall = 10756/10756+244 = 0.97 \quad (8)$$

- F_1: The harmonic mean of Recall and accuracy are denoted by F_1 [14], and the equation is [29]:

$$F_1 = 2\times TP\frac{1}{2}*TP+FP+FN \quad (9)$$

$$10756/2\times 10756+1551+244 = 0.92 \quad (10)$$

- *Plagdet*: It is described as follows: precision, Recall, and granularity, and the Equation (11) is [12]:

$$Plagdet(S,R) = \frac{F1}{\log 2}(1+Gran(S,R)) \quad (11)$$

In PAN plagiarism detection competitions, the planet metric was used to rate the competitors. Calculate using the mean average after applying the equation to all documents. The equation is average:

$$Precision = \sum Plagdet \sum n = 4770.78015500 = 0.867 \quad (12)$$

- An accuracy is an approach to determining how often the algorithm successfully classifies a data point. The number of correctly anticipated data points out of all the data points is called accuracy [15]. And the equation is [15]:

$$\begin{aligned} &Accuracy = TP+TN/TP+TN+FP+FN = \\ &= (1056+120000000)/ \quad (13) \\ &/(1056+120000000+1551+244) = 0.99 \end{aligned}$$

Table 4 shows summary of the result of the system and Figure 5 shows accuracy of the best model in the system.

Table 4. Result summary

Precision	Recall	F1	Planet	Accuracy
0.98	0.97	0.92	0.87	0.99

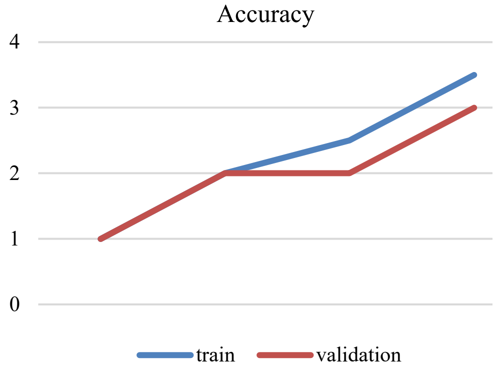

Figure 5. The accuracy results of the proposed system

5. THE PROPOSED SYSTEM INTERFACE

Most of the previous researchers relied on designing algorithms to evaluate and improve scientific plagiarism algorithms and neglected the user side. From this point of view, the proposed system provides a program to check plagiarism through simplified graphic interfaces for the user, supports the user with an un-modifiable PDF report, and allows the user to upload a text or word file. Figure 6 explains the main interface of the application.

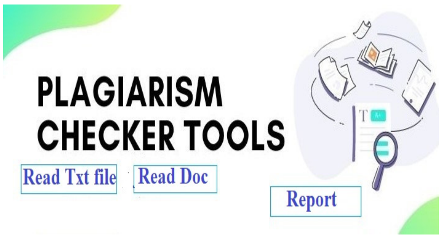

Figure 6. The main interface of the proposed application

The application supports both text and word formats, and it is the first application to work on a dataset that supports word formats. If the file to be inspected has the.txt extension, the file to be measured is chosen through the program interface by clicking on the Read text file window. Alternatively, if the file has a Doc extension, through the Read Doc file window. The file to be identified is first chosen, and then it is automatically compared to a group of source files. The proportion of plagiarism will be calculated.

Following the document comparison, the application generates a report in PDF format for the user that includes the percentage of total plagiarism and colored extracted sentences and gives a complete indexed list of the parts that were plagiarized. Table 5. Illustrates the final report.

Table 5. The final report on plagiarism

The Plagiarism Ratio is 59.18%
Master math and reading, often helping them earn their best grades ever, Tablet and desktop compatible app provide greater accessibility
Reduce homework stress and test anxiety, Songs, animations, and rewards make learning to read fun
The instructor will prepare an individualized lesson plan for your kids
Guided Lessons are easy to follow and match your child's ability, Building Confidence Older children can continue building key literacy skills

The image above shows the form of the final report, which appears to the user as a pdf, and contains the percentage of infiltration and contains coloring that shows the areas of abuse. In the proposed system, the first step was to examine the suspicious files in several source files and find a percentage of their stealing to verify the correctness of the algorithm work.

6. COMPARISON TO PREVIOUS RESEARCH

Propose an algorithm to investigate the problem of Plagiarism detection. The algorithm exploits factorized matrices [30]. Several Plagiarism detection techniques and their outcomes have been explained in the literature in the recent past. The proposed approach's conclusions are compared to those of other methodologies. Table 6. Compares the suggested method's detection measurement within the confusion matrix to that achieved in previous studies.

Table 6. Compares results with previous studies

RF.	Dataset	Number of documents used	Result				
			P	R	F1	Plag	Accuracy
Proposed system	PAN 2011	(11,000) document that mean all dataset	0.98	0.97	0.92	0.87	0.99
24	PAN-PC-2011	(11,000) document that mean all dataset	0.95	0.86	0.86	0.86	0.96
25	PAN-PC-2011	1000 doc just used suspected doc	0.65	0.19	0.29	0.28	0.88
27	PAN-PC-2011 PAN-PC-2010	200 docs in PAN-PC-2011	0.90	0.70	0.79	0.78	Not Used
28	PAN-PC-2011	200 docs in PAN-PC-2011	0.94	0.71	0.81	Not Used	Not Used
29	PAN-PC-2011	800 docs of suspect and source document	0.92	0.62	0.73	0.73	Not Used
30	PAN-PC-2011	A few docs and not tell the number	0.93	0.86	0.87	0.87	Not Used

7. CONCLUSION

The system's success in detecting plagiarism depends on how text is processed inside documents. Additionally, how tamper-proof unprocessed data and codes are gathered to measure the degree of similarity between the (source and suspicious documents). Deep learning methods were chosen for their efficiency and accuracy of results in categorization and detection of scientific plagiarism in scientific articles, based on past research, to ensure an accurate comparison procedure. LSTM algorithm was chosen because of the feature of dealing with texts and support dealing with converting documents into fragments, and the results of the system were somewhat high.

Based on the algorithm used, pre-processing steps, and the technique of relying on the weight of the word in the texts, the previous steps produced the highest accuracy and the lowest loss value of the methods previously investigated, so we've created a way that takes advantage of the expanded features of previous studies, to identify plagiarism effectively and fairly, must compare the document to previously published materials on the web. Propose relying on this technology because of the accurate findings achieved, and it can be developed in the future to detect plagiarism in images, charts, and tables.

REFERENCES

[1] A. Kocak, M.C. Taplamacioglu, H. Gozde, "General Overview of Area Networks and Communication Technologies in Smart Grid Applications", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 46, Vol. 13, No. 1, pp. 103-110, March 2021.

[2] H.A. Chowdhury, D.K. Bhattacharyya, "Plagiarism: Taxonomy, Tools and Detection Techniques", Cornell University, No. 1, 2018, http://arxiv.org/abs/1801.06323.

[3] M.H. Al Bayed, S.S. Abu Naser, "Intelligent Multi-Language Plagiarism Detection System", Int. J. Acad. Inf. Syst. Res., Vol. 2, No. 3, pp. 19-34, 2018.

[4] F.K. AL Jibory, "Hybrid System for Plagiarism Detection on a Scientific Paper", Turkish J. Comput. Math. Educ., Vol. 12, No. 13, pp. 5707-5719, 2021.

[5] M. Najm Mansoor, M.S.H. Al Tamimi, "Computer-Based Plagiarism Detection Techniques: A comparative Study", Int. J. Nonlinear Anal. Appl, Vol. 13, pp. 2008-6822, 2022.

[6] A.M. El Tahir Ali, H.M. Dahwa Abdulla, V. Snasel, "Overview and Comparison of Plagiarism Detection tools", CEUR Workshop Proc., Vol. 706, pp. 161-172, 2011.

[7] N.N. Chaubey, N.K. Chaubey, "Automatic Plagiarism Detection and Extraction in a Multilingual: A Critical Study and Comparison", Journal of Tianjin University of Science and Technology, Vol. 55, No. 01, pp. 284-304, 2022.

[8] C.F.O. Umareta, S. Mariyah, "Fuzzy Semantic-Based String Similarity Experiments to Detect Plagiarism in Indonesian Documents", The 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc., 2019.

[9] F.K. Abdul Jabbar, S. By, "Plagiarism Screening in Scientific Publication based on Fuzzy Semantic", 2021.

[10] P. Gupta, K. Singhal, P. Majumder, P. Rosso, "Detection of Paraphrastic Cases of Mono lingual And Cross Lingual Plagiarism", IR-Lab,DA-IICT, pp. 1-6, India, 2011.

[11] F.C. Buruiana, A. Scoica, T. Rebedea, R. Rughinis, "Automatic Plagiarism Detection System for Specialized Corpora", The 19th Int. Conf. Control Syst. Comput. Sci. CSCS, No. June, pp. 77-82, 2013.

[12] A. Abdi, N. Idris, R.M. Alguliyev, R.M. Aliguliyev, "PDLK: Plagiarism Detection Using Linguistic Knowledge", Expert Syst. Appl., Vol. 42, No. 22, pp. 8936-8946, 2015.

[13] M. Franco Salvador, P. Gupta, P. Rosso, R.E. Banchs, "Cross-Language Plagiarism Detection over Continuous-Space- and Knowledge Graph-Based Representations of Language", Knowledge-Based Syst., Vol. 111, pp. 87-99, 2016.

[14] A. Abdi, N. Idris, R.M. Alguliyev, R.M. Aliguliyev, "PDLK: Plagiarism Detection Using Linguistic Knowledge", Expert Syst. Appl., Vol. 42, No. 22, pp. 8936-8946, 2015.

[15] S. Ruuska, W. Hamalainen, S. Kajava, M. Mughal, P. Matilainen, J. Mononen, "Evaluation of the Confusion Matrix Method in the Validation of an Automated System for Measuring Feeding Behavior of Cattle", Behav. Processes, Vol. 148, pp. 56-62, 2018.

[16] M. Davoodifard, "Automatic Detection of Plagiarism in Writing", Stud. Appl. Linguist. TESOL, Vol. 21, No. 2, pp. 54-60, 2022.

[17] P.J. Burns, "Ensemble Lemmatization with the Classical Language Toolkit", Stud. e Saggi Linguist., Vol. 58, No. 1, pp. 157-176, 2019.

[18] P. Bafna, D. Pramod, A. Vaidya, "Document Clustering: TF-IDF Approach", Int. Conf. Electr. Electron.

Optim. Tech. ICEEOT, No. March, pp. 61-66, 2016.
[19] S. Qaiser, R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents", Int. J. Comput. Appl., Vol. 181, No. 1, pp. 25-29, 2018.
[20] P. Bafna, D. Pramod, A. Vaidya, "Document Clustering: TF-IDF Approach", Int. Conf. Electr. Electron. Optim. Tech. ICEEOT, No. November, pp. 61-66, 2019.
[21] Imamah, F.H. Rachman, "Twitter Sentiment Analysis of Covid-19 Using Term Weighting TF-IDF and Logistic Regresion", The 6th Inf. Technol. Int. Semin. ITIS, pp. 238-242, 2020.
[22] P. Goel, S.S. Kumar, "Certain Class of Starlike Functions Associated with Modified Sigmoid Function", Bull. Malaysian Math. Sci. Soc., Vol. 43, No. 1, pp. 957-991, 2020.
[23] K. Smagulova, A.P. James, "A Survey on LSTM Memristive Neural Network Architectures and Applications", Eur. Phys. J. Spec. Top., Vol. 228, No. 10, pp. 2313-2324, 2019.
[24] H. Salman, J. Grover, T. Shankar, "Hierarchical Reinforcement Learning for Sequencing Behaviors", Neural Comput., Vol. 2733, No. March, pp. 2709-2733, 2018.
[25] S. Bird, "NLTK: The Natural Language Toolkit", The 21st Int. Conf. Comput. Linguist. 44th Annu. Meet. Assoc. Comput. Linguist. Proc. Interact. Present. Sess., pp. 69-72, 2006.
[26] K.S. Tai, R. Socher, C.D. Manning, "Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks", The 53rd Annu. Meet. Assoc. Comput. Linguist., The 7th Int. Jt. Conf. Nat. Lang. Process., Asian Fed. Nat. Lang. Process. Proc. Conf., Vol. 1, pp. 1556-1566, 2015.
[27] A. Elomari, L. Hassouni, A. Maizate, "Deep Learning for Optimization of Chunks Placement on Hadoop/Hdfs", Int. J. Tech. Phys. Probl. Eng., Vol. 13, No. 4, pp. 194-200, 2021.
[28] T. Pedersen, J. Michelizzi, "[Selecionado] WordNet: Similarity, Measuring the Relatedness of Concepts Measures of Relatedness", Processing, No. Patwardhan 2003, pp. 1024-1025, 2004.
[29] J. Lever, M. Krzywinski, N. Altman, "Points of Significance: Classification Evaluation", Nat. Methods, Vol. 13, No. 8, pp. 603-604, 2016.
[30] S.M.M. Salehi, A.A. Pouyan, "Detecting Overlapping Communities in Social Networks Using Deep Learning", Int. J. Eng. Trans. C Asp., Vol. 33, No. 3, pp. 366-376, 2020.

BIOGRAPHIES

Marwah Najm Mansoor was born in Baghdad, Iraq in 1985. She received a B.Sc. degree from University of Technology, Baghdad, Iraq and a Diploma degree from Informatics Institute for Postgraduate Studies, Baghdad, Iraq. She is an employee at Research and Development Department, The Iraqi Ministry of Higher Education and Scientific Research. She is also a member of the committee to develop the skills of primary and postgraduate students in English and computer. She is currently a master's student at Institute of Informatics for Graduate Studies, Iraq. Her research interests are e-learning and the network approach.

Mohammed Sabbih Hamoud Al Tamimi was born in Iraq on June 16, 1975. He received his B.Sc. degree in Computer Science from University of Al-Fateh, Tripoli, Libya in 1998, M.Sc. degree in Computer Science from University of Science and Technology Sana'a, Yemen in 2000, and Ph.D. degree in Computer Science from Faculty of Computing, University of Technology Malaysia, Johor Bahru, Malaysia in 2015. Currently he is an Assistance Prof. at Department of Computer Science, College of Science University of Baghdad, Baghdad, Iraq. His research interests are deep learning technologies and branches of artificial intelligence.