A proposed approach for plagiarism detection in Article documents

Ayoub Ali M. Saeed [1)*], Alaa Yaseen Taqa [2)]
[1)] College of Basic Education, University of Mosul, Mosul, Iraq
[2)] College of Education for Pure Sciences, University of Mosul, Mosul, Iraq
[1)] ayobali-1980@uomosul.edu.iq , [2)] alaa.taqa@uomosul.edu.iq

Submitted : Apr 15, 2022 | **Accepted** : Apr 27, 2022 | **Published** : Apr 28, 2022

Abstract: According to the scientific institutes, Plagiarism is defined as claiming someone else's ideas or efforts as one's own without citing the sources. Systems of plagiarism detection typically use a text similarity algorithm in a text document to look for common sentences between source and suspicious documents, either by directly matching the sentences or by embedding the sentences into a vector using TFIDF-like or other methods, and then calculating the distance or the similarity between the source and suspect sentence vectors. The cosine similarity method is one of the methods for determining that distance. To cluster the documents and choose only related documents for detection, an unsupervised Machine learning technique such as K-means could be utilized. In this paper, a plagiarism detecting application was created and tested on many text document types, including doc, docx, and pdf of research papers that collected from the web to build the source corpus. To calculate the level of similarity between the suspicious article and the corpus of source articles, the TFIDF text encoding approach is used with NLP , K-means clustering and cosine similarity algorithms. The proposed application were carried out with five different documents and result in different ratios of plagiarism , the first document has 0.27 ratio, second document has 0.15 ratio, third document has 0.19 ratio while document 4 has 0.42 ratio and finally document 5 has 0.37 ratio of plagiarism. The generated detailed plagiarism ratio report presents the percentage of plagiarism in the suspicious article document. Depending on the threshold value, the application will decide if the suspicious document is acceptable or not.

Keywords: Plagiarism, Plagiarism Detection, Clustering, TFIDF, Cosine similarity

INTRODUCTION

Because scientifically structured publications contain human ideas or thoughts, scientific articles are always linked to the academic world. This effort must be ethically and legally safeguarded (Gupta, 2016). The most important prerequisite for researchers is to write research articles. The uniqueness and innovation of that articles is one of the aspects that determines its excellence (Hiten , Mohd. , Rutuja , & Nikita 2021).

Plagiarism is defined as the act of obtaining or attempting to obtain credit or value from a scientific work by quoting part or all of the work of other parties recognized as a scientific production without accurately and appropriately crediting its source (AL-Jibory, 2021).

Plagiarism can be occur in several forms (Hunt et al., 2019; Jiffriya, Jahan, & Ragel, 2021), Copy and Paste, pastes every text after copying it without changing it. Disguised exists in four techniques, shake & paste, costly plagiarism, contractive plagiarism, and mosaic plagiarism. It is the practice of covering the copied element. While Technical Disguise entails using the flaws in the basic text analysis approach, such as replacing letters with foreign letter symbols, to disguise plagiarism information from automatic detection. Excessive paraphrasing means intentionally rewritten foreign thoughts using plagiarized phrases and techniques while concealing original sources. Plagiarism that has been translated from one source language to another. Idea Plagiarism, which involves the appropriation of foreign ideas without citing the originating sources and finally Self-plagiarism, which is defined as the use of a portion or all of one's own writing for purposes that are not scientifically justified.

*name of corresponding author

Regarding scientific research articles it could be categorize plagiarism into(Kharat, Chavan, Jadhav, & Rakibe, 2013; Rosu, Stoica, Popescu, & Mihăescu, 2021):

a) Using a source's keywords, words, sentences, facts, and/or information without citing the source in the citation notes or referencing the source appropriately.
b) Referring to and/or randomly quoting keywords, words, sentences, data, and/or information from a source without citing the source and/or without completely identifying the source in the citation notes
c) Citing a source for ideas, opinions, points of view, or hypotheses without properly citing the source.
d) Formulate thoughts, opinions, perspectives, or theories in one's own words and/or sentences from a source of words and/or sentences without citing trustworthy sources.
e) Submit scientific works that were created and/or published by third parties as scientific works..

Detection of plagiarism is the process of identifying which parts of the suspicious text is plagiarized from other source texts or corpus (Kharat et al., 2013; Lahitani, Permanasari, & Setiawan, 2016). As a result, a transparent plagiarism detection system that can assist in checking suspicious document is required so that researchers can determine if the written text not previously released (Balani & Varol, 2021; Nurlybayeva, Akhmetov, Gelbukh, & Mussabayev, 2021).

The goal of this study is to utilize Term-Frequency Inverse document frequency (TF-IDF) text encoding, K-means clustering , Natural Language Processing (NLP) techniques, cosine similarity method , regular expressions and text processing to detect plagiarized parts in research articles in order to avoid duplication..

LITERATURE REVIEW

The originality of the researches is a major aspect that impact the admission of a research at an institution. For this field of information technology a number of studies have been conducted to avoid plagiarism and achieve the originality (Marjai, Lehotay-Kéry, & Kiss, 2021).

(da Costa & Mali, 2021) built a Tetun language plagiarism detection tool using a Text Mining approach that uses Tokenizing and Filtering to extract and pick a word list from the title of the students' thesis. To obtain the letter characters in the document to be matched and calculate the proportion of similarities in the processed thesis title, the n-grams and Jaccard Similarity Coefficient methods are utilized.

To semantically find plagiarism, (Kharat et al., 2013) Latent Semantic Indexing (LSI) and Term Frequency-Inverse Document Frequency (TFIDF) are used in the suggested system. Text mining and optimization algorithms are used to cluster research publications based on commonalities. The proposed method can be used to improve and speed up the grouping of papers.

In order to prevent plagiarism, a paper is presented by (Resta, Aditya, & Purwiantono, 2021), it employed data mining tools to detect similarities in the titles, abstracts, and themes of students' final scientific articles. In this paper, the cosine similarity approach is paired with the preprocessing method and TF-IDF to calculate the level of similarity between a student's final scientific paper title and abstract. The results are then displayed and compared to the existing final project repository to see if scientific work can be approved or refused depending on the threshold value..

The efficiency of prediction based on Euclidean distance, Jaccard similarity, and cosine similarity of rows before restarts is investigated by (Marjai et al., 2021). In addition to these distance metrics, characteristics such as TFIDF, Doc2Vec, LSH, and others have been applied. They utilized Spark for Big data computing since networking equipment generate a lot of log files.

Using the distance of cosine and K-means clustering algorithms, (Usino et al., 2019) presented a computerized system for detecting plagiarism information. The procedure begins with an innovative step of validating an Indonesian large lexicon, creating a vector space model, and combining K-means and cosine distance computations from 17 test items.

(Vani & Gupta, 2014) Investigated and compared several document categorization approaches used in external plagiarism detection. The main goal of study is to utilize several methods for unsupervised document categorization/clustering, the K-means algorithm with the general N-gram based approach and the Vector Space Model based method were used.

METHOD

- **The Corpus**

The corpus is a collection of source documents to which the suspicious document will checked against it (Gunawan, Sembiring, & Budiman, 2018). This application built a corpus of 300 scientific papers as text files after converting the Pdf and docx files to text files by utilizing python libraries. Furthermore , a Comma

*name of corresponding author

Separated values (CSV) file is created to save the Title, Keywords and authors names of the scientific papers in it after parsing and extracting them from the docx and pdf files as illustrated in Fig. 1.

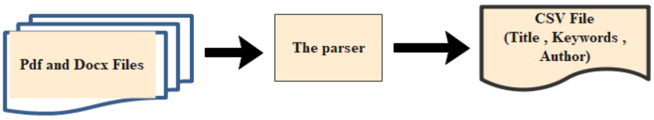


Fig. 1. Creating the CSV file in the proposed application

- **Pre-process the text**

Tokenization, lowercasing, removal of punctuations and stop-words, sentence segmentation and other improvements are intelligent NLP techniques that applied to the source and suspicious documents during pre-processing (Ahmed et al., 2021).

Because the obtained data is unsuitable for processing, we must first pre-process our corpus before implementing the word embedding approach. It has a lot of punctuation marks, stemming, and stop words, among other things. The most critical step during implementation is pre-processing, as seen in Fig. 2. We need to clean the data first by eliminating punctuation, stop words, and stemming (Wadud, Mridha, & Rahman, 2022).

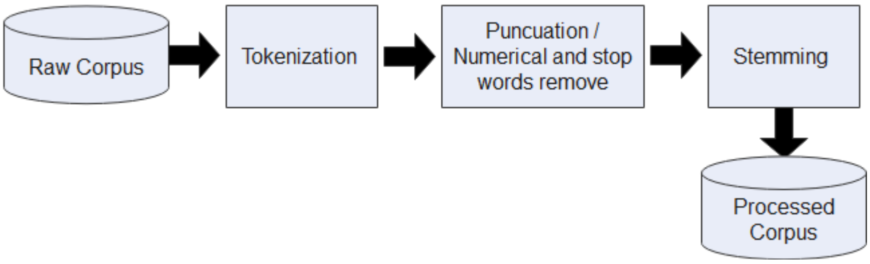


Fig. 2.Steps of text Pre-processing in the proposed application.

○ **Tokenization**

Tokenization is the process of breaking down a sentence into smaller, more relevant components. Each token is a single unit that might be a number, punctuation symbol, or a word.

○ **Removing Punctuations, numerical and stop words**

Words have been recognized using spaces, and superfluous things such as punctuation marks, hash tags, emoji, emoticons, and other symbols have been removed.

Every text contains a numeric character or word like ‘1’ ‘4768’ or a stop word, which is a non-essential term. Stop words in the English language include "the," "a," "an," "of," "my," and other words. These words are meaningless when it comes to document processing so it will be removed from the text.

○ **Lemmatization**

Lemmatization is the process of minimizing a word's variety. Depending on the situation, a word might take on several different forms.

This paper propose to eliminate the text before Abstract heading and after References heading in each document as a first stage of the pre-processing. This helps to reduce the quantity of actual text and thus decreases the required time.

N gram splitting is the process of sliding window on the text for chunking, the gram may be character , word or sentence (Hunt et al., 2019). This will be used in checking the similarity between documents.

To perform the previous tasks, the NLTK and Regulation Expression (re) libraries were employed in Python language.

- **Clustering**

A strategy for organizing a huge count of documents is document groupings or clusters. Unsupervised machine learning, in which no classes are identified, is commonly used for clustering (Vani & Gupta, 2014). The similarity of the data is used to divide it into categories. A relevant categorization can be deduced from the cluster. Within the same subject, any document has the potential for word similarity. The documents in the cluster are quite similar (Usino et al., 2019).

K-means algorithm is one of the efficient algorithms that is targeted to the unsupervised learning. It separates existing data into one or more clusters (Lydia, Govindaswamy, Lakshmanaprabu, & Ramya, 2018). This

*name of corresponding author

application used this algorithm for clustering and candidate retrieval for retrieving a selection of source documents that are similar to the suspicious document.

- **Text Encoding**

Text encoding or word embedding is the process of converting text into numerical vectors (Zen, Susanto, & Finaliamartha, 2021). The Term Frequency – Inverse Document Frequency (Tf-idf) vectorizer is one of the efficient statistical methods used for word embedding, i.e., conversion of textual data into an array of numbers (Resta et al., 2021).

TF-IDF places a premium on the frequency of a repeated word as well as its significance in the context of the input (Lahitani et al., 2016).

Term Frequency = (count of words in the document) / (total count of words in the document)
Inverse Document Frequency = log ((count of documents / count (docs containing keyword))
The formula TF-IDF is:

$$T_{i,j} = tf_{i,j} * \log\left(\frac{N}{df_i}\right) \qquad (1)$$

Where:

$tf_{i,j}$ = count of occurrences of term i in document j

df_i = count of documents containing i
N = total count of documents.

- **Text similarity measure**

This process is used to measure the similarity between the vectors and the converted form of textual data into the form of a vector is now utilized to detect the similarity between two text files (Gunawan et al., 2018).

The cosine similarity, which is one of the most used method for text similarity takes the vectors as an input. Using the angle between the two vectors, this technique determines the cosine of the data vector. The output is in a range between 0 and 1, indicating the similarity score. Given the vectors X and Y, The magnitude of the cosine similarity, cos (θ), is expressed by a dot product as following (AL-Jibory, 2021):

$$similarity = \cos(\theta) = \frac{X\,.Y}{\|X\|\;\|Y\|} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}} \qquad (2)$$

where X_i and Y_i are segments of the vectors X and Y.

- **The proposed application**

The proposed application for plagiarism detection will processed in several stages as the following and as illustrated in Fig. 3:

1. User will upload the suspicious document as PDF or docx file format and select n value of n-gram. The processing and generating the plagiarism report are differ between PDF and docx files but the stages are the same.
2. Extract the Title, Keywords and names of authors from the suspicious document. The titles and keywords of documents must first be pre-processed. Apply k-means Clustering algorithm according to the extracted title and keywords, then the related documents will be used for plagiarism detection. Similar documents are grouped together in clusters under this project's design.

The procedure for applying k-means clustering will be processed as illustrated in Fig. 4 :

- Read the Titles and Keywords of source documents from CSV file.
- Generation of Input vectors of Titles and keywords using TF-IDF values in n-dimensional term-document matrix.
- Fit the vectors to K-means clustering algorithm.
- Selection of similarity measure for generating similarity matrix using Cosine similarity.
- Predicate the title and keyword of suspicious document and get the documents of the specified cluster.

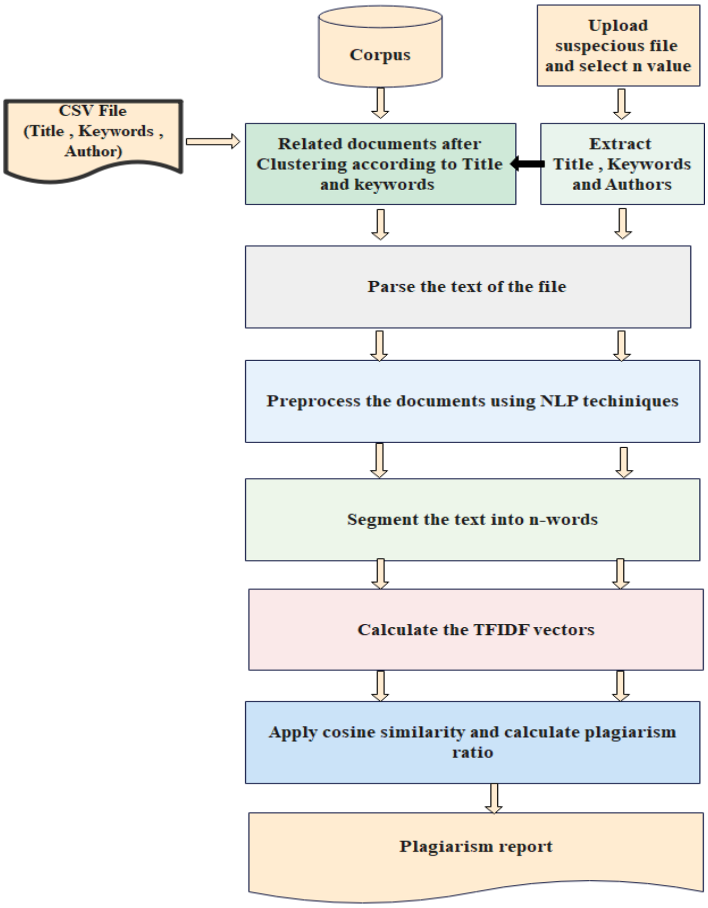

Corpus
Upload suspecious file and select n value
CSV File (Title , Keywords , Author)
Related documents after Clustering according to Title and keywords
Extract Title , Keywords and Authors
Parse the text of the file
Preprocess the documents using NLP techiniques
Segment the text into n-words
Calculate the TFIDF vectors
Apply cosine similarity and calculate plagiarism ratio
Plagiarism report

Fig. 3. Diagram of the proposed PD application

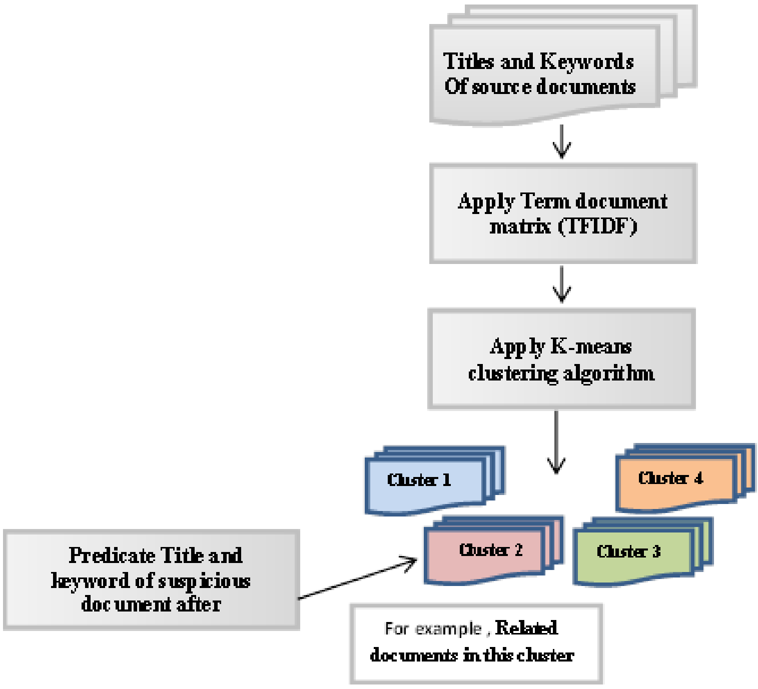

Titles and Keywords Of source documents
Apply Term document matrix (TFIDF)
Apply K-means clustering algorithm
Cluster 1
Cluster 4
Cluster 2
Cluster 3
Predicate Title and keyword of suspicious document after
For example , Related documents in this cluster

Fig. 4. Document clustering using K-means algorithm in the proposed application

3. Parse the text of the suspicious document and remove text before abstract heading and list of references.

*name of corresponding author

4. Preprocess the related documents and parsed suspicious document through applying NLP techniques.
5. Segment the texts according to the selected n value to n – words.
6. Calculate the TFIDF vectors of source and suspicious texts.
7. Apply cosine similarity and calculate the partial plagiarism ratio from each document and total plagiarized ratio.
8. Generate the plagiarism report . on the plagiarism report, he user could see total ratio , partial ratio from each source document , total number of sentences , words and characters of the suspicious document and that every plagiarized sentence will be highlighted with a color according to the its source document .

After that and according to the total plagiarism ratio the document will be accepted if the total ratio ≤ 25 otherwise it will be rejected.

RESULT

The steps of proposed application in this paper was applied to several suspicious documents using the Dell Latitude E6410 laptop, Intel Core i7 and 2.80 GHz CPU , 4096 MB RAM , Windows 10 Pro 64-bit and with Python 3.8. No matter what the type of document, pdf, doc or docx, for any case, it will be converted to text to be handled according to the application. Meanwhile, the application will extract the title and keywords from the input document and use it for clustering.
Table 1 illustrates information about five suspicious documents, including types, size in kilobytes and count of sentences.

Table 1. Information of suspicious documents

Doc. No.	Type	Size in K. bytes	Count of sentences
1	Pdf	292	3519
2	Pdf	428	4379
3	Docx	262	5326
4	Docx	347	4302
5	doc	324	4867

The percentages of plagiarism, total run times in minutes and seconds and related document of each suspicious document are listed in Table 2. Each document is accepted if its plagiarism percentage less than a threshold value for example 20.

Table 2. Percentages of PD and its Run time.

Doc. No.	Related documents Retreived after clustering	Plagiarism Percentage.	Time m:s	Accept
1	41	0.27	2:12	no
2	26	0.15	3:20	yes
3	32	0.19	3:43	yes
4	45	0.42	3:18	no
5	39	0.37	3:24	no

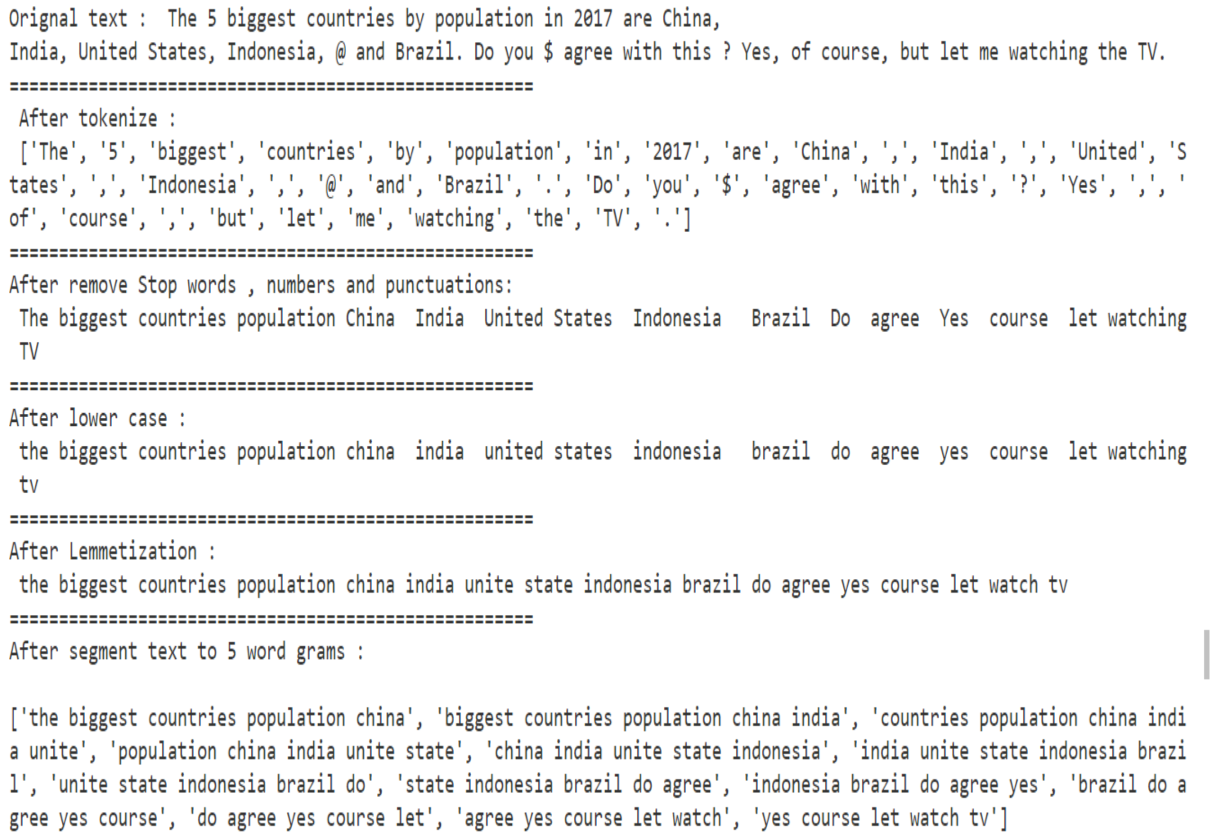
The percentage of plagiarism and run time will be affected by the clustering process, for example document 1 has related documents 41, PD percentage of 0.27 and 2:12 run time while document 5 has related documents 39, PD percentage of 0.37 and 3:24 run time.

Since the application check the plagiarism in each suspicious document sentence by sentence as 5-gram words each sentence, it’s impossible to analyze the whole processes in each document. Instead if that an example of one paragraph from document 1 will be analyzed.

Fig. 5 shows the execution of text pre-processing steps and segmenting the pre-processed text into 5 gram words as an example of source original text which is a paragraph in document 1.

Original text :
“The 5 biggest countries by population in 2017 are China, India, United States, Indonesia, @ and Brazil. Do you $ agree with this ? Yes, of course, but let me watching the TV. “

*name of corresponding author



```
Orignal text :  The 5 biggest countries by population in 2017 are China,
India, United States, Indonesia, @ and Brazil. Do you $ agree with this ? Yes, of course, but let me watching the TV.
====================================================
 After tokenize :
 ['The', '5', 'biggest', 'countries', 'by', 'population', 'in', '2017', 'are', 'China', ',', 'India', ',', 'United', 'S
tates', ',', 'Indonesia', ',', '@', 'and', 'Brazil', '.', 'Do', 'you', '$', 'agree', 'with', 'this', '?', 'Yes', ',', '
of', 'course', ',', 'but', 'let', 'me', 'watching', 'the', 'TV', '.']
====================================================
After remove Stop words , numbers and punctuations:
 The biggest countries population China  India  United States  Indonesia   Brazil  Do  agree  Yes  course  let watching
 TV
====================================================
After lower case :
 the biggest countries population china  india  united states  indonesia   brazil  do  agree  yes  course  let watching
 tv
====================================================
After Lemmetization :
 the biggest countries population china india unite state indonesia brazil do agree yes course let watch tv
====================================================
After segment text to 5 word grams :

['the biggest countries population china', 'biggest countries population china india', 'countries population china indi
a unite', 'population china india unite state', 'china india unite state indonesia', 'india unite state indonesia brazi
l', 'unite state indonesia brazil do', 'state indonesia brazil do agree', 'indonesia brazil do agree yes', 'brazil do a
gree yes course', 'do agree yes course let', 'agree yes course let watch', 'yes course let watch tv']
```

Fig. 5. Execution of text pre-processing and 5 gram words.

The computed TFIDF values of the original 5-gram text and the suspicious text using python Tfidf vectorizer are as the following,

The original text :
['the biggest countries population china', 'biggest countries population china india', 'countries population china india unite', 'population china india unite state', 'china india unite state indonesia', 'india unite state indonesia brazil', 'unite state indonesia brazil do', 'state indonesia brazil do agree', 'indonesia brazil do agree yes', 'brazil do agree yes course', 'do agree yes course let', 'agree yes course let watch', 'yes course let watch tv']

Computed TFIDF values of each document as (index of sentence, index of word)   TFIDF value of the word in the sentence.
Text0: 'the biggest countries population china'
 (0, 3)   0.42312422196478766
 (0, 9)   0.46488503840923534
 (0, 4)   0.5159961469970481
 (0, 1)   0.5818898264569827
Text1: 'biggest countries population china india'
 (1, 6)   0.38967693726274444
 (1, 3)   0.38967693726274444
 (1, 9)   0.4281366287786281
 (1, 4)   0.47520748698220294
 (1, 1)   0.5358923777636557
Text2: 'countries population china india unite'
 (2, 12)   0.4190637950712312
 (2, 6)   0.4190637950712312
 (2, 3)   0.4190637950712312
 (2, 9)   0.46042386220050013
 (2, 4)   0.5110444932663556
*name of corresponding author

```
Text3: 'population china india unite state'
  (3, 10)	0.4382285186628361
  (3, 12)	0.4382285186628361
  (3, 6)	0.4382285186628361
  (3, 3)	0.4382285186628361
  (3, 9)	0.4814800740656934
Text4: 'china india unite state indonesia'
  (4, 7)	0.4472135954999579
  (4, 10)	0.4472135954999579
  (4, 12)	0.4472135954999579
  (4, 6)	0.4472135954999579
  (4, 3)	0.4472135954999579
Text5: 'india unite state indonesia brazil'
   (5, 2)	0.4472135954999579
Text7: 'state indonesia brazil do agree'
  (7, 2)	0.5
  (7, 7)	0.5
  (7, 10)	0.5
Text8: 'indonesia brazil do agree yes'
  (8, 14)	0.5
  (8, 0)	0.5
  (8, 2)	0.5
  (8, 7)	0.5
Text9: 'brazil do agree yes course'
  (9, 5)	0.5356542147727775
  (9, 14)	0.48753617376770414
  (9, 0)	0.48753617376770414
  (9, 2)	0.48753617376770414
Text10: 'do agree yes course let'
  (10, 8)	0.5628511157331026
  (10, 5)	0.5070988690110534
  (10, 14)	0.46154596659797176
  (10, 0)	0.46154596659797176
Text11: , 'agree yes course let watch'
  (11, 13)	0.5358923777636558
  (11, 8)	0.47520748698220305
  (11, 5)	0.4281366287786282
  (11, 14)	0.3896769372627445
  (11, 0)	0.3896769372627445
Text12: , 'yes course let watch tv'
  (12, 11)	0.5593371712298097
```

The Tfidf of the suspicious text : “china is the biggest country of population”

```
  (0, 9)	0.46488503840923534
  (0, 4)	0.5159961469970481
  (0, 3)	0.42312422196478766
  (0, 1)	0.5818898264569827
```

Finally , the values of cosine similarity measure between suspicious text and original source text are:

```
The suspicious text  :
 'china is the biggest country of population'

The cosine similarity :
the biggest countries population china	1.0000000000000
biggest countries population china india	0.9209516190145
countries population china india unite	0.6550571965762
population china india unite state	0.4092579837273
```

*name of corresponding author

```
china india unite state indonesia     0.18922690464799
india unite state indonesia brazil     0.0
unite state indonesia brazil do     0.0
state indonesia brazil do agree     0.0
indonesia brazil do agree yes     0.0
brazil do agree yes course     0.0
do agree yes course let     0.0
agree yes course let watch     0.0
yes course let watch tv     0.0
```

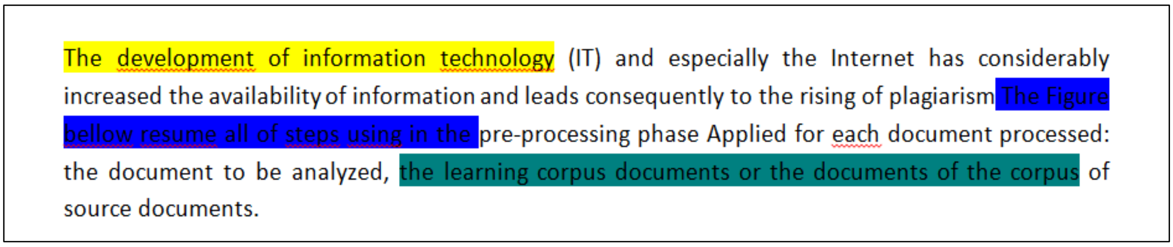
Based on the cosine similarity values, the suspicious text is plagiarized from the first original text, since it has the highest cosine similarity value 1 and it’s greater than a threshold value which is 0.5.

After executing the application for any PDF or docx suspicious document, the result will be a plagiarism ratio report like that of Turnitin and iThenticate platforms for plagiarism detection. The report will be a PDF file containing the text of suspicious document with highlighting the plagiarized sentences with colors according to the source of each sentence.

The report also shows the total ratio of plagiarism of the suspicious document, partial ratio from each source documents, count of sentences , count of words in suspicious document and other information.

After generating the report, the application will check if the total ratio of plagiarism is less than 20 then the document will be accepted, otherwise the document will be rejected.

A snapshot from one plagiarism report of one docx document is shown in Fig. 6, it shows a three plagiarized sentences which are colored according to their sources. Each highlight color refer to one source document.



The development of information technology (IT) and especially the Internet has considerably increased the availability of information and leads consequently to the rising of plagiarism The Figure bellow resume all of steps using in the pre-processing phase Applied for each document processed: the document to be analyzed, the learning corpus documents or the documents of the corpus of source documents.

Fig. 6. A snapshot from a plagiarism report in the proposed application

.

DISCUSSIONS

After several experiments, its noticed that the time of performing the plagiarism detect is affected by the total number of related source documents that retrieved from the corpus through clustering, as the retrieved documents increases the run time will increase, but the plagiarism ratio is not affected by the number of retrieved documents, it’s not necessarily that the plagiarism ratio will increase as the retrieved documents increases.

After caring out five experiments with different five documents (d1,d2,d3,d4,d5), the ratio of plagiarisms in each document was 27%, 15%,19%, 42% and 37% respectively. Because the ratios of d2 and d3 are less than the selected threshold 20%,so they will be accepted while other documents have ratios greater than 20% so the will be rejected.

The run time of for detecting plagiarism of each document were 2:12, 3:20, 3:43, 3:18 and 3:24 in minutes and seconds respectively. The run time depend on many factors like clustering process, count of sentences , preprocess , speed of CPU, and so on.

CONCLUSION

This paper intends to use data mining techniques to find similarities in the textual parts of articles in order to prevent plagiarism. The findings of this study can help uncover similarities between each suspect document and source documents in the corpus, as well as shorten the run time for plagiarism detection by removing specific parts of the suspect document's text, pre-processing, and clustering.

The proposed application will check a suspicious document against a corpus of source document and calculate the total plagiarism ratio of that document utilizing TFIDF, k-means clustering and cosine similarity techniques.

The generated plagiarism ratio report after executing the application for any PDF or docx suspicious document likes that report which generated by Turnitin and iThenticate platforms for plagiarism detection. The report will be a PDF file containing the text of suspicious document with highlighting the plagiarized sentences with colors according to the source of each sentence. The report also shows the total ratio of plagiarism of the suspicious document, partial ratio from each source documents, count of sentences , count of words in suspicious document and other information.

The total plagiarism ratio aids in determining whether or not a suspicious document satisfies the requirements and make a decision whether the document is accepted or not. The results of this application can assist the institution in managing of the checking of research articles in order to prevent plagiarism.

The suggested approach, however, is limited because it only uses a corpus of 300 research articles, which is a small sample size. The approach can be expanded by adding many article documents into the corpus and also adding the ability to search the web for the related documents and use other methods for text encoding and similarity measure.

More research can be done to look at the possibilities of employing a more condensed corpus. This research also advises adding ways to improve the application performance so that the application can function at its best when the plagiarism checking is carried out.

REFERENCES

Ahmed , A. E., Mohamed, G., Amar , F., Basma , M., Omar , M., & Mohamed , S. (2021). Plagiarism Detection Algorithm Model Based on NLP Technology. *Journal of Cybersecurity and Information Management (JCIM) 5*(1), 43-61.

AL-Jibory, F. K. (2021). Hybrid System for Plagiarism Detection on A Scientific Paper. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(13), 5707-5719.

Balani, Z., & Varol, C. (2021). Combining Approximate String Matching Algorithms and Term Frequency In The Detection of Plagiarism. *International Journal of Computer Science and Security (IJCSS), 15*(4), 97-106.

da Costa, E., & Mali, V. S. (2021). Tetun Language Plagiarism Detection With Text Mining Approach Using N-gram and Jaccard Similarity Coefficient. *Timor-Leste Journal of Engineering and Science, 2*, 11-20.

Gunawan, D., Sembiring, C., & Budiman, M. A. (2018). *The implementation of cosine similarity to calculate text relevance between two documents.* Paper presented at the Journal of physics: conference series.

Gupta, D. (2016). Study on Extrinsic Text Plagiarism Detection Techniques and Tools. *Journal of Engineering Science & Technology Review, 9*(5).

Hiten , C., Mohd. , T., Rutuja , K., & Nikita , C. (2021). Plagiarism Detector Using Machine Learning, *International Journal of Research in Engineering, Science and Management, 4*(4).

Hunt, E., Janamsetty, R., Kinares, C., Koh, C., Sanchez, A., Zhan, F., Dahal, B. (2019). *Machine learning models for paraphrase identification and its applications on plagiarism detection.* Paper presented at the 2019 IEEE International Conference on Big Knowledge (ICBK).

Jiffriya, M., Jahan, M. A., & Ragel, R. (2021). Plagiarism detection tools and techniques: A comprehensive survey. *Journal of Science-FAS-SEUSL, 2*(02), 47-64.

Kharat, R., Chavan, P. M., Jadhav, V., & Rakibe, K. (2013). Semantically Detecting Plagiarism for Research Papers. *International Journal of Engineering Research and Applications, 3*, 077-080.

Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). *Cosine similarity to determine similarity measure: Study case in online essay assessment.* Paper presented at the 2016 4th International Conference on Cyber and IT Service Management.

Lydia, E. L., Govindaswamy, P., Lakshmanaprabu, S., & Ramya, D. (2018). Document clustering based on text mining K-means algorithm using euclidean distance similarity. *Journal of Advanced Research in Dynamical & Control Systems, 10*(02-Special Issue).

Marjai, P., Lehotay-Kéry, P., & Kiss, A. (2021). Document similarity for error prediction. *Journal of Information and Telecommunication, 5*(4), 407-420.

Nurlybayeva, S., Akhmetov, I., Gelbukh, A., & Mussabayev, R. (2021). *Plagiarism Detection in Students' Answers Using FP-Growth Algorithm*, Cham.

Resta, O. A., Aditya, A., & Purwiantono, F. E. (2021). Plagiarism Detection in Students' Theses Using The Cosine Similarity Method. *Sinkron: jurnal dan penelitian teknik informatika, 5*(2), 305-313.

Rosu, R., Stoica, A. S., Popescu, P. S., & Mihăescu, M. C. (2021). *NLP based Deep Learning Approach for Plagiarism Detection.* Paper presented at the RoCHI - International Conference on Human-Computer Interaction, Romania.

Usino, W., Prabuwono, A. S., Allehaibi, K. H. S., Bramantoro, A., Hasniaty, A., & Amaldi, W. (2019). Document Similarity Detection using K-Means and Cosine Distance. *International Journal of Advanced Computer Science and Applications*.

Vani, K., & Gupta, D. (2014). *Using K-means cluster based techniques in external plagiarism detection.* Paper presented at the 2014 international conference on contemporary computing and informatics (IC3I).

Wadud, M. A. H., Mridha, M. F., & Rahman, M. M. (2022). Word Embedding Methods for Word Representation in Deep Learning for Natural Language Processing. *Iraqi Journal of Science, 63*(3), 1349-1361. doi:10.24996/ijs.2022.63.3.37

Zen, B. P., Susanto, I., & Finaliamartha, D. (2021). TF-IDF Method and Vector Space Model Regarding the Covid-19 Vaccine on Online News. *Sinkron: jurnal dan penelitian teknik informatika, 6*(1), 69-79.