A Study on Improving the Accuracy and Effectiveness of Similarity Detection Processes in Text Files Using NLP Techniques

Assist. Lect. Noor Abdulmuttaleb Jaafar

Administration & Finance Department, College of Admin. and Economics - University of Diyala, Diyala / Iraq

noor84abd84@uodiyala.edu.iq

دراسة حول تحسين دقة وفعالية عمليات اكتشاف التشابه في الملفات النصية باستخدام تقنيات البرمجة اللغوية العصبية

مدرس مساعد نور عبد المطلب جعفر

قسم الإدارة والمالية، كلية الادارة و الاقتصاد – جامعة ديالى، ديالى \ العراق

Abstract

The rapid expansion of the Internet has revolutionized access to information, especially in the area of unstructured data, most of which consists of textual content. While instant access to information brings many advantages, it has also given rise to a prevalent problem – plagiarism. Copying and reusing materials without proper permission poses a significant threat to academic integrity and integrity. Rates of plagiarism, especially in academic and scientific publications, have risen with the advent of the Internet, reaching alarming levels, such as 60% in student projects. This study examines the proposed model that includes computation of similarity using cosine coefficients, Euclidean similarity, and Jaccard similarity between training and test texts, providing a variety of metrics for comparison and analysis. These sequential steps combine automated analysis with human interpretation, enhancing the effectiveness and accuracy of the plagiarism checker and making it easier to use in many different fields and applications. The results showed that it is possible to accurately determine the similarity between texts.

Keywords: NLP, Cosine, Euclidean similarity, Jaccard , Text similarity.

المستخلص

أحدث التوسع السريع للإنترنت ثورة في الوصول إلى المعلومات، خاصة في مجال البيانات غير المنظمة، والتي يتكون معظمها من محتوى نصي. في حين أن الوصول الفوري إلى المعلومات يجلب العديد من المزايا، فإنه أدى أيضًا إلى ظهور مشكلة سائدة - الانتحال. يشكل نسخ المواد وإعادة استخدامها دون الحصول على إذن مناسب تهديدًا كبيرًا للنزاهة الأكاديمية والنزاهة. ارتفعت معدلات السرقة الأدبية، خاصة في المنشورات الأكاديمية والعلمية، مع ظهور الإنترنت، حيث وصلت إلى مستويات مثيرة للقلق، مثل 60% في مشاريع الطلاب. تتناول هذه الدراسة النموذج المقترح الذي يتضمن حساب التشابه باستخدام معاملات جيب التمام، والتشابه الإقليدي، وتشابه جاكارد بين نصوص التدريب والاختبار، مما يوفر مجموعة متنوعة من المقاييس للمقارنة والتحليل. تجمع هذه الخطوات المتسلسلة بين التحليل الآلي والتفسير البشري، مما يعزز فعالية ودقة مدقق الانتحال ويجعل استخدامه أسهل في العديد من المجالات والتطبيقات المختلفة. وأظهرت النتائج أنه من الممكن تحديد التشابه بين النصوص بدقة.

الكلمات المفتاحية: البرمجة اللغوية العصبية، جيب التمام، التشابه الإقليدي، الجاكار، تشابه النص.

1. Introduction

Plagiarism is the use of ideas, words, and expressions without proper reference to them, and is a common problem in various fields such as academia, scientific research, publishing, inventions, etc. The methods of plagiarism vary. It may be self-plagiarism, as in publishing an article in several magazines, or using other authors' texts without their permission, and this is considered a form of plagiarism. The occurrence of plagiarism can be observed in both academic and non-academic fields [1].

Plagiarism is among the most serious forms of research violations, as it negatively affects the reputation of the researcher and the university and harms the public. Research articles that include plagiarism hinder the scientific research process. Incorrect results can spread and negatively affect future research or scientific applications. For example, in the field of medicine or pharmacy, descriptive studies are vital tools for evaluating the effectiveness and safety of drugs and medical treatments. Plagiarized research articles can distort meta-studies and jeopardize patient safety.

In addition, plagiarism drains scholarly resources. Even in cases where plagiarism is discovered, reviewing plagiarized articles, requesting an apology, and holding them accountable poses a challenge o affected institutions, sponsors, and referees. If plagiarism goes undetected, its negative effects can be more serious, as plagiarists can obtain rewards and promotions in unfair ways, such as sponsoring agents giving allowance for stolen ideas or accepting plagiarized research as the results of research projects [3].

This research deals with integrating natural language processing techniques from NLP with a variety of methods for detecting plagiarism and calculating similarity scores, including using cosine similarity and measuring

the cosine of the angle between word frequency vectors, in addition to calculating Euclidean similarity by representing texts as frequency vectors and calculating Euclidean distance. The content of the paper is organized as follows: Section 2 provides a review of the literature related to the proposed work and previous research in this area. Then Section 3 provides an overview of the proposed system. Section 4 includes a presentation of the different stages of the working methodology, including a description of the dataset used. Section 5 presents the experimental results, while Section 6 discusses observations and results. Finally, the work is concluded in Section 7.

2. Literature review

Many studies have been conducted using various methodologies to diagnose and detect plagiarism in text files, as well as to predict it. Below, we will discuss a number of such studies that have been explored and implemented using a variety of algorithmic techniques.

Related work:

1. representation for similarity detection in Persian text documents. It uses word embedding's to represent words in an N-dimensional space, and calculates the similarity between the source and suspicious documents based on the cosine similarity between these vectors. The model converts text data into word embedding's, calculates similarity using cosine similarity, and classifies documents as plagiarized or not. Results from the PAN2016 dataset indicate a success rate of 94.37% with a running time of 00:01:27 per document pair, outperforming SVM and deep learning methods. In addition,

results on the PAN2015 dataset show a success rate of 96.94% with a running time of 00:01:21, outperforming graph-based methods by 9.94% in detecting plagiarism in Persian text documents, highlighting the effectiveness and efficiency of the proposed method.

2. In this work [5](2022) by Delfina Malandrino, *et al.*, we propose two new methods: text similarity-based method and clustering-based method, and demonstrate their integration into an enhanced (hybrid) approach to improve music plagiarism detection. To evaluate the effectiveness of the proposed methods, individually and in combined inference, tests were conducted on a large set of confirmed plagiarism and non-plagiarism cases. Results show that the combined inference outperforms current methods. Additionally, we deployed the combined inference into a web application and evaluated its effectiveness, utility, and overall user acceptance through a study involving 20 participants divided into two groups, one with access to the tool. The study involved participants deciding which pair of songs, from a predefined set of pairs, should be considered plagiarism and which should not. The study demonstrates that the group using our tool succeeded in identifying all plagiarism cases and performed all tasks without errors. The entire sample agreed on the benefit of having an automated tool providing a degree of similarity between two songs, highlighting the importance and efficacy of the proposed approach in addressing the issue of music plagiarism detection.
3. This paper [6] (2021) by Jirapond Muangprathope, *et al.*, An algorithm is proposed for document plagiarism detection using formal concept analysis (FCA) with a concept similarity filter introduced to retrieve

relevant source documents. The proposed similarity metrics use concept approximation using formal concept networks and have been mathematically proven to be formal similarity metrics. Source document processing and retrieval are performed using the proposed algorithm to demonstrate the performance of the proposed similarity measure in detecting document plagiarism through the implemented web applications. This work proposes three plagiarism prevention formats: (1) detection between documents within a document set, (2) detection between the suspicious document and source documents, and (3) detection between the suspicious document and other documents from the Internet. The three proposed formats were implemented in a system using PHP with a MySQL database. In addition, in the latter format, the presented system implements Google services. The efficiency and effectiveness of the proposed system was demonstrated through a case study of news and academic documents. The model was able to detect stolen text files and documents with an accuracy of up to 94.01%.

4. Both the study [7] (2018) conducted by Vani and colleagues and the research paper [8] (2021) point out the importance of using advanced natural language processing techniques to detect and treat plagiarism, given the increasing cases of plagiarism in electronic research and articles. The first study focused on analyzing linguistic features to detect plagiarized passages and plagiarism with varying degrees of complexity and their impact on the extracted features. In addition to using measures of syntactic-semantic similarity. The plagiarism suite from the PAN competition was used between 2009 and 2014 for pilot testing, which showed significant improvement in

manually identifying plagiarized data. On the other hand, the second study highlighted the possibility of obscuring the scientific content of publications by replacing words, removing or adding material, or rewriting, arranging, or rewriting the original articles. This paper also made a comparison between different plagiarism detection techniques, as well as carefully detailing the literature on plagiarism detection types, strategies, and tools.

In considering the studies presented in this section, we notice different methods for detecting similarities in text files and identifying plagiarism. This is not limited to the English language, but is intended for many languages such as Arabic and Persian, as well as providing methods that support similarity in musical texts. All studies emphasized the role and importance of natural language processing techniques to detect similarities and prevent plagiarism. Table 1 shows a summary of the above studies.

No.	Reference	Year	Method	Result
1.	Hadi Vaisi, et al.. (4)	2022	Word vector representation, cosine similarity	High success rates in detecting similarity in Persian text documents
2.	Delfina Malandrino, *et al.*, (5)	2022	Hybrid approach integrating text similarity and clustering methods	Outperformed existing methods in music plagiarism detection, high user acceptance in web application
3.	Jirapond Muangprathope, *et al.*, (6)	2021	Formal Concept Analysis, concept similarity metrics	High accuracy in detecting stolen text files and documents through case studies
4.	Fani, *et al.*, (7)	2018	Advanced natural language processing techniques	Emphasized the importance of NLP techniques in plagiarism detection

3. Background

The proposed model presented in this research focuses on analyzing uploaded texts through the use of the Google Collab repository, using natural language programming techniques that process these files and eliminate the impurities in them, such as removing symbols or correcting spelling errors, and then calculating the similarity percentage using... A set of similarity metrics. Like the cosine scale. Measuring similarity using calculating the Euclidean distance between data points in space and finally measuring Jaccard similarity. It is worth noting that each similarity measure has a unique way of analyzing and calculating the similarity between texts. As described in Section 3.3 of this paper.

3.1 Neuro-Linguistic Programming (NLP)

Neuro-Linguistic Programming (NLP) is an important technology in the modern digital age and is one of the branches of artificial intelligence that is concerned with analyzing and understanding human languages. Neuro-Linguistic Programming (NLP) appeared in the 1950s and has a wide range of applications in text analysis and similarity detection, such as text understanding and generation, machine translation, information extraction, semantic analysis and others, and there are many techniques provided by this technology, for example Work on converting clips. Converting text to digital representation (numeric arrays) such as Word Embedding's and deep conversion techniques such as BERT and GPT.

These representations also include the analysis of textual sentences, which contributes to a precise understanding of the sentence structure and linguistic structure of texts. NLP technology also provides many search and

comparison techniques, which include text search and analysis, as it works to identify similarities very effectively. The TF-IDF and Cross Comparison algorithms are among the most effective. Techniques used to determine similarity between texts.

Finally, using deep neural networks to analyze texts more deeply and accurately, such as convolutional networks such as (CNNs) and neural networks capable of calculating temporal consequences such as (RNNs), as their nature allows for precise analysis of texts and understanding similarities.

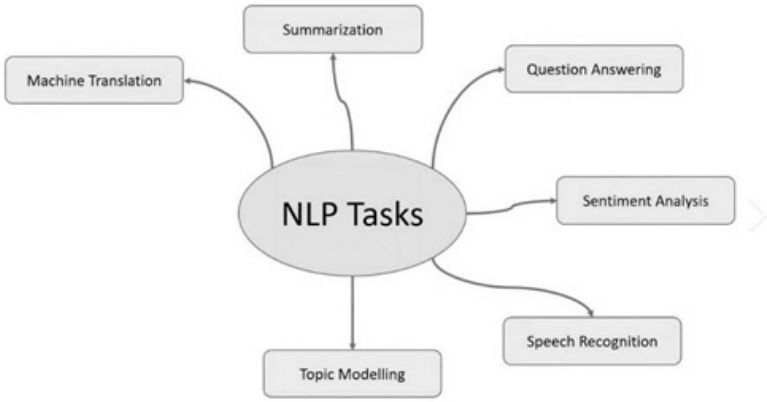

Fig. 1. NLP Tasks.

There are many real world applications such as search engines using Neuro-Linguistic Programming (NLP) techniques to improve search results by understanding queries and text similarity.

Also, it can Forensic analysis uses NLP techniques to analyze testimonies and texts related to criminal cases to identify links and similarities between them. Figs. (1 and 2) shows the process of analyzing texts using NLP techniques. [9].

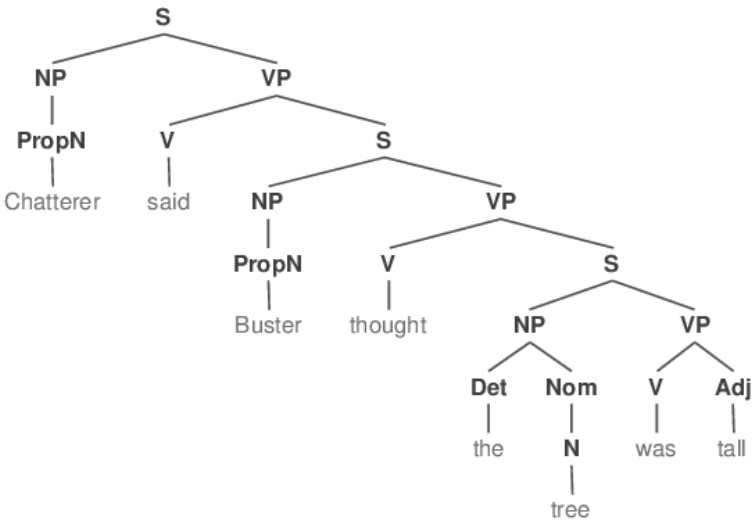

Fig. 2. Analyzing texts and sentences using NLP.

3.2 Text Similarity

Text similarity is the distance between two vectors, where the dimensions of the vectors represent the features of the objects. In simple terms, similarity is a measure of difference or similarity between two data objects. If the distance is small, the objects are said to have a high degree of similarity, and vice versa. In general, it is measured on a scale from 0 to 1. This score within the range (0, 1)is referred to as the similarity score. Despite its simplicity, similarity forms the basis of many machine learning techniques.

For example, the K-Nearest-Neighbors classifier uses similarity to classify new data objects, and similarly, K-means clustering uses similarity metrics to assign data points to appropriate clusters. Even recommendation engines use similarity-based collaborative filtering methods to identify users' neighbors.

The use of similarity metrics is particularly prominent in the field of natural language processing. Everything from information retrieval systems, search engines, paraphrase detection, text classification, document linking, and spell checking rely on similarity metrics [10].

3.3 Similarity Measures

The similarity between two excerpts of the same text is determined by the amount of cognates, which increases with the number of similarities, and vice versa. Almost all NLP-based tasks, such as information retrieval, automatic question answering, machine translation, dialog systems, and document matching, now leverage text similarity as a key factor in the operation of text systems [11].

In the past 30 years, new methods for semantic similarity using metrics have been introduced, the latest of which is divided into semantic similarity techniques, where most scholars have divided their methods. Statistical measurements of text similarity, or comparisons with databases and knowledge databases such as Wikipedia, using statistical methods. Spacing between text is not taken into account in these classifications, it only takes into account the representation of the text. The importance of neural network representation learning has been highlighted in recent years due to the development of computational models for neural network representation learning. Semantic matching, semantic matching methods and techniques, semantic matching methods, and systematic matching methods. Chart [12].

Fig. (3) shows that similarity between texts is not limited only to their semantic similarity, but also includes a comprehensive examination of the semantic features of two common words. For example, the words “king” and

"man" may be semantically similar, but in fact, the terms are not semantically similar. Semantic similarity is an essential part of semantic relatedness, and the semantic distance of this relationship is measured to be inversely related to the distance between the compared concepts [13].

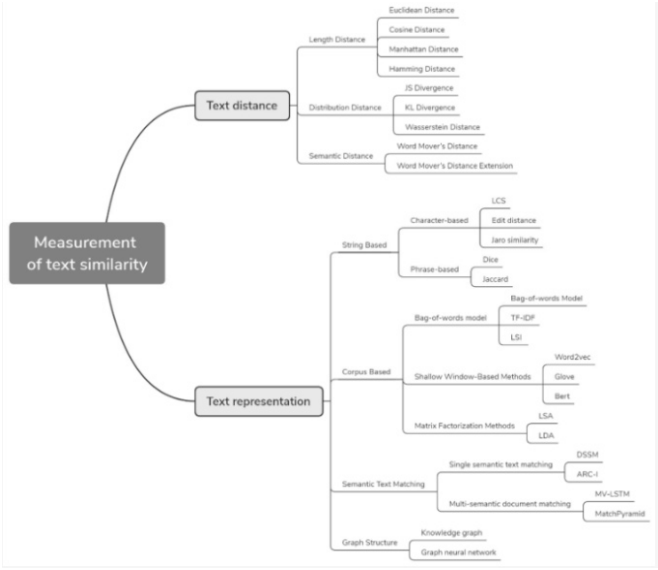

Fig. 3. Flowchart for the measurement of text similarity[13].

3.4 Text Distance

The first part of this research will address the concept of "text distance," an evaluation technique that describes the semantic relationship between two words or texts from the perspective of dimension metrics. Three main methods are used to measure text distance based on different aspects of the text object, its distribution, and semantics.

3.4.1. Length Distance

The percentage of similarity is measured by converting texts into digital matrices that express the presence or absence of different words and their frequency in each text.

In this paper, the most popular methods for evaluating text similarity will be presented in more detail, including Gaussian distance and Euclidean distance. Each of these methods will be briefly explained and their advantages and disadvantages in text similarity estimation analyzed [14].

3.4.1.1 Cosine Distance

Cosine similarity measures the degree of similarity between two vectors by taking the cosine of the vector angle as a measure. If two vectors point in a similar direction, it can be determined by similarity that they both have vectors pointing to two vectors.

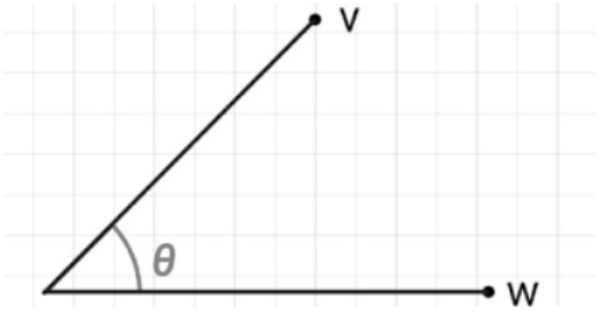

Fig. 4. The cosine distance [15].

The cosine distance is measured using the vector space of vector space and is converted to the angle problem in vector space. The similarity measure is determined by calculating Eq. Determine the cosine to measure the angle between the two vectors as shown in Fig. (4). Although the angle between the two vectors is 0°, the cosine similarity of the two vectors is equal to 1 [15]. As shown in the equation below:

$$\mathrm{Sim}\,(S_a, S_b) = \cos\Theta = \frac{\overrightarrow{S_a} \cdot \overrightarrow{S_b}}{\|S_a\| \cdot \|S_b\|}$$

3.4.1.2 Euclidean Distance

Euclidean distance, also known as the L2 rule, is one of the most widely used forms of Minkowski distance. When distance is referred to without specifying a specific type, the reference is usually to Euclidean distance. The Pythagorean principle is used to calculate the distance between two points, where the difference between the values of the two dimensions at the two points represents the opposite sides of the right-angled triangle as shown in Fig. (5) and the equation below:

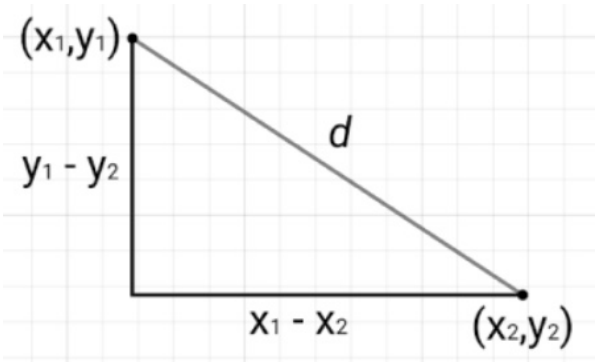

Fig. 5. The Euclidean distance [16].

The greater the distance (d) between the two vectors, the lower the degree of similarity between them, and vice versa[16].

$$d(S_a, S_b) = \sqrt{\sum_{l=1}^{n} (S_a^{(l)} - S_b^{(l)})^2}$$

3.5. Text Representation

In this section of the paper, we will focus on how texts can be represented in the form of numerical features that can be directly computed, so as to enable effective comparison and analysis processes. Texts can be similar in different ways, both in terms of vocabulary (lexicon) and in terms of meaning.

For example, lexical similarity is a common way to represent text, where the literal sequences of words in texts are compared. Semantic similarity can also be represented using methods such as the context-based method, where words, phrases and sentences are compared in context to determine the extent of similarity between texts.

Other methods for representing semantic similarity include the meaning-based method, where meanings and concepts in texts are compared. The texts are represented by graphical models to analyze the similarity between them.

Using these methods, we can accurately and comprehensively understand and analyze the similarity between texts, enabling us to reliably extract information and verify relationships between texts. [17].

3.5.1 String-Based

The benefits of string-based methods are that they are easy to compute. String similarity metrics analyze and group characters, which measures the similarity or difference (distance) between two text strings to achieve a match or comparison. These metrics represent one of the most common similarity metrics, which are included in specialized packages. As shown in Fig. 1, different approaches have been proposed to classify character- and

phrase-based methods according to their basic units. The most popular methods for each type will be displayed for a limited time.

3.5.1.1 Character-Based

Jaccard similarity is defined as the size of the intersection divided by the size of the union of two sets. The Jaccard index, also known as the Jaccard similarity coefficient, is used to treat data as groups. Calculates the intersection size of two sets and divides it by the union size. When using Jaccard similarity to calculate similarity between texts, texts are usually normalized initially to reduce words to their roots, which facilitates the comparison process.

Mostly, words are displayed with their roots in English, and their figurative meanings in other languages such as Chinese. For example, let us take two sentences as a model for calculating Jacquard similarity: “The bottle is empty” and “There is nothing in the bottle.” After normalization is applied, a Venn diagram is drawn for the remaining words in the two sentences to determine how similar they are [18]. Fig. (6) illustrates Jaccard similarity.

Fig. 6. Jaccard similarity [18].

The Jaccard similarity ratio between the two groups is 0.42, where the size of the intersection between them is 3, and the size of the union is 7 (1 + 3 + 3).

$$S(S_a, S_b) = \frac{S_a \cap S_b}{S_a \cup S_b}$$

3.6 Google Collab Hardware Specifications

Google Colaboratory, commonly referred to as Google Colab, is an open source service offered by Google to individuals with Gmail accounts. It provides access to GPU resources for research purposes, especially for those who may lack sufficient resources or cannot afford dedicated hardware. With Google Colab, users benefit from 12.72GB of RAM and 358.27GB of hard disk space during a single runtime. Each playback session lasts for 12 hours, after which it resets, requiring users to reconnect. This measure is implemented to prevent misuse of GPU resources for activities such as cryptocurrency mining or other illegal purposes. When opening a Google Colab file, users are asked to select the runtime type, offering three options: None (using the computer's CPU), GPU, and TPU (specifically for tensor processing). This selection can be made under the Runtime menu under Change Runtime Type[19]. Fig. (7) demonstrates the Google Colab Notebook Setting.

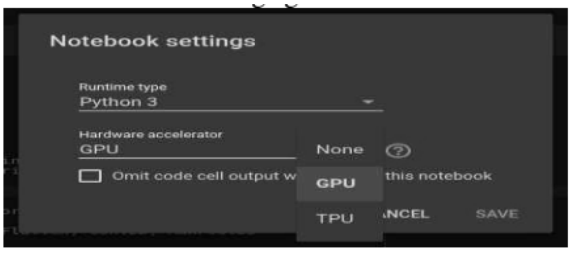

Fig. 7 . Google Colab Notebook Setting.

4.. Methodology

The methodology used in this research starts with loading the training and testing data as a .txt file. The decision was made to rely on Google's calculator rental service to conduct the research, as this service includes renting cloud processors and storage space. The proposed system uses Google Colab to increase the processing speed using TensorFlow and the TensorFlow Processing Unit (TPU). Taking advantage of the Jupyter user interface to create a drop-down menu containing files uploaded to Google Colab, making it easier to upload and select text files (training and testing). The steps also include text cleaning and pre-processing, such as removing unnecessary punctuation and text. After formatting the files. The similarity calculation phase begins.

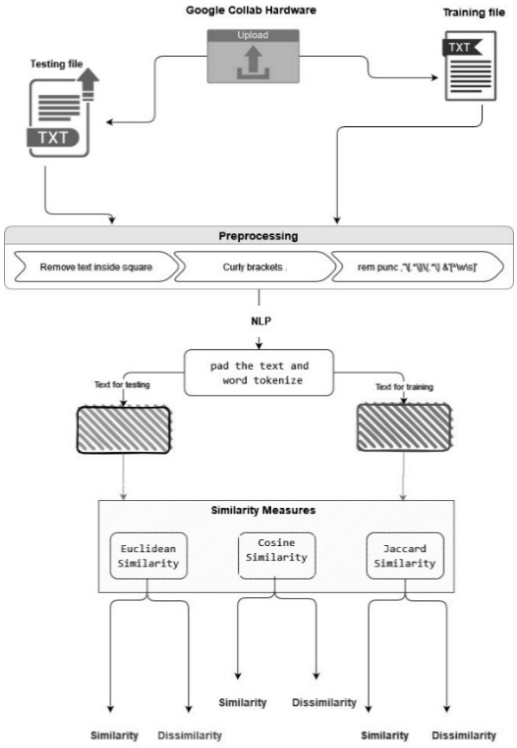

Fig. 8 Block diagram of proposed system.

The proposed model calculates similarity using cosine, Euclidean similarity, and Jaccard similarity coefficients between training and test texts, providing a variety of metrics for comparison and analysis. . These sequential steps combine automated analysis with human interpretation, enhancing the effectiveness and accuracy of the plagiarism checker and making it easier to use in many different fields and applications. Fig. (8) shows the schematic diagram of the proposed model methodology.

5.. Experimental results

This section describes the use of similarity metrics on text files uploaded via the Google Colab platform. Various similarity metrics, such as Jaccard, cosine, and Euclidean distance, were applied after implementing the necessary steps to process texts using natural language processing techniques.-

5.1 Cosine similarity

Figure (9) shows the application of the cosine similarity measure to two text files uploaded through the Google Colab platform. NLP techniques were performed to remove impurities from the two files, and then the similarity percentage was calculated in real time. The cosine similarity measure recorded a similarity rate of 5.4%.

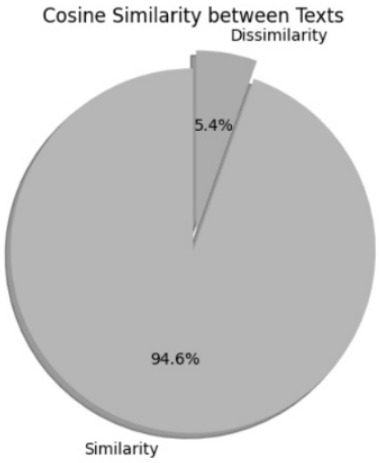

Fig. 9, Cosine similarity

5.2 Euclidean similarity

Figure (10), represents the calculation of the similarity percentage for two text files in real time, where the similarity percentage was 0.7%.

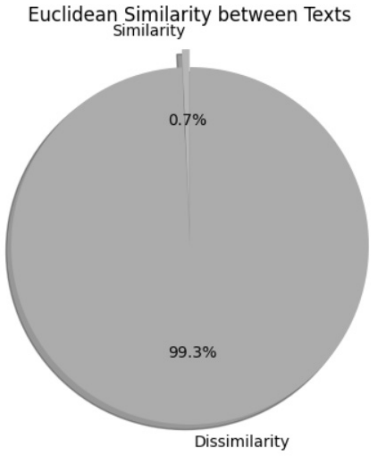

Fig. 10. Euclidean similarity.

5.3 Jaccard similarity

The method of calculating and representing similarity percentages varies from one measure to another, as the Jaccard similarity measure recorded a rate of 3.9%, as demonstrated in Figure (11).

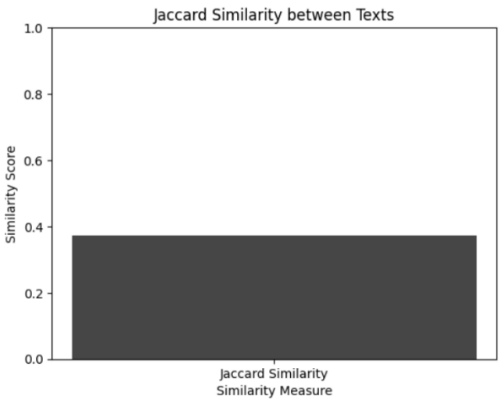

Fig. 11. Jaccard similarity.

Based on the results shown, we notice the ability to determine similarity between texts using different similarity measures. When looking at calculating similarity percentages in different measures, we see a difference in percentages, and this is due to the difference in methods for calculating similarity from one method to another. On the other hand, we note the ability of the metrics to accurately calculate the similarity between text files.

The proposed approach was also tested on the text similarity dataset, available on Kaggle and accessible at: https://www.kaggle.com/datasets/rishsankineni/text-similarity. This set consists of the data described in the next section.

In each row of the included datasets (train.csv and test.csv), the product is not exactly identical. You can make use of these recipes to predict whether each pair in the test set also refers to the same security.

Dataset information:

- Train data: includes descriptions (description_x, description_y), indicators (ticker_x, ticker_y), and rating (same_security).
- Test data: includes descriptions (description_x, description_y), and the rating will be predicted (Same_security).

The generated forecasts are compared with actual values using similarity metrics Jaccard similarity , cosine similarity and euclidean similarity to determine whether the forecasts are correct or incorrect, as described in the next section:

	test_id	description_x	description_y	same_security	jaccard_sim	cos_sim
0	0	semtech corp	semtech corporation	False	0.333333	0.336097
1	1	vanguard mid cap index	vanguard midcap index - a	False	0.285714	0.411207
2	2	spdr gold trust gold shares	spdr gold trust spdr gold shares	True	1.000000	0.956183

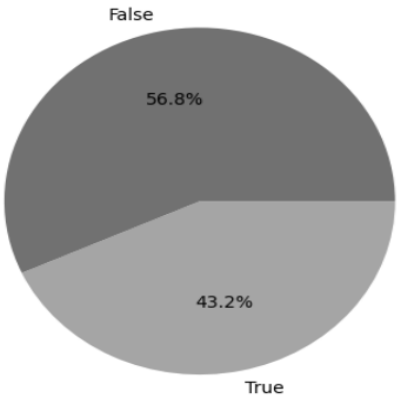

Fig. 12, Cosine similarity

Figure (12) shows a way to measure similarity between texts using the cosine similarity measure, which calculates the angle between vector spaces representing texts.

	test_id	description_x	description_y	same_security	euclidean_sim
0	0	semtech corp	semtech corporation	NaN	1.152305
1	1	vanguard mid cap index	vanguard midcap index - a	NaN	1.085166
2	2	spdr gold trust gold shares	spdr gold trust spdr gold shares	NaN	0.296031

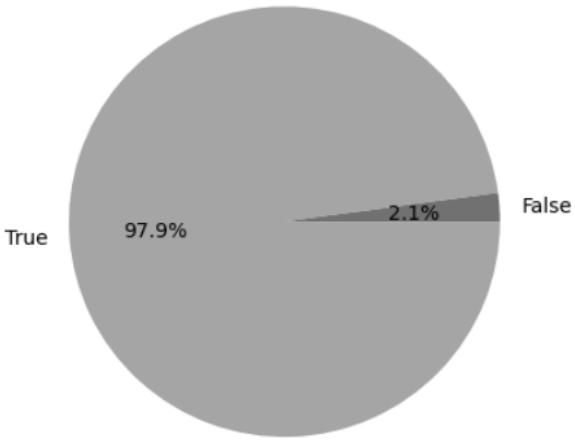

Fig. 13, Euclidean similarity

It uses the Euclidean similarity measure, which calculates the shortest distance between two points in the space defined by its perception of texts, as demonstrated in Figure 13.

	test_id	description_x	description_y	same_security	jaccard_sim
0	0	semtech corp	semtech corporation	NaN	0.333333
1	1	vanguard mid cap index	vanguard midcap index - a	NaN	0.285714
2	2	spdr gold trust gold shares	spdr gold trust spdr gold shares	NaN	1.000000

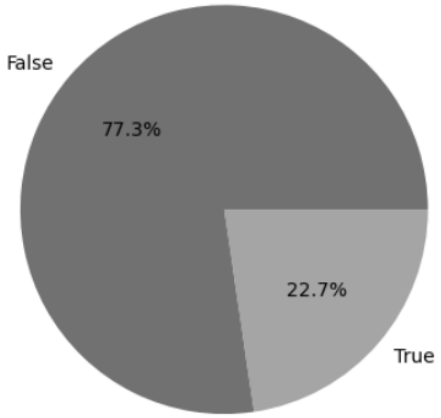

Fig.14. Jaccard similarity.

Finally, Figure (14) shows the Jaccard similarity measure, which measures the relative affiliation between groups of words in texts. A threshold (threshold) was determined to determine whether the texts refer to the same security or not. When the similarity value exceeds the specified threshold, the pair of texts is considered to refer to the same security. Next, it is determined whether the results are correct or not by comparison with the test data.

6.. Conclusion

The paper presents a methodology for calculating similarity between text files using Google Colab. Next, cleaning and pre-processing operations are performed, including removing punctuation marks and unnecessary tet. The proposed model then calculates similarity using cosine, Euclidean similarity, and Jaccard similarity coefficients between training and test texts, providing a variety of metrics for comparison and analysis. These sequential steps combine automated analysis with human interpretation, enhancing the effectiveness and accuracy of the plagiarism checker and making it easier to use in many different fields and applications. The results showed that it is possible to determine similarity between texts using different similarity metrics. When looking at calculating similarity ratios on different scales, we see a difference in the ratios, because the methods for calculating similarity differ from one method to another. On the other hand, we note the ability of metrics to do this.

References

[1] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," Comput. Biol. Med., Vol. 121, No. April, pp. 103792, 2020, doi: 10.1016/j.compbiomed.2020.103792.

[2] C. Huang, *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," Lancet, Vol. 395, No. 10223, pp. 497–506, 2020, doi: 10.1016/S0140-6736(20)30183-5.

[3] Brychcín, T. (2020). Linear transformations for cross-lingual semantic textual similarity. Knowledge-Based Systems, 187, 104819. https://doi.org/10.1016/j.knosys.2019.06.027

[4] Veisi, H., Golchinpour, M., Salehi, M. Multi-level text document similarity estimation and its application for plagiarism detection. Iran J Comput Sci 5, 143–155 (2022). https://doi.org/10.1007/s42044-022-00098-6.

[5] Malandrino, D., De Prisco, R., Ianulardo, M. An adaptive meta-heuristic for music plagiarism detection based on text similarity and clustering. Data Min Knowl Disc 36, 1301–1334 (2022). https://doi.org/10.1007/s10618-022-00835-2.

[6] Newscatcher API. (2020, 11 ,22). Ultimate Guide to Text Similarity with Python. Newscatcher API Blog. https://www.newscatcherapi.com/blog/ultimate-guide-to-text-similarity-with-python.

[7] Wang, J.; Dong, Y. Measurement of Text Similarity: A Survey. Information 2020, 11, 421. https://doi.org/10.3390/info11090421.

[8] Mansoor, M. N., & Al-Tamimi, M. S. H. (2022). Computer-based plagiarism detection techniques: A comparative study. *International Journal of Nonlinear Analysis and Applications*, 13(1), 3599-3611. DOI: http://dx.doi.org/10.22075/ijnaa.2022.6140.

[9] Gillioz, A., Casas, J., Mugellini, E., & Abou Khaled, O. (2020). Overview of the Transformer-based Models for NLP Tasks. In Proceedings of the Federated Conference on Computer Science and Information Systems (pp. 179–183). ACSIS, Vol. 21. DOI: 10.15439/2020F20. ISSN 2300-5963.

[10] Einstein, A., B. Podolsky, and N. Rosen, 1935, "Can quantum-mechanical description of physical reality be considered complete?", Phys. Rev. 47, 777-780.

[11] Prakoso, D.W., Abdi, A., & Amrit, C. (2021). Short text similarity measurement methods: a review. Soft Computing, 25, 4699–4723. https://doi.org/10.1007/s00500-020-05479-2.

[12] Pawar, A., & Mago, V. (2018). Calculating the similarity between words and sentences using a lexical database and corpus statistics. CoRR, abs/1802.05667.

[13] Wang, J., & Dong, Y. (2020). Measurement of Text Similarity: A Survey. Information, 11(9), 421. https://doi.org/10.3390/info11090421.

[14] Gomaa, W. H., & Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. International Journal of Computer Applications, 68(13).

[15] Chapman, S. (2006). SimMetrics : a java & c# .net library of similarity metrics,http://sourceforge.net/projects/simmetrics/.

[16] Tan, Z., Yang, X., Ye, Z., Wang, Q., Yan, Y., Nguyen, A., & Huang, K. (2023). Semantic Similarity Distance: Towards better text-image consistency metric in text-to-image generation. Pattern Recognition, 144, 109883.

[17] Pandya, A., Bhattacharyya, P. (2005). Text Similarity Measurement Using Concept Representation of Texts. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (Eds) Pattern Recognition and Machine Intelligence. PReMI 2005. Lecture Notes in Computer Science, Vol. 3776. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11590316_109.

[18] Bag, S., Kumar, S. K., & Tiwari, M. K. (2020). An efficient recommendation generation using relevant Jaccard similarity. Information Sciences. Retrieved from www.elsevier.com/locate/ins.

[19] Kanani, P., & Padole, M. (2019). Deep Learning to Detect Skin Cancer using Google Colab. International Journal of Engineering and Advanced Technology (IJEAT), 8(6), ISSN: 249 – 8958.