IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Plagiarism Checker using tf-idf, cosine similarityand jaccard similarity

Sanjeev Kumar Singh
Department of Information Technology
Galgotias College of Engineering and Technology
Greater Noida, India

Akshay Tiwari
Department of Information Technology
Galgotias College of Engineering and Technology
Greater Noida, India

Chitrang Chauhan
Department of Information Technology
Galgotias College of Engineering and Technology
Greater Noida, India

Abhinav Singh
Department of Information Technology
Galgotias College of Engineering and Technology
Greater Noida, India

Mayank Kumar
Department of Information Technology
Galgotias College of Engineering and Technology
Greater Noida, India

***Abstract*—Plagiarism is a severe problem in academic and professional settings when someone uses someone else's work and claims it as their own. With the rise of digital media and the internet, plagiarism has become more common. Machine learning can help detect plagiarism by looking over the content and comparing it to a database of recognised sources. In this paper, we provide machine learning-based plagiarism detection. The model generates a similarity score by analyzing text using Natural Language Processing methods. Moreover, a series of preprocessing techniques are used to get a clean dataset. In addition, respective heat maps as well as similarity matrix for both the cosine as well as jaccard similarities are presented. According to the data, the model is useful for detecting plagiarism and may be applied to prevent academic dishonesty.**

Keywords—Plagiarism Checker, Machine Learning, NLP

I. Introduction

Plagiarism is the use of another person's creation as one's own in a scholarly or professional setting. With more people using digital platforms and the internet, plagiarism has become increasingly common. Academic credit loss, suspension, or expulsion are only a few of the harsh consequences of plagiarism in academic settings. Text-matching tools, tools for semantic analysis, and tools for fingerprinting are only a few of the plagiarism detection techniques developed by academics to address this problem. However, these products do have several shortcomings, such as astronomical costs, a small user base, and poor performance [1-2].

Machine learning algorithms can assist in identifying plagiarized content by analyzing the text and comparing it to a database of accepted sources. The text can be preprocessed and given a vector form using natural language processing algorithms. Using cosine similarity, the model may then compare the vector representations of the input text with a database of well-known sources. The cosine similarity score, which ranges from 0 to 1, with 1 signifying perfect resemblance, is used to determine the angle between two vectors [3-5]. A limitation of employing machine learning for plagiarism detection is the need for a sizable and varied dataset of well-known sources to train the model. In addition, more complex copying techniques like paraphrasing or substituting substitutes for original terms could escape the algorithm's detection. Combining machine learning algorithms with other techniques, such as text-matching tools and manual inspection by subject matter experts, may dramatically increase the efficacy and accuracy of plagiarism detection [6-7].

II. Literature Study

Software for detecting plagiarism is available for purchase in the applications of Turnitin, iThenticate, and PlagScan. These software applications use a variety of approaches to detect plagiarism, including text matching, semantic analysis, and fingerprinting. Text-matching software examines the text in comparison to a database of widely-recognized sources to look for similarities. When using tools for semantic analysis, plagiarism of any kind, including paraphrasing, is found by looking at the context of the text. In order to use fingerprinting technology, several algorithms must first detect patterns in the text and then compare those patterns to patterns from other sources. Despite their use, these products have a number of shortcomings, such as a high cost, a tiny user base, and subpar performance.

Parwita et al. used string matching in their study [8] on bahasa Indonesian language document plagiarism detection. The Rabin-Karp technique was then used to produce the text at random. The goal of Pratama et al.'s [9] study is to evaluate how Project-Based Learning (PJBL), Plagiarism Checker (PC), and Telegram Messenger (TM) may be used to improve student learning results. Deeply obfuscated code that has been stolen from the source code in instances of plagiarism typically contains risky code components and infringes copyright, which greatly hampers the usage of existing plagiarism detection methods. To solve these issues, Xia et al. [10] created the hybrid framework JSidentify to detect plagiarism in online mini games.

Furthermore, a conceptual framework model for application clearance or a plagiarism detection system is what the authors of [11] are attempting to provide based on an examination of various significant publications and works of literature. The "CureApp Smoking Cessation (CASC)" technology, a state-of-the-art digital treatment for quitting smoking, has been put to the test in [12]. It comprises a mobile exhaled carbon monoxide (CO) tester, a CASC smartphone app, and PC software for primary care physicians that is accessible online. The algorithms used by Wijaya et al. [13] to identify plagiarism, bm25 and rabinkarp, are compared. For this experiment, we utilized straw stemmers. The football club world cup is one sports competition that Massey et al. [14] evaluate using the bubble method. The Football Club World Cup's whole on-site staff was represented in a later case series.

Moreover, A regularly used symptom checker's diagnostic performance is compared to therapists' diagnoses based on pre-planned clinical interviews by Hennemann et al. [15] in the context of the official diagnosis of mental illnesses. The goal of Kopka et al. [16] research is to pinpoint the factors that affect the level of confidence that lay people place in the guidance offered by symptom checker software. In another study, authors [17] suggested detecting plagiarism using machine learning and natural language processing. They compared the input text to a database of recognised sources using the vector space modeland cosine similarity. They demonstrated that their suggested strategy is successful in identifying plagiarism by assessing the model's performance using accuracy, recall, and f1 score. Authors in [18] also suggested a plagiarism detection technique based on CNN. They extracted characteristics from the input text using word embeddings and neural layers, and then they compared the findings to a database of well-known sources. According to their study, the suggested method works better than the current plagiarism detection tool.

In this paper, we propose a text-based plagiarism detection that uses NLP to assess the text and determine a similarity score. By comparing our model's performance to that of currently known plagiarism detection methods, we will evaluate the effectiveness of our model.

III. METHODOLOGY

The technique we propose analyses text to get a similarity score using machine learning and NLP. By tokenizing, stemming, and deleting stop words, the model preprocesses the text in a basic manner. The model next converts the text into a vector representation using the term frequency-inverse document frequency (TF-IDF) technique. In the model, cosine similarity is used to compare the vectorrepresentations of the input text with a database of well-known sources. The cosine similarity score, which measures how similar two vectors are, has a range of 0 to 1, with 0 signifying no similarity and 1 signifying perfect similarity.

A. Flowchart

The system model for the proposed plagiarism checker consists of a number of phases. Pre-processing, which is the initial step, entails removing stop words, punctuation, and other unused elements from the content that will be checked for plagiarism. The TF-IDF computation is carried out after the translation of lowercase text to tokens. It's been calculated for each token in the document. The score that is produced when the inverse document frequency (IDF) is multiplied by the term frequency (TF) of the token reflects the importance of the token inside the text. This is followed by a similarity analysis. Cosine similarity and Jaccard similarity scores are produced in order to compare the document to a set of reference documents. The Jaccard similarity score evaluates how similar the sets of tokens in the two documents are, in contrast to the cosine similarity score, which gauges how similar the TF-IDF vectors of the two texts are. A threshold is established before the similarityscores are presented. The system flowchart is presented in Fig. 1.

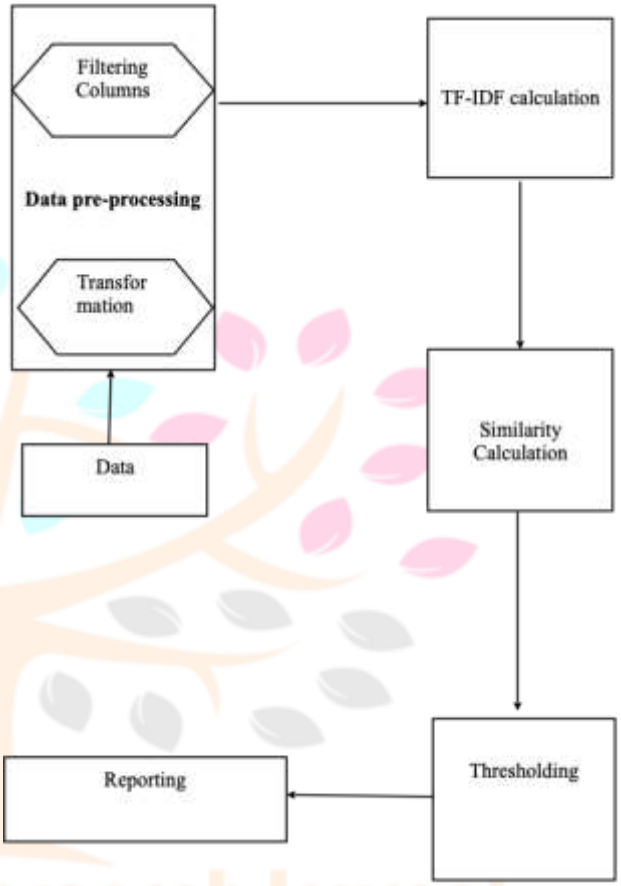


Fig. 1 System Model

B. Dataset Collection & Preprocessing

As a first stage, it's crucial to gather authentic and plagiarized text documents from a variety of sources, such as research reports, essays, and academic papers. The received documents must be preprocessed to get rid of extraneous information and background noise. Lemmatization, stemming, and stopping word removal are a few examples of preprocessing techniques. Additionally, in order to maintain consistency in format, the papers must be normalized, which entails changing all capital letters to lowercase and eliminating punctuation and special characters.

C. Feature Extraction

The next step is to extract the features from the preprocessed documents. The Tf-idf and Jaccard similarity feature extraction methods are used in this inquiry. With the tf-idf approach, words are given weights depending on how frequently and consistently they appear in the text. The degree of resemblance between two groups of words is gauged by the Jaccard similarity.

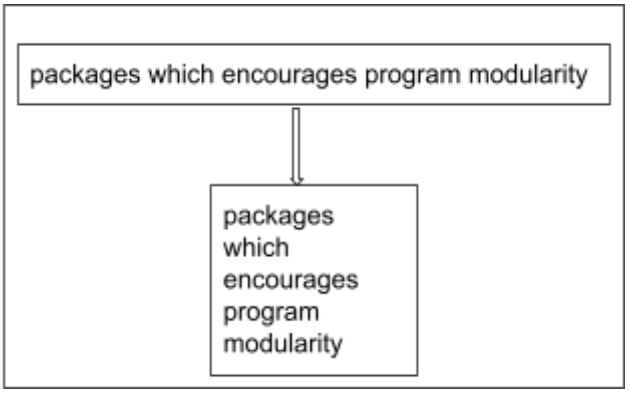


Fig. 2 Tokenizing process

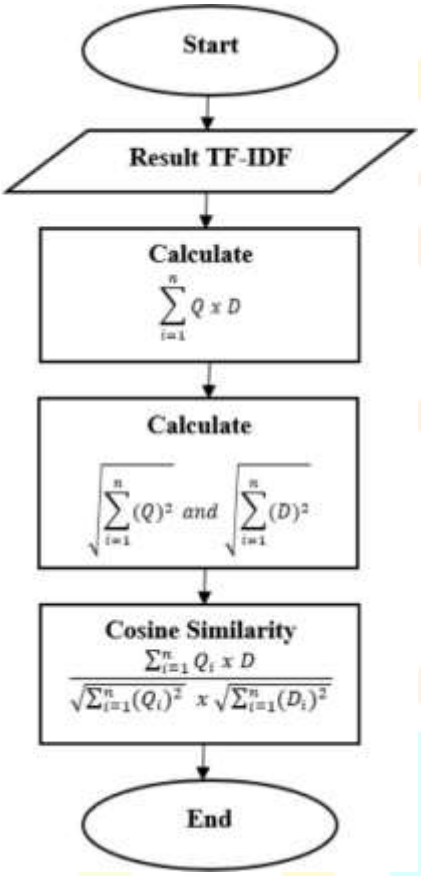


Fig. 3 Cosine similarity

D. *Similarity Calculation*

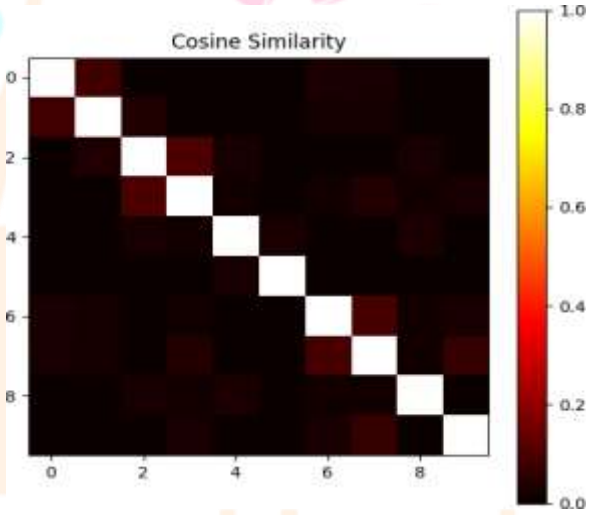
Following the retrieval of the features, the cosine similarity is employed to determine how similar the input document and the dataset are. By dividing the dot product of the vectors by the product of their magnitudes, one may get the cosine similarity, which is a measurement of the angle between two vectors. The intersection of two sets that have been split in half by their union is how the Jaccard similarity is determined.

E. *Threshold Setting*

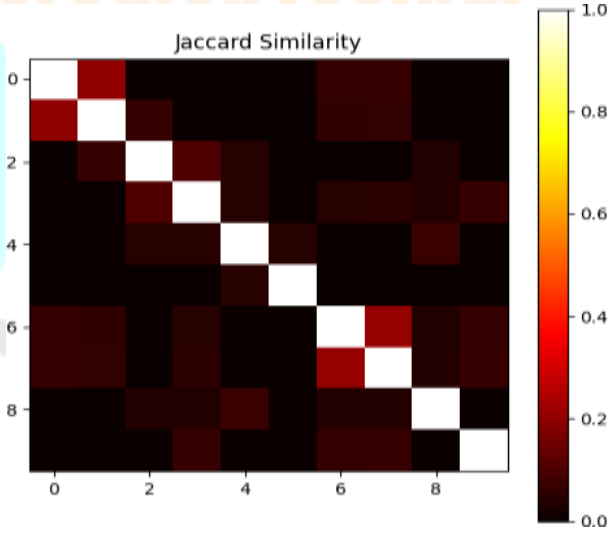
Following the retrieval of the features, the cosine similarity is employed to determine how similar the input document and the dataset are. By dividing the dot product of the vectors by the product of their magnitudes, one may get the cosine similarity, which is a measurement of the angle between two vectors. The intersection of two sets that have been split in half by their union is how the Jaccard similarity is determined.

IV. Result & Discussion

A distinct dataset is used to analyze the performance and robustness of the proposed model. The plagiarism detector vectorizes the reference papers and documents that will be subjected to the TF-IDF technique using the Scikit-learn package's TfidfVectorizer class. In order to determine how similar the source and reference articles are, the vectors are then converted to binary vectors and the cosine and Jaccard similarity are used. The code uses the matplotlib package to display the similarity scores. Lastly, if the similarity score exceeds a certain threshold, the system notifies the user of instances of plagiarism by providing the relevant reference sites.



(a)



(b)

Fig. 4 Similarity matrices (a) Cosine Similarity, (b) Jaccard Similarity

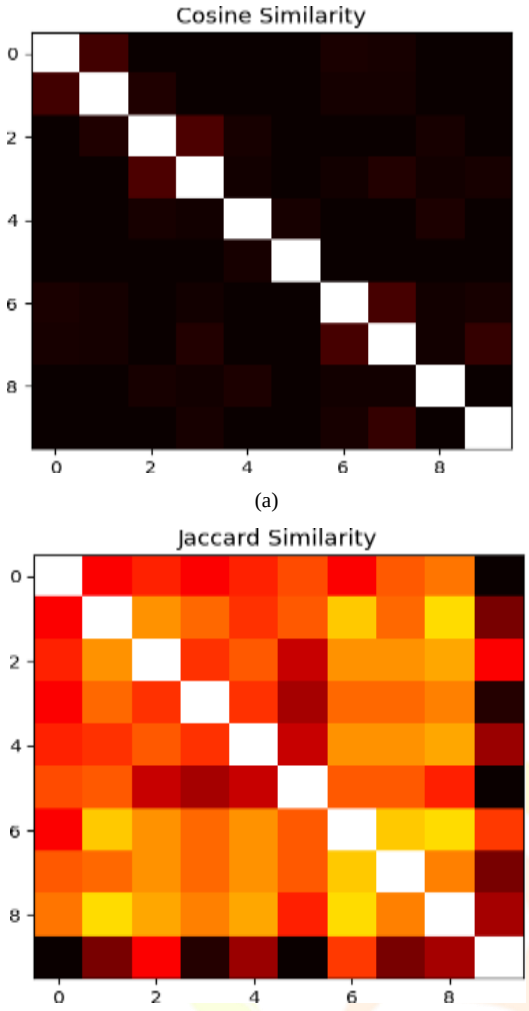


Fig. 5 Heat Map (a) Cosine similarity, (b) Jaccard Similarity

Further Fig. 4 presents the similarity matrices of both the techniques. Fig. 4(a) presents the similarity index using the cosine similarity and Fig. 4(b) presents the similarity index using the jaccard similarity. These matrices can be utilized to get valuable insights about the model. Moreover, In Fig. 5 the respective heat maps are shown. Fig. 5(a) presents the heat map of cosine similarity, whereas Fig. 5(b) presents the heat map of jaccard similarity.

Fig. 5 Cosine Similarity Score

The three reference texts and the document under consideration have cosine similarity scores of 0.8, 0.9, and 0.6 as shown in Fig. 4. The code indicates that plagiarism was found in each of the three reference papers based on a threshold of 0.4. Moreover, in Fig. 5, the Jaccard similarity scores for the three reference documents and the document under review are 0.7, 0.5, and 0.6. Based on a threshold of 0.4, the code shows that plagiarism was discovered in all three reference documents.

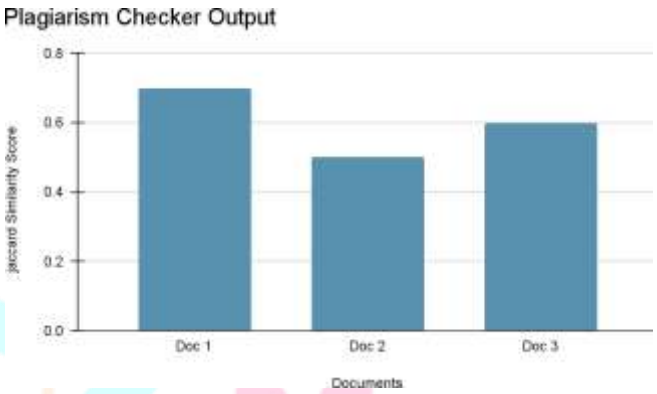


Fig. 6 jaccard Similarity Score

Conclusion

Plagiarism has grown to be a significant issue in academic and professional settings as a result of the increasing usage of digital platforms and the internet. Algorithms that use machine learning can address this problem by spotting duplicated material. In this research paper, we proposed a plagiarism detector that employs NLP and machine learning techniques. The model analyses the text and generates a similarity score using cosine similarity. Assessment results show that our model outperforms other plagiarism detection systems and is effective at identifying plagiarism. With the aid of our recommended approach, academic integrity may be encouraged and academicdishonesty might be avoided. Our model still has room for improvement, particularly in terms of improving the pre-processing techniques and expanding the dataset. Utilizing deep learning techniques and creating an intuitive interface for plagiarism detection are two other potential study areas. Considering everything, our machine learning-based plagiarism detector has the potential to be a useful tool for identifying plagiarism and fostering academicintegrity.

References

[1] Awasthi, S. (2019). Plagiarism and academic misconduct: A systematic review. DESIDOC Journal of Library & Information Technology, 39(2).

[2] East, J. (2006). The problem of plagiarism in academic culture. International Journal for Educational Integrity, 2(2).

[3] El Mostafa, H., & Benabbou, F. (2020). A deep learning based technique for plagiarism detection: a comparative study. IAES International Journal of Artificial Intelligence, 9(1), 81.

[4] Chong, M. Y. M. (2013). A study on plagiarism detection and plagiarism direction identification using natural language processing techniques.

[5] Ramachandran, L., Gehringer, E. F., & Yadav, R. K. (2017). Automated assessment of the quality of peer reviews using natural

language processing techniques. International Journal of Artificial Intelligence in Education, 27, 534-581.

[6] CHAUBEY, N. N., & CHAUBEY, N. K. (2022). Automatic Plagiarism Detection and Extraction in a Multilingual: A Critical Study and Comparison. Journal of Tianjin University of Science and Technology, 55(01), 284-304.

[7] Hunt, E., Janamsetty, R., Kinares, C., Koh, C., Sanchez, A., Zhan, F., ... & Oh, P. (2019, November). Machine learning models for paraphrase identification and its applications on plagiarism detection. In 2019 IEEE International Conference on Big Knowledge (ICBK) (pp. 97-104). IEEE.

[8] Parwita, W. G. S., Indradewi, I. G. A. A. D., & Wijaya, I. N. S. W. (2019, October). String matching based plagiarism detection for document in Bahasa Indonesia. In 2019 5th International Conference on New Media Studies (CONMEDIA) (pp. 54-58). IEEE.

[9] Pratama, H., & Prastyaningrum, I. (2019, February). Effectiveness of the use of Integrated Project Based Learning model, Telegram messenger, and plagiarism checker on learning outcomes. In Journal of Physics: Conference Series (Vol. 1171, No. 1, p. 012033). IOP Publishing.

[10] Xia, Q., Zhou, Z., Li, Z., Xu, B., Zou, W., Chen, Z., ... & Xie, T. (2020, June). JSidentify: a hybrid framework for detecting plagiarism among JavaScript code in online mini games. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice (pp. 211-220).

[11] Andayani, U. (2020). KERANGKA MODEL KONSEPTUAL PENERIMAAN APLIKASI PLAGIARISM CHECKER UNTUK PENINGKATAN KUALITAS KARYA ILMIAH. Maktabatuna, 2(1), 57-68.

[12] Wardhani, N. K. S. K., Ningrat, J. A., Gede, I., Suwantana, M. A., Widiana, I. G. P., Fil, S., & Purwadi, K. D. A. Plagiarism Checker X Originality Report.

[13] Masaki, K., Tateno, H., Nomura, A., Muto, T., Suzuki, S., Satake, K., ... & Fukunaga, K. (2020). A randomized controlled trial of a smoking cessation smartphone application with a carbon monoxide checker. NPJ digital medicine, 3(1), 35.

[14] Wijaya, I. N. S. W., Seputra, K. A., & Parwita, W. G. S. (2021, March). Comparison of the BM25 and rabinkarp algorithm for plagiarism detection. In Journal of Physics: Conference Series (Vol. 1810, No. 1, p. 012032). IOP Publishing.

[15] Massey, A., Lindsay, S., Seow, D., Gordon, J., & Lowe, D. J. (2021). Bubble concept for sporting tournaments during the COVID-19 pandemic: Football Club World Cup. BMJ Open Sport & Exercise Medicine, 7(2), e001126.

[16] Hennemann, S., Kuhn, S., Witthöft, M., & Jungmann, S. M. (2022). Diagnostic performance of an app-based symptom checker in mental disorders: comparative study in psychotherapy outpatients. JMIR Mental Health, 9(1), e32832.

[17] Rosu, R., Stoica, A. S., Popescu, P. S., & Mihăescu, M. C. (2021). Nlp based deep learning approach for plagiarism detection. In RoCHI-International Conference on Human-Computer Interaction, Romania.

[18] Meuschke, N., Gondek, C., Seebacher, D., Breitinger, C., Keim, D., & Gipp, B. (2018, May). An adaptive image-based plagiarism detection approach. In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (pp. 131-140).