IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Plagiarism Checker

Guided by: prof. Anand Donald

[1]Dipti Bangde, [2]Afiya Mohammad, [3]Gayatri Burande, [4]Pratiksha Kande

[1, 2, 3, 4]Department of Computer Science Engineering, Rajiv Gandhi College of Engineering Research and Technology, Chandrapur, Maharashtra, India

Abstract: Our research aims to detect plagiarism in computer programming assignments. Plagiarism has been a problem for a long time and the problem has evolved with time. With the rise of the internet the theft of intellectual property has risen significantly and recognizing these thefts has become difficult. With this survey we have identified techniques used in detecting plagiarism. With changing times, the tools needed to detect plagiarism have to be evolved. However, to develop a tool with an ability to achieve high accuracy and greater accessibility of data has always been a demand. A comparative study on plagiarism checking tools with the technology used is presented in this paper. This study would help us determine the algorithm and methodology to proceed with the development of code to detect plagiarism.

Key Words: plagiarism detection; machine learning; artificial intelligence; deep learning; neural network

1 INTRODUCTION

Programming assignments play an important role to hone and evaluate programming skills of a student. But the process of finding a solution to the assignments seems frustrating to the majority of students that they borrow solutions from classmates or from external sources. 'Plagiarism' is presenting someone else's work or ideas as your own, with or without their consent, by incorporating it into your work without full acknowledgement. The increasing plagiarism cases in academics hinders fair evaluation of students, hence raises the need for plagiarism detection. Manual plagiarism detection takes a lot of time and effort, in the case of a large number of assignments. Hence, reliable automatic plagiarism detection techniques are necessary.

During this literature review, we came across several approaches that previous researchers have developed. The approaches were similarity-based, logic-based, machine learning based etc. Several comparison algorithms such as the Greedy String Tiling algorithm, winnowing algorithm etc. have been used in similarity-based detection. While logicbased algorithms try to find out one dissimilarity in terms of output and execution paths. For a system, which involved multiple submissions of a single exercise, similarity of consecutive submissions made by a student were considered which eliminates the need for finding similarity between all the source codes submitted by all the students. Machine learning techniques such as XGBoost, SVM, random forest, decision trees have also been used in some approaches. The take away and limitations of each of them have been briefed in this review. This review gives us an idea of what areas to work on while building a plagiarism detection system.

2 MOTIVATION

The practice of plagiarism is not a strange thing anymore. Students, programmers and even lecturers plagiarize or copy the source code from different sources before submitting it to the evaluator. This isn't just limited to schools and colleges but also in the industries. Detecting plagiarism practices is a solution that should be done so that the fraudulent actions can be minimized. Detection of plagiarism clusters is very important to find out how many students or groups accomplishing program homework independently. It gives instructors more opportunity to enhance or modify education. Using the software can be a deterrent for students to plagiarism. However, using this software does not provide the final answer, which is why the authors have come up with an idea of using two approaches and comparing them to find out which performs better, the first technique is to use machine learning techniques like XGBoost algorithm, SVM classifier and the second technique is to make use of Artificial Neural Network and backpropagation on the dataset to identify if there is plagiarism in code

3 LITERATURE REVIEW

The literature surrounding plagiarism checkers underscores their pivotal role in the academic landscape, with an increasing emphasis on technological solutions to counteract the rise of plagiarism. Scholars have extensively explored the methodologies underpinning these tools, investigating the efficacy of various algorithms in detecting similarities and ensuring the accuracy of results. Considerations of precision and recall metrics have been central, aiming to strike a balance between identifying instances of plagiarism and minimizing false positives. Furthermore, researchers delve into the challenges posed by evolving forms of plagiarism, such as paraphrasing and mosaic writing, urging the continual refinement of detection mechanisms to address these nuances.

The impact of plagiarism checkers on educational practices is a recurring theme in the literature, as educators grapple with maintaining academic integrity in a digital age. Studies delve into the integration of these tools into educational workflows, discussing their influence on teaching strategies and student learning outcomes. Scholars also scrutinize the ethical implications of plagiarism detection, considering issues of privacy, fairness, and the potential for unintended consequences. The literature reveals an ongoing discourse on the role of plagiarism checkers in fostering a culture of originality, discouraging academic dishonesty, and shaping pedagogical approaches.

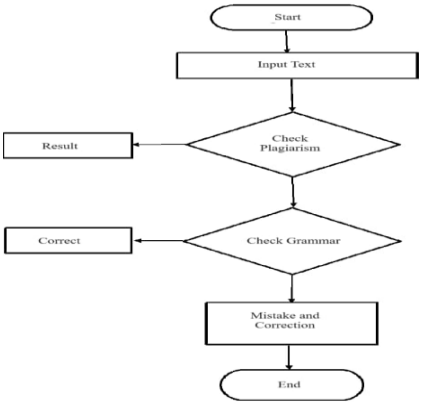
While the literature acknowledges the progress made in plagiarism detection technology, it consistently advocates for continued research and development. Comprehensive evaluations of existing tools, comparative analyses of different systems, and investigations into their impact across diverse academic disciplines remain crucial focal points. The evolving nature of plagiarism and the dynamic academic landscape necessitate ongoing efforts to enhance the precision, adaptability, and user-friendliness of plagiarism checkers, ensuring their effectiveness in upholding academic integrity.

4 METHODOLOGY

A plagiarism checker typically employs a multifaceted methodology to identify and flag potential instances of plagiarism within a given text. The process involves several key steps.

Firstly, the system utilizes advanced algorithms to break down the text into smaller units, such as sentences or phrases. These units are then compared against an extensive database containing a wide range of academic papers, articles, books, and other sources. The comparison involves analyzing the linguistic and structural features of the text units, looking for similarities with existing content. Sophisticated algorithms assess not only exact matches but also variations in wording and sentence structure. Additionally, the system considers the contextual relevance of the identified similarities to distinguish between common phrases and potential instances of plagiarism. Furthermore, many plagiarism checkers integrate machine learning techniques to enhance their accuracy over time. They continuously learn from new data, adapting to evolving language patterns and ensuring the system remains effective against various forms of plagiarism.

Overall, the methodology combines linguistic analysis, pattern recognition, and database comparisons to provide a comprehensive assessment of the text's originality. It's important to note that while plagiarism checkers can efficiently identify potential matches, human judgment is often necessary to interpret the results and determine whether the similarities constitute plagiarism or permissible overlap in content.



5. Working

A plagiarism checker is a tool designed to identify and flag instances of plagiarism in written content. Its functioning involves several key steps. First, the user submits the text or document to be checked. The plagiarism checker then breaks down the content into smaller units, such as sentences or phrases, and creates a unique fingerprint for each unit using algorithms Next, the tool compares these fingerprints against a vast database that contains a wide range of sources, including academic papers, articles, books, and websites. The database may be compiled from publicly available information or licensed from various publishers and online repositories. The comparison helps identify similarities between the submitted content and existing sources.

Plagiarism checkers use advanced algorithms, including text-matching and natural language processing techniques, to assess the level of similarity. Some tools provide a percentage of similarity, indicating how much of the content matches existing sources. Additionally, they often highlight specific passages or sentences where similarities are found, allowing users to review and address potential plagiarism to enhance accuracy, plagiarism checkers may employ machine learning algorithms that continuously learn and adapt to evolving patterns of plagiarism. Regular updates to the database also ensure that the tool remains effective in identifying content from new publications and online sources.

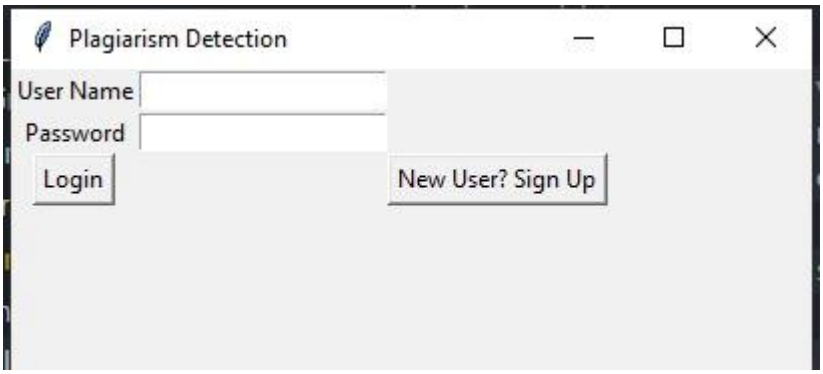
Institutions such as educational organizations, publishers, and businesses use plagiarism checkers to uphold academic integrity, protect intellectual property, and maintain content authenticity. It's essential to note that while these tools are valuable in identifying potential plagiarism, human judgment is crucial for interpreting results and considering factors like proper citation and fair use. Overall, plagiarism checkers play a vital role in promoting ethical writing practices and ensuring the originality of content in various domains.

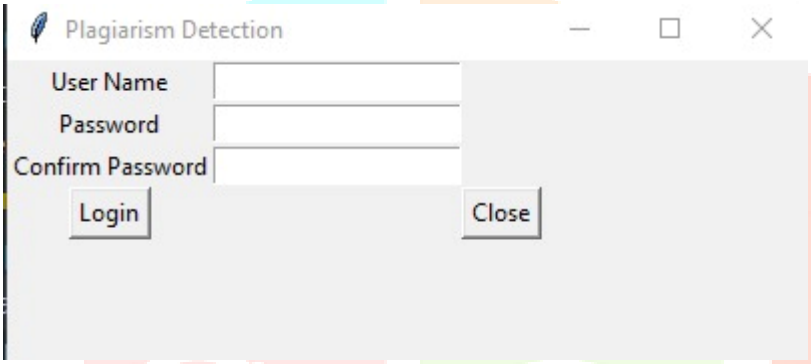
6. CONCLUSION

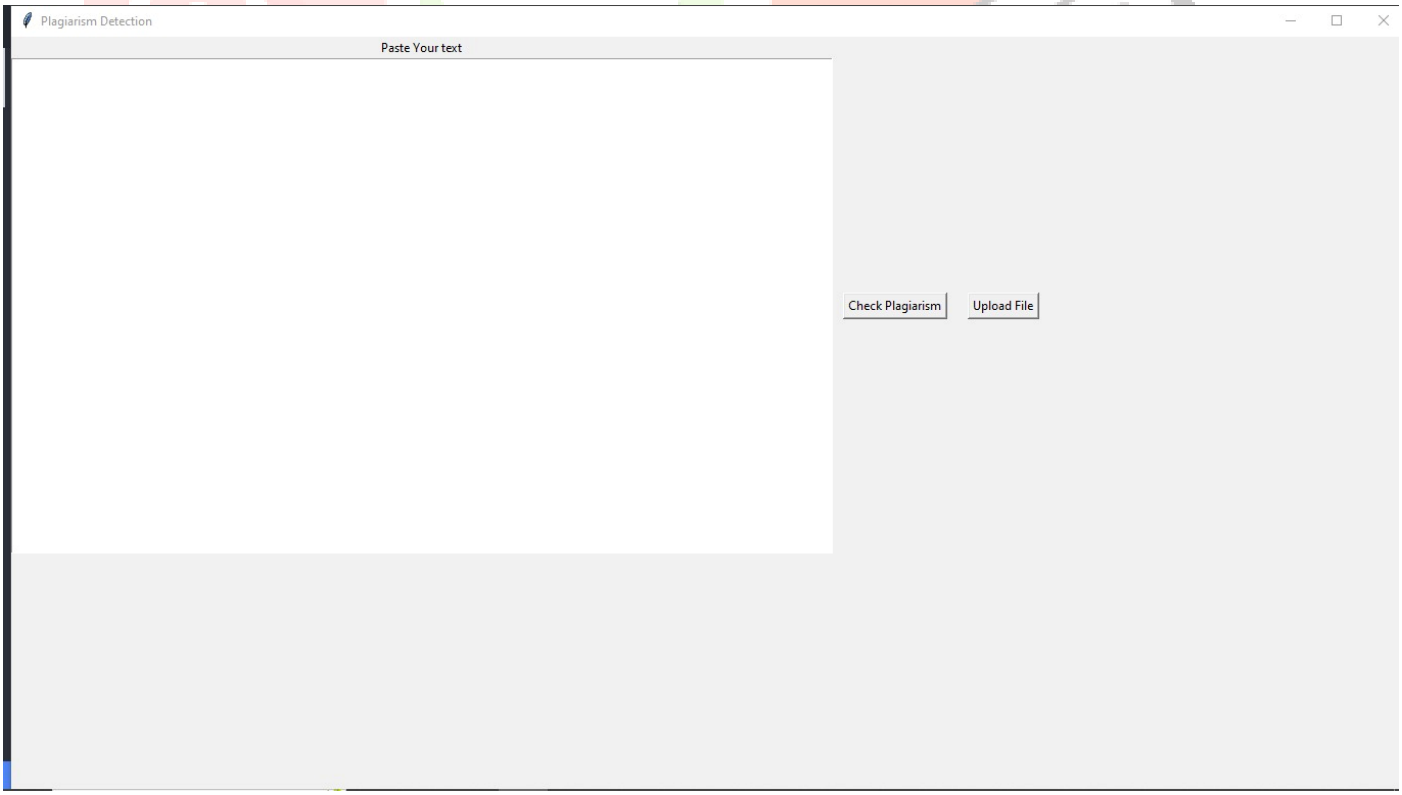
Plagiarism is a ubiquitous problem faced by practitioners of different fields like academia, journalism, literature, art and so on for decades. The field has been researched intensively since the 1970's. With the advances in technology and the pervasiveness of the world wide web, everyone has all the information they need at their fingertips. Especially in academia, this poses a problem to fair evaluation of the students and also inhibits the student's learning process. While many efforts were concentrated towards detecting textual plagiarism, significant strides have been made in the field of source code plagiarism detection. We can observe the leap from manual plagiarism checking to algorithm-based, automatic plagiarism checking made possible by advancements in technology. We can also observe the shift from using local client applications to web-based applications and now to cloud-based applications, making plagiarism detection systems easy to use and

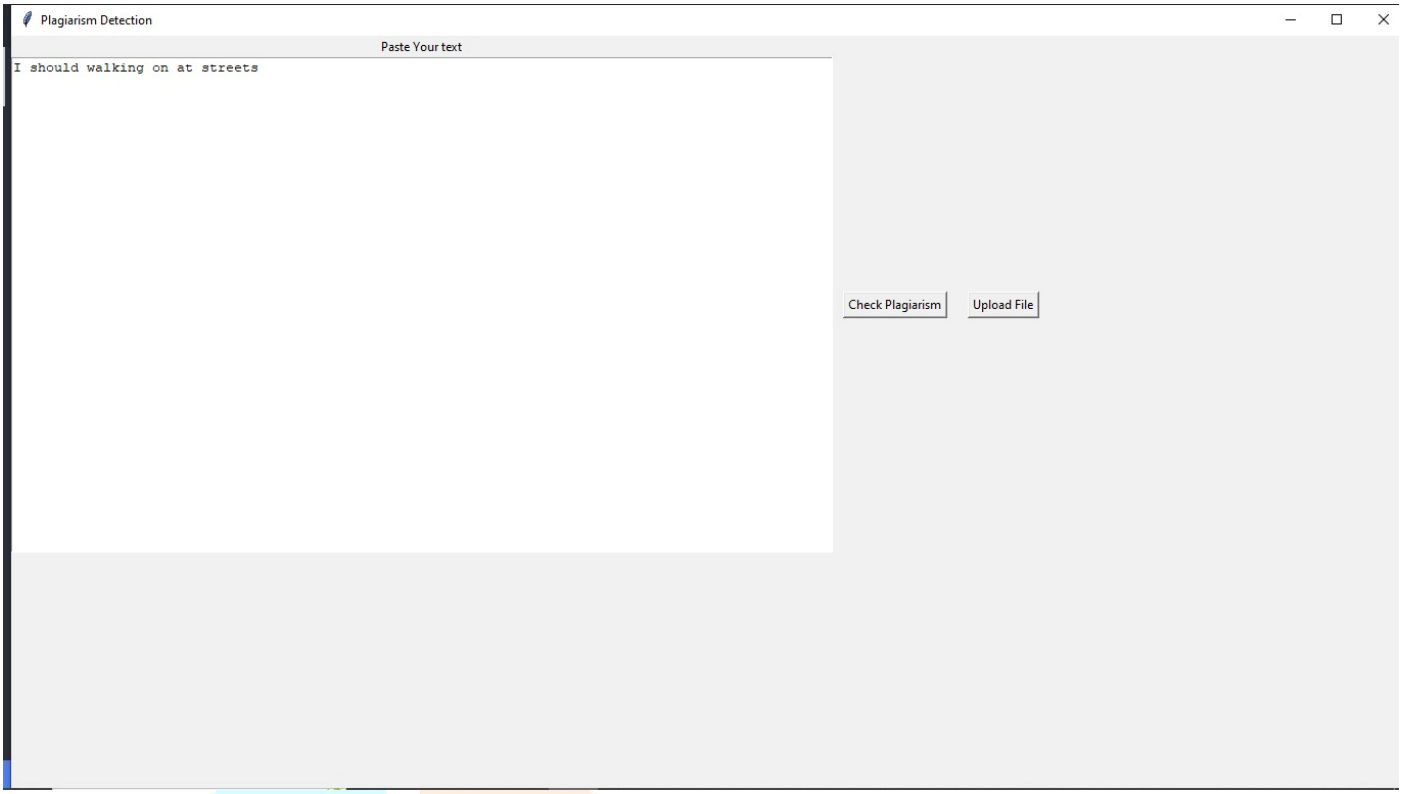
available everywhere. The detection methods started from simple feature-based approach, structure-based approach, then moved on to hybrid approaches that used similarity measurement and string-matching algorithms as students started to evade these systems. Recent approaches using Machine Learning and Deep Learning algorithms and techniques have shown some promising results to improve the accuracy and automating the process of plagiarism detection.

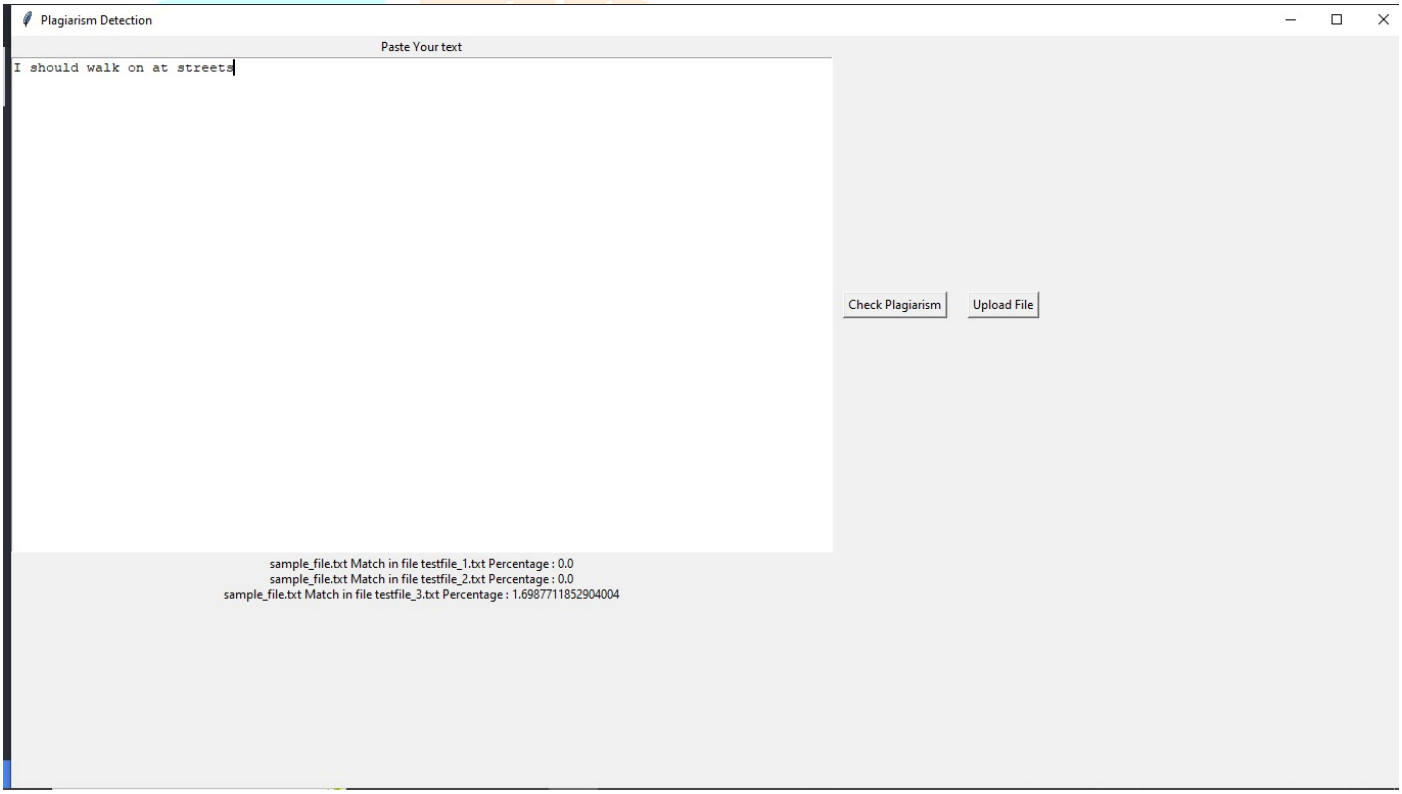
7. Output








Plagiarism Detection
Paste Your text
I should walking on at streets
Check Plagiarism
Upload File


Plagiarism Detection
Paste Your text
I should walk on at streets
Check Plagiarism
Upload File
sample_file.txt Match in file testfile_1.txt Percentage : 0.0
sample_file.txt Match in file testfile_2.txt Percentage : 0.0
sample_file.txt Match in file testfile_3.txt Percentage : 1.6987711852904004

8. REFERENCES

[1] Karl J Ottenstein, “An algorithmic approach to the detection and prevention of plagiarism”, ACM SIGCSE Bulletin, Volume 8, Issue 4, Dec. 1976, pp 30–41.

[2] Joseph L.F. De Kerf, “APL and Halstead's theory of software metrics”, APL '81: Proceedings of the international conference on APL, October 1981, Pages 89–93.

[3] John L Donaldson, Ann Marie Lancaster and Paula H Sposato, “A plagiarism detection system”, SIGCSE '81: Proceedings of the twelfth SIGCSE technical symposium on Computer science education, February 1981, Pages 21–25.

[4] Alan Parker and James O Hamblen “Computer Algorithms for Plagiarism Detection”, IEEE Transactions On Education, Vol. 32, No. 2. May 1989.

[5] Michael J Wise, “YAP3: improved detection of similarities in computer program and other texts”, SIGCSE '96: Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education, March 1996, Pages 130–134.

[6] Saul Schleimer, Daniel S. Wilkerson and Alex Aiken, “Winnowing: Local Algorithms for Document Fingerprinting”, SIGMOD 2003, June 9-12, 2003, San Diego, CA. Copyright 2003 ACM 1-58113-634-X/03/06.

[7] Richard M. Karp and Michael O. Rabin, “Efficient randomized pattern-matching algorithms”, Published in: IBM Journal of Research and Development (Volume: 31, Issue: 2, March 1987), Page(s): 249 - 260.

[8] Lutz Prechelt and Guido Malpohl, “Finding Plagiarisms among a Set of Programs with JPlag”, March 2003, Journal Of Universal Computer Science 8(11).

[9] Sven Meyer zu Eissen and Benno Stein, “Intrinsic Plagiarism Detection”, M. Lalmas et al. (Eds.): ECIR 2006, LNCS 3936, pp. 565–569, 2006.

[10] Liang Zhang, Yue-ting Zhuang and Zhen-ming Yuan, “A Program Plagiarism Detection Model Based on Information Distance and Clustering”, Published in: The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), Date Added to IEEE Xplore: 22 January 2008,Print ISBN:978-0-7695-3006-2