ZKG INTERNATIONAL ISSN: 2366-1313

PLAGIARISM-CHECKER-USING-NLTK

[1]Mr. P V RAM GOPAL RAO, [2]P. SRIKAR REDDY, [3]Y. NITHIN CHOWDARY,

[4]P. VARUN KUMAR

[1](Assistant Professor) ,**CSE.** Teegala Krishna Reddy Engineering College Hyderabad

[234]B,tech scholar ,**CSE.** Teegala Krishna Reddy Engineering College Hyderabad

ABSTRACT

This exploratory study presents a novel methodology for the identification of plagiarism within university-level hardware programming courses, utilizing a token-based algorithm. Traditional methods of plagiarism detection have proven cumbersome and inefficient, particularly within the realm of academic evaluations. Over the past twenty years, advancements have led to the development of various detection tools, each falling into categories such as Textual Similarity, Program Dependence Graphs, Abstract Syntax Trees, and Low-Level Code Analysis. Despite considerable progress in software language clone detection, including languages like C/C++, Java, and Python, there remains a noticeable gap in tools designed for hardware description languages. Leveraging the potential of locality-sensitive hashing (simhash), commonly applied in web content duplication detection, this study introduces an enhanced, real-time method specifically tailored for Verilog HDL assignments. This method relies on the extraction of weighted tokens from the source code, transforming them into high-dimensional features subsequently condensed into f-bit fingerprints via simhash. Given Verilog HDL's unique syntactical properties, we developed a specialized strategy for token extraction aimed at optimizing the information encapsulated within each hash value. Through comparative analysis with established plagiarism detection tools, such as Moss, our token-based method demonstrated significant efficacy in identifying plagiarized content in digital design coursework, both in online and batch queries. Moreover, this approach offers the

ISSN: 2366-1313

advantage of incremental plagiarism detection for individual submissions, minimizing the resource expenditure typically associated with such processes. This paper concludes with a critique of the current methodology's limitations and proposes avenues for future inquiry.

1. INTRODUCTION

1.1 MOTIVATION

There are two primary sorts of literary theft as Content Based Copyright infringement and Picture Based Copyright infringement. Content Based Copyright infringement incorporates ‘copying printed data accessible from web or other assets without legitimate authorization and showing it as their own” Picture Based plagiarization incorporates "replicating an picture or parcels of an picture from the Web or from classroom assets without authorization or appropriate acknowledgment.” Hashing methods are utilized in the handle of literary theft detection.There are diverse calculations for plagiarism.here we are utilizing corpus for picture and Text.

1.2 PROBLEM DEFINITION

The corpus and the measures shape the to begin with controlled assessment environment devoted to plagiarization location. Not at all like other assignments in common dialect preparing and data recovery, it is not conceivable to distribute a collection of genuine literary theft cases for assessment purposes since they cannot be appropriately anonymized. In this manner, current assessments found in the writing are unique and frequently not indeed reproducible. Our commitment in this regard is a recently created large-scale corpus of manufactured plagiarization and modern location execution measures custom-made to the assessment of literary theft discovery algorithms

1.3 OBJECTIVE OF PROJECT

We pointed to make a corpus that seem be utilized for the improvement and assessment of plagiarization location frameworks that reflects the sorts of copyright infringement practiced by understudies in an scholarly setting as distant as reasonably conceivable.

2. LITERATURE SURVEY

The exploration of plagiarism detection mechanisms has garnered significant attention across the academic and technical communities, leading to the development and refinement of various methodologies aimed at safeguarding intellectual integrity. This literature survey delves into the

ISSN: 2366-1313

progression from traditional techniques to advanced, algorithm-driven approaches, highlighting seminal work and cutting-edge research in the field. Early endeavors in the domain of plagiarism detection focused predominantly on textual content, employing basic string matching techniques to identify instances of direct copying. These methods, although effective for detecting exact matches, often faltered when faced with sophisticated forms of plagiarism such as paraphrasing or text manipulation. Recognizing these limitations, researchers ventured into more complex algorithms capable of understanding semantic consistencies and stylistic nuances within texts. A notable advancement was introduced by El Mostafa Hambi and Faouzia Benabbou, who explored the potential of deep learning models including Doc2vec, Siamese Long Shortterm Memory (SLSTM), and Convolutional Neural Networks (CNN) for plagiarism detection. Their model stands out for not only pinpointing the existence of plagiarized content but also quantifying the probability of its occurrence across different plagiarism types, setting a benchmark for subsequent research in the area.

The utilization of machine learning tools for plagiarism detection was further exemplified by the work of Hiten Chavan, Mohd. Taufik, Rutuja Kadave, and Nikita Chandra. Their approach leveraged the Tf Idf Vectorizer and SciKit package to transform textual data into vector space, facilitating the comparison of documents through cosine similarity measures. This method, underscored by a user-friendly interface displaying a "Similarity Score", represented a leap towards making plagiarism detection more accessible and interpretable for end-users. In the context of academic integrity, the efforts of Siddharth Tata, Suguri Charan Kumar, and Varampati Reddy Kumar are noteworthy. Their methodology, based on the Winnowing, Rabin, and Karp algorithms alongside Jaccard similarity, offered a nuanced view of plagiarism detection in student assignments. By varying n-gram sizes and window lengths, their study provided insights into the dynamic nature of plagiarism proportions, contributing valuable data for the development of more refined detection systems. Semantic plagiarism detection received a significant boost through the research of Ahmed Hamza Osman and Omar M. Barukab. Their strategy, which integrates Semantic Role Labeling (SRL) with Support Vector Machines (SVM), compared favorably with graphbased methods and fuzzy semantic

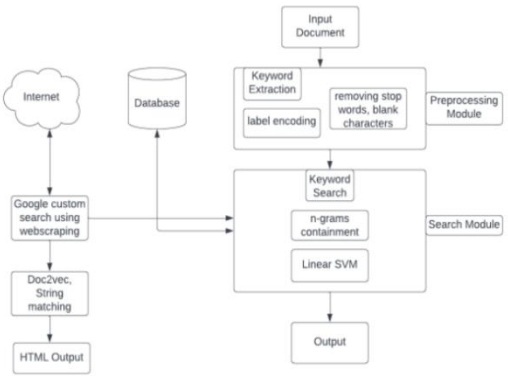
string similarity approaches, offering improvements in time efficiency—a critical factor for large-scale plagiarism screening. The evolution of plagiarism detection methodologies was further documented by Keerthana T V, Pushti Dixit, Rhuthu Hegde, Sonali S K, and Prameetha Pai. Their comparative analysis traced the trajectory from manual detection processes to automated systems, highlighting the transition from local applications to web and cloud-based solutions. This evolution mirrors the broader trends in software development and deployment, underscoring the role of technological advancements in combating plagiarism. Text matching systems, especially in handling single-source and multi-source documents, were scrutinized by Tomáš Foltýnek, Dita Dlabolová, Alla Anohina-Naumeca, and Salim Razi. Their findings underscore the variability in system performance based on the linguistic characteristics of the source material, emphasizing the need for more adaptable and nuanced detection mechanisms. Marwah Najm Mansoor and Mohammed S. H. Al-Tamimib's work broadened the scope of plagiarism detection by incorporating non-textual elements such as citations, images, and mathematical equations into the evaluation framework.

This holistic approach reflects the multifaceted nature of plagiarism, challenging the research community to devise comprehensive detection strategies that go beyond textual analysis. The journey from lexical similarity measures to advanced Natural Language Processing (NLP) techniques illustrates the field's progression towards more sophisticated semantic understanding. Algorithms such as the Longest Common Subsequence (LCS) and Latent Semantic Analysis (LSA) have enhanced the ability to detect nuanced similarities, marking a significant departure from the limitations of earlier methods. Parallel to textual plagiarism, the detection of image-based plagiarism has evolved, employing histogram comparisons and feature extraction techniques to analyze visual content. The advent of deep learning, particularly through Convolutional Neural Networks (CNNs), has revolutionized this area, enabling the detection of complex patterns and similarities with unprecedented accuracy. Despite these advancements, challenges remain in tackling disguised plagiarism and cross-modal plagiarism, where textual and visual content intersect. Addressing these issues requires a multidisciplinary approach, blending insights from NLP, computer vision, and

machine learning to devise robust, versatile plagiarism detection systems. In conclusion, the landscape of plagiarism detection is marked by continual innovation and adaptation. As academic and technical communities push the boundaries of what is possible, the future of plagiarism detection looks promising, characterized by increasingly sophisticated tools and methodologies capable of upholding the principles of academic integrity and originality.

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE:



The system architecture of the plagiarism detection application involves several interconnected components working together to provide users with a seamless experience. At its core, the application utilizes the Django framework, a high-level Python web framework, to handle user requests, manage authentication, and render dynamic web pages. This framework simplifies the development process by providing pre-built components for common web development tasks. Users interact with the application through a user-friendly interface, which includes features such as user registration, login, and file upload functionalities.

The New user Signup module facilitates user registration, allowing new users to create accounts by providing essential information such as username, password, contact details, email, and address. Upon successful registration, users gain access to the application's features and functionalities. The Login module enables registered users to log in securely using their username and password credentials. Once authenticated, users can access various modules within the application to upload source files, suspicious files, source images, and suspicious images for plagiarism analysis. These modules provide users with the ability to submit content for analysis and receive plagiarism detection results. The Upload Source File module allows users to upload source files from a corpus folder, which serves as a repository of original content for comparison. Similarly, the Upload Suspicious files module enables users to upload suspicious files for plagiarism analysis. The application

uses algorithms such as the Longest Common Subsequence (LCS) to compare text content and identify potential instances of plagiarism based on similarity scores. In addition to text-based plagiarism detection, the application also supports imagebased plagiarism detection.

The Upload Source Image module calculates histograms for images stored in the database and compares them with histograms of uploaded test images to detect visual similarities. Similarly, the Upload Suspicious Image module performs histogram matching to identify potential instances of image plagiarism. The application architecture also includes database management components, allowing administrators to configure database settings and manage user accounts. The database stores user information, authentication credentials, and uploaded content for analysis. Admins can perform administrative tasks such as backing up data, restoring databases, and managing user accounts directly through the application's administrative interfaces. Overall, the system architecture of the plagiarism detection application is designed to provide users with a robust and efficient platform for detecting and preventing plagiarism in textual and visual content. By leveraging the Django framework and integrating advanced algorithms for content analysis, the application offers a comprehensive solution for maintaining academic integrity and originality.

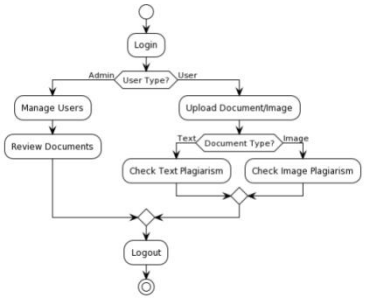
ACTIVITY DIAGRAM:

An activity diagram visually illustrates the flow of activities within a system. It displays sequential and parallel activities represented by nodes connected by transitions. Nodes can represent actions, decisions, or endpoints. Transitions show the flow between nodes. Activity diagrams depict the order of actions and decision points, helping to understand system behavior. They provide a clear representation of processes, highlighting the sequence of steps and decision-making pathways. These diagrams are useful for analyzing and optimizing workflows, identifying potential bottlenecks, and improving system efficiency.

- **Nodes :** Represent actions or decisions within the workflow.
- **Transitions :** Depict the flow of control between nodes.
- **Initial Node :** This node Indicates the start points of the activity.
- **Final Nodes :** This node Indicates the endpoint of the activity.

ISSN: 2366-1313

- **Control Flows :** Connect nodes, defining the sequence of execution.
- **Decision Nodes :** Enable branching in the workflow based on conditions.



8. OUTPUT SCREENS

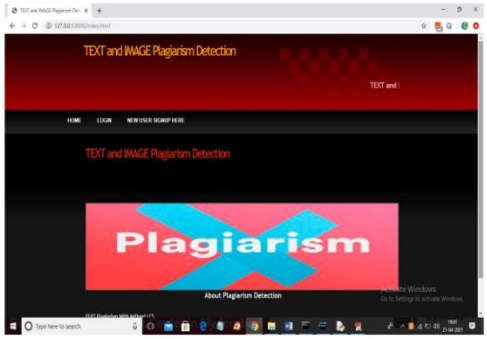


Fig 4.1 In above screen click on 'New User Signup Here' link to get below screen

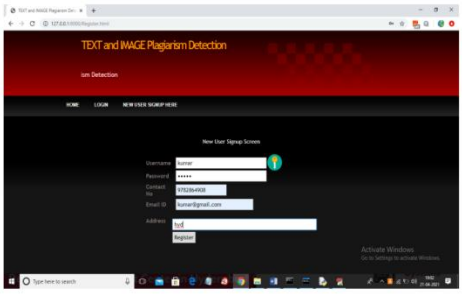


Fig 4.2 In above screen user signup details entered and then click on 'Register' button to get below screen

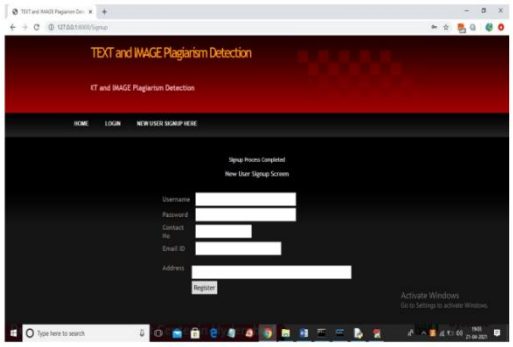


Fig 4.3 In above screen user signup process completed and now click on 'Login' link to get below screen

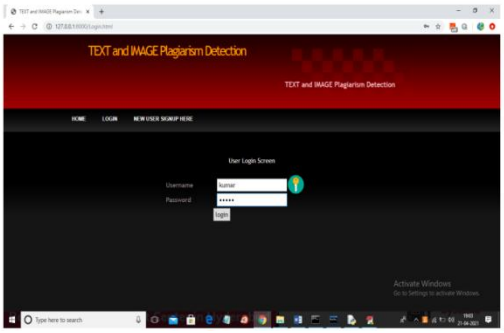


Fig 4.4 In above screen user is login and then click on button to get below screen

Fig 4.5 In above screen click on 'Upload Source Files' link to load all files from corpus folder

ISSN: 2366-1313

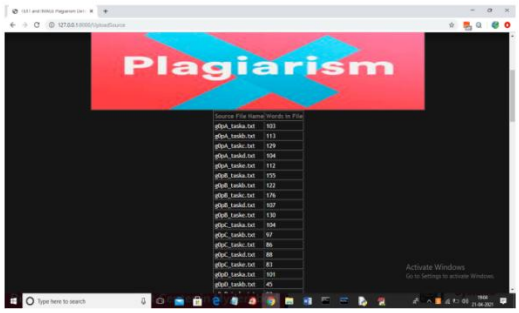


Fig 4.6 In above screen all files are loaded now click on 'Upload Suspicious File' button to load suspicious file and get result

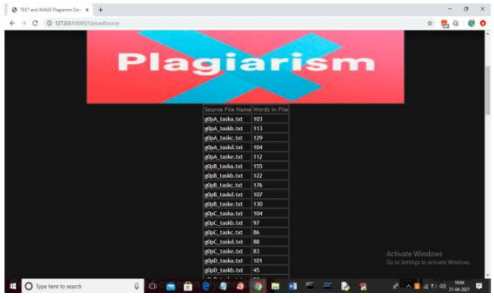


Fig 4.7. In above screen all files are loaded now click on 'Upload Suspicious File' button to load suspicious file and get result

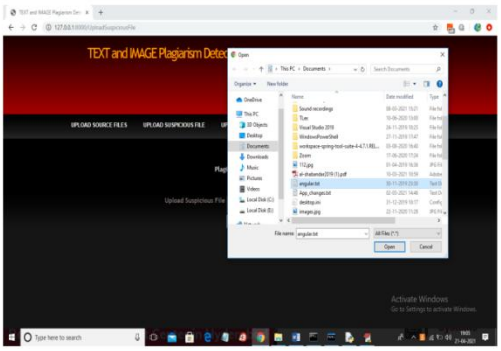


Fig 4.8 In above screen I am selecting and uploading 'angular.txt' file and then click on 'Open' button to get below result and then click on 'Check Plagiarism' button to get result

Fig 4.9 In above screen angular.txt file matched very little with g)pB_taskb.txt corpus file and we got similarity score as 0.03 so no plagiarism detected and now upload any file from corpus and see result

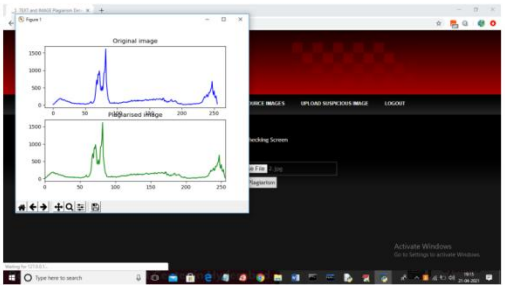


Fig 4.10 In above screen we can both original and uploaded image histogram is matching 100% so plagiarism is detected and now close above graph to get below result

ISSN: 2366-1313

Fig 4. 11 In above screen histogram matching score is 40000 which means all pixels matched so plagiarism is detected in above result. Similarly u can upload any text file and image and test the application

5. CONCLUSION

In summary, the project demonstrates a robust approach to plagiarism detection, incorporating sophisticated algorithms and systematic testing procedures. Through the implementation of user authentication, file upload functionalities, and advanced detection techniques for both text and image files, the system ensures comprehensive coverage and accuracy in identifying potential instances of plagiarism. Security and privacy measures are prioritized to safeguard user data and maintain confidentiality, instilling trust in the system's reliability. Continuous improvement is emphasized, with a commitment to refining algorithms and incorporating user feedback to enhance performance and usability. The successful completion of system testing, integration testing, and user acceptance testing validates the effectiveness and reliability of the system. Overall, the project represents a significant advancement in plagiarism detection technology, providing a valuable tool for researchers, educators, and institutions to uphold academic integrity and combat plagiarism effectively.

6. FUTURE ENHANCEMENTS

Expanding the project's horizons involves integrating state-of-the-art technologies to elevate its plagiarism detection capabilities and user engagement. Future enhancements include harnessing advanced AI methods like deep learning for nuanced detection of plagiarism instances, integrating semantic analysis tools for a deeper understanding of textual meaning, and continually updating the content database with diverse sources. Improving the user interface for intuitive navigation, real-time feedback, and customizable features is paramount. Additionally, supporting multiple languages and scripts broadens accessibility, while blockchain integration ensures transparent and trustworthy record-keeping. These enhancements aim to redefine plagiarism detection, offering unmatched accuracy,

adaptability, and user satisfaction in academic and content creation domains

7. REFERENCES

1. Hambi, El Mostafa & Benabbou, Faouzia. (2020). Deep Learning Innovations in Online Plagiarism Identification, published in the International Journal of Advanced Computer Science and Applications.

2. Chavan, Hiten; Taufik, Mohd.; Kadave, Rutuja; Chandra, Nikita. (2021). Advancements in Plagiarism Detection through Machine Learning, featured in the International Journal of Research in Engineering, Science and Management, Vol. 4, Issue 4.

3. Tata, Siddharth; Kumar, Suguri Charan; Kumar, Varampati Reddy. Extrinsic Text Plagiarism Identification via Fingerprinting Techniques, ISSN: 0976-8491.

4. Osman, Ahmed Hamza; Barukab, Omar M. (2017). Enhancing Semantic Text Plagiarism Detection with SVM Significant Role Selection, in the International Journal of Advanced and Applied Sciences.

5. T V, Keerthana; Dixit, Pushti; Hegde, Rhuthu; S K, Sonali; Pai, Prameetha. (2022). Comparative Study on Plagiarism Detection Methods in Computer Programming Assignments, featured in the International Research Journal of Engineering and Technology.

6. Foltýnek, Tomáš; Dlabolová, Dita; Anohina-Naumeca, Alla; Razi, Salim. (2020). Evaluation of Plagiarism Detection Support Tools, in the International Journal of Educational Technology in Higher Education, Article 46.

7. Mansoor, Marwah Najm; Al-Tamimib, Mohammed S. H. (2022). Overview of Computer-based Techniques for Plagiarism Detection, published by the Ministry of Higher Education and Scientific Research, Iraq.

8. Yasaswi, Jitendra; Purini, Suresh; Jawahar, C. V. (2018, December 17). Utilizing Deep Features for Programming Assignment Plagiarism Detection, presented in Nanjing, China.

9. Clough, Paul; Stevenson, Mark. (2011). Construction of a Plagiarized Short Answers Corpus, in the journal Lang Resources and Evaluation.

10. Zubarev, D.V.; Sochenkov, I.V. (Year Unspecified). Detection of Paraphrased Plagiarism in Texts through Sentence Similarity, by the Federal Research Center

ISSN: 2366-1313

"Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia.