Plagiarism Checker for Online Assessment

Ambigeswari R [1], Aruna Ramalakshmi P [1], Guru Gayathri V [1], Navedha Evanjalin R [2]

[1] *Student, Department of CSE, National Engineering College, Kovilpatti, Tamilnadu, India.*

[2] *Assistant Professor, Department of CSE, National Engineering College, Kovilpatti, Tamilnadu, India.*

Corresponding Author: 1912052@nec.edu.in

Abstract: - Plagiarism is a big problem in academics and every department in the education sector. Students plagiarize in different areas like assessments, projects, etc. Academics know that information can support valuable learning experiences, but these experiences are diminished when students plagiarize by copying assessments and getting credit for work they have yet to do. To avoid this issue, this project will develop a plagiarized tool. Plagiarism is a severe problem that needs to be monitored and controlled. Plagiarism refers to the act of blindly copying someone's unique work. In this proposed project, a plagiarism detector is designed to detect the percentage of plagiarism and the similarities of the copied text from other students. The plagiarism detector uses the Bag Of Words (BOW) technique to find the similarity between contents to get the proposed output. Initially, the user should register their identity to access the software. Each user is provided with a unique user id and password based on successful verification. Once the file has been uploaded, the developed software for plagiarism detection starts checking whether the assignment is plagiarized by comparing it with other student assignments and the Internet. The tool checks the content of the file line by line and detects if there is any plagiarism. It finds its applicability in exams that are conducted in online mode. This project's main aim is to identify each student's individuality and unique work.

Key Words: - ***Bag of Words (BOW), Natural Language Processing (NLP).***

I. INTRODUCTION

Plagiarism is when you use someone else's words or ideas in your own work without giving them proper credit. It is possible to do this with or without the other person's consent. This definition includes all published and unpublished materials, whether manuscript, print, or electronic. Plagiarism can be intentional or careless, or unintentional. According to the rules and regulations, deliberate or negligent plagiarism is a disciplinary offence.

Manuscript revised May 19, 2023; accepted May 20, 2023. Date of publication May 21, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

An online course that gives students a useful overview of the issues with plagiarism and instructs them on how to avoid it will be beneficial to them. In addition to text, other materials like computer code, drawings, schematics, etc. must also be cited when using their ideas or works.. This applies to both published text and information obtained from books and journals, such as unpublished texts and information from other students' lectures, theses, or essays. Any text, information, or other materials that you downloaded from websites must also be mentioned. Learning how to avoid plagiarism is the greatest method to apply the rules of good academic practice as soon as your academic career at the institution begins. Avoiding plagiarism is about more than just ensuring all references are correct and changing enough words so a researcher won't notice your paraphrasing. It means using your academic skills to make your work the best possible. There are many reasons to avoid plagiarism. In general, you attend college to learn how to form and convey your own beliefs, not to rehash the opinions of others, at least not with proper credit. At first, it may seem very difficult to develop your own opinions, and you will probably

find yourself comparing the writings of others as you try to understand and absorb their arguments. But you need to work on honing your voice. You are not necessarily expected to be an original thinker. Still, you are expected to be independent - to learn to evaluate the work of others critically, weigh various arguments, and draw your conclusions. Plagiarising students defy the spirit of academic scholarship and skip a crucial step in the learning process.

II. Literature Review

The beagle tool, created by Juan et al. [1], uses some collusion strategies to identify plagiarism. This programme analyses the pertinent text and finds plagiarism. The Internet has changed the lives of students and also their learning styles. Through this, the student deepens his understanding of learning. Through this, the student deepens his understanding of learning. Many methods are employed in detecting plagiarism. Plagiarism is usually done using the text mining method. They use the text mining method. Text mining is the process of extracting information and insights from textual content using techniques including text categorization, entity extraction (also known as entity identification), and sentiment analysis. Large-scale raw data analysis via text mining enables the discovery of pertinent insights. It can produce text analysis models that learn to extract certain information depending on prior training when combined with machine learning.

A plagiarism detection method that is automatic was described by Steve et al. [2]. This system develops a feature-based plagiarism detector by weighing the relative value of each feature in the available assessment using neural network techniques. assessment and create a feature-based plagiarism detector. This article focuses on only two different aspects: the copyand-paste type and the plagiarised type. The results were compared with the commercially available online software Article Checker. They used Semantic Text Similarity (STS) method. Semantic Text Similarity (STS) tries to compare two texts and decide if they are similar in meaning. This was a notoriously tricky problem due to the nuances of a natural language in which two texts can be similar even though they have no words in common. Semantic similarity or semantic text similarity is a natural language processing (NLP) task that evaluates the relationships between texts or documents using a defined metric. The applications of semantic similarity are diverse, such as information mining, text summarization, sentiment analysis, etc.

Nathaniel et al.[3] defines Plagiarism is a serious issue that violates the rights of copyrighted writings and resources. They proposed a new plagiarism detection method called SimPAD. This method aims to find similarities between two documents by comparing sentences. Tests show that SimPAD detects plagiarised documents more accurately than existing plagiarism detection methods. They use a machine learning algorithm. Machine learning refers to machine learning algorithms that learn from the results of other machine learning algorithms. Machine learning algorithms are typically related to group learning algorithms, such as stacking, which learn to combine predictions from group members. Machine learning is an emerging field in machine learning that explores approaches to learning better learning algorithms. The approaches aim to improve algorithms from various perspectives, including information efficiency and generalizability.

Jinan et al. [4] focused on the educational environment and faced similar challenges. They describe how cases of plagiarism are controlled. The simplest description of plagiarism is either a 'copy and paste' for a text, even if the source was cited or a change in some words to make meaning without a source, where determining what to mean is the most challenging and complex task. Plagiarism detection is a technique to detect the theft of research papers, literary works, source code, etc. This article provides software to detect plagiarism in Java student assignments. They are used for semantic plagiarism and tokenization.

All known renaming techniques used in plagiarism in computer programmes (such renaming variables and altering the type of loop structure) are rendered useless by tokenization [5]. Tokenization algorithms replace program code elements with individual characters. Any identifier, for instance, can be changed to an ID, as can any numerical number. Now, the string will replace the line =+; in a programme if it contains the line a = b + 45;. Since each line of the format "identifier = identifier + value;" is converted to the same tokenized sequence (the example stated above is taken from [6]), attempting to rename the variables will not be helpful.

Most existing plagiarism detectors are specifically designed to handle software source code or natural language texts. In the first case, the system usually treats the sent document collection as airtight and makes a pairwise comparison only between individual deliveries. Such efforts make use of cutting-edge methods to identify periodic changes in the code structure

(Tokenization [5], pmatching [9]), partial matches (RKS-GST [7], matching in the repository [8]), and partial matches. Systems designed to find similarities in natural language texts primarily search the Internet for possible matches. They generally do not use sophisticated benchmarking methods, mainly aimed at processing speed and comprehensive coverage. (For instance, the Turnitin [10] system's creators assert that they keep "a huge database of books and journals, as well as a database of the millions of papers already submitted").

"Hermetic" systems for plagiarism detection in natural language texts also exist, though they are little known. We can mention, e.g. CopyCatch Gold [11], YAP3 [12], and WCopyfind [13]. As a rule, detection software only finds an exact partial match: paraphrasing and formatting can hide signs of plagiarism. By only taking into account hapax legomena words (those that appear just once in the text) during the comparison, CopyCatch Gold lessens the impact of rewording, however this method is not very reliable.

III. PROPOSED SYSTEM

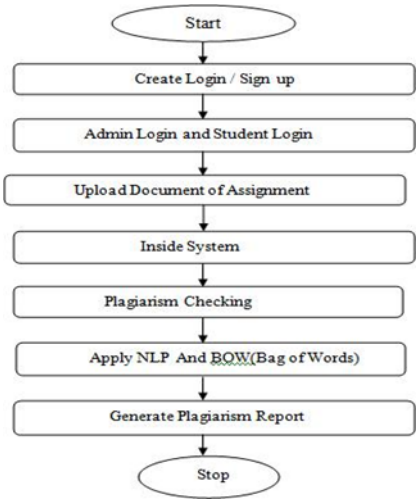

Fig.3.1 Architecture

This is the architecture of our project and it explains step-by-step execution and what techniques are used to find the similarity of the document and finally display the result.
There are three modules Create module, Student module, and Admin module.

3.1 Create a Login/Sign up module

In Create module the student has to create a login by giving the following details like Name, Course, Semester, Form no, Email id, and password. Once the student has created the login and their details are stored in the excel sheet.

3.2 Student Login module

In the student module, the student has to give a Name and Password if they don't give the correct name and password, they will not log in. Once the student logs in, it will ask to choose a file. The file should be in the document format accepted. Once the file is uploaded it will display the file is uploaded successfully on the screen.

3.3 Admin module

The admin module checks the similarity of the uploaded document. In this module, there is a name and password. Once the admin has given the correct name and password it will log and there is a start button to find the similarity between the document. It uses the bag of words technique and sequence Matcher ( ) inbuilt function to find document similarity and it will display as percentage and similarity of the document with each student and store it in an excel sheet.

IV. METHODOLOGY

4.1 Natural Language Processing (NLP)

A natural language processing technique for modelling text is a bag of words model. If you use any algorithm in NLP, it works with numbers. It cannot feed our text directly into this algorithm. Therefore, the text is preprocessed using the Bag of Words model, which turns it into a bag of words and counts the overall number of instances of the most commonly used terms. This model can be represented graphically by a table that lists the number of words that correlate to each individual word.

4.2 Bag of Words

Only a set of vectors containing a word's frequency in a document (reviews) is produced by Bag of Words. The TF-IDF model, in comparison, includes data on both essential and less important terms.

Bag of Words (BOW) encodings provide a straightforward representation of text data as they count the frequency of each word in a given document. This leads to a vectorized text representation, which is easy to interpret and analyze.

On the other hand, Term Frequency-Inverse Document Frequency (TF-IDF) encoding is a more sophisticated approach that considers the relevance of a given word in a corpus of documents. By multiplying the BOW counts by an inverse document frequency term, TFIDF encodings can effectively weigh the importance of each word in a document, leading to more accurate machine learning models. In summary, while Bag of Words vectors are easy to interpret, TF-IDF generally outperforms BOW in machine learning applications due to its ability to capture the importance of words in a larger context.

While both Bag-of-Words and TF-IDF were prevalent in this regard, there was a lack in understanding the context of words. It takes far more knowledge about the documents to translate them into another language or identify the similarities between the phrases "scary" and "spooky."

4.3 Applying the Bag of Words model

STEP 1:

First, preprocess the data in order to:

- Convert text to lowercase.
- Remove all non-word characters.

STEP 2:

Obtaining the most frequent words in our text. It will apply the following steps to generate our model.

- Mark the dictionary to hold our bag of words.
- Next, tokenize each sentence into words.
- Now, for each word in the sentence, check if the word exists in our dictionary.
- If it does, then increment its count by 1. If it doesn't, add it to our dictionary and set its count as 1.

STEP 3:

Building the Bag of Words model

- Here, constructs a vector to show whether each word in the sentence is common or not.
- If a word in a sentence is a frequent word, set it as 1; else, we set it as 0.
- In this Bag Of Words technique, SequenceMatcher( ) inbuilt functions use this inbuilt function to find similarities between documents.

4.4 Sequence Matcher ( )

The difflib Python library has a class called Sequence Matcher. The difflib module comes up with functions and classes for comparing strings. It can be used and classes for comparing strings. It can be used to compare files and can produce information about the differences between files in different formats. This class can be used to compare two input strings. In other words, this class is useful for looking for character-level similarities between two strings. The main idea of SequenceMatcher ( ) is to find the longest consecutive matching sequence (LCS) that does not contain "junk" elements. Spam is things the algorithm doesn't want to match, like blank lines in plain text files, lines in HTML files, etc...

It doesn't result in minimal changes, but usually produces hits that "look right" to humans. The Content similarity is
calculated using the formula:
Ratio = 2.0 * M/T
Where,
M= “matches”

T= “total number of elements” in both sets.

V. RESULT AND DISCUSSION

This work uses the Bag of Words Algorithm to introduce the "Plagiarism checker for online Assessment." Python GUI was mostly used in this work for UI interaction.

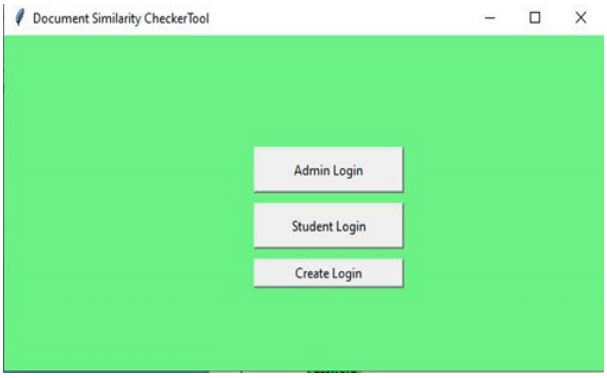

Fig.5.1 Front Page

This Page shown in Fig 5.1, displays the modules in the tools named Admin login, the student login, and Create Login. The Create login shown in Fig 5.2, is used to register the student details. Once the registration was successful the student can log in with their credentials shown in Fig 5.3.

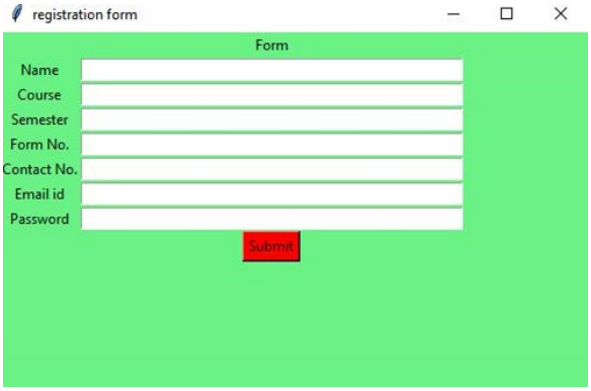

Fig.5.2 Registration Form

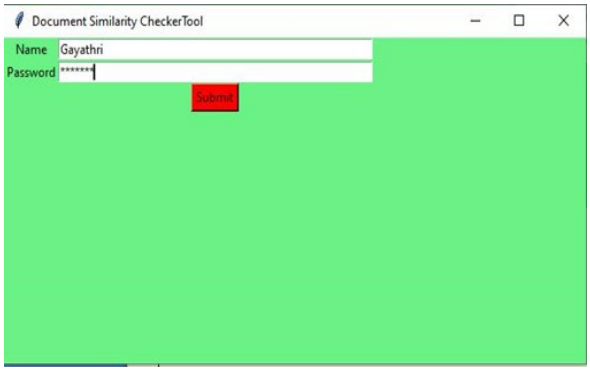

Fig.5.3 Student Login

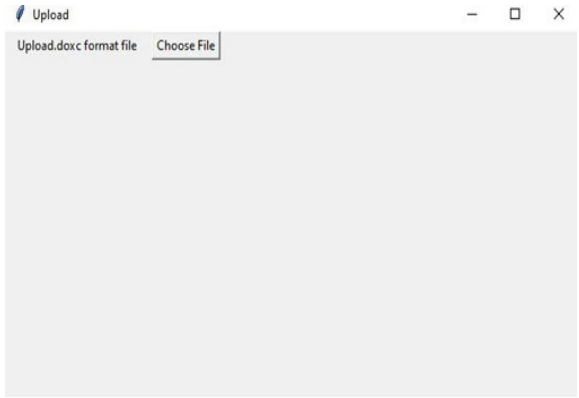

Fig.5.4 Upload File Page

After successful login, the document uploaded file page appears to the student, shown in fig 5.4.

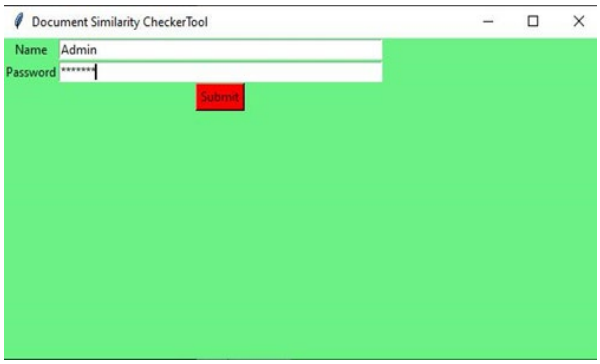

Fig.5.5 Admin Login

Here Admin can be represented as Faculty. Admin had own credential respectively. After the successful login of faculty, they can able to see the page like shown in fig 5.6. There is one start button. If the admin wants to check plagiarism between submitted student Assignments, they can press the start button to start the plagiarism-checking process.

Fig.5.6 Start Checking Plagiarism

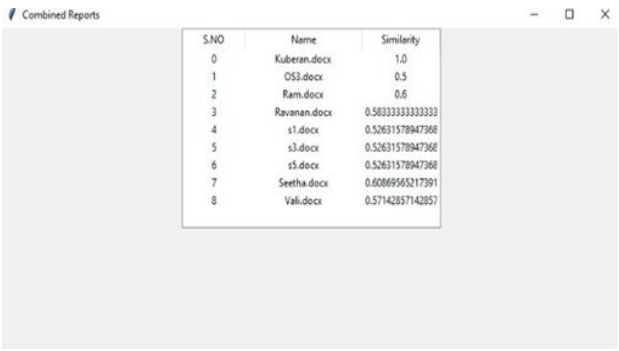

Fig.5.7 Report Generation

Once the plagiarism process was completed, thereport will be generated as shown in fig 5.7. It will show the plagiarism report for each student who submitted the assignment in this portal.

VI. CONCLUSION

This work has presented Plagiarism Checker for Online Assessment. As far as can see someone will only do the work given to them by the teachers and others will buy it from them and just submit the work. So, though everyone should have they are contributing to their work. This tool helps to make the students own responsibility for their work. Plagiarism is a ubiquitous problem faced by practitioners of academic fields like Assessments, projects, etc. Especially in academia, this poses a problem with a fair evaluation of the students and also inhibits the student's learning process. Although many efforts have focused on detecting textual plagiarism, significant progress has been made in detecting source code plagiarism. It can observe the leap from manual plagiarism checking to algorithm-based, automatic plagiarism checking made possible by advancements in technology. Recent approaches using Machine Learning and Deep Learning algorithms and techniques have shown some promising results to improve the accuracy and automation of the process of plagiarism detection.

REFERENCES

[1]. Mochol, Malgorzata; Oldakowski, Radoslaw; Heese, Ralf (2004) 'Ontology based Recruitment Process', FreieUniversität Berlin, and Humboldt-Universitätzu Berlin.
[2]. Microsoft ASP.NET Official Site.
[3]. Williams, Mickey (2002), 'Microsoft Visual C# .NET', Redmond, Wash.: Microsoft Corporation.
[4]. Martin, Tony.; Selly, Dominic (2002), 'Visual Basic.NET At Work: Building 10 Enterprise Projects', New York John Wiley & Sons, Inc.
[5]. Tatnall, Arthur (2005), 'Web Portals: The New Gateways.to Internet Information and Services', Hershey, PA: Idea Group Publishing.
[6]. Yahoo! HotJobs website.
[7]. Kanter, Rosabeth Moss (2000), 'Evolve! Succeeding in the Digital Culture of Tomorrow', Boston, Mass. Harvard Business School Press.
[8]. Das, Souripriya; Chong, Eugene Inseok; Eadon, George; Srinivasan, Jagannathan (2004), 'Supporting Ontology-based Semantic Matching in RDBMS'. Paper presented at Very Large DataBase conference, Oracle Corporation.
[9]. Pell, Arthur R., (2000), 'The Complete Idiot's Guide to Recruiting the Right Stuff', Indianapolis, Ind. Alpha Books.
[10]. Vinarski.H-Peretz, Binyamin.G, and Carmeli.A,(2011). "Subjective relational "Subjective relational innovative behaviors in the workplace." Journal of Vocational Behavior, vol. 78, no. 2, pp. 290-304.