iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XII **Month of publication:** December 2022

DOI: https://doi.org/10.22214/ijraset.2022.48080

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

AI Based Student's Assignments Plagiarism Detector

Ms. Minakshi Lanke[1], Ms. Aishwarya Porwal[2], Ms. Sejal Sarolkar[3], Mr. Gaurang Palaskar[4], Prof. Shital Kakad[5]
Marathwada Mitra Mandal College Of Engineering, Maharashtra, India

Abstract: ***The increase in plagiarizing academic contents led to efficient plagiarism detectors. In the conventional plagiarism detection tools, we have to put the contents from our paper or assignment and paste it into the input box for checking plagiarism or we have to load our files separately. Software under the domain artificial intelligence is moving towards the creation of a systems that can make tasks hassle-free and it also saves the time.***
"Smart Assignment Plagiarism Detector" can provide easy and efficient way of checking plagiarised data for teachers just in a single click by uploading a folder. This paper explains how our system is solving the problem of plagiarism through the use of algorithm in artificial intelligence.
Keywords: Artificial intelligence, Academic plagiarism, Text Similarity, Cosine similarity.

I. INTRODUCTION

The increase in plagiarising academic contents led to efficient plagiarism detectors. In the existing plagiarism detection tools, we have to put the contents from our paper or assignment and paste it into the input box for checking plagiarism or we have to load our files separately. Software under the domain artificial intelligence is moving towards the creation of a systems that can make tasks hassle-free and it also saves the time. Students copying assignments from one another is a problem that might not always get detected through conventional plagiarism scanners. An effective scanning system is required to cut the rise of taking short cuts by copying ideas from other students.

"Smart Assignment Plagiarism Detector" can provide easy and efficient way of checking plagiarised data for teachers just in a single click by uploading a folder. This paper explains how our system is solving the problem of plagiarism through the use of algorithm in artificial intelligence.

There are many string matching algorithms available one of which is Cosine similarity algorithm. This algorithm is also explained in the paper. Students copying assignments from one another is a problem that might not always get detected through conventional plagiarism scanners. To check the assignments manually for plagiarised data is not an easy task. To make it easy and time saving we are introducing "Smart Assignment Plagiarism Detector". In this, there is no need to upload each and every file separately. To check the plagiarism of the assignments, we just have to upload the folder where all the files are stored and the system will compare each file with another file in the folder and gives the final output. The system makes unique comparisons every time eliminating repeated comparisons.

II. LITERATURE SURVEY

Sr. no	Title	Publication	Year	Technology Used	Introduction
1.	Measuring Text Similarity Based on Structure and Word Embedding.	Science direct (Elsevier)	2020	structural similarity and word embedding	The proposed approach combines structural similarity, word-to-word similarity and word order similarity approaches
2.	A New Hybrid Technique for Detection of Plagiarism from Text Documents	Springer	2020	Extrinsic plagiarism detection · Semantic resemblance · Linguistic features, Plagiarism detection, Natural language processing.	The proposed system uses multiple linguistic features and matrix similarity measures for computing the resemblance.

3.	Exact String Matching Algorithms: Issues, and Future Research Directions.	IEEE	2019	String matching, Boyer-Moore, Rabin-Karp, exact string matching, pattern matching, pattern recognition	The work focuses only on software-based pattern string matching algorithms and their applications
4.	Text similarity analysis for evaluation of descriptive answers.	IEEE	2021	Computational Intelligence · MaLSTM	The paper proposes a text analysis based automated approach for automatic evaluation of the descriptive answers in an examination.
5.	Plagiarism Detection on Electronic Text based Assignments using Vector Space Model	IEEE	2014	Jaccard similarity, Cosine similarity	The paper focuses on an effective plagiarism detection tool to identify suitable plagiarism detection for text based assignments by comparing unigram, bigram, trigram of vector space model with cosine similarity measure.
6.	An improved plagiarism detection scheme based on semantic role labeling	Elsevier	2012	Character-based methods, Cluster-based methods, Syntax-based methods, Cross language-based methods.	The paper introduces a plagiarism detection technique based on the Semantic Role Labeling (SRL).
7..	A New Approach for Cross-Language Plagiarism Analysis	Springer	2010	J48 classification algorithm	The paper presents a new method for Cross-Language Plagiarism Analysis for detecting the plagiarized passages in the suspicious documents and their corresponding fragments in the source documents

In [1] the author proposed the approach which combines different similarity measures in the calculation of sentence similarity. In addition to traditional word-to-word similarity measure, the proposed approach exploits sentence semantic structure. This proposed method outperforms the current approaches on a standard benchmark dataset achieving 0.8813 Pearson correlation with human similarity.

In [2] the author represented a survey on single pattern exact string matching algorithms. The author proposed this to understand new classification, identify new directions and highlight the possible challenges, current trends, and future works in the area of string matching algorithms with a core focus on exact string matching algorithms.

In [3] the author introduced an extrinsic plagiarism detection approach inspired by cognition because it utilizes semantic knowledge to detect the plagiarized part from the text without human involvement. The author used the Dice measure as similarity measure for finding the semantic resemblance between the pair of sentences. This article is capable of identifying cases like restructuring, paraphrasing, verbatim copy, and synonymized plagiarism. The proposed system was having innovative approach, but the results were somehow close and reasonably better than the existing systems.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

In [4] the author proposes the text analysis based approach for evaluation of the descriptive answers in an examination. This research focuses on the concepts of Natural Language Processing and data mining .In this paper the text similarity model is based on the MaLSTM.
In [5] the paper focuses on an effective plagiarism detection tool to identify suitable plagiarism detection for text based assignments by comparing unigram, bigram, trigram of vector space model with cosine similarity measure
In [6] the author proposed an automatic plagiarism detection system for obfuscated text based on a support vector machine classifier that exploits a set of lexical, syntactic and semantic feature.
In [7] the author introduced Weighting for each argument generated by Semantic Role Labelling (SRL) to study its behaviour. The experimental results on PAN-PC-09 data sets showed that the proposed method significantly outperforms the modern methods for plagiarism detection in terms of Recall, Precision and F-measure.
In [16] Semantic web technology seems to be in the infant stage as only little efforts have been taken on ontology construction with cross-domain application. The core concern of this work is on two decision-making processes namely data filtering and data annotation. Certain process is followed in this work: (i) Pre-processing (ii) Proposed Jaccard Similarity Evaluation (iii) Data filtering and Outlier Detection (iv) Semantic annotation and clustering.
In [17] Industry and Institute both are equally responsible to develop quality students. In this paper, cross domain (Industry domain and Institute domain) ontology based semantic models are developed to bridge the institute-industry gap using Protégé 5.5.0 editor. The classes and sub classes of Industry Institute ontology are designed with the help of domain experts.

III. METHODOLOGY

A. System Architecture

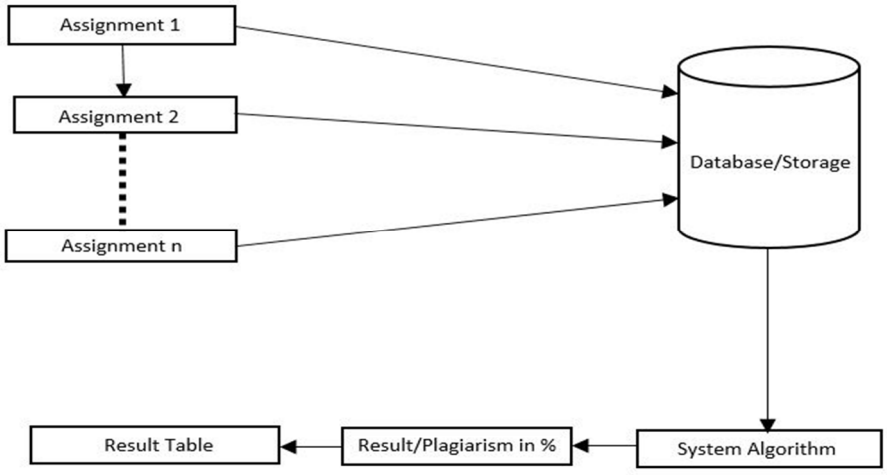


Fig. 1: Proposed System Architecture

In the given system architecture (Fig. 1), students can upload the n number of assignments in the folder which will work as a database/ storage. When teacher wants to check for the plagiarism, he/she just have to upload a folder in the system by clicking on the upload button. System will check plagiarism for every file in that folder by uniquely comparing them with each other. The system will eliminate duplicate comparisons from the result table. After comparing every file, it will generate the result table with compared file names, plagiarism percentage between them and the remark.
We have set the limits for giving the remark as highly plagiarised, medium plagiarised, low plagiarised by keeping in mind the grading system of Savitribai Phule Pune University.

B. Algorithm

1) Cosine Similarity

$$\text{Cosine Similarity} = \cos\theta$$

Here,

Θ = the angle between two projected vectors.

The value of Cos θ ranges from -1 to 1 and hence the value of cosine similarity also ranges from -1 to 1.

To calculate the plagiarism in percentage we just have to multiply the result obtained from cosine similarity algorithm by 100.

$$\text{Similarity}(M, N) = \frac{M.\ N}{||M|| * ||N||}$$

Here,

M = Vector M

N = Vector N

||M|| = square root of square of vector M

||N|| = square root of square of vector N

We can calculate similarity between vector M and vector N by using the above formula.

2) System Algorithm

```
import os
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from pathlib import Path

vectorize = lambda Text: TfidfVectorizer().fit_transform(Text).toarray()
similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])



def check_plagiarism(folder):

    student_files = [folder+"\\"+doc for doc in os.listdir(folder) if doc.endswith('.txt')]

    student_notes = []
    for File in student_files:
        student_notes.append(open(File).read())

    vectors = vectorize(student_notes)
    s_vectors = list(zip(student_files, vectors))

    plagiarism_results = set()
    for student_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((student_a, text_vector_a))
        del new_vectors[current_index]
        for student_b, text_vector_b in new_vectors:
            sim_score = similarity(text_vector_a, text_vector_b)[0][1]
            student_pair = sorted((student_a, student_b))
            score = (Path(student_pair[0]).name, Path(student_pair[1]).name, float(str(sim_score*100)[:5]))
            plagiarism_results.add(score)
    return plagiarism_results
```

C. Description

1) First of all we have declared two variables viz. vectorize and similarity. These variables vectorize and similarity are defined as

Vectorize = lambda Text: TfidfVectorizer ().fit_transform (Text).toarray ()

Similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2]

2) Now we have defined check plagiarism method and passed folder as input where all the assignments are stored.

3) All the assignments with ".txt" extension in the specified folder are stored in the variable named student files.

4) We have defined one list name student notes.

5) The function given below reads the text files and stores it into **student notes = []**

student_notes.append (open (File).read ())

6) The file gets converted into vectors using vectorize () function and stores it into variable named vectors.

7) The function **vectorize ()** assigns a number to each word and creates a 1-D array of it.

8) We are creating a list by zipping the **student files** and **vectors** and assigning it to **s_vectors** variable.

9) The function **set ()** used is avoiding the duplicate comparisons.

10) The index of zipped file gets passed to a **current vectors** variable.

11) Using for loop, we are calculating the similarity score between all the stored files one by one.

12) Now we are passing the name of file 1, name of file 2 and percentage of similarity score to the **score** variable for displaying it in result.html page.

13) Finally we are returning the final result to app.py file.

IV. RESULT AND ANALYSIS

A. Result

After comparing files in the folder, we will get the plagiarism in percentage. There are total 4 columns in the result table namely:

1) File 1

2) File 2

3) Plagiarism in % and

4) Remark.

Remark is given on the basis of grading system of Savitribai Phule Pune University. Remark includes highly plagiarised, medium plagiarised and low plagiarised.

On the basis of remark obtained, rows in the table gets highlighted in the colour given below:

- For highly plagiarised.
- For medium plagiarised.
- For low plagiarised.

B. Analysis

After comparing Cosine similarity with many other similarity measures, we understood that Cosine similarity is better than other similarity measures.

1) Why Cosine similarity is better than Euclidean Distance?

Ans: Cosine is better than Euclidean because there is possibility that two documents can measure far apart by Euclidean distance due to size of the document but it may have small angle between them which gives accurate similarity.

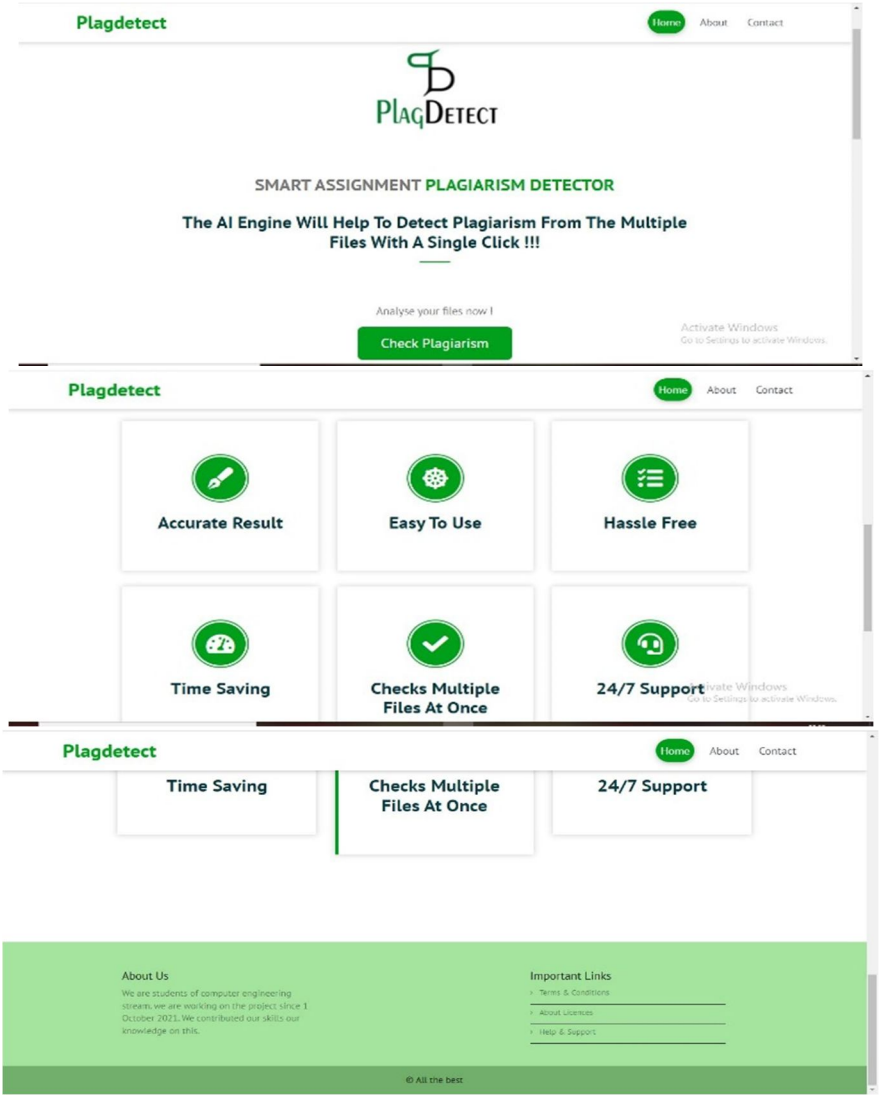
2) Why Cosine similarity is better than Jaccard?

Ans: Jaccard similarity is not good for cases where duplication do matter, cosine similarity is good for cases where duplication matters while analysing text similarity. After researching more about the plagiarism tools, we found that there is no tool available who checks for multiple files at a time. There are tools available who checks 2 files at a time with each other. But our system checks more than 2 files simultaneously. We also analysed the time for comparing 4 files using existing plagiarism tool and using "Smart Assignment Plagiarism Detector" and we understood that time taken by existing system to compare 4 files is more than the time taken by our system for the same number of files.

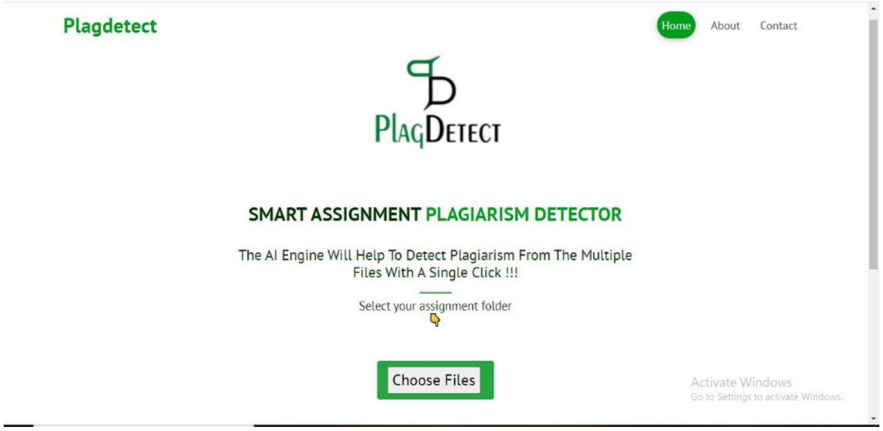
V. SYSTEM SCREENSHOTS

Here screenshots of the system are attached.

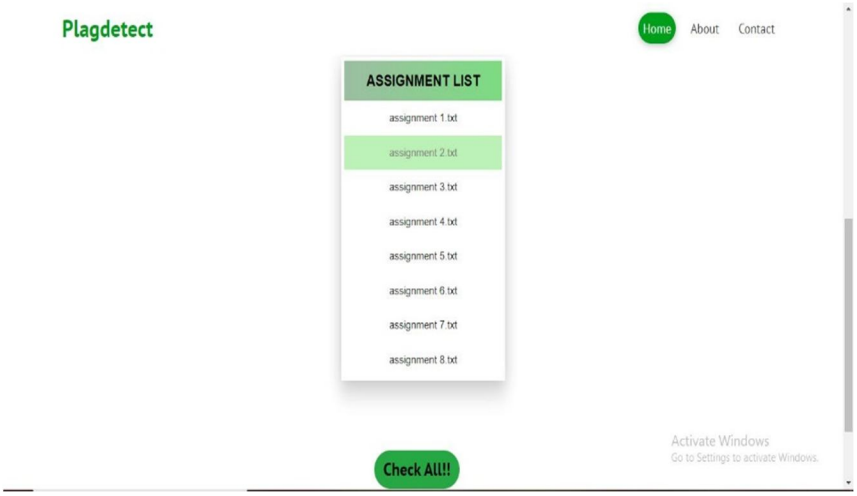
1) *Front Page:* This is the front page. The page contains Nav bar, Footer, one button which will navigate us to the 2nd page of the system and we have also included some features of the system.



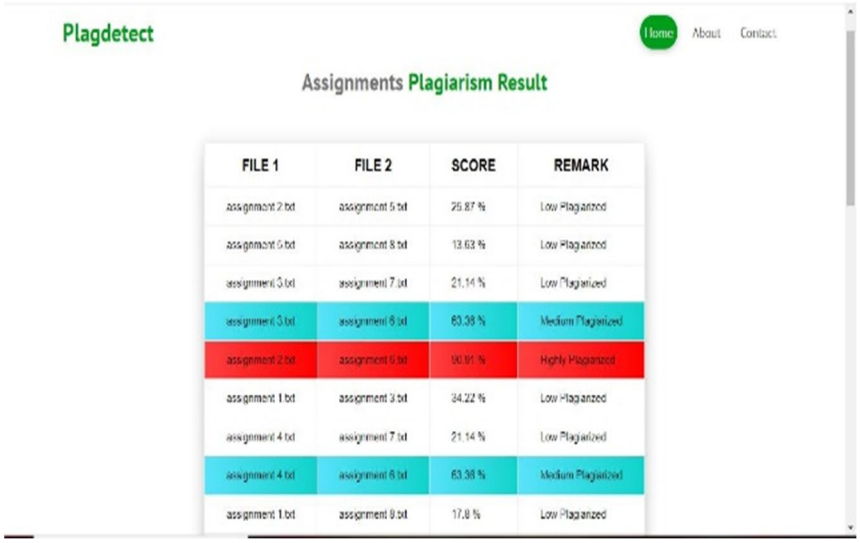
2) *Home Page 1:* This is the home page. It includes Nav bar and one button to choose the folder.



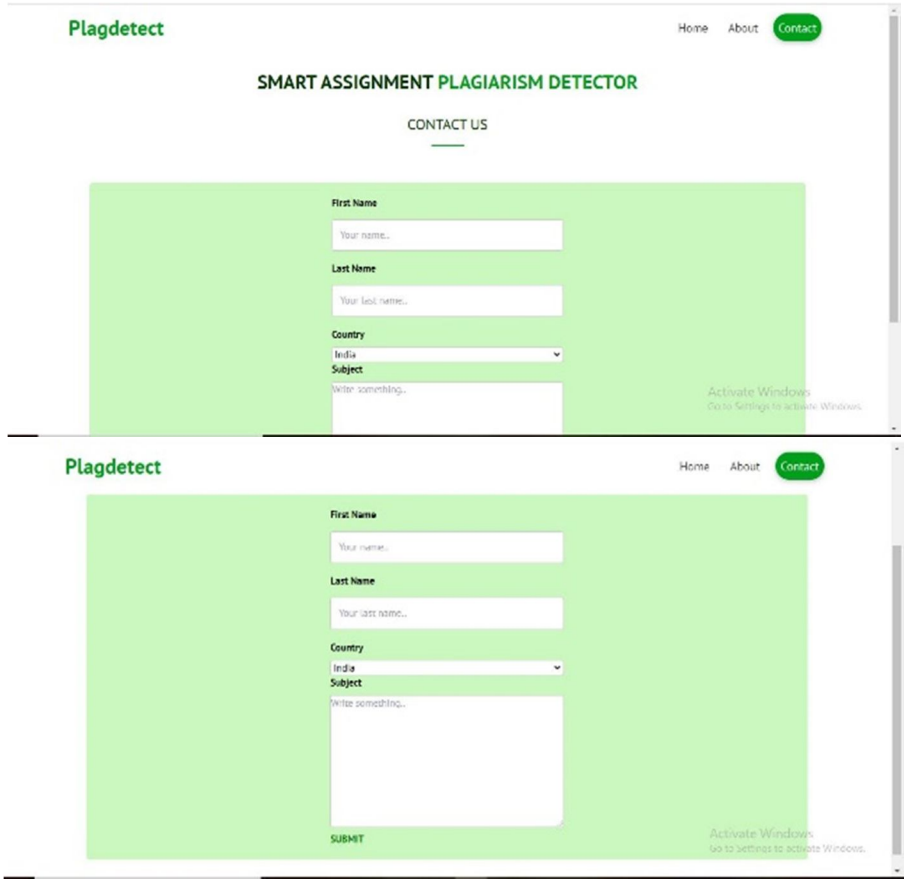
3) *Home Page 2:* This page displays the list of assignments stored in the specific folder.



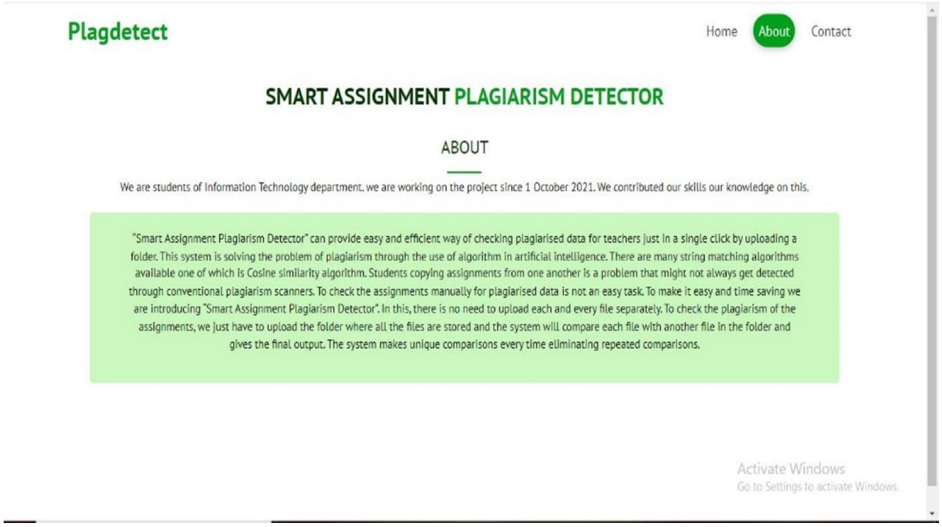
4) *Result Page:* This page shows the result table with 4 columns namely File 1, File 2, Plagiarism in % and Remark.



5) Contact us Page: After hitting the contact we button on Nav bar it will redirect to a feedback form.



6) About Page: The page describes a brief introduction of our system



VI. CONCLUSION

This paper explains the approach to solve the academic plagiarism problem. The algorithm used in the system is comparing multiple files for the calculation of plagiarised data with a single folder upload. Also, it is giving the unique comparisons by eliminating the duplicate ones. Here, we have used Cosine similarity measure to calculate similarity as, cosine similarity measure shows slightly higher results than Jaccard similarity measure and therefore cosine similarity measure is more preferable than the other approach [5].

VII. FUTURE WORK

The future work for this research is to improve the UI, focus on adding the OCR technology in the system to find out the plagiarism in pictures, one can modify the system to recognize the handwritten assignments also.

The future work for this research is to detect the plagiarism in flowcharts and videos. Along with that to develop it from business point of view into a possible product for use.

REFERENCES

[1] Farouk, Mamdouh (2020). Measuring text similarity based on structure and word embedding. Cognitive Systems Research, 63(), 1–10. doi:10.1016/j.cogsys.2020.04.002

[2] Ahuja, Lovepreet; Gupta, Vishal; Kumar, Rohit (2020). A New Hybrid Technique for Detection of Plagiarism from Text Documents. Arabian Journal for Science and Engineering, (), –. Doi: 10.1007/s13369-020-04565-9

[3] S. I. Hakak, A. Kamsin, P. Shivakumara, G. A. Gilkar, W. Z. Khan and M. Imran, "Exact String Matching Algorithms: Survey, Issues, and Future Research Directions," in IEEE Access, vol. 7, pp. 69614-69637, 2019, Doi: 10.1109/ACCESS.2019.2914071.

[4] arXiv: 2105.02935 [cs.LG] (or arXiv: 2105.02935v1 [cs.LG] for this version)https://doi.org/10.48550/arXiv.2105.02935

[5] (). . , (), –. doi:10.1109/iciafs.2014.7069593

[6] Ahmed Hamza Osman; Naomie Salim; Mohammed Salem Binwahlan; Rihab Alteeb; Albaraa Abuobieda (2012). An improved plagiarism detection scheme based on semantic role labeling. , 12(5), 1493–1502. doi:10.1016/j.asoc.2011.12.021

[7] Ahmed Hamza Osman; Naomie Salim; Mohammed Salem Binwahlan; Rihab Alteeb; Albaraa Abuobieda (2012). An improved plagiarism detection scheme based on semantic role labeling. , 12(5), 1493–1502. doi:10.1016/j.asoc.2011.12.021

[8] 1877-0509 © 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/) Peer-review under responsibility of KES International. 10.1016/j.procs.2019.09.303

[9] Jiffriya, M. A. C., et al. "Accelerating text-based plagiarism detection using gpus." 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS). IEEE, 2015.

[10] D. Soyusiawaty and Y. Zakaria, "Book Data Content Similarity Detector With Cosine Similarity (Case study on digilib.uad.ac.id)," 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2018, pp. 1-6, doi: 10.1109/TSSA.2018.8708758.

[11] Osman, Ahmed Hamza, et al. "An improved plagiarism detection scheme based on semantic role labeling." Applied Soft Computing 12.5 (2012): 1493-1502.

[12] Liu, Ming, et al. "Measuring similarity of academic articles with semantic profile and joint word embedding." Tsinghua Science and Technology 22.6 (2017): 619-632.

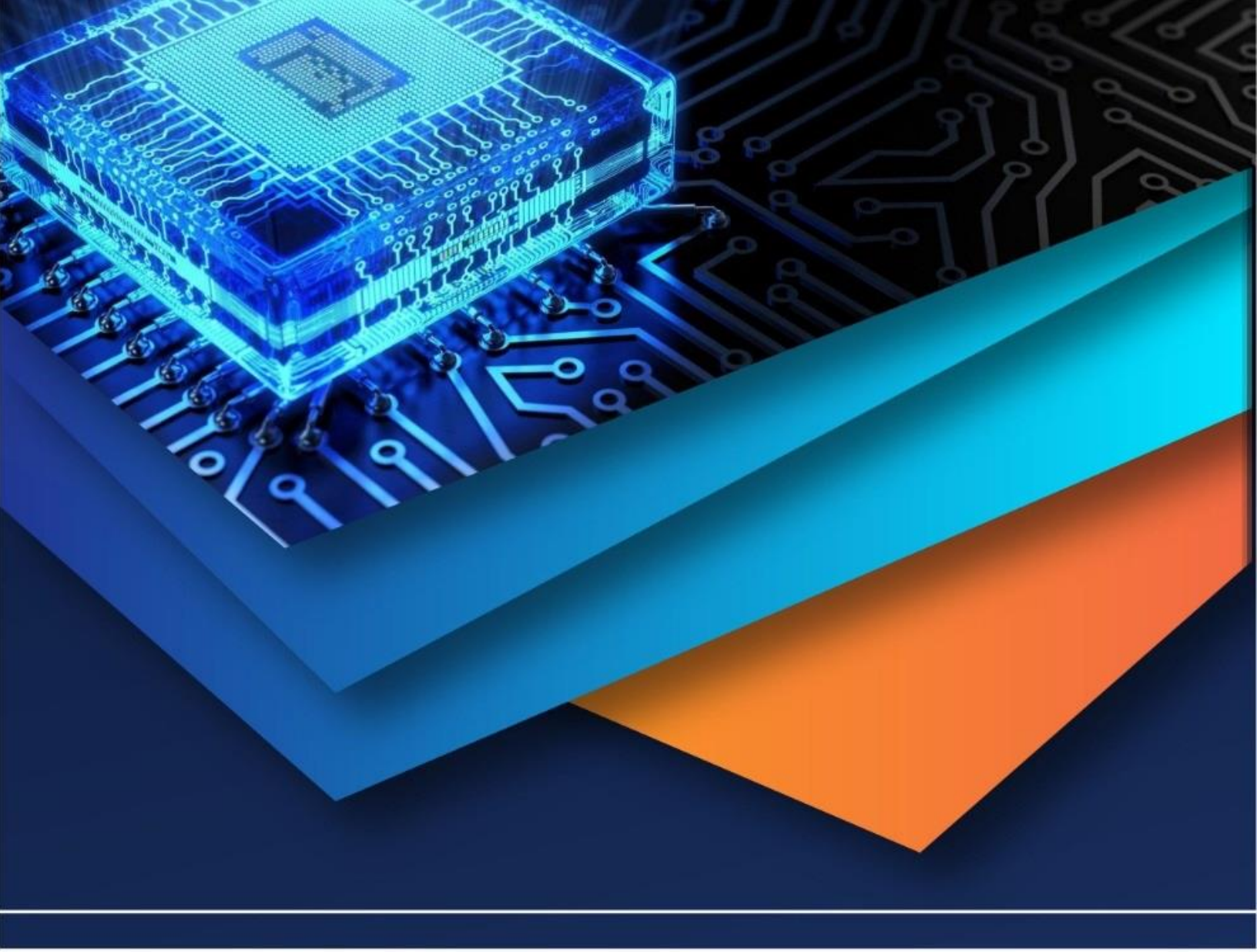
[13] Bahel, Vedant, and Achamma Thomas. "Text similarity analysis for evaluation of descriptive answers." arXiv preprint arXiv:2105.02935 (2021).

[14] S. Zhu, J. Wu and G. Xia, "TOP-K cosine similarity interesting pairs search," 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, 2010, pp. 1479-1483, doi: 10.1109/FSKD.2010.5569212.

[15] Shajalal, Md, and Masaki Aono. "Semantic textual similarity between sentences using bilingual word semantics." Progress in Artificial Intelligence 8.2 (2019): 263-272.

[16] Shital Kakad, Sudhir Dhage,"Cross domain-based ontology construction via Jaccard Semantic Similarity with hybrid optimization model," Elsevier ,Expert Systems with Applications,Volume 178,2021,115046,ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.115046.

[17] Shital Kakad, Sudhir Dhage," Knowledge Graph and Semantic Web Model for Cross Domain", Journal of Theoretical and Applied Information Technology 100 (16)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY