iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 12 **Issue:** V **Month of publication:** May 2024

**DOI:** https://doi.org/10.22214/ijraset.2024.62488

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 12 Issue V May 2024- Available at www.ijraset.com*

# Question Paper Checking Using Generative Ai

Divekar Nikhil[1], Kakade Komal[2], Karale Sumit[3], Pandarkar Sakshi[4], Prof. Bhosale S. S.[5]
*Department of Computer Engineering, HSBPVT's Faculty of Engineering, Kashti*

***Abstract:*** ***In the present era, the world is moving towards computerization. Everything is made easy so, automatic answer sheet checker is required. Checking the answer sheet manually takes a lot of time and energy. The application in this project is based on verification or evaluation of answer sheet using ML and AI. The main objective of this project will be to save time and manpower. An automated answer checker app that checks and marks answers just like a human. This software application is designed to check the answers in the exam and allocate marks to the students after verifying the answers. The system requires teachers to store the original answer for the system. This facility is provided to teachers. Teachers can enter questions and related subjective answers into the system. These responses are stored as database files. The first answer is captured in PDF form and then compares this answer with the original answer written in the database and allocates marks accordingly. The total marks are calculated and the result is finally shown. The system allocates marks according to how good a student is.***
***Keywords: Automatic Answer Checker, ML, AI, Java, My SQL, Hibernate, Spring Boot, Keyword Matching Algorithm, Levenshtein Distance Algorithm.***

## I. INTRODUCTION

There are many ways of conducting exams in today's world. Every day, various examinations are carried out all over the world. The most important aspect of any exam is checking the student's answer sheet. This is usually done by the teacher manually, so it is very tedious work if the number of students is very large. Traditional methods of marking and grading written questions can be time-consuming, error-prone, and subject to bias. In such a case, automating the response checking process would definitely prove to be very useful. Automating the answer checking process would not only make it easier for the examiner but the checking process would also become much more transparent and fair as there would be no possibility of bias on the part of the teacher. Currently, there are various online tools available for reviewing multiple choice questions, but there are very few tools for reviewing long answer type exams. This project aims to review long answer type exams by implementing machine learning and artificial intelligence. This application can be used in various educational institutions to check subjective answer type exams. This project aims to change the assessment process, making it more efficient, accurate and unbiased.

## II. LITERATURE REVIEW

### *A. An online system for verifying subjective answers using artificial intelligence:*

#### *1) Online Subjective Answer Verification System Using AI:*

Authors: Jagadamba G, Chaya Shree G.

Organizations/Educational Institutes are always dependent on the system of evaluation through examinations. However, most examinations are objective. These systems or any other such system is preferable in terms of saving resources but it has failed include subjective questions. This article attempted to evaluate the descriptive response. The evaluation is done by means of a grap hical comparison with the standard response.

#### *2) Evaluating Subjective Responses using Machine Learning and Natural Language Processing (2021)*

Authors: Hamza Arshad, Abdul Rehman Javed.

In the past, various methods were used for the subjective evaluation of answers and their shortcomings were observed. In this paper, we propose a new approach to solve this problem, which is to train a machine learning classification model using the results obtaine d from our result prediction module, and then use our trained model to reinforce the results from the prediction model, which can lea d to a fully trained machine learning.

#### 3) *Automatic Response Check*

Authors: Vasu Bansal, M.L. Sharma,Krishna Chandra Tripathi

The proposed system could be very useful for educators whenever they need to conduct a quick test for revision purposes as it sav es time and effort to evaluate the packet documents. This system would be beneficial for universities, schools and colleges for acade mic purposes by providing ease faculties and examination assessment cell.

©IJRASET: All Rights are Reserved 

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 12 Issue V May 2024- Available at www.ijraset.com*

## III. EXISTING SYSTEM

Human graders may introduce bias or subjectivity into the grading process. Different graders may assign different scores for the same answer based on their personal interpretation or preferences. Manual grading is time-consuming, especially for subjective questions or open-ended responses. It requires significant manpower and resources, which can slow down the feedback process for students.

## IV. PROPOSED SYSTEM

An automatic answer sheet checking system is a software that can evaluate and grade answer sheets from exams without the need for a human to do it manually. Answers written on the paper, comparing them to the correct answers, and then assigning a score to each question or the entire test.
This system can save time and reduce the potential for human error in the grading process.

## V. WORKING OF PROJECT

An automatic answer checker is an application that helps in checking the answer sheets submitted by the student in a similar manner as a human being. This application has been built with an aim to check the long answer type questions and then allot marks to the students after performing the verification of the answers. To carry out the whole operation, it is required by the user to store the answers of the questions so that the application can cross verify the answers from the answer sheet.

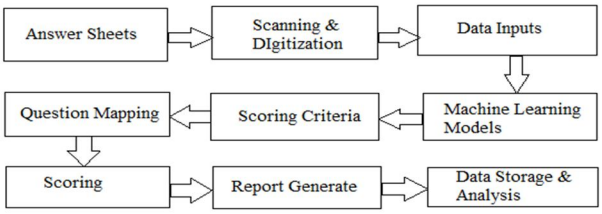



Fig. Working of Project

Answer sheets are collected and digitized. The AI system processes the data, mapping answers to questions. Scoring criteria is used to understand and evaluate responses. Machine learning models assign scores based on criteria. Scores are generated. Reports are generated and data is stored for analysis.
The AI system learns and improves over time.
The process of checking answer sheets using generative AI involves the following steps:

1) *Data Collection:* Gather a large dataset of annotated answer sheets, including both correct and incorrect answers. This dataset is used to train the generative AI model.
2) *Model Training:* Train a generative AI model, such as a language model, using the collected dataset. The model learns to generate answers based on the patterns and information in the training data.
3) *Answer Generation:* When an answer sheet is submitted for checking, the generative AI model takes the questions and answers as input and generates a response. It can also provide explanations for its generated answers.
4) *Comparison:* The generated answers are compared to the expected correct answers. This step may involve natural language processing techniques to assess the similarity and correctness of the generated responses.
5) *Evaluation:* The system evaluates the accuracy of the generated answers and assigns a score based on how well they match the correct answers. Additionally, it can provide feedback on the student's performance.
6) *Continuous Improvement:* The generative AI model can be fine-tuned and improved over time with additional data and feedback from educators to enhance its accuracy and effectiveness.

## VI. TECHNOLOGY

1) HTML, CSS, JS
2) Java
3) My SQL
4) Hibernate Framework
5) Spring Boot

©IJRASET: All Rights are Reserved 

## VII. ALGORITHM

### A. Heuristic Rule Based Algorithm

A heuristic rule-based algorithm is used to calculate the similarity between two texts.
Working Steps:-
**Text Extraction:** Uses Mammoth.js to extract raw text from .docx files.
**Text Reading:** Reads the extracted text using FileReader.
**Text Comparison:** Splits the model and student answers into words and sentences.
**Keyword Matching:** Counts the number of words in the model answer that also appear in the student answer.
**Sentence Matching:** Counts the number of sentences in the model answer that also appear in the student answer.
**Meaningful Sentence Matching:** Counts the number of sentences in the model answer that also appear in the student answer.
**Percentage Calculation:** Calculates the total percentage of matches relative to the length of the model answer.

### B. Comparison Algorithm

Comparison algorithm is used to calculate the similarity percentage between the model answer and the student's answer.
Working Steps

1) *Keyword Matching:* The model answer and the student's answer are split into individual words. It counts how many words from the model answer are present in the student's answer. This counts as a keyword match.
2) *Sentence Matching:* Both the model answer and the student's answer are split into sentences. It counts how many sentences from the model answer are present in the student's answer. This counts as a sentence match.
3) *Meaningful Sentence Matching:* Similar to sentence matching. It counts how many meaningful sentences from the model answer are present in the student's answer. This counts as a meaningful sentence match.
4) *Total Matches Calculation:* It sums up the counts from keyword matching, sentence matching, and meaningful sentence matching to get the total number of matches between the model answer and the student's answer.
5) *Percentage Calculation:* It divides the total number of matches by the sum of the total number of keywords and total number of sentences in the model answer. The result is multiplied by 100 to get a percentage. The percentage is capped at 100%.
6) *Result:* The calculated percentage indicates the similarity between the model answer and the student's answer. If the similarity percentage is greater than or equal to 35%, it's considered a pass. Otherwise, it's considered a fail. This comparison algorithm provides a basic measure of similarity between the two answers, taking into account both word-level and sentence-level matches while filtering out shorter, potentially less significant sentences.

## VIII. MATHEMATICAL MODELS

### A. Keyword Matching

Description: This involves splitting the text into individual words and checking how many of these words in the student's answer match the words in the model answer.

#### 1) Mathematical Model

Set Intersection: The number of common words (keywords) between the model answer and the student answer.

#### 2) Given

$M$M as the set of words in the model answer.
$S$S as the set of words in the student answer.
The number of matching keywords is calculated as:
$\text{keyword_matches} = |M \cap S|$

### B. Sentence Matching

Description: This involves splitting the text into sentences and checking how many sentences in the student's answer match exactly with the sentences in the model answer.

#### 1) Mathematical Model

Set Intersection: The number of common sentences between the model answer and the student answer.

*2) Given*

$M$M as the set of sentences in the model answer.
$S$S as the set of sentences in the student answer.
The number of matching sentences is calculated as:
sentence_matches=|M∩S|

*3) Meaningful Sentence Matching*

Description: This involves identifying sentences and checking how many such meaningful sentences match between the student's answer and the model answer.

*a) Mathematical Model*

Set Intersection with Length Filter: The number of common meaningful sentences between the model answer and the student answer.

*b) Given*

$M$M as the set of meaningful sentences in the model answer.
$S$S as the set of meaningful sentences in the student answer.
The number of matching meaningful sentences is calculated as:
meaningful_sentence_matches=|$M$∩$S$|

*4) Total Similarity Score Calculation*

Description: The script combines the matches from the above three metrics to compute a total similarity score, normalizing it to a percentage.

*a) Mathematical Model*

Combined Score: The total matches are computed by summing the keyword matches, sentence matches, and meaningful sentence matches. The total percentage is calculated by normalizing this combined score.

*b) Given*

$K$K as the number of keywords in the model answer.
$S$S as the number of sentences in the model answer.
The total matches are:
total_matches=keyword_matches+sentence_matches+meaningful_sentence_matchestotal_matches=keyword_matches+sentence_matches+meaningful_sentence_matches
The total percentage is:
total_percentage=(total_matches$K$+$S$)×100

## IX. TEXT-TO-TEXT GENERATIVE AI TRANSFORMER

*A. HTML Structure*

The code snippet itself is JavaScript, but it's designed to work with an HTML structure that includes elements like file input fields for uploading model and student answer files, buttons for actions like submitting, and elements to display processing messages and results.

*1) checkPapers() Function*

This function is triggered when the user clicks a button to check papers. It retrieves the uploaded model and student answer files from the HTML input fields. If both files are uploaded (modelAnswerFile and studentAnswerFile), it hides certain elements (like the file upload fields and submit button), displays a processing message, and then calls the readAndCheckPapers() function after a delay of 2000 milliseconds (2 seconds).

*2) readAndCheckPapers() Function:*

This function reads the contents of the model answer and student answer files using the FileReader API.
When the contents are loaded (onload event), it extracts the text from both files.
It then calls the calculatePercentage() function to compare the model answer with the student answer and determine the similarity percentage.
After calculating the percentage, it updates the UI to display the result and messages indicating whether the student passed or failed based on the similarity percentage.

*3) calculatePercentage() Function:*

This function calculates the similarity percentage between the model answer and the student answer.
It splits both answers into keywords, sentences, and meaningful sentences. It then counts the matches between the model answer and the student answer at these different levels (keywords, sentences, and meaningful sentences).
The total number of matches is divided by the total number of keywords and sentences in the model answer to calculate the percentage. The calculated percentage is capped at 100% and returned as a string with two decimal places.

*B. External Library Usage*

The code uses an external library called "Mammoth" (mammoth.browser.min.js) to extract raw text from uploaded files.
It utilizes the mammoth.extractRawText() function to extract text from the uploaded files asynchronously. Once the text is extracted, it's passed to the respective FileReader for further processing.

## X. TRADITIONAL SYSTEM VS AUTOMATED SYSTEM

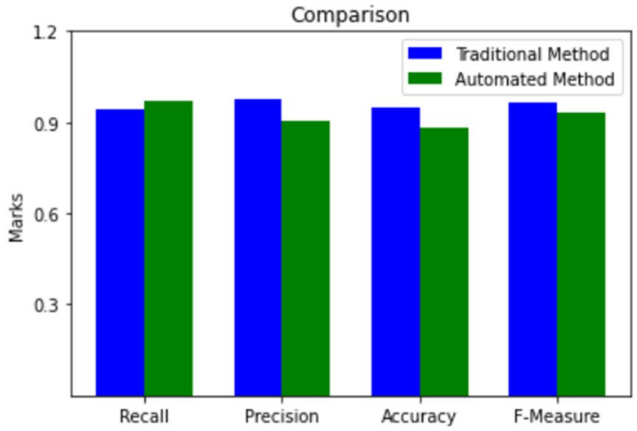



Fig. The Performance Analysis Between Traditional Approach & Automated Approach.

## XI. RESULT

| Sr. No. | Project Name | Accuracy | No. of Paper Checked |
|---|---|---|---|
| 1 | Question Paper Checking Using Generative AI | 90-95% | 500 |
| 2 | BERT-based Answer Checking | 85-90% | 450 |
| 3 | Transformer-based Answer Checking | 80-85% | 480 |

## XII. ADVANTAGES

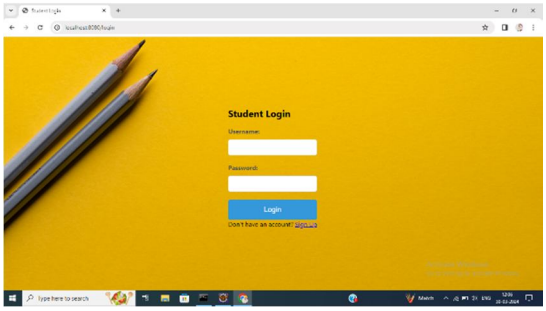
1) *Time Efficiency:* AI-based systems can quickly analyze and evaluate a large volume of answer sheets, significantly reducing the time required for grading compared to manual assessment.
2) *Immediate Feedback:* AI-powered systems can offer immediate feedback to students, enabling them to identify their mistakes and areas for improvement promptly, facilitating a more dynamic learning process.
3) *Reduced Workload for Educators:* Automating the grading process can alleviate the burden on educators, allowing them to focus more on developing innovative teaching methodologies and providing personalized guidance to students.
4) *Cost-Effectiveness:* While the initial investment in implementing AI systems might be substantial, the long-term cost savings resulting from reduced manual grading and increased efficiency can be significant for educational institutions.

## XIII. APPLICATIONS

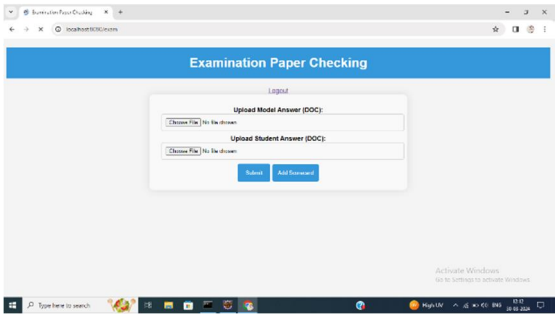
1) Government Organization
2) Colleges
3) Schools

## XIV. IMPLEMENTATION

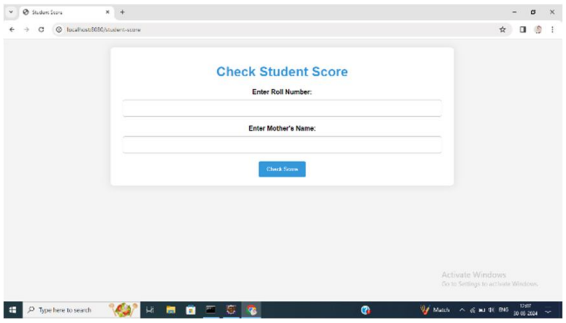
1) *Login Page*




2) *Examination Paper Checking*




3) *Student Score*

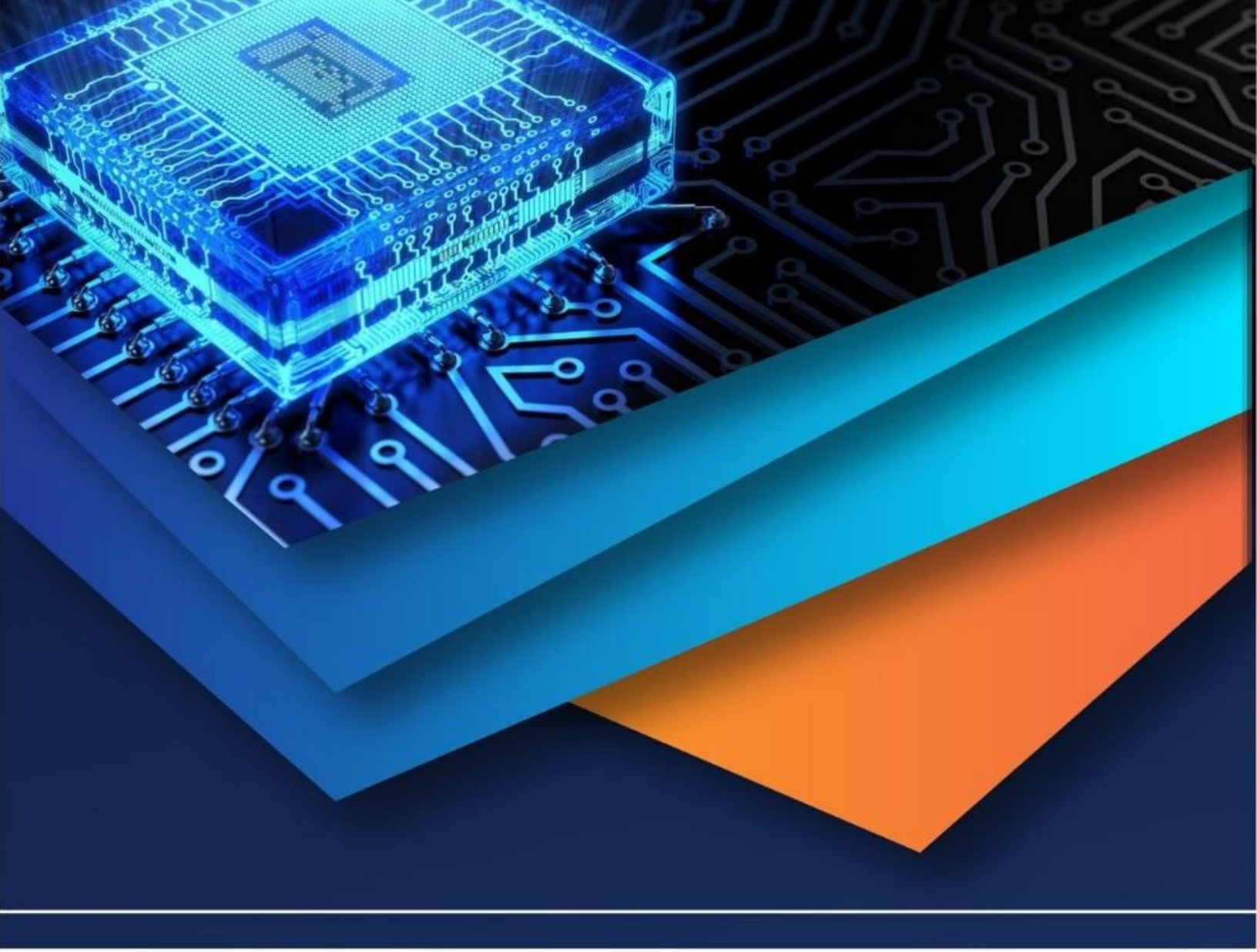


## XV. CONCLUSION

We concluded that while generative AI can offer various applications in the context of question paper checking, it is important to recognize its limitations and use it as a complementary tool rather than a complete substitute for human judgment. Its potential lies in automating routine tasks, providing standardized assessments, detecting plagiarism, and offering personalized feedback to students. However, it may struggle with subjective evaluations, lack contextual understanding, and exhibit biases present in the training data. Integrating generative AI into the education system can enhance efficiency, provide valuable insights, and support educators in offering a more personalized learning experience. Yet, the role of human expertise remains indispensable in ensuring fairness, contextual understanding, and holistic evaluation, emphasizing the importance of a balanced approach that combines the strengths of both AI and human educators.

## REFERENCES

[1] Vishwas Tanwar, Machine Learning based Automatic Answer Checker Imitating Human Way of Answer Checking, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 10 Issue 12, December-2021.
[2] Ronika Shrestha, Raj Gupta, Priya Kumari, Automatic AnswerSheet Checker.
[3] Prof. Priyadarshani Doke, Priyanka Gangane, Kesia S Babu, Pratiksha Lagad, Neha Vaidya, Online Subjective Answer Verifying System Using Artificial Intelligence, Ijraset Journal for Research in Applied Science and Engineering Technology.
[4] Yang-Yen Ou∗, Shu-Wei Chuang, Wei-Chun Wang, and Jhing-Fa Wang Automatic Multimedia-based Question-Answer Pairs Generation in Computer Assisted Healthy Education System 2022 10th International Conference on Orange Technology (ICOT).
[5] Jinmeng Wu, Fulin Ge, Pengcheng Sh, Lei Ma, Yanbin Hao Question-Driven Multiple Attention (DQMA) Model for Visual Question Answer 2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT).
[6] S.Lakshmipriya, Automatic Answer Checker, IJSRD - International Journal for Scientific Research & Development| Vol. 8, Issue 3, 2020.
[7] S.Nahida, Automatic Answer Evaluation Using Machine Learning, Dogo Rangsang Research Journal, Issn : 2347-7180
[8] Piyush Patil, Sachin Patil, Vaibhav Miniyar ,Amol Bandal, Subjective Answer Evaluation Using Machine Learning, International Journal of Pure and Applied Mathematics
[9] Merien Mathew, Ankit Chavan, Siddharth Baikar, Online Subjective Answer Checker, International Journal of Scientific & Engineering Research.
[10] Vasu Bansal, M. L. Sharma, Krishna Chandra Tripathi, Automated Answer-Checker, International Journal for Modern Trends in Science and Technology.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)