
*Asian Journal of Research in Computer Science*

***Volume 17, Issue 6, Page 36-57, 2024; Article no.AJRCOS.113528***
***ISSN: 2581-8260***

# Exam Assessor Tool: An Automated System for Efficient Answer Sheet Evaluation

**Vaibhav Shikhar Singh [a], Avni Verma [a], Garima Srivastava [a*] and Sachin Kumar [a]**

*[a] Department of Computer Science, ASET, Amity University Uttar Pradesh, Lucknow Campus, India.*

***Authors' contributions***

*This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.*

***Article Information***

DOI: 10.9734/AJRCOS/2024/v17i6455

**Open Peer Review History:**
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://www.sdiarticle5.com/review-history/113528

***Original Research Article***

***Received: 05/02/2024***
***Accepted: 09/04/2024***
***Published: 15/04/2024***

## ABSTRACT

With Education 4.0 and four quadrant approach number of innovations have gone into academics for efficient, experiential, and outcome-based education however assessment schemes are still very much dependent on manual assessment methods which are time-consuming and cumbersome. The grading system can sometimes be irrational, with diversified schemes for the same course and can also be biased. Covid 19 pandemic caused a global economic avalanche like we've never experienced in our lifetime. Many countries have implemented control measures such as blockades and curfews. The education system in this chaos saw a silver lining with academics shifting to online mode, with paradigm shift in teaching, assessment techniques too need to evolve. Work done is an effort to ease the process of assessment, a machine learning assisted model is developed that automates subjective answer evaluation in the education sector. Our project involved several crucial steps, including grayscale conversion, Natural Language Processing (NLP) for data cleansing, data splitting, and training an artificial neural network (ANN) to predict scores based on extracted features. ANN-based system grades subjective responses without human

**Corresponding author: E-mail: gsrivastava1@lko.amity.edu;*

intervention, reducing the workload of teachers and professors. Model constructed an ANN architecture with three layers using Rectified Linear Activation Unit (ReLU) and Sigmoid activation functions. Trained model was incorporated into a user-friendly web application using the Streamlit library. Model design gives a major boost in grading efficiency and accuracy while providing valuable feedback to students. Research surveys were conducted, and a dataset was constructed for training and testing the model. study yielded an accuracy of 83.14% after employing techniques such as text cleaning, preprocessing, and feature extraction.

*Keywords: Automated evaluation; handwritten answers; deep learning; natural language processing; subjective answer evaluation; artificial neural networks.*

## 1. INTRODUCTION

Assessing subjective answers is a crucial element of the education system, but conventional evaluation techniques may have limitations, especially in evaluating unrestricted written responses. In these cases, human graders can be prone to biases and errors, leading to inconsistencies and subjectivity in the evaluation process. Furthermore, the grading process can be time-consuming, leading to delayed feedback for students and an increased workload for teachers. With the growth of online learning, the need for more efficient and consistent evaluation methods has become more pressing. Automated systems, specifically subjective automatic answer evaluators (SAEs), have emerged as a potential solution to these challenges. SAEs are computer programs that use algorithms to assess the quality and accuracy of written responses to open-ended questions. These systems offer a faster, more objective, and consistent way of evaluating open-ended responses, allowing for instant feedback to students and reducing the workload of teachers. As a result, the use of SAEs has gained significant attention in recent years.

Tool developed is with objective to create a SAE that can improve the overall educational experience for both educators and learners. Tool development is based on existing research and future advancements in the field, emphasizing the significance of SAEs in modern education. The primary contribution in this field is centered around an automated answer evaluation system which leverages natural language processing (NLP) and machine learning methods to ensure precise and effective assessment. To validate the effectiveness of the system, research surveys were conducted among several faculties in college campus to determine the key factors to consider while evaluating answer sheets. By involving multiple stakeholders, diverse perspectives were established on the evaluation process, which helped us in creating a more robust and reliable system. From the surveys, a dataset was constructed for training and testing the model. This ensures that the system meets the specific requirements of the project and provides a more accurate evaluation of open-ended responses. Throughout the development process, algorithm proposed relied on extensive research and surveys to underpin the effectiveness and reliability of system.

Overall, evaluation systems make an important contribution to the field of education by easing the process of evaluation, specifically the use of SAEs. Research and analysis have enabled us to develop an automated system that can contribute to improving the educational experience for both teachers and students.

### 1.1 Existing Techniques

There are several existing techniques for evaluating answers online, ranging from rule-based systems to machine learning algorithms. One commonly used approach is to assign scores to answers based on predefined rubrics that assess various aspects such as relevance, clarity, and correctness. Another approach is to use machine learning models, such as natural language processing techniques, to evaluate answers based on the similarities between the student's answer and a model answer. Additionally, some platforms use peer-review mechanisms, where students evaluate each other's answers, or crowdsourcing methods, where answers are evaluated by a large number of people. Each of these techniques has its advantages and limitations, and choosing the appropriate one depends on several factors, such as the context of the question and the desired level of accuracy.

## 2. LITERATURE REVIEW

Automated grading of short answers has been an active area of research in recent years. Several studies have explored different techniques for automating the grading of short answers, such as grading based on predefined model answers and using natural language processing (NLP) and deep learning algorithms for short answer grading. This literature review discusses some of these studies and their findings. One study proposed a grading method that compared a student's response to a set of predefined model answers and calculated a score based on the similarity between each model answer and the student's response. The results of the study showed high levels of accuracy, but the method was limited by the reliance on pre-defined model answers that may not encompass all possible correct answers. Additionally, the method did not account for potential variations in accuracy due to cultural and linguistic differences [1]. Another study focused on the use of deep learning approaches for automated short answer grading (ASAG). Attention mechanisms were found to be effective, and the study suggested the need for more diverse datasets, including programming languages. The paper also suggested potential next steps for future research, including the development of ASAG models that can learn from context, provide immediate feedback, and take less training and inference time [2].

Another study analyzed the performance of automatic grading systems for long and descriptive answers using various techniques like data recovery, mapping, and natural language processing. The study found that manage-based techniques and information extraction strategies were more effective than corpus-based strategies or AI frameworks. The study highlights the need for continued innovation in this field to enhance the evaluation of student answers [3]. One study discussed the development of two natural language processing tools for scoring open-ended essay questions and short-answer content-based responses. The first tool, e-rater, has been successfully implemented in the Graduate Management Admissions Test (GMAT), with over 750,000 essays scored and an agreement rate of over 97%. The second tool is still in development and aims to automate the scoring of short-answer content-based responses. The use of automated essay scoring technology reduces the time and costs associated with having multiple human readers score essay responses. However, the drawback of automated scoring is that it may not always capture the nuances and complexities of human writing [4].

Another study proposed an innovative system that employs a semantic learning algorithm, Google's Universal Sentence Encoder algorithm, to generate sentence embeddings for analyzing subjective answers. The system offers students essential feedback to improve their responses by highlighting key aspects of the model answers provided. However, the system has limitations in handling longer answers or additional points not present in the model answer [5]. One study designed a system for subjective or descriptive examinations that utilizes cognitive and computation-based algorithms to extract patterns from candidate answers and compare them with model answers. The study found that one-word answers had a higher percentage of correct responses compared to one-line answers, which were often copied from tutorials. However, the system had limitations in handling multi-line answers, figures, examples, abbreviations, and contextual references [6].

In a research study, they used a nifty tool called Optical Character Recognition (OCR) to extract text from handwritten answer sheets. To grade these sheets, they applied machine learning and natural language processing (NLP) techniques. They measured the similarity of each sentence in the answer paper using cosine set measures, which helped determine the score [7]. One study proposed an approach for evaluating textual assignments using NLP techniques in e-learning systems [8]. The study used Python with NLTK toolkit and WordNet to match and paraphrase candidate answers with model answers. The research found that while the system had limitations in handling complex answers, one-word answers had a higher percentage of correct responses compared to one-line answers copied from tutorials. The study suggests that further development and refinement of the system may be necessary to improve its accuracy and effectiveness. Another paper explored the use of semantic analysis to auto-evaluate handwritten text by combining IRE and NLP methods to derive useful features [9].

The study focused on determining the matching score between keywords associated with a TRA and handwritten document images written in an unconstrained setting. Limitations of the prototype include its inability to comprehend complex equations and struggles with segmenting text with little spacing and excessive scribbling. Future enhancements include Self-Supervised Word Spotting, Contextual Query Expansion, POS Tagging, and NER. In another study, the development of an automated handwritten text grading system using a deep learning approach was discussed [10]. The system was developed and tested on a dataset of 450 handwritten answers. The model achieved a training accuracy just above 90% and an average test accuracy of nearly 80%. The study suggests that the system may not be suitable for longer texts with figures and equations, which requires a higher level of analysis and study. The authors plan to continue their research in this area with the aim of developing a model like an expert human grader.

Another research paper discussed the use of NLP to extract text from answer scripts and various similarity measures are calculated to assign marks [11]. Different similarity measures including cosine similarity, Jaccard similarity, bigram similarity, and synonym similarity are employed. Weight values are assigned to each parameter after conducting a survey to estimate the best weight. However, one limitation of this study is that weight values are assigned manually. One study investigated the problem of summarization faithfulness in neural abstractive summarization models [12,13]. The researchers proposed a QA-based metric for evaluating summary faithfulness, which correlates better with human judgment. However, the metric is limited by the quality of the QA model, and the final evaluation still needs to rely on human annotation. The study highlights the need for new inductive bias or additional supervision for learning more reliable models in abstractive summarization.

The education system uses evaluation to assess students' knowledge, but as the number of students increases, evaluation becomes more complex and manual assessment can be biased. To address this, computer-aided grading methods have been developed, such as automated assessment systems. Multiple-choice questions are widely used, but short answers are better for online examinations. Automated evaluation systems eliminate the tedious process of manual evaluation and reduce errors and biases. Research in this field has been ongoing since the 1970s, with various techniques available for different purposes. Recent deep learning techniques have outperformed traditional methods for data classification [14].

Automated systems are commonly employed to score written responses in tests, and they are trained and assessed using a set of responses from test-takers. To ensure consistency, important tests are scored by several evaluators. To assess the accuracy of the evaluators, model responses are used, and including them in the evaluation process improves the correlation between human and machine scores. While the performance of the scoring model is significantly impacted by the selection of the evaluation set, the selection of the training set is not as crucial, provided that the training set is adequately extensive and has a certain amount of noise [15]. In recent years, there has been a significant interest in using machine learning and natural language processing techniques for automatic text-based and essay type assessment grading systems in open and distance learning (ODL) institutions Blessing et al. [16] highlights the importance of such grading systems for providing efficient and quick feedback to students George et al. [17] proposes an automated system that uses deep learning and natural language processing to evaluate descriptive answer papers, which outperforms existing systems and can assign scores to non-evaluated descriptive answers Shashavali et al. [18] proposes a methodology to improve the performance of goal-oriented conversational agents by finding similarity between user input and representative text using N-gram and Sliding Window with FastText Word Embeddings technique Brill et al. [19] discusses Microsoft Research's participation in the TREC QA track and their use of data-driven techniques for web question answering. Finally, [20] provides an overview of the essential components of a QA system and highlights the need for the next generation of QA systems to consider multimedia data available in various forms. Authors in Samadi et al. [21] presented a unique clustering method for labeling documents. A multi-label text classification of words was presented based on weights using Nave Bayes classifier. Das, Bidyut et al presented an automatic question generation and answer assessment for subjective examination using multi criteria decision making approach [22,23].

Proposed model addresses the limitation and challenge faced in previous version and can improve upon the existing literature on automated grading systems. Firstly, the proposed system can incorporate a larger and more diverse set of model answers, taking into account cultural and linguistic differences. This can be achieved by leveraging natural language processing techniques, as demonstrated in Bonthu et al. [2,6]. Additionally, the system can utilize attention mechanisms and contextual learning, as suggested in Bonthu et al. [2], to improve its ability to evaluate short and long-form answers.

To address the limitations highlighted in Burstein et al. [4], the proposed system can incorporate advanced natural language processing tools and technologies that capture the nuances and complexities of human writing. The system can also provide personalized feedback to individual students, as suggested in Johri et al. [5], to enhance their learning and performance. Furthermore, to handle the limitations of existing systems in evaluating descriptive answers, as discussed in Singh et al. [3,6], the proposed system can utilize manage-based techniques and information extraction strategies, as well as cognitive and computation-based algorithms.

Finally, to improve the accuracy and efficiency of the grading process, the proposed system can utilize Optical Character Recognition (OCR) and machine learning techniques, as suggested in Sanuvala et al. [7]. By incorporating these advancements, the proposed system can provide a more accurate, efficient, and personalized approach to automated grading, addressing the limitations of existing systems and improving upon the state-of-the-art in the field. In conclusion, the reviewed literature suggests that while automated grading of answers has made significant progress, there is still a need for further refinement and improvement of the existing systems. The proposed system should aim to address the limitations of the current systems and provide accurate grading, even for longer answers or additional points not present in the model answer. Additionally, the system should be able to handle complex answers, examples, abbreviations, and contextual references.

## 2.1 System Architecture

Evaluation scheme involves number of tasks working in tandem supported not only by hardware but modified algorithm for the assessment.

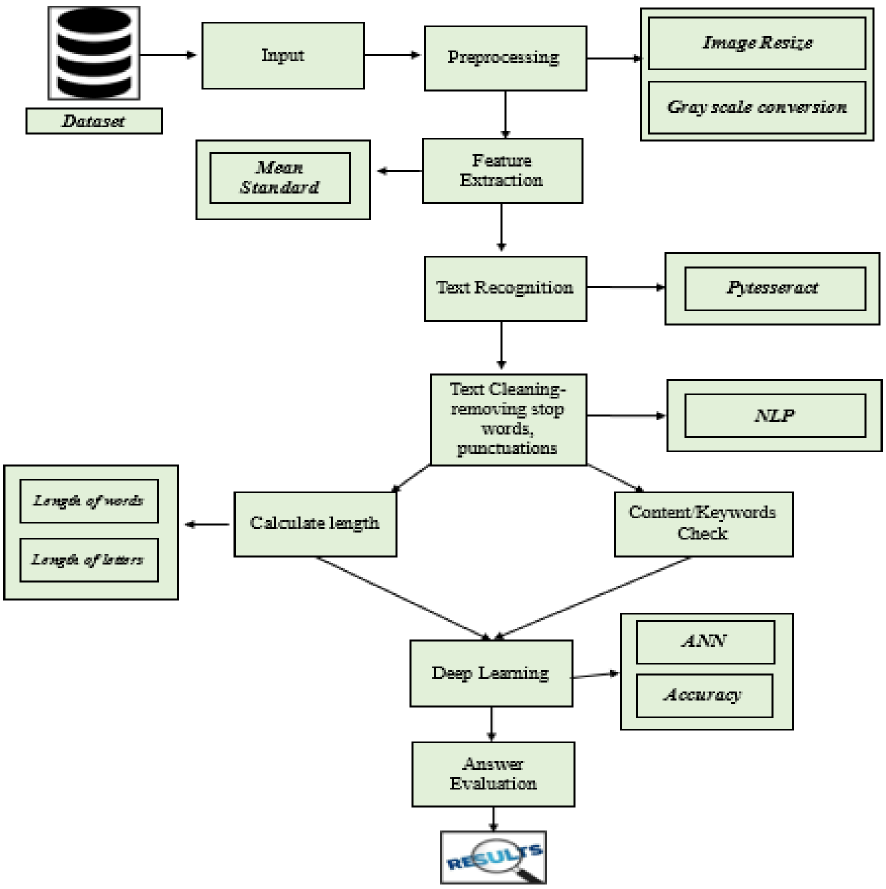



**Fig. 1. System design**

Fig. 1 presents a comprehensive illustration of the architecture of the system, which is developed in Python. The system starts by taking input from the user in the form of PDF files, as most answer sheets are scanned in this format. Preprocessing techniques such as image resizing and conversion to grayscale are applied to the extracted images. These techniques help to enhance the quality of the images and make them suitable for further processing. After preprocessing, features are extracted from the input. These features may include aspects such as text size, font type, and spacing, among others. These features are then passed through the Pytesseract library, which is a popular optical character recognition tool, to recognize the text from the extracted images. Once the text has been extracted, text cleaning techniques are applied to improve the accuracy of the input. Natural language processing techniques are used to remove stop words and punctuation, which do not contribute to the meaning of the text. This helps to improve the accuracy of the extracted text and ensure that it reflects the student's actual response.

After text cleaning, the length of the extracted text is calculated, and keywords are checked to evaluate the extracted text's quality. This helps to ensure that the extracted text is relevant and accurate and can be used for further evaluation. Finally, the extracted features, along with the text length and keywords, are fed into a machine learning model, specifically an artificial neural network (ANN). The ANN is trained on a set of labeled data to evaluate subjective answer sheets and provide marks. The ANN uses the extracted features and text parameters to assess the student's response and provide a mark that reflects the quality of the response. Overall, this workflow demonstrates how the system can effectively evaluate subjective answer sheets and provide accurate marks.

## 3. PROPOSED METHODOLOGY

At the university level, the requirement for automatic subjective response assessors is motivated by the need for increased efficiency in grading and delivering feedback to students. Professors can save time by using automatic assessors, and students can receive evaluation more quickly, which can enhance learning results. They can also lessen the possibility of prejudice and assure uniformity in grading. We conducted a comprehensive survey across seven diverse departments within the university, which comprised faculties from a wide array of fields such as Science, Technology, Engineering, Mathematics (STEM), Law, Applied Sciences, and Languages. To ensure broad participation, work utilized an online form which allowed for unlimited responses. Efforts were met with an excellent response, receiving feedback from nearly 52 faculty members with varying qualifications such as PhDs and MTech degrees to gather their feedback on the criteria they use to evaluate academic answer sheets. Many of them mentioned factors such as keywords, accuracy, and length as important considerations. As a result, platform developed implements a new grading system that takes into account both the use of keywords and the length of the answers. When queried about their willingness to adopt a software that integrated these attributes, a majority of the faculty members exhibited a favorable response, expressing their enthusiasm towards the potential usage of such a tool. Some even provided suggestions for additional features they would like to see included.

Fig. 2 to Fig. 6 shows diversified insights from responses of faculty members. Work done employed dataset of images from various answer sheets and divided it into training and testing sets. The aim is to enable machine to learn from the training data and test it on the testing data to evaluate its performance. Findings showed that the accuracy of the model is quite promising. Fig. 7 depicts Handwritten answer sheet images that are utilized in both training and testing datasets, encompassing a diverse range of representations.

### 3.1 Survey Insights

Based on the insights gathered from survey, it can be concluded that there is a strong demand for a software or tool that can automatically evaluate subjective answers. Survey results, as depicted in Fig. 2 to 6, indicate that 51.9% of the surveyed faculty members wanted such a system, while 26.9% were open to the idea opting for the “maybe option”. This highlights the need for a system that can provide automated evaluation of subjective answers. Furthermore, survey also revealed that respondents from STEM branches were particularly interested in the topic. Another key finding was that respondents preferred a user-friendly interface for the system, with an option for re-evaluation of answers. In terms of evaluation metrics, keywords were the most popular choice, with

92.3% of faculty members selecting it as their preferred option. Length of answers was the second-most popular metric, with 51.9% of faculty members selecting it. Other popular options included grammar and presentation of answers.

**SURVEY FOR PROJECT**

We are conducting a survey on the subjective answer evaluation methods used by teachers, faculties, and evaluators in Amity University Lucknow. The purpose of this research is to understand the different factors that are taken into consideration while evaluating a student's response to subjective questions.

Your participation in this survey is greatly appreciated as it will help us gain valuable insights into the subjective answer evaluation process at our university. We request you to take a few minutes to complete the following questionnaire.

Your responses will be kept confidential and only used for research purposes. Thank you for your participation in this study.

Best regards,

Vaibhav Shikhar Singh

Avni Verma

Amity University Lucknow.

**Fig. 2. The introductory declaration made prior to conducting the survey**

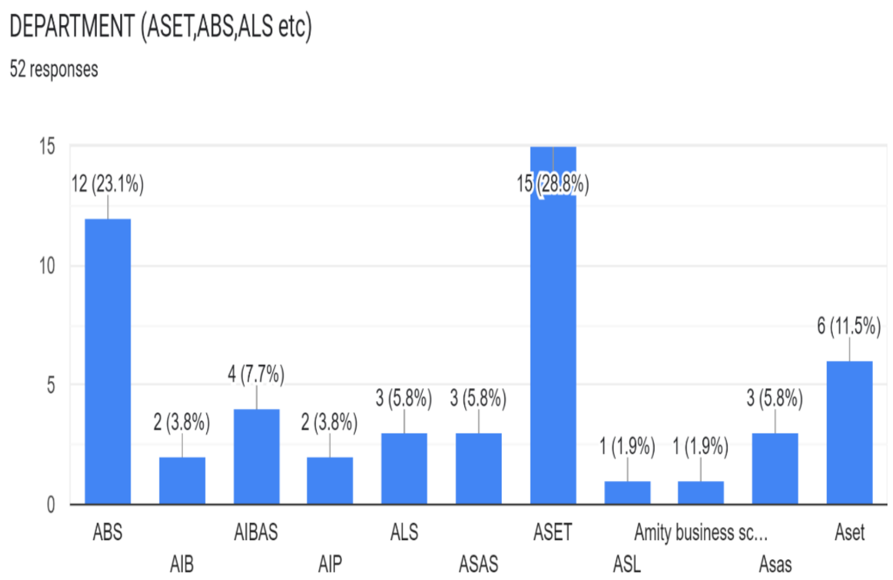



**Fig. 3. Survey was conducted across several departments ranging from S.T.E.M, Arts,Law etc.**

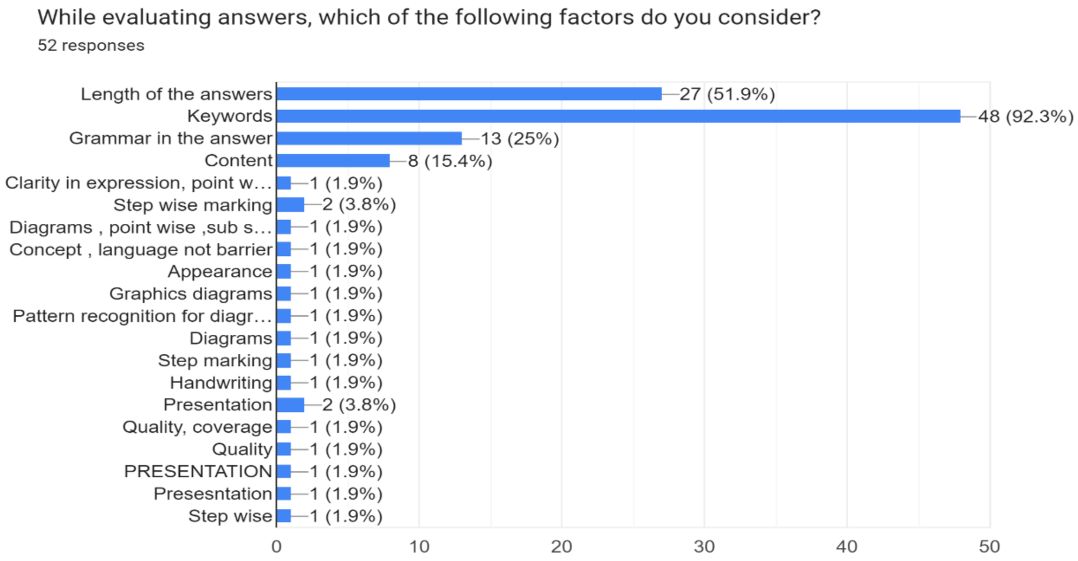



**Fig. 4. Factors that faculties consider when evaluating answer sheets According to recent survey results, the use of keywords received the maximum votes at 92.3%, while the length of answers received 51.9%**

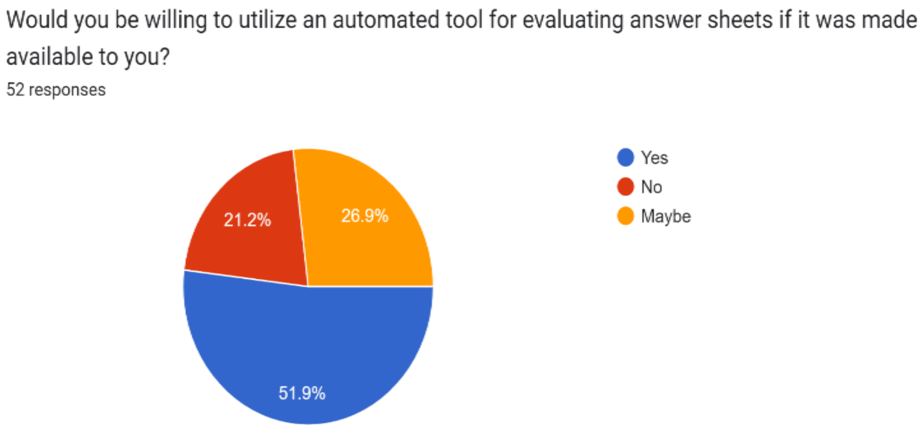



**Fig. 5. Over 50% of faculty members expressed a willingness to utilize the tool**

Overall, objective now as deduced from the survey results would be a system that considers keywords and length of answers for evaluation would be highly desirable. Additionally, the system should be customizable to meet the needs of different organizations and users, while also providing a user-friendly interface. Python is a versatile programming language that can be employed for diverse purposes, including web development, data analysis, machine learning, and other tasks. Its adaptability makes it ideal for creating a tool that automatically evaluates subjective answers and can handle a range of inputs while providing accurate and trustworthy findings. Python provides an extensive collection of libraries and frameworks, making it incredibly convenient and efficient to develop intricate applications with ease. This offers tools for developing an automated subjective response evaluation tool, including libraries for natural language processing, machine learning, data visualization, and more.

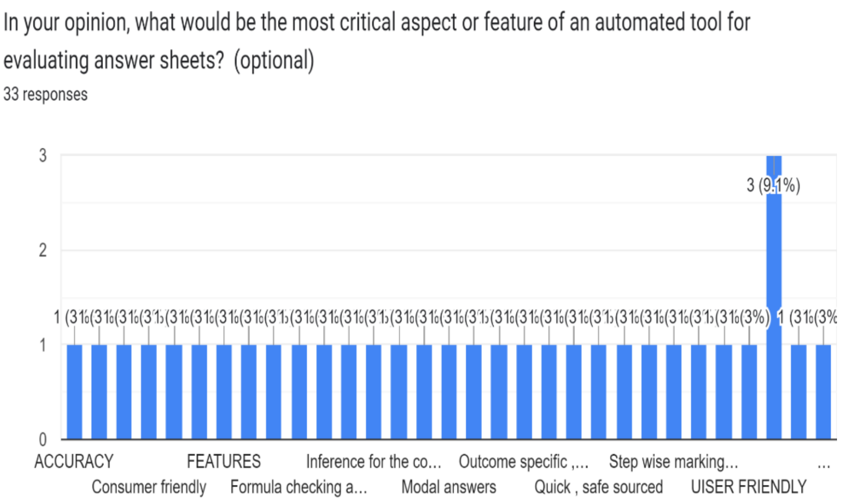



**Fig. 6. Suggested feature requirements for a tool based on user opinions for maximum benefit**

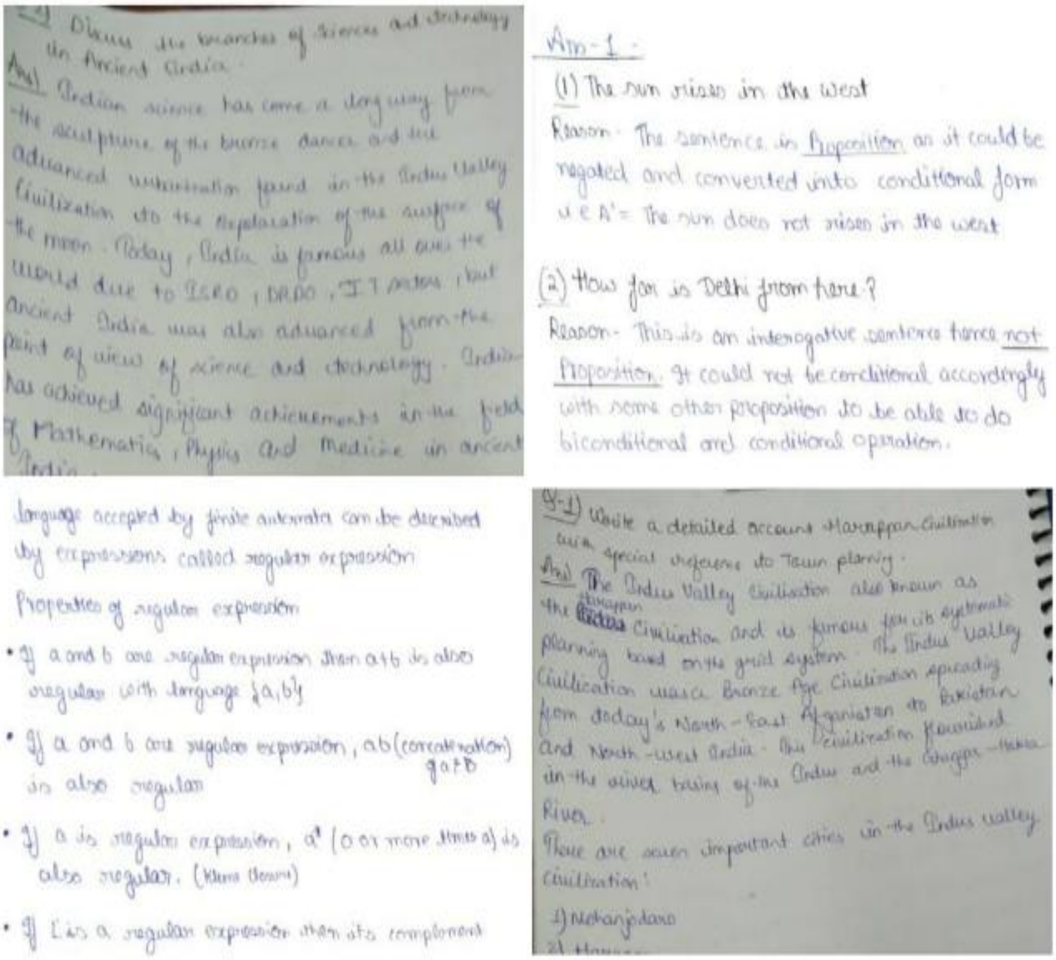


**Fig. 7. Handwritten answer sheet images are utilized in both training and testing datasets, encompassing a diverse range of representations**

Overall, these elements show why Python is frequently chosen over other programming languages for this kind of application and make it a great option for creating an automated subjective response evaluation tool. The automated subjective answer evaluator tool uses several libraries to perform its tasks efficiently. Here is a brief summary of each of the libraries and why they are used in the tool:

- Pandas: Pandas is a powerful data manipulation library that is used for data analysis and cleaning tasks. It is used in the tool to read and manipulate the data collected from the user's input.
- NumPy: NumPy is a fundamental scientific computing library in Python that is used for numerical operations such as array manipulation, mathematical operations, and linear algebra. It is used in the tool for image processing and manipulation tasks.
- Matplotlib.pyplot: Matplotlib is a data visualization library that is used to create plots, histograms, and other visualizations. It is used in the tool to visualize the results of the evaluation.
- Skimage.io: Scikit-image is an image processing library that is used to process, analyze, and transform images. It is used in the tool to load and read images.
- Skimage.transform: Skimage.transform is a submodule of scikit-image that is used for image transformations such as resizing, cropping, and rotation. It is used in the tool to resize and crop the images.
- Pytesseract: Pytesseract is a handy tool that enables Python developers to use Google's Tesseract OCR engine. With this tool, you can easily extract text from images without having to write complex code.
- Tkinter.filedialog: Tkinter is a Python GUI toolkit that is used to create graphical user interfaces. The filedialog submodule is used in the tool to allow the user to select input files.
- Keras.models: Keras is a high-level neural network API that is used to build and train deep learning models. It is used in the tool to evaluate the answers and generate scores based on the model's predictions.

After ensuring that all the necessary libraries have been imported, the next step in the project is to read the input. We took into consideration the existing system, where most answer sheets are stored in the form of PDFs. This entails that students write their answers, which are later evaluated by a separate authority. The evaluation process involves clicking images of each page of the answer sheet and converting them into PDFs for further evaluation. Therefore, the system was designed with this process in mind, making it capable for accepting PDF files as input. To enable the system to process PDF files, work employed a third-party Python module known as "fitz." This module is specifically designed for working with PDF documents and enables us to perform various operations, including creating, reading, and editing PDF files in Python. With this module, the process can efficiently extract and process the pages of PDF files, which are critical components of project.

In extracting the pages of a PDF file, algorithm utilizes the "fitz.Matrix" class to define a matrix that scales the size of the PDF page. This step is essential to ensure that the pages are well-aligned, and the images are of good quality. Then, we open the PDF document using the "fitz.open" method, and algorithm iterates through each page of the document. On each page, "get_pixmap" method is utilized to render the page as an image with the specified scaling matrix. Algorithm saves each rendered image as a PNG file using the "save" method. This way, page number of each image is retained by naming the resulting PNG files as "page-1.png," "page-2.png," and so on. Fig. 8 depicts the procedure adopted for isolating blanks and pages with answer from a PDF document.

Overall, the system’s ability to handle PDF input files is essential to the success of the project. The "fitz" module is instrumental in extracting and processing the pages of PDF files, allowing us to evaluate the students' answer sheets efficiently. Fig. 9 depicts extracted image in PNG form.

### 3.2 Preprocessing

Having obtained the input image using the aforementioned code, the next logical step would be to perform preprocessing on the image. Preprocessing is a crucial step that involves transforming the raw image data into a format that is more amenable for analysis and feature extraction. Numerous techniques exist that one can use for preprocessing data, such as resizing the image to a fixed size, converting the image to grayscale or binary format, and applying filters such as blur or edge detection. By implementing such preprocessing techniques, the image can be better prepared for subsequent stages of analysis, enabling more accurate and effective feature extraction. In subjective answer sheet evaluation tool , Work utilizes two important techniques to prepare the images for analysis: resizing and grayscale conversion. These techniques are crucial in ensuring that the data is of high quality and consistency, which makes it easier to extract relevant information from the answer sheets. Making sure all answer sheet images are of the same size and scale is crucial, and that's why resizing is necessary. This way, no matter how big or small the original document is, all the images will have a uniform look. This is helpful for subsequent analyses such as optical character recognition (OCR), which relies on images being consistent in size and resolution to accurately detect and extract text. By resizing, computational load required can be reduced for subsequent analyses, as smaller images require less memory and processing power. Fig. 10 depicts resized image in PNG format.

```
import glob, sys, fitz

zoom_x = 2.0
zoom_y = 2.0
mat = fitz.Matrix(zoom_x, zoom_y)
doc = fitz.open(filename)

for page in doc:
    pix = page.get_pixmap(matrix=mat)
    pix.save("page-%i.png" % page.number)
```

**Fig. 8. The procedure of isolating specific pages from a PDF document**


**Fig. 9. The extracted image in png form**

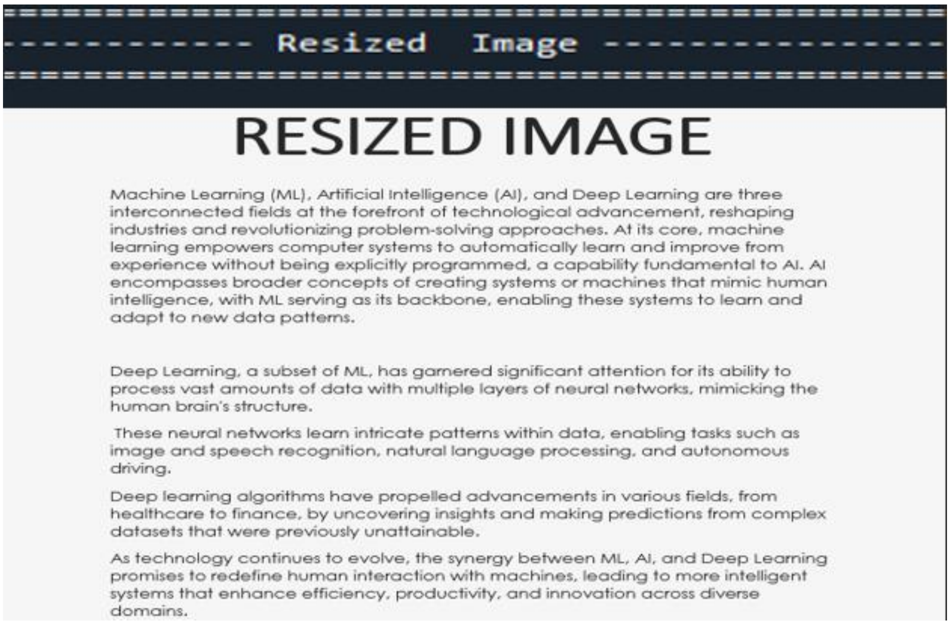



**Fig. 10. The resized image in png form**

### 3.3 Grayscale Conversion

Grayscale conversion is also important because it simplifies the image data by reducing each pixel to a single value representing its brightness. This simplification makes it easier to process the image data and reduces the risk of errors caused by variations in color or lighting conditions. Grayscale conversion is particularly useful in subjective answer sheet evaluation, as it allows researchers to extract the written text and other handwritten features of the answer sheet, which are typically black or dark in color. Fig. 11 depicts gray scale converted image extracted from the answer sheet pdf.

The formula used to convert the resized image from RGB to grayscale is:

$$gray = 0.2989 * red\_channel + 0.5870 * green\_channel + 0.1140 * blue\_channel \quad (i)$$

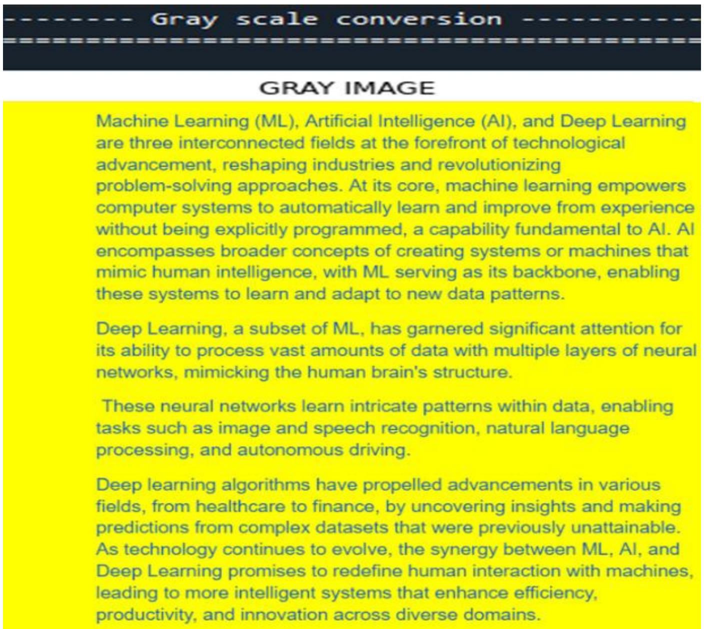



**Fig. 11. A Gray scale converted image extracted from the answer sheet pdf.**

### 3.4 Feature Extraction

Once the pre-processing step is complete, the next important step is to extract some features. Feature extraction is a technique used in machine learning and image processing to identify significant characteristics or patterns in the input data. In image processing, it involves recognizing specific attributes or traits of an image that differentiate it from other images.

In the context of subjective answer evaluation, feature extraction plays a vital role as it helps to extract relevant features or attributes from the student's answer. These features can then be used to assess the quality of the answer and provide feedback to the student.

Fig.12 depicts feature extraction using NumPy library. To better understand the grayscale values of the image, the work analyzes its average, middle, and variability by calculating the mean, median, and variance. These values are considered to be essential features that distinguish the input image from other images. The extracted features can then be fed as input to a machine learning algorithm, which can learn to differentiate between good and bad answers. For instance, these features can be used as input to a neural network, which can classify the student's answer as either correct or incorrect. The output of the code displays the mean, median, and variance of the grayscale values of the input image. This information can be used to gain insight into the characteristics of the image and can guide further analysis.

### 3.5 Text Recognition

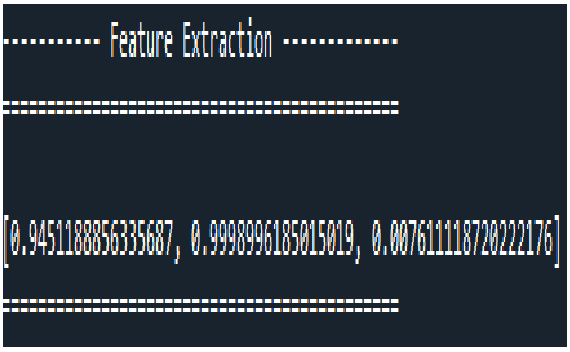
The algorithm proposed employs Optical Character Recognition (OCR) as a method to recognize and extract text from images. OCR technology is a powerful tool that enables us to convert non-textual media, such as images or handwritten documents, into machine-readable text. The OCR process involves a series of steps, including preprocessing the image, feature extraction, and text recognition. Preprocessing the image involves adjusting the image's contrast, brightness, and removing any noise to optimize the image for text recognition. Feature extraction is the process of identifying unique features of the text, such as the shape of the letters, which enables OCR software to recognize text accurately. To extract the text from an image, special tools called OCR engines like Tesseract are utilized. This process is commonly known as "text recognition" or "OCR text recognition". Its job is to identify the text from the image and extract it for further use.

The Tesseract OCR engine is a tool that can recognize and read text in images. It was first created by Hewlett-Packard (HP), but later on, Google released it as an open-source project, which means anyone can use and modify it for their own purposes. Tesseract OCR engine is widely used for OCR tasks, and it supports more than 100 languages. It utilizes deep learning algorithms to perform OCR tasks and has a high level of accuracy in recognizing text from images.

In evaluation, work uses Tesseract OCR engine to recognize and extract text from images. The recognized text can then be further processed, analyzed, or stored for downstream applications. Fig. 13 depicts recognized text from the answer pdf.

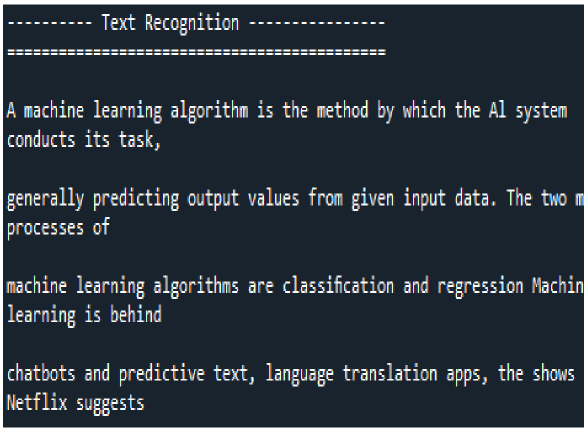
### 3.6 Text Cleaning

NLP, which stands for Natural Language Processing, is a fascinating subfield of artificial intelligence that focuses on how computers and humans interact through language. By using NLP techniques, Model can analyze and comprehend the structure, meaning, and context of human language in written text. These techniques are extremely valuable for cleaning up unstructured text data, as they can transform it into a more organized and machine-friendly format. In short, NLP is a powerful tool that helps us bridge the gap between computers and human language. NLP techniques can help to identify and remove irrelevant or redundant information, such as stop words, punctuation, and numerical characters. Work employs NLP specifically, the NLTK library for text cleansing removing stop words and special characters and converting text to lowercase, these are important steps in text preprocessing as stop words are common words that do not carry much meaning in a sentence, such as "the," "a," "an," and "in," and removing them can help reduce noise in the data and improve text analysis accuracy. By removing stop words, focus can be more on the most important words in a sentence, which can help us identify key themes or topics. Special characters, such as punctuation marks and numbers, can also be removed during text preprocessing. These characters may not add much value to the analysis and can cause complications when trying to analyze text data. By removing them, algorithms ensure that the text data is clean and standardized, which can make it easier to analyze.



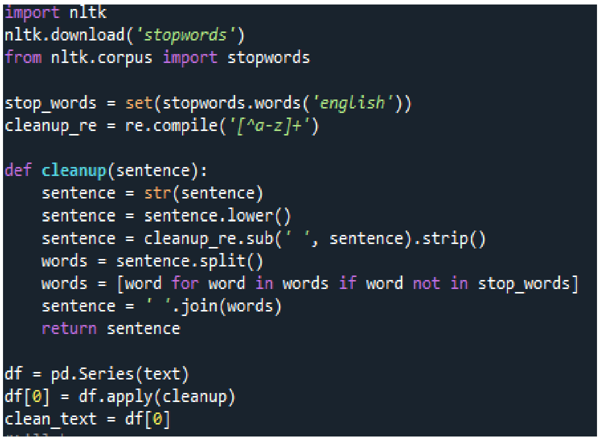
```
---------- Feature Extraction ------------
==========================================
[0.9451188856335687, 0.9998996185015019, 0.007611118720222176]
==========================================
```

**Fig. 12. Features obtained through a process of extraction**



```
---------- Text Recognition ---------------
==========================================

A machine learning algorithm is the method by which the AI system
conducts its task,

generally predicting output values from given input data. The two m
processes of

machine learning algorithms are classification and regression Machin
learning is behind

chatbots and predictive text, language translation apps, the shows
Netflix suggests
```

**Fig. 13. Recognized text from the pdf**



```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))
cleanup_re = re.compile('[^a-z]+')

def cleanup(sentence):
    sentence = str(sentence)
    sentence = sentence.lower()
    sentence = cleanup_re.sub(' ', sentence).strip()
    words = sentence.split()
    words = [word for word in words if word not in stop_words]
    sentence = ' '.join(words)
    return sentence

df = pd.Series(text)
df[0] = df.apply(cleanup)
clean_text = df[0]
```

**Fig. 14. Cleaning the text using NLP**

Converting text to lower case is also important for several reasons. To begin with, using it can make the text data more consistent and simplify the process of comparing various text documents. Secondly, it can help avoid errors caused by case sensitivity, where the same word may be treated as different words if it appears in different cases. Lastly, it can help us identify important words in a sentence more easily, this can avoid treating words that appear in different cases as separate entities.

To prepare the text data, the first step is to download the stopwords list from the nltk.corpus package. This list contains common words that are not very useful for analysis purposes, such as "the," "a," and "an." Fig. 14 depicts code snippet for cleaning the text using NLP. Next, a regular expression pattern is defined using the re library to remove any non-alphabetic characters from the text data. This ensures that only words remain in the text data, which can make analysis easier and more accurate. The cleanup () function is then used to preprocess the text data. This function takes in a sentence as input and applies several techniques to clean it up, such as converting the text to lowercase, removing non-alphabetic characters, and removing stop words using the stop words list downloaded earlier. Once the text has been purified, it is saved into a pandas Series object. To perform the cleanup() function on each item in the Series, the apply() method is utilized. This produces a cleaned version of the original text data, which is then stored in a separate variable called clean_text. These preprocessing techniques are essential for ensuring that text data is clean, standardized, and ready for further processing.

## 3.7 Implementing the Key Evaluators

From the survey it was concluded that most of the faculties considered the keyword and length of the answer an important metric while evaluating the answers hence algorithm designed implemented both in SAE tool

### 3.7.1 Keywords

Keywords help to identify the main ideas and concepts discussed in the answer. By focusing on specific keywords related to the question, you can quickly determine if the answer is relevant and directly addresses the topic at hand. Furthermore, keywords provide a way to check if the answer covers all the essential aspects of the question.

### 3.7.2 Length of the answer

A short answer may be sufficient for some questions, but in other cases, a more detailed response may be necessary. The length of the answer can also provide an indication of the depth of understanding of the person providing the answer. If an answer is too brief, it may indicate a lack of knowledge or effort. Conversely, if an answer is too lengthy, it may suggest that the person hasn't adequately focused on the key ideas or hasn't organized their thoughts coherently For instance, if a question carries a weightage of 15 marks, faculties expect the answer to be of a certain length and in a fully elaborative manner. They do not expect mere two or three lines of an answer, which will not be able to provide enough depth and detail to merit full marks.

To evaluate the answers, the proposed technique used different methods. A dictionary of keywords specific to each subject was created and integrated it with python script. This allowed us to automatically deduct marks for answers that did not include the necessary keywords. Additionally, an if-else ladder was developed to assess the length of each answer. After carefully considering the appropriate number of marks for each answer based on its length, a system customizable check was made so that users or evaluators could adjust how many marks they wanted to assign for a certain length of answer. This ensured that the evaluation system was fair and flexible for all users. Fig. 15 depicts the number of extracted words.

## 3.8 Data Splitting

Splitting a dataset into test and train sets is an essential step in building a machine learning model for automated subjective answer evaluation. The goal with this system is to build a tool that can reliably judge the quality of students' written responses on tests and assignments. To accomplish this, the algorithm uses a training dataset to teach the model how to evaluate the answers. Then, we use a separate test dataset to see how well the model performs on new, unfamiliar material. This approach helps prevent the model from becoming too focused on the training data and ensures it can make accurate judgments in practical settings. By analyzing the model's performance on the test dataset, the model can identify and address any problems, like errors due to a lack of information or an overly narrow focus. This evaluation helps to improve the model's accuracy and robustness before deployment in a real-world setting. Fig. 16 depicts a dictionary of domain keywords is provided to verify the context of responses. So, work done implemented a data splitting methodology that involved importing essential libraries and reading the dataset images, which were then resized to a predetermined size. The Scikit-Learn library was utilized to divide the dataset into distinct training and testing subsets. In addition, we employed one-hot encoding techniques to represent the categorical data. The resulting code produced accurate and efficient data splitting, which was a critical aspect of work done.

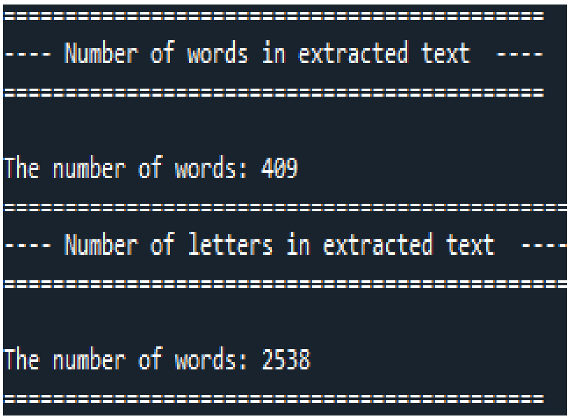



**Fig. 15. Number of Extracted words**

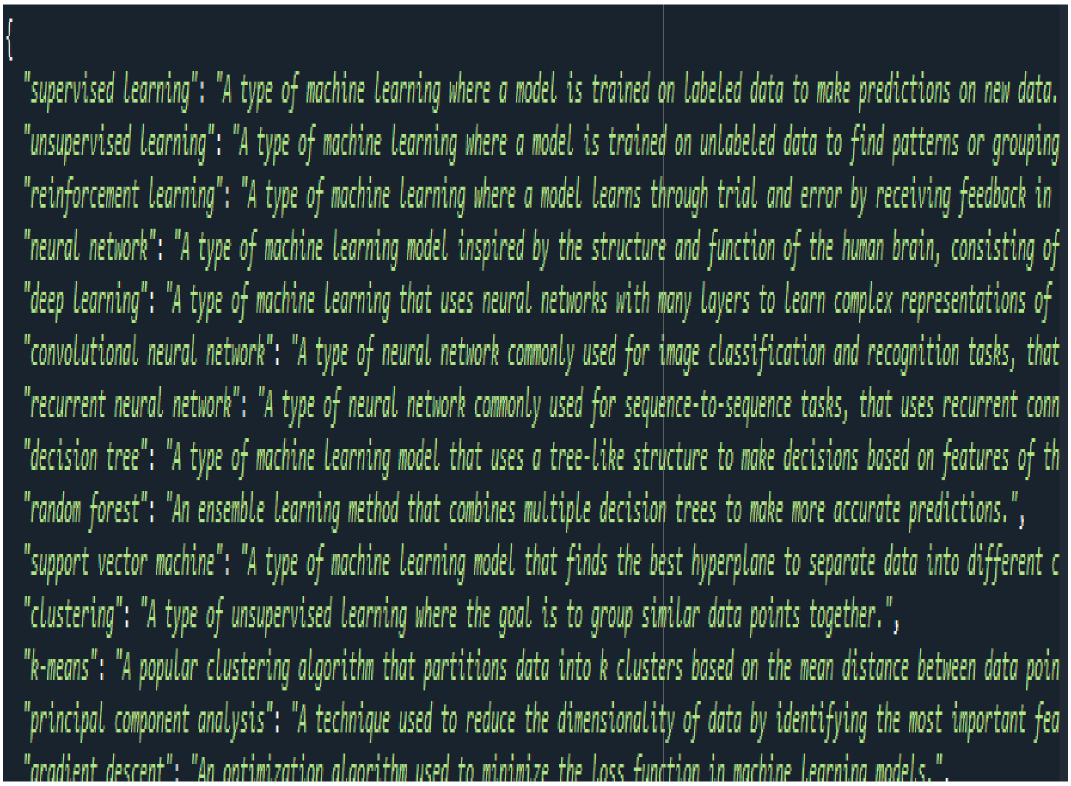



**Fig. 16. A dictionary of domain keywords is provided to verify the context of responses**

### 3.9 Training an Artificial Neural Network

Deep learning is an area within the domain of machine learning that focuses on teaching artificial neural networks to gain knowledge from extensive datasets. In the context of an automated subjective answer evaluation project, deep learning is used to train neural networks to identify patterns in textual responses that indicate whether an answer is correct or not.The process of deep learning requires several steps such as preprocessing the data, creating and training an appropriate neural network structure, and refining the model's efficiency through techniques such as hyperparameter tuning and regularization. By using deep learning techniques, an automated subjective answer evaluator can provide accurate and reliable evaluations of student responses, reducing the need for human grading and providing timely feedback to learners. Therefore, the system includes instructing a machine learning algorithm to forecast the rating of a reply by utilizing the features that were extracted. Artificial Neural

Networks (ANN) are a powerful tool for solving complex problems that involve large amounts of data. One application of ANN is in the field of automatic subjective answer evaluation in education. This technology can be used to grade subjective responses to questions, such as essay questions, without human intervention. The ANN-based system works by analyzing the text of the answers given by students and comparing them to a predetermined set of criteria. Afterward, the network evaluates and allocates a score according to how closely the response corresponds to the established criteria. This can greatly reduce the workload of teachers and professors who would otherwise have to grade each answer by hand.

Employing an Artificial Neural Network (ANN) system can potentially decrease grading bias and save educators time, as the system can be programmed to evaluate answers objectively based on predetermined criteria. Additionally, students can receive feedback on their responses more quickly, allowing them to improve their writing skills and better understand how to answer subjective questions effectively. Overall, the use of ANN in automatic subjective answer evaluation is a promising development in education technology. The utilization of an ANN system has the potential to considerably enhance grading precision and efficiency, while also furnishing valuable feedback to students. For this the Sequential() function is utilized for initializing the model. Subsequently, the architecture of the Artificial Neural Network (ANN) is defined, which comprises three layers. In the proposed model, the initial two layers utilize the Rectified Linear Unit (ReLU) activation function, and the final layer applies the Sigmoid activation function. This approach aims to enhance the accuracy of the prediction by leveraging the nonlinear transformations performed by the activation functions. Fig. 17 depicts training of Artificial Neural Network (ANN) to facilitate evaluation.

### 3.10 Rectified Linear Activation Unit (ReLU)

The ReLU activation function is defined as

$$f(x) = \max(0, x) \quad \text{(ii)}$$

where x is the input to the neuron. It returns the input if it is positive, and 0 if it is negative. The ReLU function is often used in deep learning models because it helps to prevent the vanishing gradient problem and can speed up training.

### 3.11 Sigmoid Activation Function

The sigmoid activation function is defined as

$$f(x) = 1 / (1 + \exp(-x)) \quad \text{(iii)}$$

The Sigmoid activation function receives an input value, x, and outputs a value between 0 and 1. This output can be interpreted as the probability of the neuron being "activated". The Sigmoid function is commonly applied in binary classification tasks, where the objective is to predict a binary output, such as "yes" or "no".

```
Epoch 1/10
WARNING:tensorflow:Model was constructed with shape (None, 50) for input
KerasTensor(type_spec=TensorSpec(shape=(None, 50), dtype=tf.float32, name='dense_input'),
name='dense_input', description="created by layer 'dense_input'"), but it was called on an
input with incompatible shape (None, 50, 50).
WARNING:tensorflow:Model was constructed with shape (None, 50) for input
KerasTensor(type_spec=TensorSpec(shape=(None, 50), dtype=tf.float32, name='dense_input'),
name='dense_input', description="created by layer 'dense_input'"), but it was called on an
input with incompatible shape (None, 50, 50).
11/11 [==============================] - 2s 9ms/step - loss: 0.4610 - accuracy: 0.7105
Epoch 2/10
11/11 [==============================] - 0s 2ms/step - loss: 0.3279 - accuracy: 0.7143
Epoch 3/10
11/11 [==============================] - 0s 1ms/step - loss: 0.2962 - accuracy: 0.7143
Epoch 4/10
11/11 [==============================] - 0s 1ms/step - loss: 0.2883 - accuracy: 0.7143
Epoch 5/10
11/11 [==============================] - 0s 2ms/step - loss: 0.2872 - accuracy: 0.7143
Epoch 6/10
11/11 [==============================] - 0s 1ms/step - loss: 0.2867 - accuracy: 0.7143
Epoch 7/10
11/11 [==============================] - 0s 1ms/step - loss: 0.2866 - accuracy: 0.7143
Epoch 8/10
11/11 [==============================] - 0s 1ms/step - loss: 0.2864 - accuracy: 0.7143
Epoch 9/10
11/11 [==============================] - 0s 1ms/step - loss: 0.2863 - accuracy: 0.7143
```

**Fig. 17. The Artificial Neural Network (ANN) model is trained to facilitate its evaluation**

## 4. RESULTS AND DISCUSSION

### 4.1 Test and Evaluate the Model

After the successful training of the model, it becomes imperative to test its efficacy by using a separate set of responses and evaluate its performance with respect to metrics such as accuracy. Further, to validate the model's performance, we compare its predictions with human graders' scores on a specific set of responses. This rigorous evaluation process ensures the reliability of the model, following which it is integrated into evaluation tool/system for generating evaluation scores. Subsequently, the trained model is deployed to predict outcomes for the training data, the test data is thoroughly analyzed to ensure the accuracy of the results. Remarkably the model performed quite well, achieving an accuracy rate of 83.14%. which serves as a testament to its effectiveness. In the final step, algorithm records the model's accuracy and error rates meticulously to facilitate its future use as a reference. Fig. 18 depicts evaluation metrics of the Trained ANN Model.

### 4.2 The Web Application

To take the model to the next level and make it accessible to a wider audience, the model was integrated into a user-friendly web application. This was achieved using Streamlit library, which enabled us to seamlessly incorporates the system and all its features into the app.Streamlit is a powerful library that allows developers to create web applications quickly and easily, without needing to worry about the underlying infrastructure. It provides a simple, intuitive interface that makes it easy for users to interact with model designed and obtain accurate results. Furthermore, Streamlit empowers us to share app across multiple platforms, providing access to a diverse community of users. Fig. 19 depicts the Web based application designed for the system and Fig. 20 depicts the overview of uploading PDFs within the web application.

Application's workflow is quite simple - a user or faculty member selects the required answer sheet PDF and uploads it to the tool. The app then performs all the necessary steps of data cleaning and preparation before feeding the input to the trained artificial neural network model. Ultimately, the model returns the evaluated marks to the user. As in Blessing et al. [16] highlights the importance of such grading systems for providing efficient and quick feedback to students.

### 4.3 Experiments and Results

With Education 4.0 and four quadrant approach number of innovations have gone into academics for efficient, experiential, and outcome-based education however assessment schemes are still very much dependent on manual assessment methods which are time-consuming and cumbersome. The grading system can sometimes be irrational, with diversified schemes for the same course and can also be biased. Covid 19 pandemic caused a global economic avalanche like we've never experienced in our lifetime. Many countries have implemented control measures such as blockades and curfews leading to a major economic slowdown. The education system in this chaos saw a silver lining with academics shifting to online mode, thereafter online education has become an integral part of the academics. With paradigm shift in teaching, our assessment techniques too need to evolve. Several challenges however do exist while designing the system, like quality of handwriting, scanning and parameters for assessment change with subject being evaluated. Work done is an effort to ease the process of assessment, a machine learning

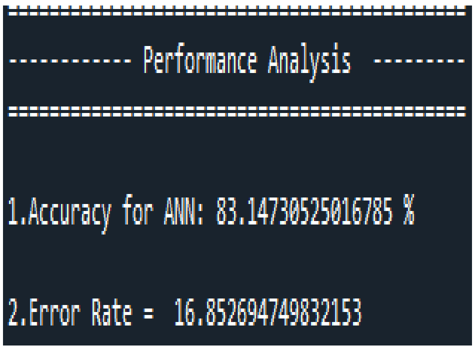



**Fig. 18. Measuring the Performance of the Trained ANN Model on the Testing Dataset**

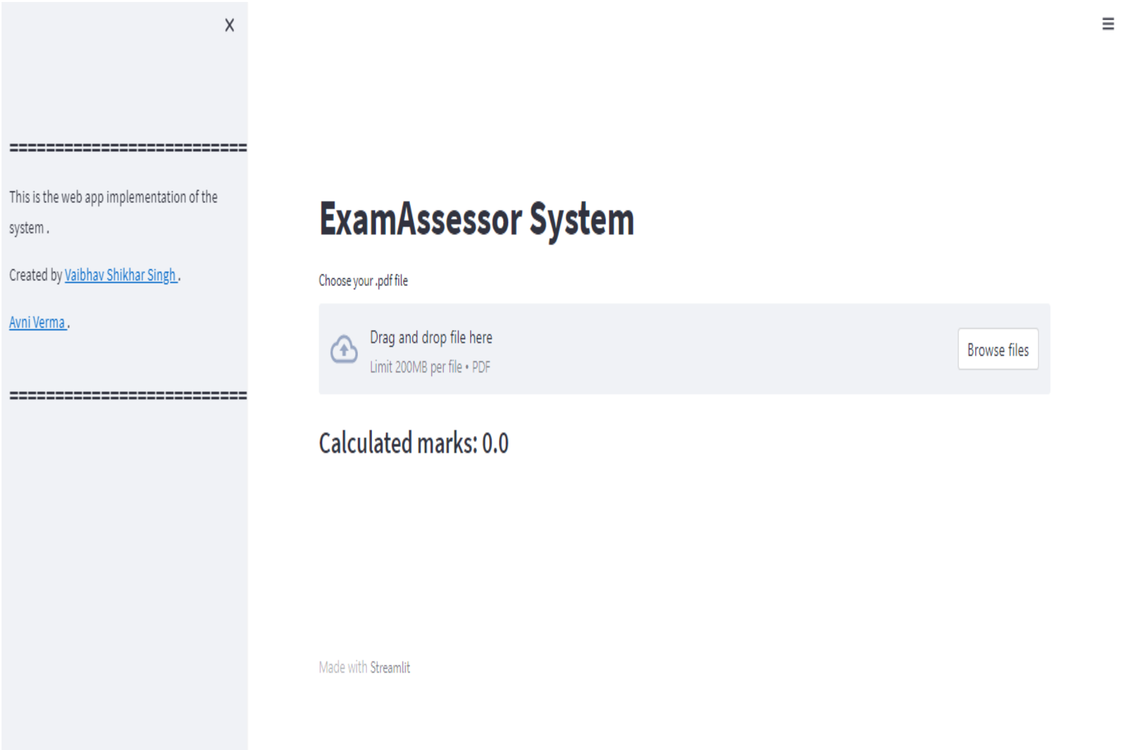



**Fig. 19. The Web based application we designed for the system**

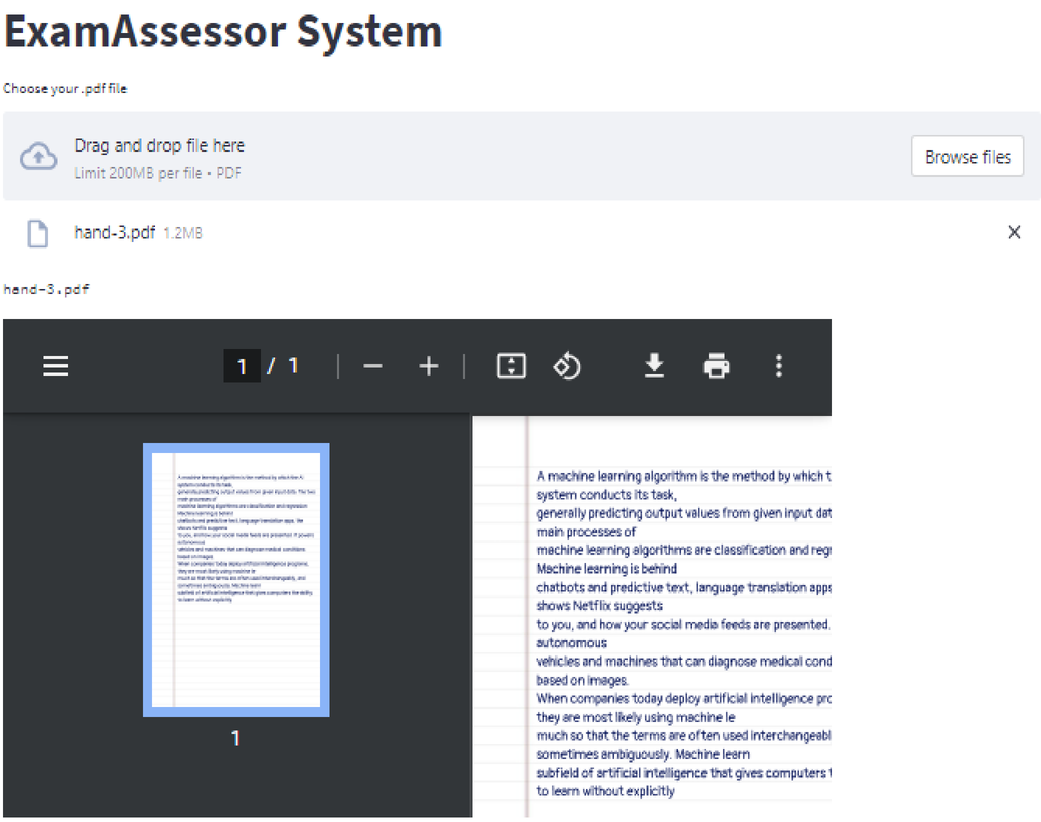



**Fig. 20. Overview of uploading PDFs within the web application**

**Extracted Text:**

A machine learning algorithm is the method by which the AI system conducts its task, generally predicting output values from given inp
two

main processes of machine learning algorithms are classification and regression

Machine learning is behind

chatbots and predictive text, language translation apps, the shows Netflix suggests

to you, and how your social media feeds are presented. It powers autonomous

vehicles and machines that can diagnose medical conditions based on images.

When companies today deploy artificial intelligence programs, they are most likely using machine le

much so that the terms are often used interchangeably, and sometimes ambiguously. Machine learn

subfield of artificial intelligence that gives computers the ability to learn without explicitly

**Keywords found!**

**Number of words in extracted text: 115**

**Number of letters in extracted text: 796**

**Calculated marks: 15.0**

**Fig. 21. A sample answer sheet, in which all the keywords were present and evaluated for grading**

easily without any burden. Another purpose of developing this software is to generate the report automatically during exams at the end of the session
or in between the

session. This project also allocates particular invigilator for particular hall.

**Keywords missing: AI, predicting, classification, regression**

**Penalty for missing keywords: 8**

**Number of words in extracted text: 594**

**Number of letters in extracted text: 3882**

**Calculated marks: 47.0**

**Fig. 22. An illustrative case of an answer sheet, where a few keywords were absent, resulting in a deduction of marks**

assisted model is developed that automates subjective answer evaluation in the education sector. After conducting extensive research and performing various tasks such as text cleaning, preprocessing, and feature extraction, study has achieved an accuracy of 83.14% and an error rate of 16.85. This demonstrates the effectiveness of our approach in analyzing text-based data. Furthermore, a web application was developed that is capable of analyzing PDF documents uploaded by users and assigning grades based on the number of keywords and length of the answers found. This application can serve as a valuable tool for educators and students, as it can provide quick and accurate feedback on written assignments. Work done has not only contributed to the field of subjective answer evaluators but also has practical implications for educational institutions. The results obtained from the study can be utilized to enhance the grading process, leading to more efficient and effective assessment of written work. Fig. 21 depicts the sample answer sheet, in which all the keywords were present and evaluated for grading and Fig. 22 depicts an illustrative case of an answer sheet, where a few keywords were absent, resulting in a deduction of marks.

## 5. CONCLUSION AND FUTURE WORK

Work done is an effort to design a unique technique to automatically grade handwritten responses. The findings have revealed that this approach is both accurate and efficient, which is quite promising. The potential benefits of the method are manifold: it can save valuable time for instructors, provide fair and consistent grading, and enable the use of machine learning algorithms for more advanced evaluations. System's success lies in its simplicity, as it uses metrics such as answer length and keyword matching to provide a model for grading. With paradigm shift in teaching, our assessment techniques too need to evolve. Several challenges however do exist while designing the system, like quality of handwriting, scanning and parameters for assessment change with subject being evaluated. One promising avenue for future research is to introduce machine learning algorithms that can predict the marks of answer scripts by training on various calculated parameters. This will allow us to provide more precise evaluations of handwritten answers, enabling better feedback for students and instructors. Moreover, algorithms aim to explore new techniques for effective and precise summary generation, detecting correct and incorrect coding answers, and checking diagrams drawn by students. By investing in these initiatives, the model designed can enhance the precision and speed of automatic grading systems while also giving teachers the tools they need to evaluate their students' progress more effectively. Overall, the potential impact of work is significant, as it can lead to more efficient and fair grading systems, ultimately benefiting both students and instructors. With further research and development, after conducting an exhaustive research process and carrying out various tasks such as text cleaning, preprocessing, and feature extraction, Model achieves an accuracy rate of 83.14% and an error rate of merely 16.85%. These outcomes suggest that the system is likely to perform efficiently and produce satisfactory results.

## COMPETING INTERESTS

Authors have declared that no competing interests exist.

## REFERENCES

1. Sultan MA, Salazar C, Sumner T, Fast and Easy Short Answer Grading with High Accuracy. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas. 2016;1565-1570.
2. Bonthu S, Sree SR, Krishna Prasad MHM, Automated short answer grading using deep learning: a survey. International Journal of Engineering & Technology. 2018;7(4):121-127.
3. Singh B, Kapoor J, Nagpure SM, Kolhatkar SS, Chanore PG, Vishwakarma MM, Kokate RB. An Analysis of Automated Answer Evaluation Systems based on Machine Learning. In: Proceedings of the 3rd International Conference on Intelligent Computing and Control Systems, Madurai, India. 2019;696-702.
4. Burstein J, Leacock C, Swartz R. Automated Evaluation of Essays and Short Answers. In: Proceedings of the 15th International Conference on Artificial Intelligence in Education, Springer, Amsterdam. 2011;21-28.
5. Johri E, Dedhia N, Bohra K, Chandak P, Adhikari H, ASSESS - Automated subjective answer evaluation using Semantic Learning. In: Proceedings of the 2020 International Conference on Machine Learning and Data Science, ACM, New York, NY. 2020;147-153.
6. Dhokrat A, Hanumant GR, Mahender CN. Automated Answering for Subjective Examination." In: Proceedings of the 2018 International Conference on Computing, Communication and Security, IEEE, Pune, India. 2018;143-147.
7. Sanuvala G, Fatima SS. A study of automated evaluation of student's examination paper using Machine Learning Techniques. International Journal of Computer Science and Information Security. 2018;16(5):13-20.
8. Dumal PAA, Shanika WKD, Pathinayake SAD, Sandanayake TC. Adaptive and Automated Online Assessment Evaluation System." In: Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, IEEE, Udupi, India. 2017;301-306.
9. Rowtula V, Oota SR, Jawahar CV. Towards Automated Evaluation of Handwritten Assessments. In: Proceedings of the. International Conference on Computer Vision and Image Processing, Springer, Singapore. 2019;179-184.

10. Rahman MA, Mahmud H. Automated Evaluation of Handwritten Answer Script Using Deep Learning Approach. In: Proceedings of the International Conference on Data Science and Information Technology, ACM, New York, NY. 2020;109-115.
11. Rahman MM, Akter F. An Automated Approach for Answer Script Evaluation Using Natural Language Processing. In: Proceedings of the 2018 2nd International Conference on Computer and Communication Systems IEEE, Nagoya, Japan. 2018;382-386.
12. Kumar C, Sunil RJ. Rama Sree. Assessment of performance of various machine learning algorithms during automated evaluation of descriptive answers" ICTACT Journal on Soft Computing: Special Issue on Soft Computing in System Analysis, Decision and Control. 2014;04:04.
13. Durmus E, He H, Diab M. FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019;6254–6264.
14. Loukina A, Zechner K, Bruno J, Klebanov BB. Using exemplar responses for training and evaluating automated speech scoring systems. Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications. 2014; 163–170.
15. Kundal R, Parekh B. Automated Assessment and Grading System for Short Answers. In 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS). 2018; 132-136.
16. Blessing G, Azeta A, Misra S, Chigozie F, Ahuja R. A Machine Learning Prediction of Automatic Text Based Assessment for Open and Distance Learning: A Review. In 2019 4th International Conference on Communication System, Computing and IT Applications (CSCITA). 2019;1-5.
17. George N, PJ S, Varghese SM. Grading Descriptive Answer Scripts Using Deep Learning. In 2018 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT). 2018;646-650.
18. Shashavali D, Vishwjeet V, Kumar R, Mathur G, Nihal N, Mukherjee S, Patil SV. Sentence Similarity Techniques for Short vs Variable Length Text using Word Embeddings. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). 2019;1-5.
19. Brill E, Lin J, Banko M, Dumais S, Ng A. Data- Intensive Question Answering. In Proceedings of the tenth international conference on World Wide Web (WWW). 2002;366-375.
20. Gupta P, Gupta V. A Survey of Text Question Answering Techniques. In 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS). 2018;295-299.
21. Samadi N, Ravana SD. Xml Clustering Framework Based on Document Content and Structure in a Heterogeneous Digital Library. Malaysian Journal of Computer Science. 2023;36(2):124–147.
Available:https://doi.org/10.22452/mjcs.vol36no2.2
22. Kaur W, Balakrishnan V, Wong KS. Improving Multi-Label Text Classification Using Weighted Information Gain and Co-Trained Multinomial Naïve Bayes Classifier. Malaysian Journal of Computer Science, 35(1), 21–36. (2023).
Available:https://doi.org/10.22452/mjcs.vol35no1.2
23. Das, Bidyut, Mukta Majumder, Arif Ahmed Sekh, and Santanu Phadikar. Automatic question generation and answer assessment for subjective examination. Cognitive Systems Research 2022;72:14-22.



*Peer-review history:*
*The peer review history for this paper can be accessed here:*
*https://www.sdiarticle5.com/review-history/113528*