SN Computer Science (2024) 5:653
https://doi.org/10.1007/s42979-024-02954-7

ORIGINAL RESEARCH

Empowering Educators: Automated Short Answer Grading with Inconsistency Check and Feedback Integration using Machine Learning

P. Sree Lakshmi[1] · J. B. Simha[2] · Rajeev Ranjan[1]

Received: 5 April 2024 / Accepted: 4 May 2024

Abstract

Automatic Short Answer Grading (ASAG) is a thriving domain of natural language understanding, focusing on learning analytics research. ASAG solutions are designed to alleviate the workload of teachers and instructors. While research in ASAG continues to advance through the application of deep learning, it faces certain limitations such as the need for extensive datasets and high computational costs. Our focus is on creating a machine-learning solution for ASAG that optimizes performance with small datasets and minimal computational demands. In this study, an ASAG framework namely Intelligent Descriptive answer E-Assessment System (IDEAS) is proposed. It uses a model answer-based approach that utilizes eight similarity metrics to compare the model's answer with student answers. These similarities are derived using the combination of both statistical and deep learning approaches. Unlike any prior work, this differs significantly because (i) the ASAG problem is conceptualized as multiclass classification rather than regression or binary classification, eliminating the necessity for extra discriminators. (ii) it aids evaluators in identifying inconsistencies in evaluation and provides comprehensive feedback. IDEAS is validated question-wise on various ASAG benchmark datasets namely ASAP-SAS, SciEntsBank, STITA Texas (Mohler). These datasets are constrained in ways such as lacking grading criteria for mark allocation. To address this limitation, a novel dataset, IDEAS_ASAG_DATA, is collected and utilized to validate the framework. Results demonstrate an accuracy of 94% when evaluating the framework on a specific dataset question. The results show that IDEAS attains comparable, and in certain instances, even superior performance when compared to human evaluators. We argue that the proposed framework establishes a robust baseline for future advancements in the ASAG field.

Keywords Automatic short answer grading · Machine learning · Multi-class classification · Clustering · Inconsistency · Feedback

Introduction

The role of assessment is vital in the process of teaching and learning. It helps in understanding the student learning, identifying any invisible barriers, and aids in improving the teaching approaches. Answer grading involves assigning a number representing the performance level or the quality of a student's answer. However, a consistent and fair assessment remains challenging as manual grading is more prone to errors, time-consuming, monotonous, tedious, and inconsistent [1]. Assessment can be done using different questions including (i) objective questions [fill in the blanks, multiple choice (MCQ), true or false] where a student has to choose one among the available options and (ii) Subjective questions (open-ended like essays and closed-ended like short answers) [2, 3]. The focus of the present work is on the automatic grading of short answers where the answer length varies from one phrase to one paragraph and the main focus is on the content, unlike essays where the focus is on the writing style and grammar.

✉ P. Sree Lakshmi
p.sreelakshmi@reva.edu.in

✉ Rajeev Ranjan
rajeev.ranjan@reva.edu.in

J. B. Simha
jb.simha@reva.edu.in

[1] School of Computer Science and Applications, REVA University, Bangalore, Karnataka, India

[2] RACE, REVA University, Bangalore, Karnataka, India

Published online: 13 June 2024

Figure 1 shows the typical answer evaluation process cycle. In Phase 1, the question paper along with the model answers (scheme), and criteria for grading are prepared by the subject matter expert. Then the exam is conducted, and students' scripts are collected for evaluation. Later in Phase 2, using the provided scheme, evaluation is carried out, followed by a second evaluator review for inconsistency check-in evaluation. Then, feedback regarding inconsistently evaluated answers is provided to the evaluator. If the inconsistency is greater than a certain threshold, then again third evaluation happens and finally, scores will be confirmed.

In the proposed work, we designed a framework namely the Intelligent Descriptive answer E-Assessment System (IDEAS), that assists the evaluator in performing the Phase 2 task as mentioned in Fig. 1 in a simplified manner. Due to legal and ethical issues, our proposed system is not intended to replace the human evaluators but in turn, aids as a screening tool in the evaluation process. The proposed framework aids the evaluator in three ways: (i) Perform automatic evaluation of concise descriptive answers in English, (ii) Identify inconsistency in the evaluation and (iii) Provide feedback concerning the inconsistently evaluated answers to the evaluator.

The machine learning methods discussed in the ASAG literature primarily fall into two categories: classification and regression. The classification approach employs classification techniques and gives scores logically like incorrect, correct, partially correct, and contradictory [4–7]. The drawback of this approach includes no validation of the result as it just provides the result as either correct or incorrect. On the other side, the regression approach uses regression techniques and assigns a score as a real number like 2.3, 3.4, etc. [8–10]. The main limitation of this method is the requirement of an additional discriminator to convert the real score to an integer value as the humans will not assign a score to an answer as a real number. To overcome the above limitations, we conceptualized ASAG as a Multiclass classification problem [11], where each mark is considered as a label.

One more important issue addressed in this work is inconsistency identification in evaluation. Inconsistency in evaluation is the variation in scores deviating from the expected standards [12]. Here, we focus on intra-rater inconsistency which occurs in situations where the evaluator assigns similar answers contrarily in diverse situations. This may be due to the mood swings of the evaluator, context, or personal bias. Inconsistency identification in evaluation is vital because it compromises the reliability of evaluation which leads to biased results. Addressing this issue will help to sustain fairness, integrity, and reliability in the assessment process. Regarding the inconsistency identification in the evaluation process, the literature is scarce [12, 13]. Moreover in literature, some systems provide feedback to the students regarding their performance in the examination [14–16]. As of our knowledge, currently, there are no systems that give feedback to the evaluator regarding inconsistently evaluated answers. Our proposed work aids in inconsistency identification and provides detailed feedback about it to the evaluator.

The Aim of the Work

In this paper, the focus is on developing a framework namely an Intelligent Descriptive answer E-Assessment System (*IDEAS*), for ASAGthat assists the evaluator in performing three vital tasks:

(i) Automatic assessment of short descriptive answers in English.
(ii) Finding inconsistency in the evaluation

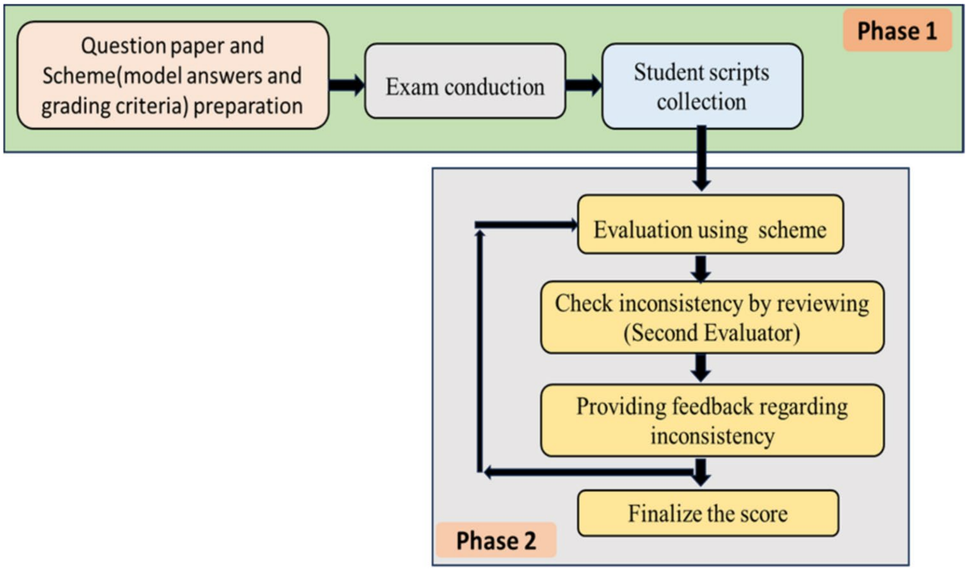
Phase 1
Question paper and Scheme(model answers and grading criteria) preparation
Exam conduction
Student scripts collection
Evaluation using scheme
Check inconsistency by reviewing (Second Evaluator)
Providing feedback regarding inconsistency
Finalize the score
Phase 2

Fig. 1 Evaluation process cycle

(iii) Provide comprehensive feedback regarding inconsistent answers to the evaluator.

We want to answer the following research questions:

- What is the best method for ASAG in the context of IDEAS? In this respect, we aim to study which kind of features/text characteristics it should take into account and which machine learning classification model is most suited for ASAG.
- How does the effectiveness of the proposed method change across different datasets?

By answering these research questions, the paper aims to fill the gaps in the literature highlighted in the previous paragraph. Upon request, we provide researchers in this field with a new dataset called *IDEAS_ASAG_DATA* for benchmarking and a methodological framework to experiment and validate ASAG solutions. To ease other researchers we offer step by step process in comparison with our tool. This process serves as a platform to build and validate future ASAG systems.

The Proposed Approach

IDEAS, the proposed framework tackles three essential tasks within the evaluation process. The first involves automatically assessing short descriptive answers in English. Here, the ASAG task is treated as a multiclass classification problem instead of binary classification or regression. The proposed framework IDEAS is a model answer-based approach. Our idea is to exploit eight varieties of features between the model and student answers for scoring. Table 1 below depicts the need/importance of each feature in ASAG. IDEAS integrates the traditional methodology of Natural Language Processing (NLP) tasks like bag-of-words, with the most modern technology of sentence embedding like Infersent [17] for the semantics of the text and t-distributed Stochastic Neighbour Embedding (t-SNE) [18] for cluster visualization. So, each answer is represented as eight similarities which are the fusion of statistical, lexical, semantic,

Table 1 Features/similarities extracted between the model and student answers and their importance

Sl. no.	Feature	Importance
1	Statistical similarity	Provides comprehensive and depth of the content
		Valuate the brevity and lucidity of writing
		Assists in determining whether students maintain sustained focus on key concepts or not
2	BoW (bag of words)/word—word similarity	Evaluating the degree to which students follow precise instructions and verify their use of accurate terminology in responses
		Assists in measuring the accuracy and uniformity of definitions
3	Keywords/unique words similarity	Carries more weight in grading subjects that prioritize particular concepts or terms
		Guarantees that students cover the essential concepts and themes relevant to the subject matter
		Provides an understanding of the degree to which a student applies precise terminology accurately. Assists in pinpointing misconceptions through the detection of improper keyword usage
4	Lemmatized words similarity	Assists in efficiently recognizing terms with similar meanings
		Lemmatization improves the accuracy of comparing underlying meanings by accommodating greater grammatical variation in responses, thereby enhancing matching precision
5	TFIDF similarity	Successfully handling stop words requires reducing their significance while emphasizing specific and informative terms pertinent to a given answer
		Adaptable across various fields and subjects, including mathematics, literature, science, and beyond
6	LSA similarity	Helps manage paraphrases and synonyms present in student responses
		Valuable for assessing answers containing ambiguous language, especially in cases where questions allow for multiple interpretations
7	Semantic similarity	Assists in identifying paraphrasing and rephrasing of model answers
		It helps in discerning situations where a student has effectively communicated the same ideas and concepts using their own words while maintaining the original meaning and accuracy
8	Summary similarity	Assists in evaluating the student's understanding of the subject matter
		Helps identify inaccuracies and misinterpretations in students' comprehension, facilitating targeted feedback

and summary characteristics. These acted as a rich feature matrix to be processed with machine learning approaches. For the validation, as priority should be given for each label, we used stratified K Fold Cross-validation. The proposed framework is applied to various benchmark datasets including ASAP-SAS, STITA, SciEntsBank, Texas (Mohler), and our novel dataset IDEAS_ASAG_DATA.

The second issue addressed is inconsistency identification in evaluation. Here, the assigned score is considered inconsistent when a student is supposed to get 1 mark but has scored 0. Inconsistency in evaluation is identified using unsupervised learning methods like K Means for clustering and t-distributed Stochastic Neighbour Embedding (t-SNE) for visualization of clusters. For each question we analyzed inconsistency in 2 ways (i) Identify inconsistency in actual marks (Evaluator marks) and (ii) Identify inconsistency in predicted marks of the best multiclass classification models. When we compare these two values of inconsistencies, the experimental results show that inconsistency is less in the predicted marks of the multiclass classification model than in the actual marks given by the evaluator. This proves the proposed method's accuracy is on par and in sometimes better than the human scoring. Finally, the identified inconsistent answers are provided as feedback to the evaluator for further review.

Methodological Contributions

The key contributions of our work are summarized as:

- Presented a novel framework IDEAS, that treats ASAG as a multiclass classification problem, departing from the conventional binary classification (correct/incorrect scores) or regression (real-number scores) approaches. In this unique framework, each mark assigned to the answer is treated as a distinct label, providing a more comprehensive and nuanced assessment.
- In Literature only a few similarities have been considered while grading [19]. But in IDEAS, we considered all important aspects that are required for grading short descriptive answers. The proposed work is novel since it jointly considers eight features of short answers including statistical, Word–Word, keyword, lexical features using TF-IDF, Lemmatized word, Contextual features computed using Latent semantic Analysis (LSA), Semantic features using Infersent, and summary features.
- We have put forward an approach based on Term Frequency-Inverse Document Frequency (TF-IDF) for inconsistency checking using K Means clustering, which can be readily applied to any dataset.
- In contrast to existing literature, validated the proposed methodology by conducting comprehensive experiments question-wise across various publicly available ASAG datasets, namely ASAP-SAS [19–22], SciEntsBank [19, 22], STITA [19], Texas (Mohler) [19, 23, 24], and on novel dataset IDEAS_ASAG_DATA. The findings indicate that our proposed method's performance is comparable to, and occasionally surpasses, state-of-the-art approaches across all datasets indicating its scalability.

The remaining sections of the paper are structured as follows: "State of Art/Literature" section provides an exploration of related works in the literature. "Proposed Methodology/Materials and methods" section offers an overview of the materials and methods employed in the study. "Results and Discussion" section presents the obtained results and includes a detailed discussion of the insights gained. Finally, "Conclusion and Future Enhancement" section concludes with final remarks and outlines potential future developments.

State of Art/Literature

In this section, we provide a comprehensive review of the previous literature that shares points of contact with the research presented in this work. We delve into the details and findings of these related studies. Research on Automatic short-answer grading commenced as early as the 1960s. Figure 2 shows the basic categorization of Automatic short answer grading methods from the literature [2].

Table 2 provides a detailed explanation of the different categories of Automatic short answer grading methods, their underlying principles, various employed techniques, limitations, and the corresponding authors. The concept mapping method considers each student's answer as a set of concepts, making it suitable for cases requiring solutions and justifications. On the other hand, information extraction methods utilize pattern-matching techniques like regular expressions and parse trees. Additionally, corpus-based methods analyse statistical properties from large corpora, while machine learning methods use extracted features from

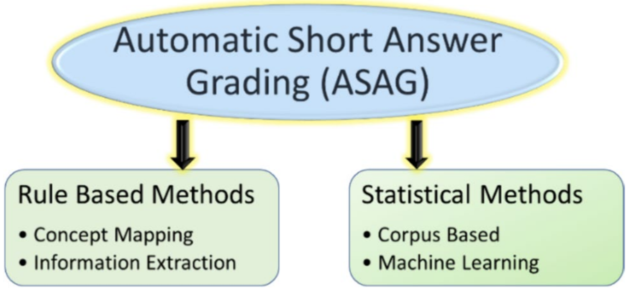

Fig. 2 Classification of automatic short answer grading methods

Table 2 Categorization of automatic grading systems, their basis, techniques used, and limitations [2]

Category	Basis	Techniques	Limitation	Author/study
Rule-based methods				
Concept mapping	Concepts	Syntactic analyzers, semantic analyzers Rhetorical parsers	Challenging to offer lexical features in Natural Language Understanding (NLU) techniques due to underlying obstacles	[25, 26]
Information extraction	Fact-finding	Parse trees, regular expressions	Expert human knowledge is necessary Failure occurs in domains abundant with correct patterns due to an inability to define and represent said patterns	[27, 28]
Statistical methods				
Corpus-based	Statistical properties	Statistical techniques	Need availability of huge corpora	[29]
Machine learning	Quantities extracted from NLP techniques	Classification methods	Utilizes classification to generate scores as either pass/fail or correct/incorrect without validation of the outcome	[6, 17, 30, 31]
		Regression methods	Utilizes regression to generate scores represented as real numbers such as 2.3, 4.5, etc., requiring a discriminator to convert floats into integer values	[9, 10, 32]

natural language processing techniques to build classification and regression models.

ASAG as a Classification Problem

The main limitation of classification methods for ASAG is that theygive scores like correct/incorrect, partially correct, and contradictory. There will be no quantification of scores. Many machine learning algorithms along with deep learning, models are used in literature. Few works from the literature that considered ASAG as a classification problem and produced results like correct/incorrect are as follows:

Zhang et al. [6] leveraged student and domain/question models to the task of ASAG. Extracted 31 features from the answer model, Question models, and student model. Explored the efficiency of the deep learning model Deep Belief Networks (DBN) in comparison with machine learning classifiers NB, LR, DT, ANN, and SVM. The main limitation here is to extract Knowledge components (KC's) from the question, human expertise is required which is labor intensive and the results are represented as either correct/incorrect only. Whereas [31] derived the word frequency, the part-of-speech tag (containing the pre tag and pos tag), the term frequency and inverse document frequency, and the entropy variation from the students' responses, Using these features built an SVM model and achieved a precision of 71.9% for two-level assessment. The main limitation here is they built only the SVM model.

Gomaa and Fahmy [33] introduced "ans2vec," a method that utilizes the skip-thought mechanism for embedding students' and reference answers to measure their similarity. The method was evaluated on three benchmarking data sets: Texas, Cairo University, and SCIENTSBANK dataset, achieving an RMSE of 0.91 and F1 scores ranging from 0.54 to 0.58. Logistic regression is used to predict the score. There is no clarity on whether the approach is applied to every question or the entire dataset.

Condor and Aubrey [30] proposed a method that utilizes BERT (Bidirectional Encoder Representations from Transformers) as a tool, educators can benefit from automated short answer grading (ASAG) without fully replacing human judgment in critical and high-stakes scenarios. However, the limitation here is the categorization of answers into correct or incorrect.

Wang et al. [17] demonstrated incorporating meta-learning with BERT enhances model performance, especially when labeled data is scarce. A comparison was made among Logistic Regression, KNN, Random Forest, BERT, and ml-BERT, with ml-BERT achieving the highest accuracy of 80%.

ASAG as a Regression Problem

The main drawback of considering ASAG as a regression problem is it generates the scores as a real number that again requires an additional discriminator to convert the real score into the integer value.

A few works from the literature that considered ASAG as a regression problem are:

Saha et al. [9] proposed a practical system for grading long or descriptive answers with limited training data. It uses expert-written reference answers and various similarity measures. The system identifies additional relevant information in student responses and achieves high accuracy with a root mean square error of 0.59 on a 0–5 scoring scale in testing. The results, initially in float values, were converted to integers using the nearest and next integer methods.

Prabhudesai et al. [34] proposed a technique to evaluate students' responses based on (i) word embedding via Glove combined with manually created features (such as the number of words, length of the answer, number of unique words, average word length, etc.) and (ii) a specially designed LSTM that accepts embeddings, manually created features, and the reference/golden answer as inputs. Additionally, they contrast several LSTM architectures, including simple, deep, and bidirectional (as well as those with and without handmade features), to determine which is best for the scoring task. The authors then contrast their answers with those that have previously been published in the literature. An MAE of 0.618 and an RMSE of 0.889 were achieved. Their research demonstrates how adding handmade characteristics enhances outcomes, particularly for MAE (of roughly 0.25). However, LSTM may not work well for small datasets.

Bahel et al. [10] presented architecture based on Siamese Manhattan LSTM (MaLSTM) for a fair evaluation of the answer sheet. In this architecture, the examiner creates a sample answer sheet for given sets of questions. By using the concepts of text summarization, text semantics, and keyword summarization, the final score for each answer is calculated. The limitation here is the scores are generated as real numbers that need additional algorithms to convert into integers.

Gobbo et al. [32] designed GradeAid, a framework for ASAG that analyses lexical and semantic features in students' answers using advanced regressors. Notably, it handles non-English datasets, undergoes robust validation, and is tested on various datasets, including a new one available for researchers. GradeAid achieves performance comparable to existing systems with root-mean-squared errors as low as 0.25 based on specific dataset-question tuples. In the process of scoring answers, humans do not assign an infinite range of scores but rather limit themselves to a predefined set of values. To address this, the output of the regressor should be rounded to the nearest integer value.

Deep learning models were also used in the literature for ASAG tasks. For example, in [35] they fine-tuned a Word-2Vec model to embed students' answers and supplemented it with handcrafted features like grammar errors, answer length, average word length, etc., for automatic scoring using an LSTM network. In their comparison between LSTM and logistic regression, they achieved an accuracy of 0.32 and aQuadratic Weighted Kappa (QWK) score of 0.94. Their findings indicated that LSTM significantly outperformed logistic regression, which yielded a QWK score of 0.65, in the task of scoring students' answers. Essay scoring methods may not be suitable for short answers with limited datasets. Whereas [36] proposed an approach using MaLSTM and sense vectors attained using SemSpace. It is tested using the Mohler dataset and CU-NLP) dataset. However, SemSpace cannot handle misspelled words or grammar errors. A few more deep learning techniques applied for ASAG tasks include attention networks [37], Recurrent Neural Networks [38], SemSpace Vectors with MaLSTM [36], BERT [11, 39, 40], Siamese Dependency tree with LSTM [35, 41], CNN [42]. Even though there is a lot of applicability of deep learning models for ASAG tasks, the main limitation is they need huge training data and computational resources. We propose a framework using Machine learning instead of deep learning that works well with small data and with fewer computational resources.

Proposed Methodology/Materials and Methods

Figure 3 depicts the proposed ASAG framework namely IDEAS. Here, we provide the details regarding the employed dataset ("Dataset" section), Various data pre-processing techniques employed ("Data Preprocessing" section), the

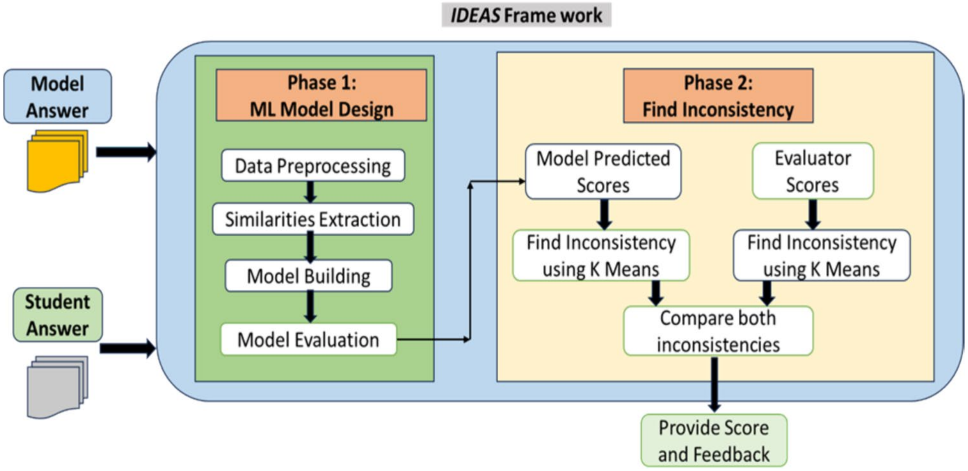

Fig. 3 Proposed methodology for IDEAS framework

methods exploited to extract similarities between the model and student answers ("Feature Extraction" section), the experimented Machine learning classification algorithms ("Build a Machine Learning Model" section), the metrics used for IDEAS performance evaluation ("Model Evaluation" section) and finally the proposed method to check the inconsistency and provide feedback in ("Identifying Inconsistency in the Evaluation and Providing Feedback to the Evaluator" section).

The IDEAS framework initiates with data collection, followed by pre-processing to improve data quality. Subsequently, eight types of similarities are computed between student and model answers. The importance of these similarities has been discussed in detail in "Feature Extraction" section. These similarities are considered as the independent features, and the evaluator-assigned score as the dependent feature. Several machine-learning classification models are then constructed and assessed using metrics such as accuracy, precision, recall, and F1 Score. Inconsistencies are identified using the unsupervised learning technique K Means. We checked the inconsistency in both actual marks and the predicted marks of the machine learning model. Experimental results indicate that there is a higher degree of variability in the actual evaluator scores than in the scores predicted by the model. This strongly suggests that the model's predicted scores are reliable.

Dataset

The proposed framework is validated on various ASAG datasets from the literature to check its scalability. These datasets include, Automatic Student Assessment Prize Short Answer Scoring (ASAP-SAS), SciEntsBank, STITA, and Texas (Mohler). Each of these datasets has drawbacks when applying them to ASAG tasks. The details regarding each dataset along with their limitations are mentioned in Table 3.

To overcome the limitations of existing benchmark datasets, a novel dataset, IDEAS_ASAG_DATA is collected and the proposed work is validated on it. This dataset is collected from a School, in Andhra Pradesh, India. The data collection and initial screening have been carried out for 6 months. Since the class strength is less, the data is collected from multiple examinations of Class VII Social Science subject. The labels given for each answer are the marks the respective evaluator provided. The dataset consists of a total of 20 questions among which there are 6 one-mark questions,

Table 3 Details on ASAG datasets exploited in the proposed IDEAS framework

Dataset	# Questions	# Samples	Language	Average word length for each answer	Subject	Score/grade range	Limitation
ASAP SAS[a]	10	17,043	English	150–550	Diverse: Biology, Science, Arts, English, etc	N [0, 3]	Essay answers, suited for essay prompts i.e. Automatic Essay Scoring (AES) systems
SciEntsBank[b]	4	139	English	1–110	Science	{0, 1}	Marks allocation lacks appropriate rubric alignment
STITA[c]	6	333	Italian/but translated to English	50–100	Statistics	N [0, 1]	Suits for essay prompts
Texas (Mohler)[d]	85	2558	English	2–100	Science	R [0, 5]	Few answers for each question Allocation of marks lacks suitable grading criteria (rubrics) forexample, A two-word answer has been awarded 5 marks
IDEAS_ASAG_DATA	20	800	English	10–50	Social science, class VII	N [0, 4]	–

[a] https://www.kaggle.com/c/asap-sas

[b] https://github.com/dbbrandt/short_answer_granding_capstone_project

[c] https://github.com/edgresearch/datasetautomaticgrading-2022

[d] https://github.com/dbbrandt/short_answer_granding_capstone_project

6 two-mark questions, and 8 four-mark questions. In the actual scenario, there is a choice for answering the questions in the exam, so every student did not attempt all the questions. Therefore, we considered only 40 answers from each question to maintain commonality. The dataset consists of a variety of factual and evaluative questions, representing the first two levels of Bloom's taxonomy Remember and Understand. Factual questions solicit simple answers and Evaluative questions where answers should be analyzed at multiple levels and from different perspectives. Table 4 shows a sample set of questions from the dataset. Table 5 shows a sample question of one mark, model answer, student answers, and its corresponding scores.

Data Preprocessing

Data pre-processing must be done before building the machine learning modelsto improve the data quality. Table 6 shows the list of pre-processing methods applied to the dataset, their description, and a sample student answer SA1 after applying a particular pre-processing method.

SA1: "Go green and eliminate global warming. Do not neglect the environment it is our basic need of life."

Feature Extraction

The proposed methodology is a model answer-based approach. Here, eight types of similarities are extracted between student answers (SA) and model answers (MA), These are capable of extracting the overall similarities between the model and the student answers. Table 6 shows the importance of each similarity concerning the ASAG task. The step-by-step process followed for extracting each similarity is explained below. Implementation is done using Python built-in modules from the Python library sklearn like Count Vectorizer, Tf-idf Vectorizer, and machine learning algorithms.

Statistical Similarity

Here, statistical data including the count of sentences, words, and unique words from both model and student answers are considered for similarity. Initially, all statistical data is collected and stored in individual arrays for each model and student answer. Then Euclidean distance between these two arrays is considered a statistical similarity. Figure 4 shows the process for statistical similarity [8, 9].

Word–Word Similarity Using BoW

Here the similarity between the model and student answer is calculated by checking the presence or absence of each word in both answers. This is implemented using the Count Vectorizer library from Scikit-learn. Initially an instance, the CV of the count vectorizer is created and it is fit and transformed on the model answer creating a model answer bag of words, and then transformed on the student answer which produces a student answer bag of words. The cosine similarity between these two is considered for BoW similarity. Figure 5 shows the procedure for Word_Word similarity [8, 9].

Keywords/Unique Words Similarity

The above word–word similarity is calculated by including the stop words. Here the similarity is calculated based on the presence or absence of a unique word/keyword in the model and student answers. This is implemented by including the parameter stop words in the count vectorizer. Figure 6 shows the process of keyword similarity.

Lemmatized Words Similarity

The above methods will have more dimensions, including all words from answers. The dimensions can be reduced by using a Lemmatization process that helps in converting the words to their respective root words. Then similarity is calculated between the lemmatized words from the model and student answers. Figure 7 shows the process for Lemmatized word similarity.

TFIDF Similarity

The disadvantage of the Bag of Words method of text representation is that it may result in high-dimensional feature space. It neither retains the information regarding

Table 4 Sample factual and evaluative questions from the dataset

Sl. no.	Factual questions	Evaluative questions
1	Mention the types of forests in India	Trees in deciduous forests shed their leaves. When and why?
2	Write two slogans on the Conservation of forests	Mangrove forests are natural protectors of the seacoast, Discuss
3	What do you know about the Universe?	Compare the climate of various climate regions
4	Write two slogans on the protection of the environment	"Nature's environment is different from man-made". Explain
5	Write a list of items made from forest products	How is the biosphere different from other elements of Earth?

Table 5 A sample one-mark Question from the dataset, model answer, student answer, and its scores by the evaluator

Question	Model answer	Student answer	Score
Write two slogans on the protection of the environment	Go green and eliminate global warming. Do not neglectthe environment, It is our basic need of life	SA1: Go green and remove the global warming. Do not neglect the environment. It is our basic need of life	1
		SA2: Save water save earth. Go trees help humans	0
		SA3: Save the environment and eliminate global warming. Save and do not neglect the environment. It is the basic need of our life	1
		SA4: We should reduce global warming. We should support the environment	1

the grammar of sentences nor maintains the word order. Term Frequency and Inverse Document Frequency (TF-IDF) reflect the importance of a word to a document in a corpus. It is implemented using the TFIDFVectorizer() library from scikit-learn. Figure 8 denotes the procedure for TFIDF similarity [8, 9, 43].

LSA Similarity

The Bag of Words or TFIDF method will not retain the context of a sentence. But while grading the answers the context plays a very important role. So, Latent Semantic Analysis (LSA) is used to compare the contextual similarity between the Model and student answers. It is implemented using TruncatedSVD () from scikit-learn. Figure 9 shows the detailed process followed for LSA similarity [8, 9].

Semantic Similarity

While grading the answers, semantics plays a vital role. The semantic similarity between the model and student answers is extracted using Infersent [17]. Infersent generates sentence encodes and finds the similarities between the model and student answer sentences. Here, the similarity of the model answer sentence with each sentence in the student's answer is calculated. This technique works for situations with a single sentence in the model and student answers, but the number and order of sentences generally differ [8]. First one-to-one mapping of sentences then the normalized sum was computed. This method fails in case of change in the order of sentences in student and model answers. Various other methods to find semantic similarity are word mover distance, smooth inverse frequency, pre-trained encoders like Google sentence encoder, and transformer-based models like DAN (Deep Averaging Network). Figure 10 depicts the process followed [8, 9].

Summary Similarity

Finally, the similarities between the summaries of the model and student answers are extracted using the extractive summarization method. Here the summaries of top K sentences in both model and student answers are calculated, then cosine similarity between those two summaries is considered for Summary similarity. Figure 11 depicts the process followed for summary similarity.

Table 6 Pre-processing techniques with their description and a student's answer SA1 after applying preprocessing

Sl. no.	Pre-processing technique	Description	Student answer, SA1 after pre-processing
1	Removingpunctuation/ special characters	Eliminate all unnecessary special characters and symbols from the data	Go green and eliminate global warming Do not neglect the environment It is our basic need of life
2	Tokenization	Divide the sentences into distinct words named tokens	['go', 'green', 'and', 'eliminate', 'the', 'global', 'warming', 'do', 'not', 'neglect', 'the', 'environment', 'is', 'our', 'basic', 'need', 'of', 'life']
3	Removestopwords	Remove the most frequently used words like a, an, the, etc	['go', 'green', 'eliminate', 'global', 'warming', 'neglect', 'environment', 'basic', 'need', 'life']
4	Stemming	Each word is reformed into its root	['go', 'green', 'elimin', 'global', 'warm', 'neglect', 'environment', 'basic', 'need', 'life']
5	Lemmatization	Each word is converted into its base form by considering the grammar	['go', 'green', 'eliminate', 'global', 'warming', 'neglect', 'environment', 'basic', 'need', 'life']

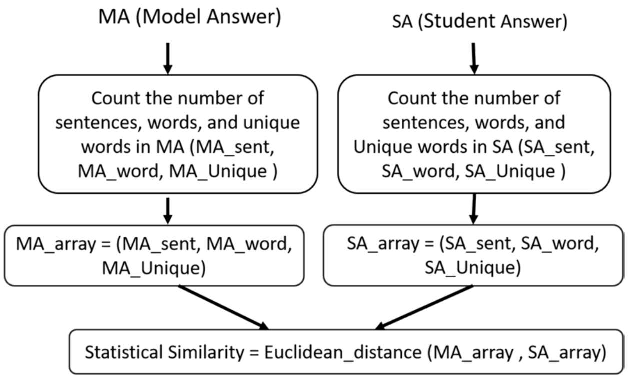
MA (Model Answer)
SA (Student Answer)
Count the number of sentences, words, and unique words in MA (MA_sent, MA_word, MA_Unique)
Count the number of sentences, words, and Unique words in SA (SA_sent, SA_word, SA_Unique)
MA_array = (MA_sent, MA_word, MA_Unique)
SA_array = (SA_sent, SA_word, SA_Unique)
Statistical Similarity = Euclidean_distance (MA_array , SA_array)

Fig. 4 Method for statistical similarity

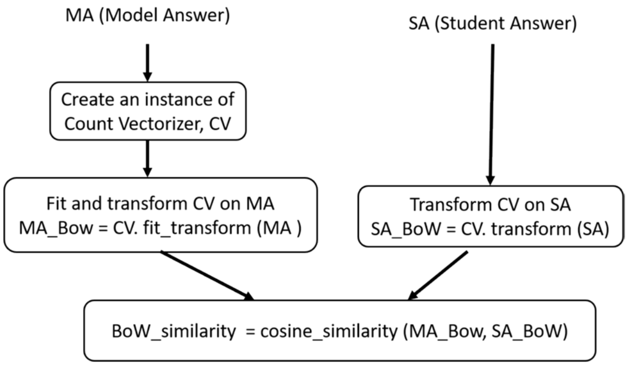
MA (Model Answer)
SA (Student Answer)
Create an instance of Count Vectorizer, CV
Fit and transform CV on MA
MA_Bow = CV. fit_transform (MA)
Transform CV on SA
SA_BoW = CV. transform (SA)
BoW_similarity = cosine_similarity (MA_Bow, SA_BoW)

Fig. 5 Method for Word_Word similarity

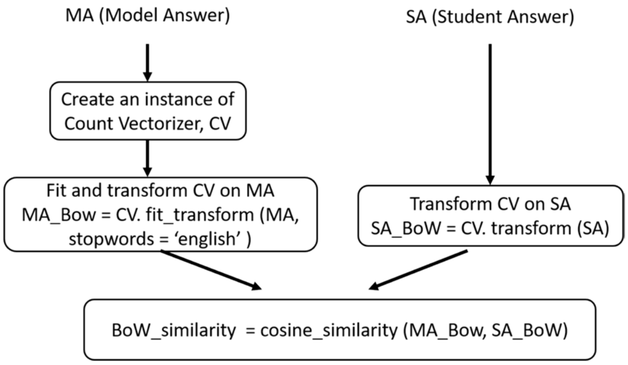
MA (Model Answer)
SA (Student Answer)
Create an instance of Count Vectorizer, CV
Fit and transform CV on MA
MA_Bow = CV. fit_transform (MA, stopwords = 'english')
Transform CV on SA
SA_BoW = CV. transform (SA)
BoW_similarity = cosine_similarity (MA_Bow, SA_BoW)

Fig. 6 Method for keyword similarity

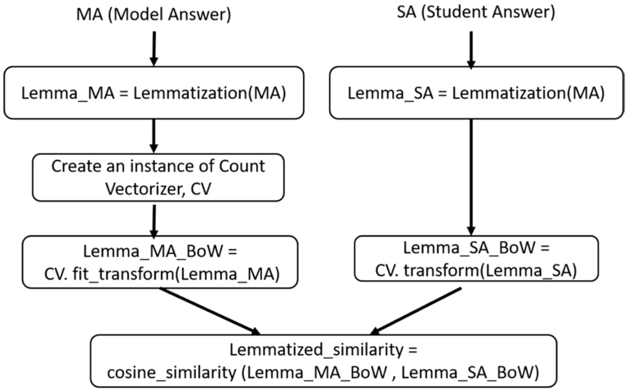
MA (Model Answer)
SA (Student Answer)
Lemma_MA = Lemmatization(MA)
Lemma_SA = Lemmatization(MA)
Create an instance of Count Vectorizer, CV
Lemma_MA_BoW = CV. fit_transform(Lemma_MA)
Lemma_SA_BoW = CV. transform(Lemma_SA)
Lemmatized_similarity = cosine_similarity (Lemma_MA_BoW , Lemma_SA_BoW)

Fig. 7 Method for lemmatized word similarity

Build a Machine Learning Model

Once the similarities are extracted, the multiclass classification machine learning models are designed by considering them as input features and the actual score given by the evaluator as the output feature. Our approach-*IDEAS*– through supervised learning- exploits classifiers (mapping function) that map model answers and student answers (input) to a label/score for each student answer. Here the ASAG problem is considered a multi-class classification problem where each score is treated as a label. For instance, the labels for the one-mark answer are 0 and 1, similarly, the labels two-mark answer are 0, 1, and 2, and so on. Here stratified K fold (5 Fold) cross-validation method of sampling is used to give priority toall classes. To build the proposed system we have

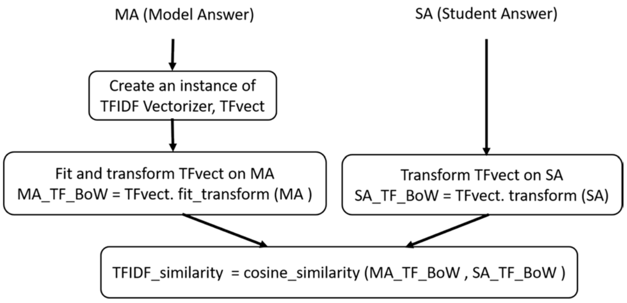
MA (Model Answer)
SA (Student Answer)
Create an instance of TFIDF Vectorizer, TFvect
Fit and transform TFvect on MA MA_TF_BoW = TFvect. fit_transform (MA)
Transform TFvect on SA SA_TF_BoW = TFvect. transform (SA)
TFIDF_similarity = cosine_similarity (MA_TF_BoW , SA_TF_BoW)

Fig. 8 Method for TFIDF similarity

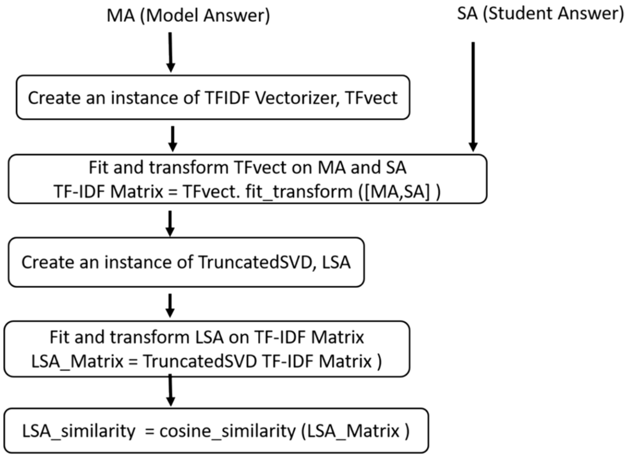
MA (Model Answer)
SA (Student Answer)
Create an instance of TFIDF Vectorizer, TFvect
Fit and transform TFvect on MA and SA
TF-IDF Matrix = TFvect. fit_transform ([MA,SA])
Create an instance of TruncatedSVD, LSA
Fit and transform LSA on TF-IDF Matrix
LSA_Matrix = TruncatedSVD TF-IDF Matrix)
LSA_similarity = cosine_similarity (LSA_Matrix)

Fig. 9 Method for LSA similarity

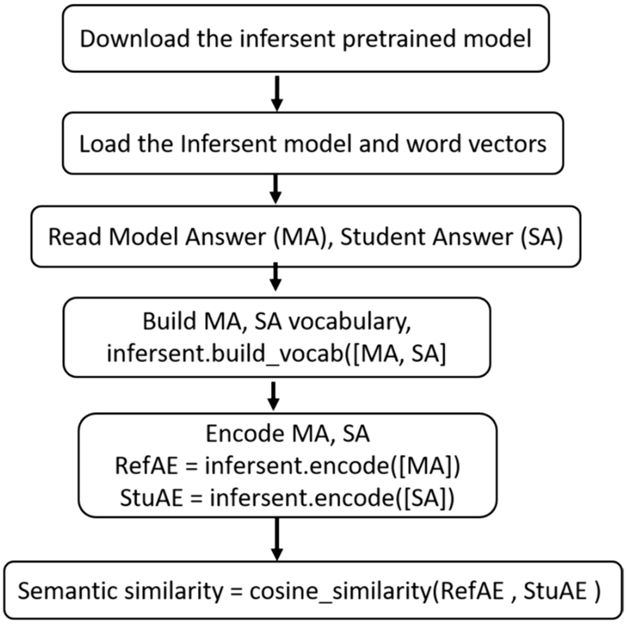
Download the infersent pretrained model
Load the Infersent model and word vectors
Read Model Answer (MA), Student Answer (SA)
Build MA, SA vocabulary,
infersent.build_vocab([MA, SA]
Encode MA, SA
RefAE = infersent.encode([MA])
StuAE = infersent.encode([SA])
Semantic similarity = cosine_similarity(RefAE , StuAE)

Fig. 10 Method for semantic similarity

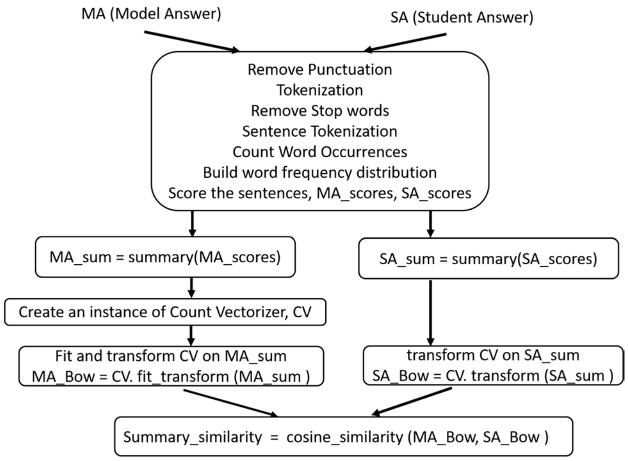
MA (Model Answer)
SA (Student Answer)
Remove Punctuation
Tokenization
Remove Stop words
Sentence Tokenization
Count Word Occurrences
Build word frequency distribution
Score the sentences, MA_scores, SA_scores
MA_sum = summary(MA_scores)
SA_sum = summary(SA_scores)
Create an instance of Count Vectorizer, CV
Fit and transform CV on MA_sum
MA_Bow = CV. fit_transform (MA_sum)
transform CV on SA_sum
SA_Bow = CV. transform (SA_sum)
Summary_similarity = cosine_similarity (MA_Bow, SA_Bow)

Fig. 11 Method for summary similarity

considered the state-of-the-art classifiers: K Nearest Neighbour (KNN) [44], Naïve Bayes (NB) [6], Support Vector Machine (SVM) [6, 45, 46], Decision tree (DT) [6], Random Forest (RF) [46, 47], XGBoost (XGB) [5]. All the modules are available in the sklearn Python library.

Model Evaluation

The following standard metrics are adopted as we model the ASAG problem as a multiclass classification task. Here TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

- Accuracy: Indicates the correctly graded answers.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{TP + TN + FP + FN} \tag{1}$$

- Precision: represents how many are positive out of all predicted positive values.

$$\text{Precision} = \frac{\text{TP}}{TP + FP} \tag{2}$$

- Recall: denotes the positive results that were predicted correctly out of all total positive values.

$$\text{Recall} = \frac{\text{TP}}{TP + FN} \tag{3}$$

- F1 score:Considers both precision and recall for effective values

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{Precision + Recall} \tag{4}$$

Identifying Inconsistency in the Evaluation and Providing Feedback to the Evaluator

While answer evaluation, inconsistency arises when similar answers are assigned different marks. Here, we validated the XGBoost model's results question-wise by comparing the inconsistency between actual marks given by the evaluator and the XGBoost model's predicted values. The second important contribution proposed is aTFIDF-based inconsistency-checking method for ASAG systems which can be directly applicable to any dataset. While checking inconsistency we focused on positive values i.e. for example in a one-mark question when an answer is supposed to get 1, but assigned a 0 mark. Here we are not focusing on answers that are supposed to get 0 marks but assigned 1 mark. These cases are given as benefit of the doubt to the students. Figure 12 illustrates the two-step proposed method for inconsistency checking using Term Frequency- Inverse Document Frequency (TF-IDF). For example, in the case of a set of one-mark answers, we get two clusters C0: which contains all zero-mark answers, and C1: which contains all one-mark answers. Our focus is mainly on positive cluster C1 because no student should be given 0 marks when he is supposed to get 1 mark.

In step 1, for each question, the clusters are created and visualized both for model-predicted marks and the actual marks using the K means clustering method. The best model predicted values were calculated using cross_val_predict(), as the models were sampled using the Stratified K Fold cross-validation method. In step 2, to identify inconsistency, the checking process involves counting the number of inconsistent answers in cluster C1 (1 mark cluster) which are labeled as 0 instead of 1 for both actual mark clusters and model-predicted mark clusters. When we compare inconsistency in actual and predicted mark clusters, experimental results discussed in "Results and Discussion" section for each question show that there is more inconsistency in actual marks i.e. evaluator marks than the model-predicted marks. By this, it is proved that the proposed IDEAS framework for ASAG achieves performance that is on par with, and occasionally surpasses, that of human evaluators. While creating Clusters using the Step 1 approach, a matrix of words is built using a well-established feature extraction method, Term Frequency- Inverse Document Frequency (TF-IDF). It helps to quantify the relevance of a word in a particular answer amongst a collection of answers.

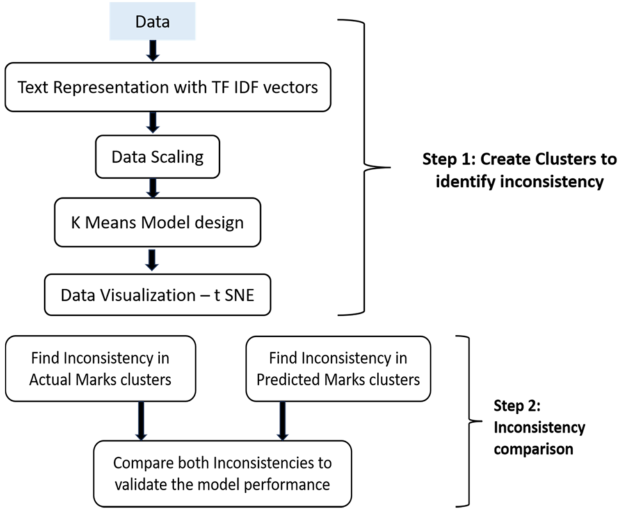
Data
Text Representation with TF IDF vectors
Data Scaling
K Means Model design
Data Visualization – t SNE
Step 1: Create Clusters to identify inconsistency
Find Inconsistency in Actual Marks clusters
Find Inconsistency in Predicted Marks clusters
Compare both Inconsistencies to validate the model performance
Step 2: Inconsistency comparison

Fig. 12 Proposed methodology for checking inconsistency using term frequency-inverse document frequency (TF-IDF)

The Matrix of TF-IDF vectors is scaled and provided as input to the K Means clustering algorithm to create clusters. Here the optimal K value is determined using the elbow method. TF-IDF vectors usually will be highly dimensional so, t-SNE (t-Distributed Stochastic Neighbor Embedding) is used for the visualization of clusters. t-SNE emphasizes capturing the relationships and local structure and handles non-linear relationships more effectively when compared to other dimensionality reduction techniques like PCA (Principal Component Analysis) and SVD (Singular Value Decomposition).

This inconsistency information is used in two ways:

- To validate the XGBoost model results concerning the actual marks. It is done by comparing the number of inconsistent answers in both actual mark clusters and XGBoost model-predicted mark clusters.
- At present, the review of the scripts after the first evaluation is done using sampling. i.e. randomly few scripts were selected and assigned to the second evaluator for review. In this process, there is very little probability of getting an inconsistent script to the second evaluator. But using our approach of inconsistency checking, for the second evaluation inconsistent scripts can be identified perfectly.

Once inconsistency is identified, it must be provided as feedback to the evaluator. The process involves extracting answers that are erroneously marked as incorrect instead of correct. A comprehensive report detailing these inconsistent responses includes the Student ID, reference answer, student answer, reference answer keywords, and student answer keywords. Additionally, it provides similarity data such as cosine similarity, statistical, semantic, and summary similarities between the student and reference answers. This feedback serves various purposes for the evaluator, including identifying patterns in their assessments and pinpointing areas prone to inconsistencies, thereby enabling corrective action. It plays a crucial role in the professional development of evaluators by aiding them in comprehending deviations from established criteria and encouraging unbiased and objective evaluations in the future.

Results and Discussion

Here, we discuss the achieved results of the proposed framework on various datasets. Initially, in "Experimental Setup" section we discuss the experimental setup used for our work, "Achieved Results of Proposed Methodology IDEAS on Various ASAG Benchmark Datasets" section represents the achieved results regarding the effectiveness of the proposed methodology on various ASAG benchmark datasets. "Achieved Results of Proposed Methodology IDEAS on Real Word Dataset" section discusses the validation of the proposed framework on areal-world dataset and "Results on the Identification of Inconsistency in the Evaluationand Provide Feedback" section shows the results regarding inconsistency checking and feedback.

Experimental Setup

Experiments were conducted on a machine equipped with an Intel(R) Core(TM) i5-1035G1 CPU @ 1.00 GHz, running at 1.19 GHz, with a 64-bit operating system and an × 64-based processor, and possessing 8 GB of RAM. The internet connection speed reaches approximately 144/72 (Mbps). The time required for data pre-processing, feature extraction, and model building for a single question from the ASAP dataset, comprising nearly 1600 answers, is under 15 min on average. Importantly, this approach doesn't rely on GPUs and can be implemented on standard machines. Therefore, IDEAS is accessible to both students and educators without extensive system modifications. Scoring can be performed on personal computers belonging to teachers or students, rather than exclusively on high-performance laboratory computers. We opted for machine learning algorithms instead of deep learning due to their promising results, contributing to a reduction in carbon footprint. Additionally, machine learning's interpretability aids in providing valuable feedback to evaluators regarding inconsistent answers. Deep learning was utilized specifically for extracting semantic features.

The entire code is implemented in Python3.6 programming language. Core libraries used include:

- NLTK: Toolkit to support text analysis
- Matplotlib: Data visualization library
- Sklearn: Library for classical machine learning support for classification, clustering
- Pandas: Library for data preparation

Achieved Results of Proposed Methodology IDEAS on Various ASAG Benchmark Datasets

The proposed framework is validated on numerous benchmark datasets including ASAP-SAS, SciEntsBank, STITA, and Texas (Mohler). Table 7 depicts the achieved results on the Texas (Mohler) dataset. As the dataset has 85 questions, even though we had implemented the proposed framework question-wise because of space limitations, in Table 7 we mentioned the average values of each classification metric. For this dataset, the Random Forest classifier gave prominent results with the highest accuracy of 76%, precision of 75%, recall of 76%, and F1 score of 75%. Table 8 shows the attained results on four questions of the SciEntsBank dataset. The KNN and Random Forest classifiers gave prominent results compared with the remaining classifiers. KNN achieved a maximum accuracy of 68%, precision of 62%, recall of 68%, and F1 Score of 64% whereas the Random forest classifier achieved maximum accuracy of 64%. precision 64%, recall 64% and F1 score of 63%. This data set also has its limitations like the allocation of marks lacking proper rubric association.

Table 9 shows the achieved results on the ASAP-SAS dataset and is not suitable for a model answer-based approach as the answer length varies from 150 to 550 words. Still, we want to test the applicability of our proposed work on this dataset. We build six multiclass classification models using stratified K fold Cross-validation for sampling. The models were built with question-wise data and then the average of individual classification metrics is calculated. The random Forest Classifier achieved the highest results when compared with other state of art classifiers with the highest accuracy of 58%, Precision of 56%, Recall of 58%, and F1 Score of 56%.

Table 10 depicts the results achieved on the STITA dataset with six questions. Here XGBoost outperforms other classifiers with a highest of 86% for all classification metrics whereas the Random Forest Classifier achieved an accuracy

Table 7 Average values of multiclass classification model results (average across 85 questions) achieved through stratified tenfold cross validation on the Texas (Mohler) dataset

Classifier	Accuracy	Precision	Recall	F1 score
KNN	0.71	0.69	0.71	0.69
Naive Bayes	0.73	0.72	0.73	0.72
SVM	0.71	0.50	0.71	0.59
Decision tree	0.70	0.71	0.70	0.71
Random forest	**0.76**	**0.75**	**0.76**	**0.75**
XGBoost	0.75	0.74	0.75	0.74

The highest scores are represented in bold

Table 8 Multiclass classification model results achieved through stratified tenfold cross validation on SciEntsBank dataset

Model	KNN				NB				SVM			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q1	0.68	0.63	0.68	0.64	0.68	0.76	0.68	0.69	0.68	0.50	0.68	0.57
Q2	0.58	0.58	0.58	0.57	0.50	0.52	0.50	0.50	0.50	0.49	0.50	0.49
Q3	0.78	0.60	0.78	0.68	0.50	0.78	0.50	0.53	0.78	0.60	0.78	0.68
Q4	0.67	0.68	0.67	0.66	0.61	0.62	0.61	0.61	0.64	0.69	0.64	0.61
Average	**0.68**	**0.62**	**0.68**	**0.64**	0.57	0.67	0.57	0.58	0.65	0.57	0.65	0.59
Model	Decision tree				Random forest				XGBoost			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q1	0.61	0.55	0.61	0.57	0.74	0.73	0.74	0.73	0.65	0.63	0.65	0.64
Q2	0.50	0.51	0.50	0.50	0.50	0.49	0.50	0.50	0.44	0.44	0.44	0.44
Q3	0.56	0.62	0.56	0.58	0.69	0.68	0.69	0.69	0.58	0.63	0.58	0.60
Q4	0.58	0.59	0.58	0.58	0.61	0.64	0.61	0.59	0.58	0.59	0.58	0.58
Average	0.56	0.57	0.56	0.56	**0.64**	**0.64**	**0.64**	**0.63**	0.56	0.57	0.56	0.57

The highest scores are represented in bold

Table 9 Multiclass classification model results achieved through stratified tenfold cross validation on the ASAP-SAS dataset

Model	KNN				NB				SVM			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Set 1	0.32	0.31	0.32	0.31	0.36	0.36	0.36	0.31	0.37	0.27	0.37	0.27
Set 2	0.31	0.29	0.31	0.29	0.36	0.38	0.36	0.32	0.37	0.27	0.37	0.27
Set 3	0.53	0.51	0.53	0.51	0.35	0.49	0.35	0.30	0.57	0.44	0.57	0.46
Set 4	0.56	0.53	0.56	0.54	0.52	0.58	0.52	0.48	0.63	0.59	0.63	0.58
Set 5	0.77	0.71	0.77	0.74	0.44	0.81	0.44	0.50	0.77	0.60	0.77	0.68
Set 6	0.83	0.74	0.83	0.78	0.41	0.84	0.41	0.53	0.84	0.74	0.84	0.77
Set 7	0.46	0.40	0.46	0.41	0.30	0.43	0.30	0.23	0.52	0.27	0.52	0.35
Set 8	0.47	0.47	0.47	0.47	0.52	0.57	0.52	0.48	0.52	0.38	0.52	0.44
Set 9	0.47	0.48	0.47	0.47	0.41	0.41	0.41	0.31	0.51	0.45	0.51	0.43
Set 10	0.58	0.58	0.58	0.58	0.53	0.58	0.53	0.49	0.61	0.53	0.61	0.55
Average	0.53	0.50	0.53	0.51	0.42	0.55	0.42	0.39	0.57	0.45	0.57	0.48
Model	Decision tree				Random forest				XGBoost			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Set 1	0.33	0.33	0.33	0.33	0.37	0.36	0.37	0.36	0.35	0.35	0.35	0.35
Set 2	0.32	0.32	0.32	0.32	0.38	0.37	0.38	0.37	0.33	0.32	0.33	0.32
Set 3	0.51	0.51	0.51	0.51	0.60	0.59	0.60	0.58	0.57	0.55	0.57	0.55
Set 4	0.54	0.55	0.54	0.54	0.60	0.59	0.60	0.59	0.59	0.58	0.59	0.58
Set 5	0.73	0.74	0.73	0.73	0.79	0.76	0.79	0.77	0.79	0.76	0.79	0.77
Set 6	0.80	0.80	0.80	0.80	0.85	0.80	0.85	0.81	0.84	0.81	0.84	0.82
Set 7	0.42	0.43	0.42	0.43	0.51	0.47	0.51	0.47	0.49	0.46	0.49	0.47
Set 8	0.47	0.46	0.47	0.47	0.55	0.52	0.55	0.53	0.53	0.52	0.53	0.52
Set 9	0.51	0.51	0.51	0.51	0.55	0.55	0.55	0.55	0.53	0.52	0.53	0.53
Set 10	0.53	0.53	0.53	0.53	0.61	0.60	0.61	0.60	0.62	0.61	0.62	0.61
Average	0.52	0.52	0.52	0.52	**0.58**	**0.56**	**0.58**	**0.56**	**0.56**	**0.55**	**0.56**	**0.55**

The highest scores are represented in bold

Table 10 Multiclass classification model results achieved through stratified tenfold cross validation on the STITA dataset

Model	KNN				NB				SVM			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q1	0.95	0.95	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.95	0.95	0.94
Q2	0.89	0.80	0.89	0.85	0.89	0.89	0.89	0.89	0.91	0.92	0.91	0.88
Q3	0.91	0.91	0.91	0.91	0.56	0.77	0.56	0.56	0.93	0.93	0.93	0.93
Q4	0.93	0.93	0.93	0.92	0.85	0.84	0.85	0.84	0.93	0.93	0.93	0.92
Q5	0.63	0.64	0.63	0.63	0.67	0.71	0.67	0.60	0.57	0.53	0.57	0.53
Q6	0.85	0.85	0.85	0.84	0.80	0.78	0.80	0.77	0.76	0.58	0.76	0.66
Average	0.86	0.85	0.86	0.85	0.79	0.82	0.79	0.77	0.84	0.81	0.84	0.81
Model	Decision tree				Random forest				XGBoost			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q1	0.91	0.90	0.91	0.91	0.93	0.92	0.93	0.92	0.93	0.92	0.93	0.92
Q2	0.86	0.86	0.86	0.86	0.91	0.90	0.91	0.90	0.95	0.94	0.95	0.95
Q3	0.82	0.83	0.82	0.83	0.86	0.86	0.86	0.86	0.91	0.91	0.91	0.91
Q4	0.85	0.85	0.85	0.85	0.87	0.86	0.87	0.85	0.87	0.86	0.87	0.86
Q5	0.69	0.70	0.69	0.69	0.67	0.66	0.67	0.66	0.69	0.68	0.69	0.68
Q6	0.80	0.80	0.80	0.80	0.80	0.78	0.80	0.78	0.83	0.83	0.83	0.83
Average	0.82	0.82	0.82	0.82	**0.84**	**0.83**	**0.84**	**0.83**	**0.86**	**0.86**	**0.86**	**0.86**

The highest scores are represented in bold

of 84%, precision of 83%, recall of 84%, and F1 score of 86%. The experimental results on various ASAG benchmark datasets prove the scalability of our proposed framework. Its performance ranges from a minimum of 58% accuracy on the ASAG-SAS dataset to a maximum of 84% for the STITA dataset. Because of the limitations mentioned in Table 2 regarding various benchmark datasets, our model answer-based approach did not suitASAG-SAS, but for other datasets, it gave decent performance.

Achieved Results of Proposed Methodology IDEAS on Real Word Dataset

To overcome the limitations of existing ASAG-SAS benchmark datasets, we collected real-world data namely IDEAS_ASAG_DATA, from a school in Andhra Pradesh, India. This dataset consists of 800 answers for 20 questions from a class VII Social science subject. Each question has a model answer, student answers, and scores given by the evaluator. This dataset has diverse questions with 6 one-mark questions, 6 two-mark questions, and 8 four-mark questions with 40 answers each. Table 11 displays the obtained outcomes from six classifiers regarding 480 responses to single-mark questions (Q1–Q6). The findings indicate that Random Forest and XGBoost classifiers perform similarly, with XGBoost achieving an accuracy of 82%, precision of 84%, recall of 81%, and an F1 Score of 81%. Similarly, Table 12 illustrates the obtained results for two-mark questions (Q7–Q12), where the Random Forest classifier achieved a respectable accuracy of 72%, precision of 69%, recall of 68%, and an F1 score of 65%. Furthermore, Table 13 showcases the outcomes for four-mark questions (Q13–Q20), highlighting XGBoost' s notable performance with an accuracy and precision of 65%, a recall of 67%, and an F1 score of 63%.

K Nearest Neighbor (KNN) did not perform well because of its sensitivity to outliers. At the same time, calculating the similarities the statistical similarity was calculated using Euclidean distance and remaining using cosine similarity. Euclidean distance ranges from [0, ∞] whereas for cosine similarity it is [− 1, 1]. Explicit scaling of values is required for KNN. As a future improvement, KNN can be modelled after scaling the values. Naive Bayes (NB), a probabilistic algorithm, assumes conditional independence among the features. But a few similarity values including word-to-word, keyword, tf-idf, and LSA may have a dependency. As a future work, can build a naïve Bayes model on a big dataset and discretized values. Support vector Machines (SVM) also can't handle outliers properly. Here we did not perform any

Table 11 Multiclass Classification model results achieved through stratified tenfold cross validation on the dataset IDEAS_ASAG_DATA one mark questions (Q1–Q6)

Model	KNN				Naïve Bayes (NB)				SVM			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q1	0.60	0.56	0.62	0.57	0.60	0.63	0.62	0.53	0.58	0.62	0.58	0.55
Q2	0.85	0.86	0.85	0.85	0.86	0.88	0.85	0.85	0.90	0.93	0.91	0.90
Q3	0.63	0.66	0.63	0.62	0.71	0.76	0.70	0.69	0.68	0.70	0.68	0.67
Q4	0.93	0.94	0.92	0.91	0.91	0.93	0.90	0.88	0.93	0.94	0.92	0.91
Q5	0.81	0.86	0.75	0.75	0.78	0.75	0.71	0.69	0.83	0.87	0.79	0.79
Q6	0.86	0.84	0.84	0.84	0.90	0.91	0.88	0.89	0.88	0.83	0.83	0.82
Average	0.78	0.79	0.77	0.75	0.79	0.81	0.78	0.76	0.80	0.81	0.78	0.77
Model	Decision tree (DT)				Random forest (RF)				XGBoost (XGB)			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q1	0.70	0.75	0.71	0.69	0.72	0.75	0.73	0.72	0.70	0.73	0.71	0.70
Q2	0.88	0.91	0.88	0.87	0.88	0.91	0.88	0.87	0.93	0.94	0.93	0.93
Q3	0.58	0.62	0.58	0.55	0.61	0.62	0.60	0.59	0.61	0.64	0.60	0.59
Q4	0.91	0.93	0.90	0.88	0.91	0.93	0.90	0.88	0.93	0.94	0.92	0.91
Q5	0.83	0.88	0.78	0.79	0.90	0.91	0.89	0.89	0.90	0.92	0.88	0.89
Q6	0.88	0.88	0.89	0.87	0.88	0.87	0.87	0.87	0.85	0.88	0.83	0.83
Average	0.80	0.83	0.79	0.78	**0.81**	**0.83**	**0.81**	**0.80**	**0.82**	**0.84**	**0.81**	**0.81**

The highest scores are represented in bold

Table 12 Multiclass Classification model results achieved through stratified tenfold cross validation on dataset IDEAS_ASAG_DATA two mark questions (Q7–Q12)

Model	KNN				Naïve Bayes (NB)				SVM			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q7	0.68	0.63	0.64	0.61	0.68	0.66	0.66	0.64	0.63	0.47	0.55	0.49
Q8	0.46	0.42	0.48	0.42	0.59	0.61	0.66	0.58	0.66	0.57	0.65	0.59
Q9	0.76	0.55	0.63	0.58	0.79	0.80	0.75	0.76	0.79	0.68	0.67	0.66
Q10	0.75	0.81	0.75	0.73	0.68	0.70	0.68	0.64	0.73	0.80	0.73	0.71
Q11	0.81	0.67	0.70	0.67	0.59	0.62	0.64	0.57	0.81	0.70	0.72	0.70
Q12	0.71	0.79	0.71	0.70	0.51	0.41	0.58	0.44	0.59	0.70	0.62	0.59
Average	0.70	0.64	0.65	0.62	0.64	0.63	0.66	0.60	0.70	0.65	0.66	0.62
Model	Decision tree (DT)				Random forest (RF)				XGBoost (XGB)			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q7	0.59	0.58	0.57	0.54	0.74	0.67	0.68	0.65	0.69	0.71	0.65	0.63
Q8	0.78	0.81	0.79	0.78	0.73	0.79	0.74	0.73	0.81	0.82	0.81	0.79
Q9	0.76	0.71	0.69	0.69	0.76	0.68	0.67	0.67	0.74	0.70	0.67	0.67
Q10	0.66	0.55	0.62	0.55	0.64	0.61	0.62	0.56	0.64	0.63	0.61	0.59
Q11	0.76	0.60	0.69	0.62	0.81	0.67	0.74	0.69	0.78	0.67	0.73	0.67
Q12	0.66	0.71	0.65	0.64	0.63	0.70	0.65	0.63	0.61	0.70	0.64	0.62
Average	0.70	0.66	0.67	0.64	**0.72**	**0.69**	**0.68**	**0.65**	**0.71**	**0.70**	**0.69**	**0.66**

The highest scores are represented in bold

hyperparameter tuning. These things have to be addressed in the future. The performance of Random Forest (RF), and XGBoost (XGB) models is good when compared to other classifiers. Because they can capture the non-linear relationship between the independent and dependent features, don't require feature scaling, are not sensitive to

Table 13 Multiclass Classification model results achieved through stratified tenfold cross validation on the dataset IDEAS_ASAG_DATA four mark questions (Q13–Q20)

Model	KNN				Naïve Bayes (NB)				SVM			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q13	0.44	0.28	0.39	0.32	0.58	0.51	0.59	0.51	0.54	0.34	0.49	0.39
Q14	0.34	0.26	0.36	0.29	0.41	0.40	0.44	0.39	0.51	0.33	0.48	0.39
Q15	0.73	0.73	0.74	0.69	0.78	0.71	0.80	0.73	0.59	0.56	0.62	0.56
Q16	0.63	0.49	0.59	0.51	0.68	0.61	0.67	0.60	0.73	0.67	0.69	0.65
Q17	0.49	0.48	0.48	0.45	0.49	0.40	0.48	0.43	0.59	0.48	0.58	0.50
Q18	0.78	0.81	0.80	0.77	0.66	0.65	0.72	0.64	0.81	0.87	0.84	0.81
Q19	0.52	0.51	0.54	0.48	0.49	0.48	0.52	0.46	0.56	0.38	0.50	0.40
Q20	0.70	0.78	0.77	0.75	0.65	0.72	0.71	0.69	0.73	0.68	0.74	0.68
Average	0.58	0.54	0.58	0.53	0.59	0.56	0.61	0.56	0.63	0.54	0.62	0.55
Model	Decision tree (DT)				Random forest (RF)				XGBoost (XGB)			
Question	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Q13	0.56	0.52	0.57	0.51	0.56	0.44	0.53	0.46	0.51	0.46	0.48	0.45
Q14	0.54	0.49	0.56	0.49	0.51	0.48	0.54	0.48	0.59	0.57	0.62	0.57
Q15	0.59	0.55	0.60	0.55	0.64	0.60	0.64	0.60	0.63	0.61	0.64	0.60
Q16	0.78	0.75	0.79	0.75	0.79	0.75	0.77	0.73	0.90	0.89	0.91	0.88
Q17	0.61	0.55	0.60	0.54	0.56	0.54	0.54	0.53	0.58	0.56	0.56	0.54
Q18	0.74	0.68	0.74	0.67	0.81	0.86	0.84	0.81	0.71	0.74	0.74	0.69
Q19	0.51	0.44	0.50	0.45	0.59	0.56	0.62	0.55	0.66	0.67	0.68	0.63
Q20	0.69	0.80	0.76	0.75	0.65	0.74	0.71	0.71	0.64	0.74	0.70	0.71
Average	0.63	0.60	0.64	0.59	**0.64**	**0.62**	**0.65**	**0.61**	**0.65**	**0.65**	**0.67**	**0.63**

The highest scores are represented in bold

the outliers, and the splits are performed based on the percentile threshold instead of raw values.

Results on the Identification of Inconsistency in the Evaluation and Provide Feedback

In this study, we explore the results of identifying inconsistencies in evaluations and provide evaluators with feedback on how to resolve these discrepancies. Inconsistencies are pinpointed on a question-by-question basis within each dataset. Initially, responses to specific questions are represented using TFIDF vectors K Means clustering is then employed on these vectors to generate two clusters: correct and incorrect, which are visualized using t-SNE visualization. An answer is deemed inconsistent if it is classified as incorrect within a cluster labeled as correct. In the case of a one-mark answer, inconsistency occurs in a situation where the answer is supposed to get 1 but has got 0. We evaluated inconsistency in the actual marks and also in the XGBoost classifier predicted marks for each question.

Initially, the proposed method for checking inconsistencies and providing feedback is applied to our novel dataset IDEAS_ASAG_DATA. The t-SNE visualization of clusters is created for all 20 questions. Here, we are showing the sample clusters created for six questions, ranging from question 1 to question 6 of the IDEAS_ASAG_DATA dataset in Figs. 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 and 24. In these figures, the color green (0) denotes the incorrect answer cluster, while orange (1) represents the correct answer cluster. The numerical labels (0, 1, 2, 3, etc.) associated with each data point indicate the student ID.

Regarding question 1, Fig. 13 reveals the presence of five inconsistent responses for actual marks whereas Fig. 14

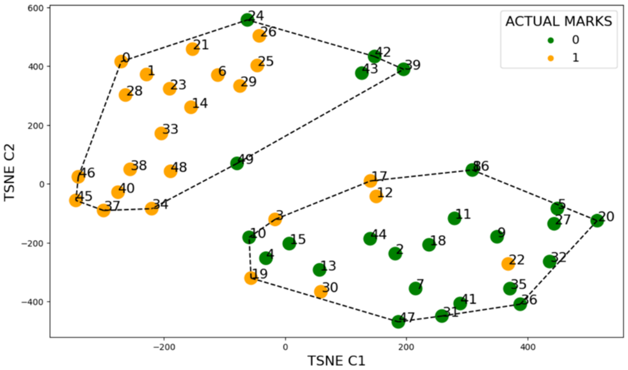

Fig. 13 t-SNE visualization of Actual Marks for Question 1 answers of dataset IDEAS_ASAG_DATA

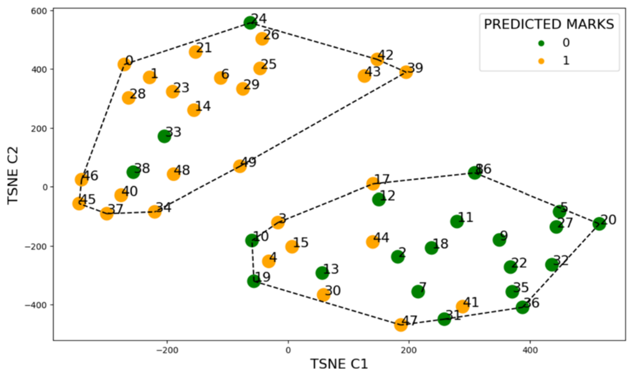

Fig. 14 t-SNE visualization of XGBoost model Predicted marks for Question 1 answers of dataset IDEAS_ASAG_DATA

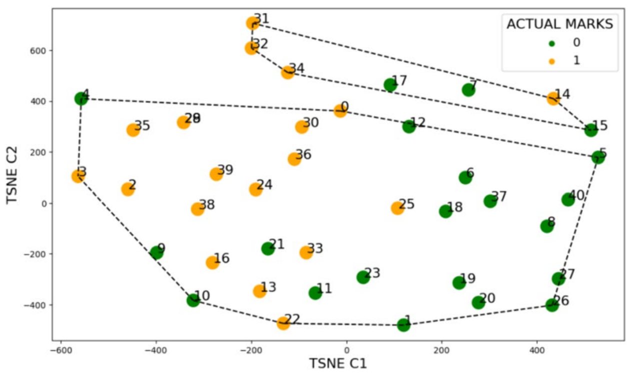

Fig. 17 t-SNE visualization of Actual marks for question 3 of dataset IDEAS_ASAG_DATA

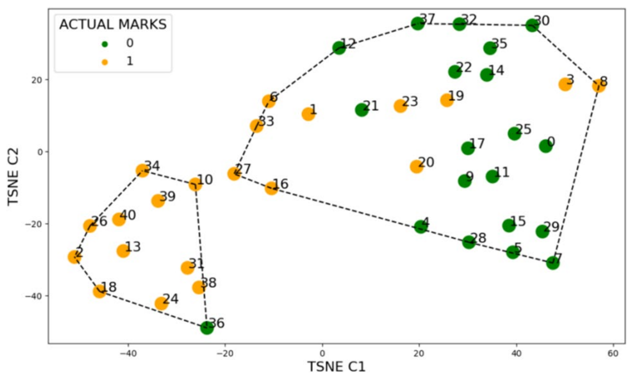

Fig. 15 t-SNE visualization of Actual Marks for Question 2 answers of dataset IDEAS_ASAG_DATA

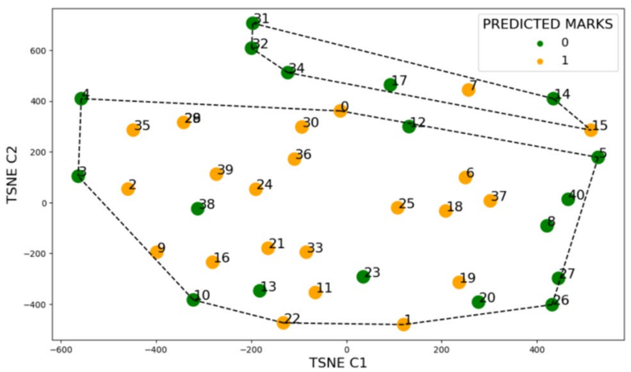

Fig. 18 t-SNE visualization of XGBoost model Predicted marks for Question 3 answers of dataset IDEAS_ASAG_DATA

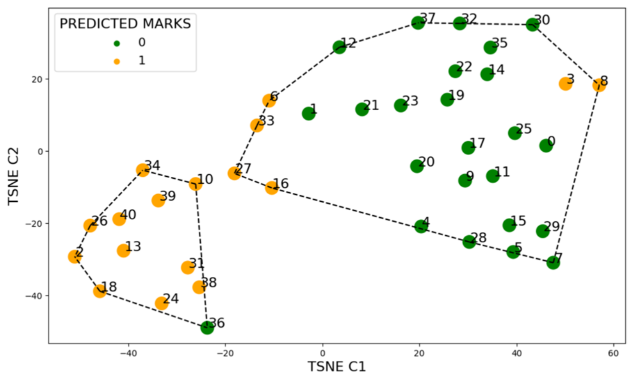

Fig. 16 t-SNE visualization of XGBoost model Predicted marks for Question 2 answers of dataset IDEAS_ASAG_DATA

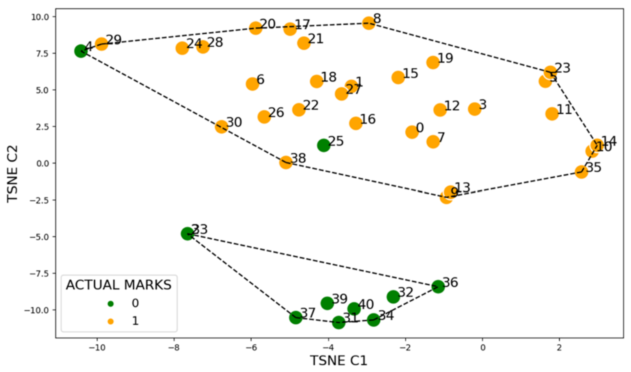

Fig. 19 t-SNE visualization of actual marks for Question 4 answers of dataset IDEAS_ASAG_DATA

shows only three inconsistent responses for predicted marks. Similarly for question 2 Figs. 15 and 16 depict one inconsistent answer for both actual marks and predicted marks. Figure 17 indicates 18 inconsistent answers for question 3 actual marks whereas Fig. 18 depicts 13 answers as inconsistent for predicted marks. Figure 19 displays 2

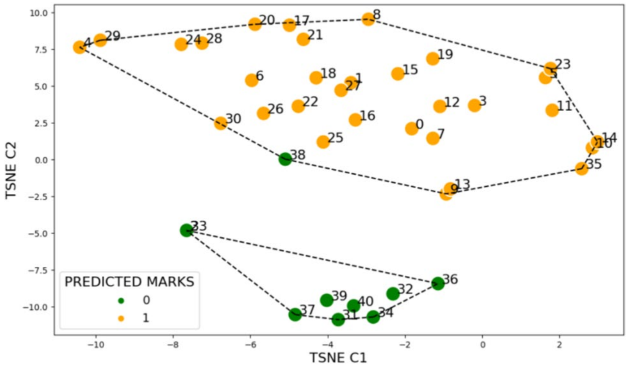

Fig. 20 t-SNE visualization of XGBoost model Predicted marks for Question 4 answers of dataset IDEAS_ASAG_DATA

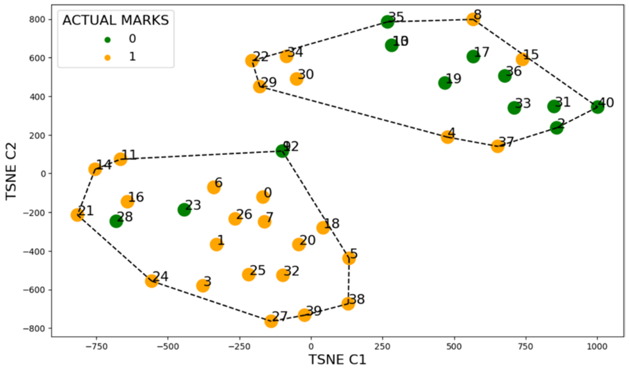

Fig. 21 t-SNE visualization of actual marks for Question 5 answers of dataset IDEAS_ASAG_DATA

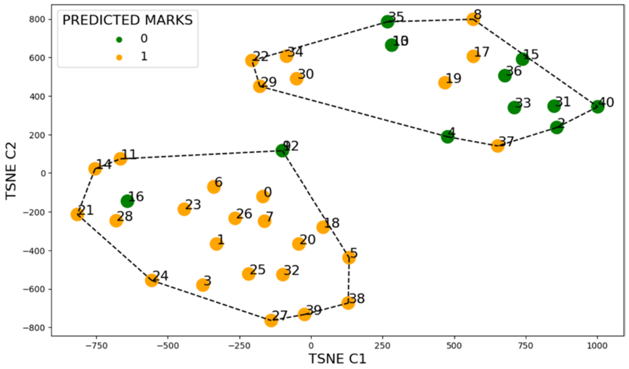

Fig. 22 t-SNE visualization of XGBoost model Predicted marks for Question 5 answers of dataset IDEAS_ASAG_DATA

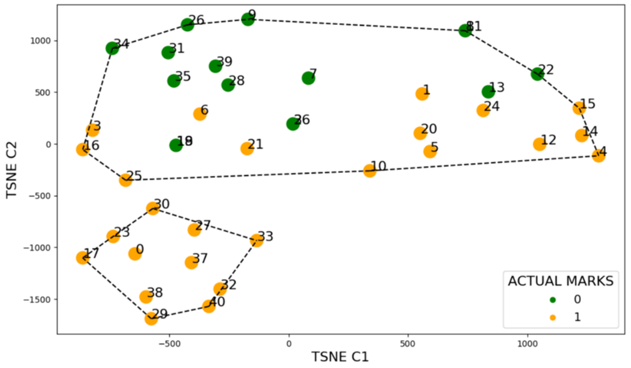

Fig. 23 t-SNE visualization of actual marks for Question 6 answers of dataset IDEAS_ASAG_DATA

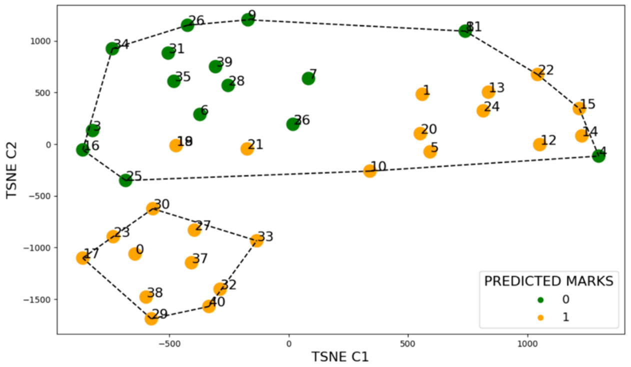

Fig. 24 t-SNE visualization of XGBoost model Predicted marks for Question 6 answers of dataset IDEAS_ASAG_DATA

inconsistencies for question 4 actual marks there as it is only two for predicted marks as shown in Fig. 20. Additionally, Fig. 21 depicts 4 inconsistent responses for question 5, while Fig. 22 establishes that there are three inconsistent answers for predicted marks. Regarding question 6, Figs. 23 and 24 show zero inconsistency in both actual and predicted marks. The above visualizations show the clusters of 6 questions for both actual and predicted marks question-wise. As we are more interested in finding the inconsistency in the positive cluster i.e. one mark cluster, when the positive clusters of both actual and predicted marks are compared, experimental results show more inconsistency in actual mark clusters than the predicted mark clusters.

Similarly, the proposed approach is applied to the SciEntsBank and STITA datasets too. The comparison between the number of inconsistent answers in both the actual and predicted marks for the datasets namely IDEAS_ASAG_DATA, SciEntsBank, and STITA is shown in Figs. 25, 26, and 27. For all the datasets, experimental results show that there is less inconsistency in model-predicted values compared to actual marks given by the evaluator. This demonstrates that the model's performance is comparable to, and in certain instances, surpasses that of the evaluator.

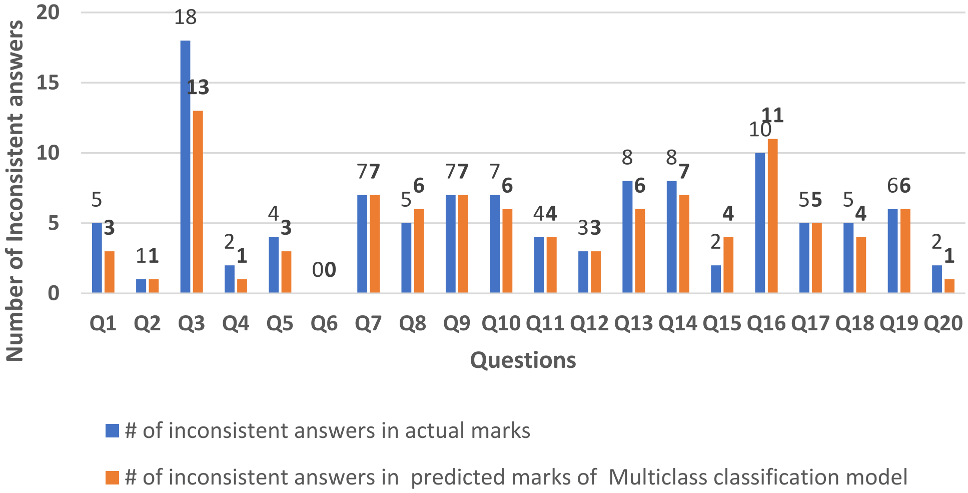

Fig. 25 Comparison between the number of inconsistent answers in actual and predicted scores for the dataset IDEAS_ASAG_DATA (20 Questions)

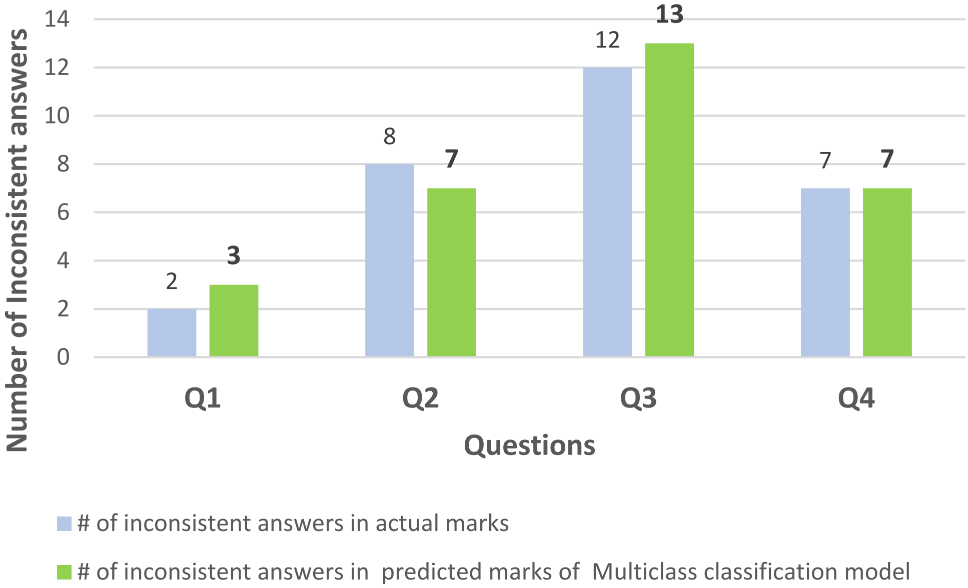

Fig. 26 Comparison between the number of inconsistent answers in actual and predicted scores for the dataset SciEnts-Bank (4 Questions)

Once the inconsistency in actual marks is identified, it is provided to the evaluator as feedback. This feedback comprises the matched (green) and unmatched (red) keywords of both model and student answers. It also includes similarity data indicating the percentage of similarity between the student's answer and the model answer, both statistically and semantically, as well as at a summary level. It also presents the actual marks and the model's predicted marks for each specific answer. These details regarding inconsistently evaluated answers were furnished to the evaluator to enable thorough review and informed decision-making. The table shows the sample feedback given to an inconsistently evaluated answer for question 2 in IDEAS_ASAG_DATA. Table 14 depicts the sample feedback concerning an inconsistent answer for Question 2 of the IDEAS_ASAG dataset.

Conclusion and Future Enhancement

Automated Short Answer Grading (ASAG), an increasingly prominent area within natural language comprehension, serves as a pivotal focus of research in the expansive realm of learning analytics. ASAG solutions are tailored to mitigate the challenges faced by educators, aiming to streamline the teaching process. The proposed ASAG framework, Intelligent Descriptive answer E-Assessment System (IDEAS) discourses three primary tasks of the evaluation process namely (i) Automatic grading of short descriptive answers in english, (ii) inconsistency identification in the evaluation, and (iii) providing detailed feedback to the evaluator regarding inconsistent answers. We conceptualized the ASAGtask as a multiclass classification problem instead of

Fig. 27 Comparison between the number of inconsistent answers in actual and predicted scores for the dataset STITA (6 Questions)

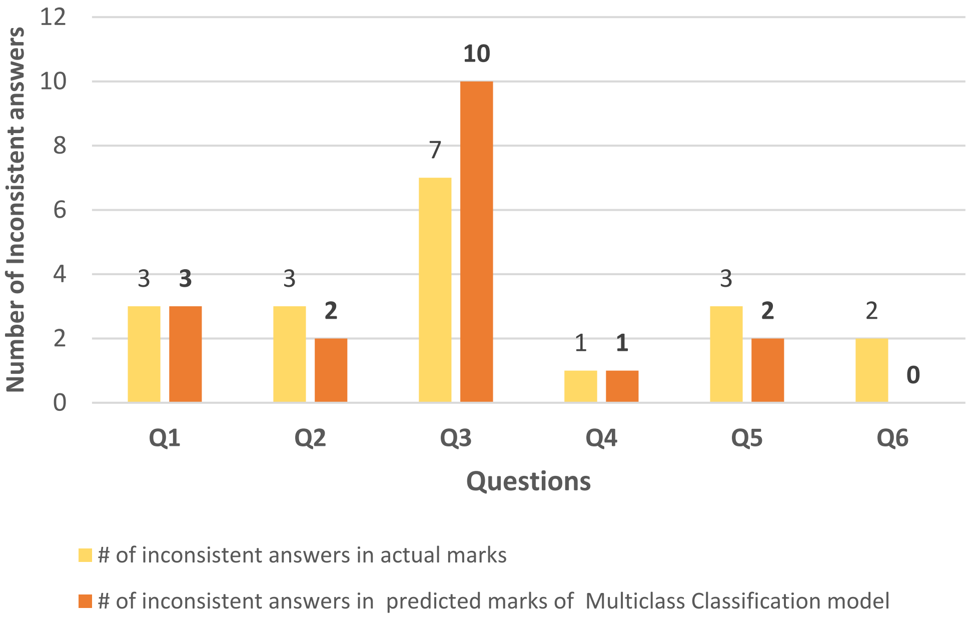

Table 14 Feedback regarding the inconsistent answer for Question 2 of the dataset IDEAS_ASAG_DATA

STDID	Model answer keywords	Student answer keywords	Cosine	Statistical	Semantic	Summary	Actual marks	Predicted marks
36	**matches**, **one**, *one*, **many**, **1.save**, **climate**, **keep**, **rare**, **we**, **million**, **care**, *tree*, **3**, *match*, **forests**, **save**, **2.**, **now**, **they**, **future.4.**, **save**, **destroy**, *make*	*tree, make, match, one*	0.361	55.145	0.92	0.44	0	0

The highest scores are represented in bold

regression or binary classification. The model answer-based approach worked well for all ASAG benchmark datasets and on the novel dataset IDEAS_ASAG_DATA. Proposed a novel method for inconsistency identification and comprehensive feedback to the evaluator regarding inconsistency in evaluation using unsupervised learning techniques. The experimental findings demonstrated a reduced level of inconsistency in the model-predicted scores compared to the actual marks provided by the evaluator, indicating the superior accuracy of the proposed approach, which occasionally exceeds human evaluation standards.

Despite the promising outcomes of IDEAS, the proposed framework has a few intrinsic limitations including integrating our framework with syntactic features and implementing newer datasets on explicit subjects. Currently, IDEAS is not proposed to provide a final score for students' responses but as a tool for supporting both students and teachers. In the future, we have planned to incorporate a feedback mechanism that aids the students in identifying their mistakes and correcting them. In addition, we are planning for a speech analytics model that takes answers in the form of speech/audio and scores accordingly. In addition to that can implement Explainable AI to justify why the model has given that score and provide it as feedback to the student. Cheating issues must be addressed regarding the automatic scoring systems. Our definitive goal for the future is to embed IDEAS in a web application with an in-built graphical user interface.

Acknowledgements The authors acknowledge the support from REVA University for the facilities provided to carry out the research.

Author Contributions P. Sree Lakshmi: Conceptualization, Methodology, Visualization, Writing-original draft. Simha J. B.: Supervision, Validation, Rajeev Ranjan: Writing-review and editing.

Funding No funding available.

Data Availability Data is provided only upon request.

Declarations

Conflict of interest All authors declare that there is no conflict of interest.

References

1. Mohler M, Mihalcea R. Text-to-text semantic similarity for automatic short answer grading. In: Proceedings of the 12th conference of the European chapter of the ACL (EACL 2009). Athens, Greece: Association for Computational Linguistics; 2009. p. 567–75.
2. Burrows S, Gurevych I, Stein B. The eras and trends of automatic short answer grading. Int J Artif Intell Educ. 2015. https://doi.org/10.1007/s40593-014-0026-8.
3. Sree Lakshm P, Kavitha. Intelligent scoring systems for descriptive answers—a review. Test Eng Manag. 2020;83:3595–600.
4. Lun J, Zhu J, Tang Y, Yang M. Multiple data augmentation strategies for improving performance on automatic short answer scoring, vol. 20; 2020.
5. Rajagede RA, Hastuti RP. Stacking neural network models for automatic short answer scoring. IOP Conf Ser Mater Sci Eng. 2021;1077:012013. https://doi.org/10.1088/1757-899x/1077/1/012013.
6. Zhang Y, Lin C, Chi M. Going deeper: automatic short-answer grading by combining student and question models. User Model User Adapt Interact. 2020;30:51–80. https://doi.org/10.1007/s11257-019-09251-6.
7. Siddiqi R, Harrison CJ, Siddiqi R. Improving teaching and learning through automated short-answer marking. IEEE Trans Learn Technol. 2010;3:237–49. https://doi.org/10.1109/TLT.2010.4.
8. Saha SK, Gupta R. Adopting computer-assisted assessment in evaluation of handwritten answer books: an experimental study. Edu Inform Technol. 2020;25:4845–60. https://doi.org/10.1007/s10639-020-10192-6.
9. Saha SK, Dhawaleswar Rao CH. Development of a practical system for computerized evaluation of descriptive answers of middle school level students. Interact Learn Environ. 2022;30:215–28. https://doi.org/10.1080/10494820.2019.1651743.
10. Bahel V, Thomas A. Text similarity analysis for evaluation of descriptive answers; 2021. arXiv:2105.02935.
11. Jamil F, Hameed IA. Toward intelligent open-ended questions evaluation based on predictive optimization. Expert Syst Appl. 2023;231:120640. https://doi.org/10.1016/J.ESWA.2023.120640.
12. Shukla A, Chaudhary BD. A strategy for detection of inconsistency in evaluation of essay type answers. Educ Inform Technol. 2014;19:899–912. https://doi.org/10.1007/s10639-013-9264-x.
13. Rico-Juan JR, Gallego A-J, Calvo-Zaragoza J. Automatic detection of inconsistencies between numerical scores and textual feedback in peer-assessment processes with machine learning. Comput Educ. 2019;140:103609. https://doi.org/10.1016/j.compedu.2019.103609.
14. Bernius JP, Krusche S, Bruegge B. Machine learning based feedback on textual student answers in large courses. Comput Educ Artif Intell. 2022. https://doi.org/10.1016/j.caeai.2022.100081.
15. Vwen YL, Luco AAC, Tan SC. A human-centric automated essay scoring and feedback system for the development of ethical reasoning. Technol Soc. 2023;26:147–59. https://doi.org/10.2307/48707973.
16. Hao Q, Smith DH IV, Ding L, Ko A, Ottaway C, Wilson J, Arakawa KH, Turcan A, Poehlman T, Greer T. Towards understanding the effective design of automated formative feedback for programming assignments. Comput Sci Educ. 2022;32:105–27. https://doi.org/10.1080/08993408.2020.1860408.
17. Wang Z, Lan AS, Waters AE, Grimaldi P, Baraniuk RG. A meta-learning augmented bidirectional transformer model for automatic short answer grading. In: Proceedings of the 12th international conference on educational data mining (EDM 2019); 2019.
18. Zhu H, Togo R, Ogawa T, Haseyama M. Prompt-based personalized federated learning for medical visual question answering; 2024. arXiv:2402.09677.
19. del Gobbo E, Guarino A, Cafarelli B, Grilli L. GradeAid: a framework for automatic short answers grading in educational contexts—design, implementation and evaluation. Knowl Inform Syst. 2023;65:4295–334. https://doi.org/10.1007/s10115-023-01892-9.
20. Kumar Y, Aggarwal S, Mahata D, Shah RR, Kumaraguru P, Zimmermann R. Get IT scored using AutoSAS—an automated system for scoring short answers. Proc AAAI Conf Artif Intell. 2019;33:9662–9. https://doi.org/10.1609/aaai.v33i01.33019662.
21. Wang T, Inoue N, Ouchi H, Mizumoto T, Inui K. Inject rubrics into short answer grading system; 2019. p. 175–82. https://doi.org/10.18653/v1/P17.
22. Riordan B, Horbach A, Cahill A, Zesch T, Lee CM. Investigating neural architectures for short answer scoring. In: EMNLP 2017-12th workshop on innovative use of NLP for building educational applications, BEA 2017—proceedings of the workshop. Association for Computational Linguistics (ACL); 2017. p. 159–68. https://doi.org/10.18653/v1/w17-5017.
23. Gaddipati SK, Nair D, Plöger PG. Comparative evaluation of pretrained transfer learning models on automatic short answer grading; 2020.
24. Sultan MA, Salazar C, Sumner T. Fast and easy short answer grading with high accuracy. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego, California. Association for Computational Linguistics; 2016. p. 1070–5.
25. Callear D, Jerrams-Smith J, Soh V. CAA of short non-MCQ answers. In: Proceedings of the 5th CAA conference, Loughborough: Loughborough University; 2001.
26. Leacock C, Chodorow M. C-rater: automated scoring of short-answer questions. Comput Hum. 2003;37:37.
27. Siddiqi Ra, Harrison C. A systematic approach to the automated marking of short-answer questions. In: IEEE INMIC 2008: 12th IEEE international multitopic conference—conference proceedings; 2008. p. 329–32. https://doi.org/10.1109/INMIC.2008.4777758.
28. Mitchell T, Russell T. Towards robust computerised marking of free-text responses understanding evolution and inheritance in the national curriculum KS2-3 view project GEMSTONE technology: optimisation of global supply chain view project; 2002.
29. Alfonseca E, Pérez D. Automatic assessment of open ended questions with a Bleu-inspired algorithm and shallow NLP. In: Vicedo JL, Martínez-Barco P, Muńoz R, Saiz Noeda M, editors. Advances in natural language processing. EsTAL 2004. Lecture notes in computer science(), vol. 3230. Berlin: Springer; 2004.
30. Condor A. Exploring automatic short answer grading as a tool to assist in human rating. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 12164 LNAI:74–79. London: Springer; 2020. https://doi.org/10.1007/978-3-030-52240-7_14.
31. Hou WJ, Tsao JH. Automatic assessment of students' free-text answers with different levels. Int J Artif Intell Tools. 2011;20:327–47. https://doi.org/10.1142/S0218213011000188.
32. del Gobbo E, Guarino A, Cafarelli B, Grilli L. GradeAid: a framework for automatic short answers grading in educational contexts—design, implementation and evaluation. In: Knowledge and information systems. Springer Science and Business Media Deutschland GmbH; 2023. https://doi.org/10.1007/s10115-023-01892-9.

33. Gomaa WH, Fahmy AA. Ans2vec: a scoring system for short answers. Adv Intell Syst Comput. 2020;921:586–95. https://doi.org/10.1007/978-3-030-14118-9_59.
34. Prabhudesai A, Duong TNB. Automatic short answer grading using Siamese bidirectional LSTM based regression. In: 2019 IEEE international conference on engineering, technology and education (TALE). IEEE; 2019. p. 1–6. https://doi.org/10.1109/TALE48000.2019.9226026.
35. Chimingyang H. An automatic system for essay questions scoring based on LSTM and word embedding. In: Proceedings—2020 5th international conference on information science, computer technology and transportation, ISCTT. Institute of Electrical and Electronics Engineers Inc; 2020. p. 355–64. https://doi.org/10.1109/ISCTT51595.2020.00068.
36. Tulu CN, Ozkaya O, Orhan U. Automatic short answer grading with SemSpace sense vectors and MaLSTM. IEEE Access. 2021;9:19270–80. https://doi.org/10.1109/ACCESS.2021.3054346.
37. Zichao Y, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego, California. Association for Computational Linguistics; 2016. p. 1480–9.
38. Cai C. Automatic essay scoring with recurrent neural network. In: Proceedings of the 3rd international conference on high performance compilation, computing and communications. New York, NY, USA: ACM; 2019. p. 1–7. https://doi.org/10.1145/3318265.3318296.
39. Sung C, Dhamecha T, Saha S, Ma T, Reddy V, Arora R. Pre-training BERT on domain resources for short answer grading. 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). Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 607074. https://doi.org/10.18653/v1/D19-1628.
40. Ghavidel HA, Zouaq A, Desmarais MC. Using BERT and XLNET for the automatic short answer grading task. In: CSEDU 2020—proceedings of the 12th international conference on computer supported education, vol. 1. SciTePress; 2020. p. 58–67. https://doi.org/10.5220/0009422400580067.
41. Wiratmo A, Fatichah C. Assessment of Indonesian short essay using transfer learning Siamese dependency tree-LSTM. In: ICICoS 2020—proceeding: 4th international conference on informatics and computational sciences. Institute of Electrical and Electronics Engineers Inc; 2020. https://doi.org/10.1109/ICICoS51170.2020.9299044.
42. Chen Z, Zhou Y. Research on automatic essay scoring of composition based on CNN and OR. In: 2019 2nd international conference on artificial intelligence and big data (ICAIBD). IEEE; 2019. p. 13–8. https://doi.org/10.1109/ICAIBD.2019.8837007.
43. Lakshmi S. Document representation methods for text categorization: a review. International Journal of Scientific Research in Computer Science Applications and Management Studies IJSRCSAMS, vol. 7; 2018.
44. Stacey B, Meurers D. Diagnosing meaning errors in short answers to reading comprehension questions. In: Proceedings of the 3rd ACL Workshop on Innovative Use of NLP for Building Educational Applications; 2008. p. 107–14.
45. Hou W-J, Tsao J-H, Li S-Y, Chen L. LNAI 6096—automatic assessment of students' free-text answers with support vector machines. IEA/AIE 2010, Part I, LNAI 6096, © Springer, Berlin; 2010.
46. Elnaka A, Nael O, Afifi H, Sharaf N. AraScore: investigating response-based Arabic short answer scoring. Proc CIRP. 2021;189:282–91. https://doi.org/10.1016/j.procs.2021.05.091.
47. Saha S, Dhamecha TI, Marvaniya S, Sindhgatta R, Sengupta B. Sentence level or token level features for automatic short answer grading? Use both. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 10947 LNAI. London: Springer; 2018. p. 503–17. https://doi.org/10.1007/978-3-319-93843-1_37.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.