Journal of Information Systems Engineering and Management
2025, 10(31s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Optimization of Answer Grading System with Deep Learning Algorithm & Optimizer

Mr. Rudragouda G. Patil[1*], Dr. Mahantesh N. Birje[2], Dr. Manisha T. Tapale[3], Dr. Nagaraj V. Dharwadkar[4]
[1*]*Research Scholar, Department of Computer Science and Engineering, Visvesvaraya Technological University, Belagavi, KA, India.*
*1*Faculty, Department of Computer Engineering, Marathwada, Mitramandal College of Engineering, Pune, MH, India.*
[1*]*Email: rudra.g.patil@gmail.com*
[2]*Professor, Department of Computer Science and Engineering, Visvesvaraya Technological University, Belagavi, India.*
[3]*Associate Professor, Department of Computer Science and Engineering, KLE MSSCET, Belagavi, India.*
[4]*Assistant Professor, Department of Computer Science, Central University of Karnataka, Kadaganchi, KA, India.*

ARTICLE INFO

Received: 28 Dec 2024

Revised: 18 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Introduction: Evaluating student answers is crucial to educational assessment, significantly impacting learning outcomes and academic success. Traditional grading methods often exhibit inconsistencies and subjectivity, leading to variations in evaluators scoring similar answers. This study proposes a comprehensive methodology employing advanced computational techniques to enhance the consistency and objectivity of grading student answers. The research primarily aims to design a standardized grading framework that reduces variability in scoring due to human subjectivity. The study explores various deep learning algorithms to automate grade prediction and recommendation, thereby streamlining the grading process and providing data-driven insights for evaluators. The proposed framework addresses the challenges of traditional grading by leveraging "Natural Language Processing" (NLP) techniques to analyze and assess answers given by students. The study utilizes a Kaggle dataset of manually graded essays for training and testing. The performance of models is compared using measures such as "Quadratic Weighted Kappa" (QWK). Results indicate that the LSTM model with Generative Pre-Training Transformer-2 tokenizer, optimized using the "Grey Wolf Optimizer" (GWO), outperforms other models, including BERT with its tokenizer. The GPT-2-LSTM model demonstrates the highest QWK, accuracy, and stability, with the lowest MSE and variance, indicating superior performance in automated grading. The research findings suggest that the proposed work can effectively enhance the consistency and objectivity of student answer grading, reducing human bias and improving the overall assessment process.

Keywords: Bidirectional Encoder Representations from Transformers_1, Long Short-Term Memory_2, Generative Pre-training Transformer_3, Grey Wolf Optimizer_4, Quadratic Weighted Kappa_5

1. INTRODUCTION

Assessing students' written answers is crucial to educational assessment, affecting learning results and achievement. The regular grading approaches have drawn out some drawbacks, including that different evaluators can grade similar answers (semantically similar) in other ways [1]. In response to these difficulties, we introduced a conceptual framework that can be implemented as a complete solution for automated grading of various forms of student answers with improved reliability and objectivity through Natural Language Processing (NLP). This paper therefore mainly aimed at developing a structure that will facilitate a means of checking bodies with the view of standardizing the grading of similar answers written by multiple students. According to this framework, difficulties that arise from scoring, which stems from subjectivity, would be reduced since there would be universally set grading criteria. The use of deep learning algorithms including "Convolutional Neural Networks" (CNN), "Long Short-Term Memory" (LSTM) networks, and "Generative Pre-trained Transformers" (GPT), to predict and recommend grades, streamline the grading process, and provide evaluators with data-driven insights [2].In education, the outcome of teaching is assessed through student knowledge, typically done by conducting tests and other exercises. Written assessments, including long and short-answer questions, significantly evaluate students' critical thinking and reasoning abilities.

Traditionally, grading has been performed by human evaluators, which can introduce biases and subjectivity, especially in free-text answers. The manual grading process is time-consuming and may also be affected by the individual pressures the graders face. To address these issues, technological advancements have led to the development of automated assessment systems, aiming for fairness, speeding up the evaluation process, and reducing the time and effort required. Certain questions, such as Multiple-Choice Questions (MCQ) or Fill-in-the-Blank, are straightforward and simple for automated systems to grade. At the same time, free-text answers and essay-kind answers present more challenges due to their complexity and criticality shown in answers [3].

Automatic Essay Scoring systems have been developed to assess longer responses, focusing on spelling, grammar, and sentence coherence. Unlike essay grading, where sentence structure is often analyzed, short answer grading primarily considers content accuracy relative to the model answer [4]. The approach is recognized as a complex task within NLP [5], particularly because students may correctly express an answer using words different from those in the model answer. Various methods for measuring textual similarity highlight the shortfalls of simple word overlap techniques in capturing semantic meaning. Text-similarity measures and examines their application in automatically grading short answer questions, acknowledging that correct answers may not always align word-for-word with the model answers [6]. The aims of the research presented in this research work are as follows:

- Develop a novel framework for Consistent Grading of Student Answers: This objective focuses on establishing a standardized grading system to ensure uniformity in evaluating similar responses from multiple students.
- Automate Grade Prediction and Suggestion Using Machine Learning and Deep Learning Algorithms: This objective involves utilizing machine learning algorithms for the prediction and recommendation of grades. The effectiveness of these algorithms will be assessed by comparing the scores generated through different methodologies, requiring a consistent grading framework for model training and evaluation.

2. LITERATURE SURVEY

The survey highlights the progression of artificial intelligence techniques used in various applications, paying attention to recent developments in machine learning (ML) algorithms and deep learning (DL) algorithms [7]. Early methods in automated grading systems relied on text similarity features and traditional ML models, which used alignment and semantic vector similarity to create a quick and accurate grading system. Numerous techniques have been adopted and utilized for automatic scoring. In recent years, ML and DL approaches have developed the most robust models for automatically grading short answer questions [8].

2.1. Related works

DL methodologies for educational assessment, focus on the application of neural networks in grading short-answer responses. In 2024 the author's [9] research illustrated that DL models can learn intricate patterns within student answers, thereby offering precise and dependable grade predictions. Building on this foundation, conducted a comparative analysis of traditional machine learning (ML) techniques and DL models, ultimately determining that DL models exhibit superior accuracy and robustness. The author, [10] reviewed the latest advancements in automated essay scoring and illustrates the integration of clustering and machine learning techniques in grading systems. They highlight the effective combination of clustering algorithms for grouping similar responses with ML models for grading, demonstrating how this synergy improves the efficiency and accuracy of automated grading systems. Hybrid models that integrate clustering and predictive algorithms for grading short answers [11]. Their results indicate that these combined approaches enhance grading consistency and offer insights into frequent student errors, thereby supporting the development of effective instructional strategies [12] produced a Root Mean Square Error of 0.057.

[13] [14] developed a "deep descriptive answer scoring model" using DL & NLP techniques. This system, included an embedded layer, LSTM layer, dropout layer, and dense layer as a model. During preprocessing, text relevant important features from student responses are extracted and transformed into "GloVe vector" indices. The embedding layer then transforms these indices into "GloVe vectors". The LSTM layer, a type of recurrent neural network, processes the "GloVe vectors" for each word in the response one by one sequentially, converting them into a semantic representation. As a result, the embedding vector for the entire response is derived from the very last word in sequence [15]. In the dense layer, the "softmax" activation function anticipates the one-hot encoded score for each student response. After 80% and 90% of training, the author's accuracy rates were 82% and 89%, respectively.

Adapting instruction to meet student needs can enhance learning and achievement. However, these studies cover a wide range of differentiation strategies, including both between-class and within-class ability grouping, computerized adaptive instruction, and individualized learning, with results varying from no effect to moderate positive outcomes [16]. Investigations encompass a broad spectrum of differentiation methodologies, such as inter-class and intra-class ability grouping, adaptive instruction through computer-based systems, and personalized learning plans. The outcomes of these strategies range from negligible to moderately positive impacts.

Combining supervised deep neural network models with unsupervised MCMC sampling technique, [17] this work suggested and implemented an innovative framework for an automated evaluation and reporting system This work specifically evaluated, in the same context, three models: CNN, CNN+LSTM, and CNN+Bi-LSTM, on AES tasks. Among the three methods, CNN+LSTM shown the best performance on the AES tasks, according to results. On the effective, and informative criteria, meanwhile, these three models all fell short. CNN+Bi-LSTM obtained a QWK score ranging from 0.64 to 0.72.

A need for teachers to adapt their education to students' various learning requirements. Their research highlighted several critical methods, including differentiated instruction, which adapts teaching methods and materials to a classroom's skills and learning styles. They also recommended regular formative evaluations to check student understanding and inform instruction. These tactics help educators build more inclusive and effective learning environments that help all students and instructors succeed academically. Personalized learning plans and collaborative teaching emphasize the importance of teacher flexibility and response in student achievement [18].

The development of data clustering methods beyond the conventional K-means algorithm examined advanced techniques capable of managing the complexities of data in various formats. Then, the authors emphasized the implementation of these clustering algorithms in educational settings, where grouping similar student responses can enable personalized feedback and enhance learning outcomes [19].

The literature underscores the potential of combining standardized grading frameworks, machine learning, deep learning, and clustering techniques to achieve consistent and objective grading. By leveraging these advanced methodologies, educators can enhance the accuracy of assessments, provide personalized feedback, and ultimately improve student learning outcomes. Further research and development in this interdisciplinary field hold promise for transforming educational assessment practices as shown in Table 1.

Table 1. Comparison of Text Analysis Techniques and Their Advantages

References	Text Analysis Techniques	Advantages
[20]	"Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and Glo-Ve Word Embedding for Sentiment Analysis"	Utilize advanced deep learning models like BERT in a simple and practical architecture.
[21] [17]	LSTM and CNN, CNN+Bi-LSTM	Outperforms traditional non-neural systems, providing better accuracy & QWK.
[22]	Siamese Neural Network	Well-suited for large datasets, offering consistent and reliable performance.
[23]	"Histogram of Partial Similarities and its Extension to Part-of-Speech Tags"	Demonstrates generalizability and effectiveness, validated across multiple testing scenarios, but suffers from limited context understanding, as it focuses on partial-string or tag matching without fully capturing semantic meaning.
[1]	Established Bag of Words and K-means Algorithm	Enables fast feedback and improves grading consistency across responses. Captures deeper meaning of the text.

[24]	WordNet Taxonomy - Trigram - Information Content	May rely heavily on predefined lexical hierarchies and may not capture the context-specific relationships between words in student essays. Leading to high inaccuracy.
[25]	Attention Mechanism, Bidirectional RNN with LSTM, and Word Embedding.	Enhances grading accuracy by using multiple reference responses and an attention mechanism.

Figure 1 presents a highlight of various approaches used in automatic grading systems. The detailed explanation of each category and their specific methods are as:

1. Manual Approach: The manual approach involves human experts evaluating and grading submissions. This approach ensures high accuracy and reliability due to the expertise of the graders but is time-consuming and not scalable for large volumes of data.
2. Traditional Approach: Traditional approaches leverage rule-based systems and simple algorithms to automate parts of the grading process. These methods are less flexible and may not handle complex or nuanced responses well. This method identifies predefined patterns in the student answers, matching patterns against correct or model answers. It analyzes the structure and correctness of sentences, focusing on syntax and basic grammar. It matches the grammatical structures and vocabulary of the response to those of the expected answers, checking for correctness and relevance.
3. Machine Learning Approach: Machine learning approaches use data-driven models to learn from examples and improve over time. These methods can handle more complex and varied responses compared to traditional approaches. Unsupervised models do not rely on labeled training data. They identify patterns and structures within the data independently. Whereas, supervised models are trained on labeled data, learning to predict grades based on examples of correct and incorrect answers. Semantic matching evaluates the meaning and context of the responses rather than just the literal content, ensuring that the intended meaning is captured [7] [26].
4. Deep Learning Approach: Deep learning approaches involve advanced neural networks capable of understanding complex patterns and representations in data. These methods are highly effective for handling large datasets and complex grading tasks. A series of RNNs stand in understanding sequences and context in text, making it suitable for grading tasks involving written responses. Advanced language models such as Generative Pre-trained Transformers (GPT) are pre-trained on a diverse and vast amount of text dataset. They can generate and understand text, making them highly effective for automatic grading by evaluating the content and context of responses [27].

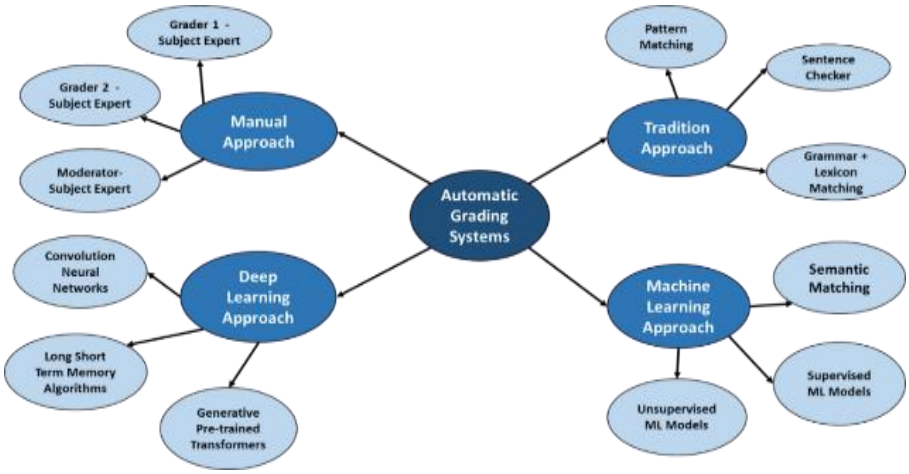

Figure 1: Overview of approaches to automatic grading system

At the center of Fig. 1 is Automatic Grading Systems, indicating that all these approaches contribute to automating the grading process to various extents and for different types of assessments.

3. AUTOMATIC ANSWER GRADING SYSTEM (AAGS)

This section briefly introduces the AAGS methodology. The objective is to develop an autonomous machine-learning system capable of predicting grades for answers. For this Essay Dataset containing a substantial number of essays focused on specific categories of topic and their human grades were taken into consideration. The dataset ensures consistency of grades among domain raters. Kaggle's dataset comprises graded essays covering various topics. In the initial phase, dataset pre-processing is carried out, cleaning the data, which includes removing inaccurate, incomplete, duplicate, or erroneous entries. Subsequently, non-alphabetical characters are stripped from the dataset. Stop words, are identified using the NLTK stop words list, and then removed from the text by tokenizing and filtering out these words.

After pre-processing word tokenizer and word embedding are applied to the entire dataset. Tokenizer converts sentences into words of the core meaning removing stop words, whitespaces, prepositions, splitters, and duplicates. Word Embedding is a numeric representation of words with similar meanings to have the same representation. Different tokenizers are employed to see the impact of results on the same data set. The large essay data set is tokenized using popular tokenizers which are used worldwide by various researchers in the Natural Language Processing (NLP) domain **[28]**. Different tokenizers and word embedders such as TF-IDF, Word2vec Glove Vector, BERT, and GPT -2 tokenizer are used to tokenize and word embeddings in this experiment. The word embedding vectors from these are passed to the LSTM, Bi-LSTM model to predict scores. The cross-fold results of these will be compared. Quadratic Kappa Score, Mean Square Error, and Variance are used as evaluation metrics for the system. The best process and model are highlighted in the results and discussion.

3.1. Proposed System (Framework)

In this section, the detailed architecture with the module description is discussed. Also, various Machine Learning modules are experimented with for the development proposed model.

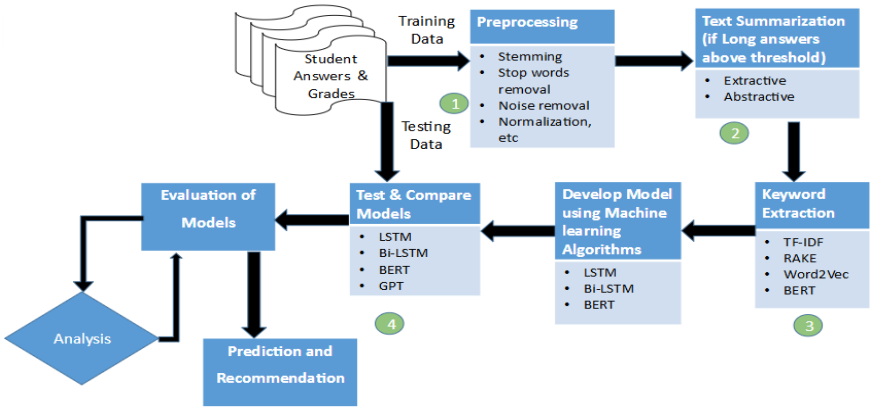

Figure 2: The proposed System shows a Block Diagram of the Answer Grading System.

The entire architecture as displayed in Figure 2 mainly has 4 phases Pre-processing Phase (Noise removal, stop words removal, stemming, tokenization) and text summarization - converting long answers text above threshold. Keyword extraction (developing a bag of words) and model Development using deep learning algorithms. In this system, the student's answer is fed to the pre-processing phase, which consists of noise removal, stop word removal, stemming, and tokenization. In noise removal, any redundant and irrelevant data is removed. In the process of stop word removal, most common words like "and," "the," and "is," which generally do not hold significant meaning, are removed. In stemming, words are brought to their root terms, and lastly, in this module, any names of persons, or places are removed to avoid potential privacy issues and ensure anonymity in data processing. In the second phase, word length is limited to a certain threshold. To eliminate extensive answers, the entire response is processed using text summarization, which summarizes large text (sentences) within a specified word limit. In this phase text summarizers like, extractive and abstractive summarizers are applied to reduce the word length and to retain the whole meaning of

answers. In the third phase of keyword extraction, tokenization is applied, adopting several different tokenizers such as Word2Vec, BERT, and GPT-2 tokenizer in an experiment to know the performance of algorithms with different tokenizes.

3.1.1. Word2Vec

The word2vec model processes text in batches, producing a vector space of hundreds of dimensions. This model generates one vector for each word. Every individual word in the corpus is represented as a high-dimensional vector. These vectors serve as a starting point for the training process; these dimensions are typically around 100-500. As training begins, vectors are updated with the words appearing in a similar context. Word2Vec is a word-based model and is a context-independent model. This model generates one vector for each word.

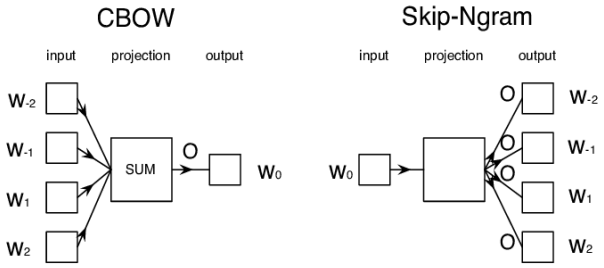

Figure 3: Word2Vec with "Continuous Bag of Words" (CBOW) and Skip-N-gram

Word2vec stands out as a widely adopted technique for embedding sequences, converting natural simple language into distributed vector forms. It excels in capturing intricate word-to-word relationships within a multidimensional space and serves as an essential pre-processing phase for predictive models in semantic analysis and information retrieval tasks. The process of Word2vec, illustrated in Figure 3, comprises two main components: "Continuous Bag of Words" (CBOW) and skip-gram. CBOW generates the target word by considering the context of surrounding words, whereas skip-gram forecasts nearby context words for a specified input word. To determine the relatedness between two sentences, the authors were instructed to perform the following operation using their respective vector representations u and w.

$$u * w = r \tag{1}$$

$$| u - w | = v \tag{2}$$

$$R + v \tag{3}$$

Equation (1) is the element-wise product between the two vectors, while (2) is the absolute value of the difference between both vectors. Those two results are then concatenated as indicated in (3). This final vector represents the sentence pair composed of u and w [29].

3.1.2. BERT Tokenizer

Introduced "Bidirectional Encoder Representations from Transformers," a continuous sequence-to-sequence model that best represents word associations regardless of sentence location. BERT developed on transformers and self-attention mechanisms, revolutionized NLP by pre-training on unlabeled text like Wikipedia using masked language modelling. This method trains BERT to predict missing words contextually rather than relying on fixed embedding. The transformer architecture allows BERT to process words concerning all others in a sentence simultaneously, capturing complex dependencies effectively. BERT's bidirectional self-attention enables it to understand how word meanings evolve within sentences, crucial for handling ambiguity in natural language. Its pre-trained model supports transfer learning, enabling adaptation to new tasks with smaller datasets, and making BERT highly versatile among others for various NLP applications **[30]**. In the fourth phase model development is done using deep learning algorithms and optimizers.

3.2. Deep Learning Models

LSTM: For natural language processing, LSTM is a sort of RNN, which interpret and generates human language. The memory cell of an LSTM lets it store and retrieve information over time. Traditional RNNs have limited memory and data storage. Figure 4 shows input, forget, and output gates in the LSTM design. These gates regulate memory cell data

flow. The input gate saves fresh information in the memory cell, the forget gate deletes irrelevant information, and the output gate applies it to the current work. LSTMs are mostly used for time series forecasting [31].

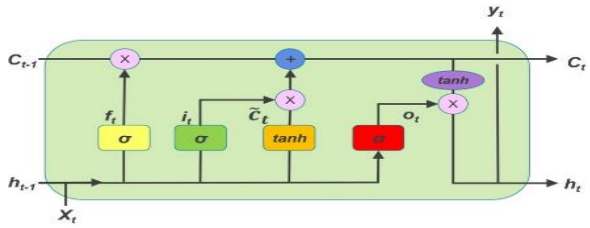

Figure 4: Long Short-Term Memory Architecture

Bi-Long Short-Term Memory (Bi-LSTM): Bi-Directional Long Short-Term Memory as an RNN that captures dependency throughout the input vector's temporal sequence. Bi-LSTM lets models draw from past and future settings. Figure 5 shows Bi-LSTM architecture with two layers, one front and one backward. The forward LSTM layer generates hidden and cell states at each time step from the input sequence. The backward LSTM generates a hidden state and cell state at each time step by processing the same input sequence in reverse order [22].

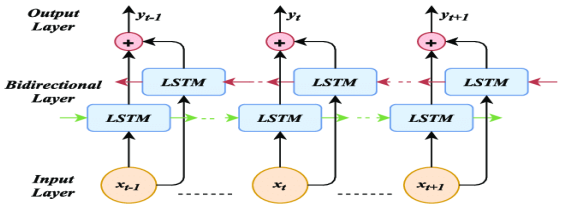

Figure 5: Bidirectional Long Short-Term Memory Architecture

Generative Pre-Transformers: A Deep Learning model, GPT employs a transformer architecture consisting of multiple self-attention layers. The GPT concerning other words within the input sequence effectively captures both short-range and long-range dependencies. During the training process, GPT is optimized to forecast the next word in a sequence based on preceding words. This training allows the model to generate text by sequentially predicting the most probable subsequent words according to the given input. GPT-1 is based on the same transformer architecture that is introduced in the paper called "Attention Is All You Need" which introduces the transformer design used in GPT-1. A 12-layer transformer block, 768 hidden units, and 12 attention heads are used. GPT-1 introduced unsupervised NLP pre-training, which worked well. GPT-2's largest transformer model included 48 layers, 1,024 hidden units, and 16 attention heads. This improved capability enables it to record more complicated text patterns. GPT-2 performed much better than GPT-1. It could write more coherently, contextually, and rationally on more themes. Applying GPT-2 for Tokenizer and Word Embedding [31].

Optimizers

The LSTM algorithm can be run with various optimizers to improve the performance of algorithms such as "Particle Swarm Optimization" (PSO) [32], "Ant Colony Optimization" (ACO) [33] "Differential Evolution" (DE) [34] and "Grey Wolf Optimization" (GWO) [35]. However, compared to other optimization techniques, Grey Wolf Optimization performs better. Details of comparison with different optimization techniques are discussed below. The final decision for selecting GWO is based on the comparisons done below.

Comparison with Other Optimization Techniques

Particle Swarm Optimization: GWO and PSO share similarities in that both are inspired by the social behaviour of animals. However, GWO offers better exploration capabilities in the early stages of the search and avoids premature convergence more effectively than PSO, making it more suitable for high-dimensional problems like those found in NLP and ASAG.

Ant Colony Optimization: ACO is more complex due to its probabilistic nature and the need for pheromone updating rules. GWO, on the other hand, is straightforward and requires fewer parameters, making it easier to implement and tune for specific tasks in NLP and ASAG. Differential Evolution: DE is robust but often slower compared to GWO. GWO's leadership hierarchy (alpha beta & gamma wolves, i.e., leader and follower technique) and hunting mechanism

provide a more balanced approach to exploration and exploitation, which can lead to quicker convergence in NLP tasks.

AAGS Algorithm 1: BERT Tokenize with Bidirectional Encoder Representations Transformer (BERT)

1. Import all the necessary library
2. Import the CSV or TSV file
3. Drop the null columns or useless information
4. Normalize the numeric data score
5. Pre-process the text essay and remove extra spaces and @ or any spatial symbols
6. Remove stop words in the essay
7. Store the clean essay in the column
8. Load the dataset
9. Split into train and validation dataset
10. Loading pre-trained BERT model and tokenizer.
11. Tokenize the essay using a BERT tokenizer.
12. Use an optimizer with a low learning rate (ex Adam)
13. Run the epochs to train the model using the BERT sequence classifier
14. After Completion of the epochs, store the BERT model
15. Using the Fine-tuned model for grading load the Fine-tuned model
16. Load the validation dataset and predict the score.
17. Based on predictions and true labels calculate Mean Score Error
18. Evaluate Models: Calculate Kappa Score

Model Development

AAGS Algorithm 2: GPT-2 Tokenizer and Long Short-Term Memory with GWO

2.1 GPT-2 Tokenizer

1. Import Necessary Libraries for GPT-2:
2. Load the GPT-2 tokenizer.
3. Add padding tokens to the tokenizer.
4. Load the GPT-2 model.

2.2 LSTM + Grey Wolf Optimizer (GWO) for Hyper-parameter Tuning

1. Import Necessary Libraries for LSTM and GWO:
2. Prepare Data for LSTM Model Training:
3. Define a fitness function to evaluate the performance of the LSTM model using different hyper-parameters
4. Define the structure of the LSTM model and the hyper-parameters that need to be optimized (e.g., units, epochs, batch size).
5. Define the Easom function to be used by the GWO for optimization.
6. Set parameters for the GWO (pack size, bounds for hyper-parameters, number of iterations, etc.).
7. Execute the GWO to find the optimal hyper-parameters for the LSTM model.
8. Train the LSTM Model with Optimized Hyper-parameters
9. Use the best hyper-parameters found by the GWO to build and train the LSTM model.
10. Load the validation dataset and predict the score.

11.	Based on predictions and true labels calculate Mean Score Error
12.	Evaluate Models using Calculate Cohen Kappa Score, Variance

In this section, a brief description of the proposed algorithms is discussed. The algorithm makes use of a Kaggle data set as provided at "The Hewlett Foundation: Automated Essay Scoring". The BERT Tokenized and GPT-2 Tokenizer respectively are employed for tokenization and the model is further trained and tested using BERT Sequence Classifier with ex Adam Optimizer, and LSTM with Grey Wolf Optimizer respectively. The results of these two models are compared in Results and Discussion. The BERT tokenizer employs a method known as Word Piece tokenization, which decomposes words into sub-word units. This enables the model to process out-of-vocabulary words by understanding their constituent sub-words.

AAGS is rebuilt differently. A complex solution using a GPT-2 Tokenizer and a GWO-optimized LSTM model for hyper-parameter adjustment is proposed in the below algorithm. The components integrate to provide semantic understanding with GPT-2 Tokenizer, temporal sequence modeling with LSTM, optimization with Grey Wolf Optimizer (GWO), and robust assessment metrics.GPT-2 tokenizes text well and preserves its semantic richness. Sentences and paragraphs are ideal for the LSTM model. It captures dependencies throughout time, making it perfect for understanding student response flow and context. The GWO is an advanced meta-heuristic algorithm inspired by grey wolf social hierarchy and hunting. It efficiently searches the hyper-parameter space for the ideal settings to improve LSTM answer grading. The GWO algorithm simulates grey wolf's leadership structure & cooperative hunting. The α wolf leads and offers the finest option, followed by β and δ, while ω wolves follow. These three best solutions lead the other wolves to the best solution. Grey wolves surround prey during hunting. Hunting is guided by α, β, and δ wolves. The ω wolves adjust their locations based on these three leaders. The GWO algorithm iterates through encircling, hunting, and attacking until reaching a halting threshold.

4. RESULTS AND DISCUSSION

4.1. Dataset

The Kaggle competition launched in 2012, a competition on Automated Essay Scoring called "Automated Student Assessment Prize" (ASAP, https://www.kaggle.com/c/asap-aes/data) which was sponsored by the "Hewlett Foundation". The essay dataset provided by the Kaggle "The Hewlett Foundation: Automated Essay Scoring" (with statistics as shown in Table 2), for the development of an automated scoring system for descriptive essays, are considered as answers of length 150 to 650 words and is used in an experiment to compare results with other rating systems. A Kaggle dataset was made available to obtain a collection of human-graded essay scores, enabling researchers to develop, train, and evaluate their scoring systems in competition with other established models. Data scientists and machine learning engineers globally utilized a dataset provided by Kaggle to devise rapid, efficient, and cost-effective solutions for automated grading systems for student-authored essays. The dataset had eight sets of essays, with each set taken from different grade students from high school grades. The essay's length ranges from 150 to 650 words per response. All these essays underwent manual grading and double-scoring by different human graders. The training and testing data set is structured in a tab-separated value (TSV) file format, having three scores: rater1 score, rater2 score, and domain score. Table 2 is a brief description of the dataset which is also used in this experiment and result comparison.

Table 2: The Kaggle's Dataset for Automated Student Assessment Prize Statistics Highlighted

Essay Grade ID	Essays Count in each set	Average Word Length of Essays	Score Range Graded by Graders
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

4.2. Evaluation Metrics Used

To evaluate the consistency among grades awarded to student answers assigned by multiple graders, In our proposed model, the "Quadratic-Weighted Kappa" (QWK) [36] is adopted, opting for it over traditional "Cohen's Kappa" because QWK can account for the ordinal nature of the scores. Take an answer that may earn a score between 0 and 2, for example. If the first grader gives a score of 0, the second grader gives a score of 1, and the third grader gives a score of 2, then the second and third graders don't agree with the first grader. "Quadratic Weighted Kappa" (QWK) as in equation 4 is a statistical way to find out how much two graders who categorize things agree with each other. Scores range from -1 to 1, with 1 meaning full agreement, 0 meaning agreement by chance, and negative values meaning complete disagreement beyond chance. Scores between 0.0 and 0.2 mean low agreement, scores between 0.2 and 0.4 mean mild agreement, scores between 0.4 and 0.6 mean good agreement, and scores between 0.8 and 1.0 mean perfect agreement.

$$\mathbf{qwk} = \mathbf{1} - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\omega_{ij}O_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\omega_{ij}E_{ij}} \quad (4)$$

where, $\boldsymbol{O_{ij}}$(Observed Agreement) is a number of times the rater pair gave scores i and j respectively.

$\boldsymbol{E_{i,j}}$ (Expected Agreement) is agreement accepted by chance,

$\boldsymbol{E_{i,j}}$ is calculated as, $\boldsymbol{E_{i,j}} = \frac{\sum_{k=1}^{N} O_{ik} \cdot \sum_{k=1}^{N} O_{kj}}{N}$ (5)

where, $\sum_{k=1}^{N} \boldsymbol{O_{ik}}$ is total number of observations for rater1=i, & $\sum_{k=1}^{N} \boldsymbol{O_{kj}}$ is the total number of observations for rater2=j.

$\boldsymbol{\omega_{ij}}$ is Weighted matrix is quadratic weight for disagreement between i & j.

$$\boldsymbol{\omega_{ij}} = \frac{(i-j)^2}{N-1^2} \quad (6)$$

where, N is number of possible rating categories.

"Mean Squared Error" (MSE) as a common way to measure the average squared difference between a dataset's real value and its expected value. MSE is a positive number, and a lower MSE means the model worked better. A zero MSE means that the expected and actual numbers were the same [37].

$$MSE = 1/n \sum_{i=1}^{n}(y_i - \hat{y}_n)^2 \quad (7)$$

where n is the sequence number of data sets, $\boldsymbol{y_i}$ is the actual value of ith data row and $\hat{\boldsymbol{y}}_{\boldsymbol{n}}$ is the anticipated value of the i^{th} data row.

Accuracy: A Short Answer Grading System measures how well the system's predictions match the correct answers. It can be computed using the following formula.

Accuracy = No of Correct Predictions/Total No of Predictions (8)

F1 Score: It is a performance metric for classification models, representing the Harmonic Mean of Precision and Recall. It offers a balanced kind of assessment by considering both metrics as shown in equation 9.

F1 Score = (2 * (Precision * Recall)) / (Precision + Recall) (9)

Performance Analysis of Existing Methods: When evaluating performance in automatic essay grading systems using different methods like LSTM, Bi-LSTM (Bidirectional-LSTM) with attention mechanism [25], it's crucial to analyse each method's strengths and limitations in the context of essay grading. Table 3 Comparing Various Existing Models in Automatic Short Answer Grading System (ASAGS) and their results.

Table 3: Existing Machine Learning Models Results in ASAGS [25]

Model	QWK	MSE
LSTM	0.94	7.86
LSTM + Bi LSTM	0.95	6.66
LSTM + Bi LSTM + Attention	0.96	6.2

Performance Evaluation of Tokenizers: This section compares how well different tokenizers work. The "Kaggle data set" is used to train the BERT and GPT-2 tokenizers, and the Kappa Score, F1 Score, training loss, and validation loss are used to see how well they did. The graphs in Figure 6 discuss the performance of various tokenizers used in this study, across all 5 folds. The tokenizers are run for 5 folds, each fold consisting of 50 epochs to train the tokenizers on a given data set, and their performance is observed. The evaluation metrics, such as Kappa Score, F1-Score, Training Loss, and Validation Loss are used to measure performances across Word2Vec, BERT, and GPT-2 Tokenizers on their epochs for all 5 folds. Kappa Score Comparison shows that GPT-2 Tokenizer achieves the highest Kappa Score across all folds, indicating strong agreement and better performance in classification tasks as compared to BERT. Though BERT and GPT-2 have slightly lower Kappa scores, BERT exhibits higher variations in Kappa scores in all folds. Then in the F1 Score, GPT-2 shows a slightly better score in few folds as compared to BERT. Training Loss shown in Figure 7, shows GPT-2 and BERT exhibits small training loss. The validation loss shown in the validation graph remains consistent across all 5 folds in GPT-2 Tokenizer as compared to BERT.

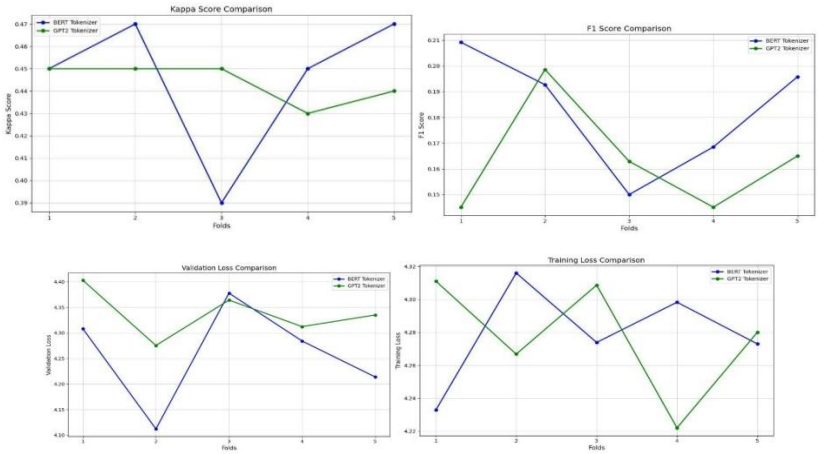

Figure 6: Graphs showing a comparison of Word2Vec, BERT, and GPT-2 Tokenizer Performance for Kappa Score, F1 Score, Validation loss, and Training loss

Fig. 7 compares the BERT and GPT-2 Tokenizes regarding training loss and validation loss over 50 epochs. During the initial five epochs, both BERT and GPT-2 exhibit comparable performance regarding training and validation loss. During the 35th epoch, BERT and GPT-2 exhibit comparable performance throughout the training phase, significantly increasing BERT's validation loss, whilst GPT-2 maintains relative stability during this interval. In the subsequent epochs (from 35 onwards), GPT-2 demonstrates more constant performance, but BERT exhibits occasional significant aberrations. GPT-2 is preferable for extended training sessions because of its more consistent performance in the validation loss graph, suggesting superior generalization to unseen datasets. BERT demonstrated initial stability, then exhibited increased variance and fluctuations in subsequent epochs, potentially affecting performance based on the validation dataset.

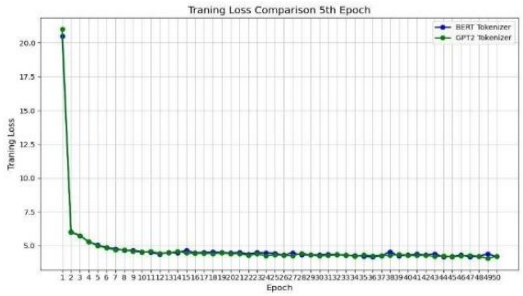

Figure 7: Graphs showing BERT and GPT-2 Tokenizer Training Loss and Validation Loss for 50 epoch

Table 4 compares the performance of the GPT-2 and BERT Tokenizers. Based on this data, the GPT-2 tokenizer performs slightly better than BERT because of its good validation loss and high Kappa Score and F1 scores.

Table 4: Comparing Average Performances of BERT and GTP-2 Tokenizer

Tokenizers	Training Loss	Validation Loss	Kappa Score	F1 Score	Variance
GPT-2	4.27	4.36	0.45	0.17	0.29
BERT	4.27	4.29	0.44	0.15	0.30

Performance Evaluation of Proposed Model: In this section, a discussion of the developed model with training loss and validation loss with the sample's first 10 epochs is discussed. F1 Score is also plotted to understand the model's predictive performance on a sample basis. Figure 8 illustrates the performance characteristics of a BERT model over 10 epochs. In this, the Training Loss, Validation Loss, and F1 Score (Weighted) on the y-axis against the number of epochs on the x-axis are plotted to see how the model is performing. The training loss (blue line) indicates the BERT model is learning and fitting the data over epochs. The validation loss (orange line) indicates the BERT model is not performing well concerning training. Over-fitting during validation is observed. The increase in F1 Score (green line) is not significant to decreasing training loss, this indicates, that the model is not generalized when testing on the validation set.

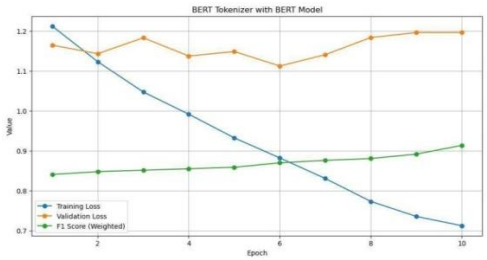

Figure 8: Performance graph of BERT Model with BERT Tokenizer

The line graph as shown in Figure 9 displays the performance metrics of our developed Machine Learning model using a GPT-2 tokenized as word embedding with an LSTM (Long Short-Term Memory) network optimized by GWO (Grey Wolf Optimizer). In this, the Training Loss, Validation Loss, and F1 Score (Weighted) on the y-axis against the number of epochs on the x-axis is plotted to see how the model is performing over 10 epochs. The consistent decline of training loss (blue line) suggests that the model is effectively learning from given training data and enhancing its performance over epochs. The validation loss (orange line) demonstrates the model's capability to generalize and stabilize well even with new data. The F1 Score (green line) with a consistent increase at the end indicates that the balance of precision and recall remains consistent with a decrease in training and validation loss. This shows that model predictions are becoming well without significance in over fitting.

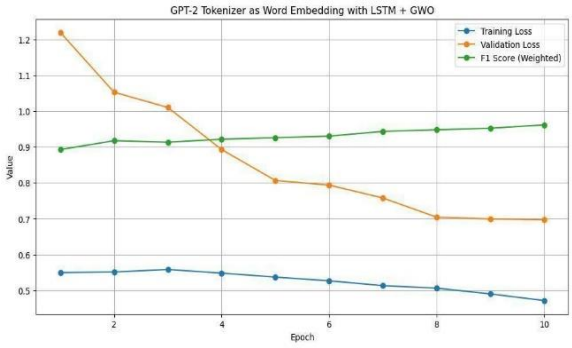

Figure 9: Performance graph of LSTM Model with GPT-2 Tokenizer

Compared with the BERT model developed with BERT Tokenizer and the LSTM model with GPT-2 Tokenizer, the developed model (LSTM with GPT-2 Tokenizer) is performing well, with less over-fitting. The BERT model is facing an increase in F1 Score with a decrease in training and consistent varying of validation loss indicating model overfitting and performance loss. The LSTM model utilizing the GPT-2 tokenizer demonstrates a gradual enhancement in F1

Score alongside a reduction in both training and validation loss, signifying effective learning without overfitting, and stability in F1 Score with consistent performance in precision and recall. Table no 5 and Figure 10 showcase a comparison of metrics for our developed model with the LSTM and BERT model used in NLP. In this the metrics QWK is used to measure the agreement between two different graders. A higher value indicates better performance. Second, we used Accuracy as a measure, which indicates the proportion of correctly predicted scores out of total scores. A third measure, MSE is the average square difference between anticipated and actual scores, lower indicates better performance. Lastly, Variance indicates the variability of model prediction, a higher value indicates, more variability.

The LSTM Model used Word2Vec word embedding with ReLu Activation function, which has a high QWK value indicating good agreement, the accuracy is quite low, which is unusual. The MSE is relatively high, at 7.86 which indicates larger errors in predictions. The variance is 0.9 which is significantly variable in predictions. The Bi-LSTM Model with BERT Tokenizer for tokenizing with an attention mechanism. This model shows some improvement over the previous model with a higher QWK of 0.9364 and better accuracy at 0.37. The MSE is lower at 6.66, indicating better prediction accuracy. The variance is slightly higher at 0.92, suggesting a bit more variability in predictions. BERT model with BERT tokenizer as word embeddings directly within a BERT model, optimized using the Adam optimizer this model achieves the better performance among the three models with a QWK of 0.9521, higher accuracy at 0.72, and the lower MSE at 3.56, indicating the more accurate predictions, and variance of 0.37, indicating the lesser variability in predictions.

Table 5: Comparing Performances of Developed Models with Other Models in ASAG

Model	QWK	Accuracy	MSE	Activation	Variance
Fine Tuned BERT [38]	0.88	0.77	~	~	~
BERT Tokenizer with	0.936398147	0.37	6.66	Relu	0.92
BERT Tokenizer as	0.952121614	0.72	3.56	~	0.37
GPT-2 as	0.976254	0.86	2.33	~	0.28

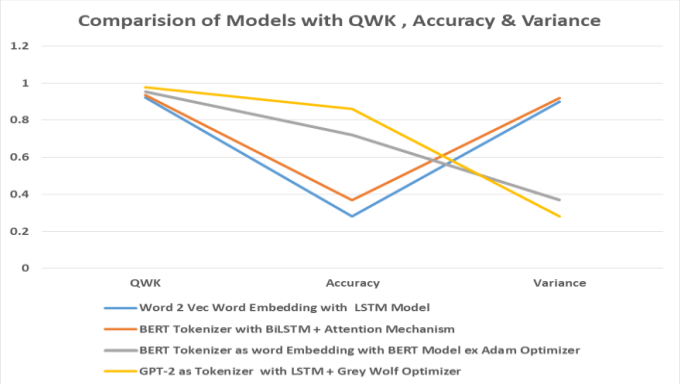

Figure 10: Graph Comparing Model QWK, Accuracy and Variance

Developed model suggesting GPT-2 as Tokenizer with LSTM model optimized using Grey Wolf Optimizer outperforms the BERT Tokenizer as Word Embedding with BERT Model ex Adam Optimizer across all performance metrics. The model has the highest QWK value of 0.9763 among all models, indicating best agreement with actual values. The highest accuracy at 0.86 reflects the most accurate predictions. The lowest MSE suggests very few errors in predictions and the lowest variance of 0.28 implies more stable and consistent predictions among all other State-of-Art-Models. As shown in Table 6 proposed methodology outperforms in highest quadratic weighted kappa score as compared with other state-of-the-art models developed by other researchers.

Table 6: Proposed Methodology Comparison with State-of-Art-Models

Citations & Year	Model Techniques	Dataset	Highest QWK
[39] (2018)	SBLSTMA	ASAP	0.861

[40] (2019)	Bi-LSTM neural network	ASAP	0.870
[17] (2021)	CNN+Bi-LSTM	ASAP	0.726
[38] (2023)	One Shot – SBERT	ASAP	0.73
[38] (2023)	Fine Tuned BERT	ASAP	0.88
[25] (2024)	Bi-LSTM with Attention	ASAP	0.936
Proposed	BERT Tokenizer with BERT Ex	ASAP	0.957
Proposed	GPT-2-LSTM-GWO	ASAP	0.976

5. CONCLUSION AND FUTURE WORK

By integrating advanced NLP techniques, such as GPT-2 tokenization, with deep learning models like LSTM networks and optimization methods such as GWO, the ASAG system offers a robust, efficient, and scalable solution for grading short answers [41-42]. This approach addresses the shortfall of traditional grading systems, which often fail to capture the complexity and delicacy of student responses. The use of the GPT-2 tokenizer ensures that the semantic content of answers is accurately represented, while the LSTM model effectively analyzes the temporal sequence of words, allowing for a more nuanced evaluation of student input. Moreover, the application of the GWO for hyper-parameter fine-tuning optimizes the performance of the LSTM model, ensuring that the system is both accurate and reliable. The developed model, utilizing GPT-2 as the tokenizer and optimized by the Grey Wolf Optimizer, outperforms other models across all metrics. It achieves the best agreement with actual values (highest QWK), the highest accuracy, the lowest MSE, and the least variance, making it the most effective and reliable model among those compared.

Develop standardized benchmarks and datasets for evaluating AAG systems, allowing for more objective comparison and validation of different approaches. Also to develop hybrid systems where teachers can interact with and adjust the automatic grading to ensure the system's decisions align with human judgment, allowing for continuous improvement. To develop Generative AI tools that can provide grades to answers, along with valid feedback to the student, which makes students agree on their mistakes and improve on them.

REFERENCES

[1] Amur ZH, Hooi YK. State-of-the-art: Assessing semantic similarity in automated short-answer grading systems. Inf. Sci. Lett. 2022;11:1851-8.
[2] Ratna AA, Purnamasari PD, Anandra NK, Luhurkinanti DL. Hybrid deep learning cnn-bidirectional lstm and manhattan distance for japanese automated short answer grading: Use case in japanese language studies. InProceedings of the 8th International Conference on Communication and Information Processing 2022 Nov 3 (pp. 22-27).
[3] Hattie J, Timperley H. The power of feedback. Review of educational research. 2007 Mar;77(1):81-112.
[4] Mrs.S.Mahalakshmi M. M. M.S M.E, "AUTOMATED ANSWER EVALUATION BASED ON DEEP LEARNING," INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT), vol. 12, no. 03, 2023.
[5] Prakoso DW, Abdi A, Amrit C. Short text similarity measurement methods: a review. Soft Computing. 2021 Mar;25:4699-723.
[6] Ibrahim SS, Elfakharany EF, Hamed E. Improved Automated Essay Grading System Via Natural Language Processing and Deep Learning. In2022 International Conference on Engineering and Emerging Technologies (ICEET) 2022 Oct 27 (pp. 1-7). IEEE.
[7] Amur ZH, Hooi YK, Soomro GM, Bhanbhro H, Karyem S, Sohu N. Unlocking the potential of keyword extraction: the need for access to high-quality datasets. Applied Sciences. 2023 Jun 16;13(12):7228.
[8] ISarker IH. Machine learning: Algorithms, real-world applications and research directions. SN computer science. 2021 May;2(3):160.
[9] M. C. S. B. R. .. S. S. ... D. G. M. S. Prerana, " Eval - Automatic Evaluation of Answer Scripts using Deep Learning and Natural Language Processing.," International Journal of Intelligent Systems and Applications in Engineering, p. 316–323, 2023.
[10] Baniata LH, Kang S, Alsharaiah MA, Baniata MH. Advanced deep learning model for predicting the academic performances of students in educational institutions. Applied Sciences. 2024 Feb 28;14(5):1963.
[11] DRamesh D, Sanampudi SK. An automated essay scoring systems: a systematic literature review. Artificial Intelligence Review. 2022 Mar;55(3):2495-527.
[12] Lui AK, Ng SC, Cheung SW. A framework for effectively utilising human grading input in automated short answer grading. International Journal of Mobile Learning and Organisation. 2022;16(3):266-86.

[13] Sokkhey P, Okazaki T. Hybrid machine learning algorithms for predicting academic performance. Int. J. Adv. Comput. Sci. Appl. 2020;11(1):32-41.
[14] George N, Sijimol PJ, Varghese SM. Grading descriptive answer scripts using deep learning. International Journal of Innovative Technology and Exploring Engineering (IJITEE). 2019 Mar;8(5).
[15] Abdullah AS, Geetha S, Aziz AA, Mishra U. Design of automated model for inspecting and evaluating handwritten answer scripts: A pedagogical approach with NLP and deep learning. Alexandria Engineering Journal. 2024 Dec 1;108:764-88.
[16] Alammary AS. BERT models for Arabic text classification: a systematic review. Applied Sciences. 2022 Jun 4;12(11):5720.
[17] Deunk MI, Smale-Jacobse AE, de Boer H, Doolaard S, Bosker RJ. Effective differentiation practices: A systematic review and meta-analysis of studies on the cognitive effects of differentiation practices in primary education. Educational Research Review. 2018 Jun 1;24:31-54.
[18] Lu C, Cutumisu M. Integrating Deep Learning into an Automated Feedback Generation System for Automated Essay Scoring. International Educational Data Mining Society. 2021.
[19] Pasira I. Assessing the effectiveness of differentiated instruction strategies in diverse classrooms. Journal of Education Review Provision. 2022 Jun 2;2(1):31-6.
[20] Hooshyar D, Padaste m. Huang YM., "Clustering Algorithms in an Educational Context: An Automatic Comparative Approach," IEEE Access, pp. 1-1, 2020.
[21] Kamyab M, Liu G, Adjeisah M. Attention-based CNN and Bi-LSTM model based on TF-IDF and glove word embedding for sentiment analysis. Applied Sciences. 2021 Nov 27;11(23):11255.
[22] Haller S, Aldea A, Seifert C, Strisciuglio N. Survey on automated short answer grading with deep learning: from word embeddings to transformers. arXiv preprint arXiv:2204.03503. 2022 Mar 11.
[23] Prabhudesai A, Duong TN. Automatic short answer grading using Siamese bidirectional LSTM based regression. In2019 IEEE international conference on engineering, technology and education (TALE) 2019 Dec 10 (pp. 1-6). IEEE.
[24] Faseeh M, Jaleel A, Iqbal N, Ghani A, Abdusalomov A, Mehmood A, Cho YI. Hybrid approach to automated essay scoring: Integrating deep learning embeddings with handcrafted linguistic features for improved accuracy. Mathematics. 2024 Oct 31;12(21):3416.
[25] Goh TT, Jamaludin NA, Mohamed H, Ismail MN, Chua H. Semantic similarity analysis for examination questions classification using WordNet. Applied Sciences. 2023 Jul 19;13(14):8323.
[26] Mahajan D, Channe P, Diwate S, Kharate S, Patil R. Smart Grading System Using Bi LSTM with Attention Mechanism. InInternational Conference on Emerging Research in Computing, Information, Communication and Applications 2023 Feb 24 (pp. 247-260). Singapore: Springer Nature Singapore.
[27] Amur ZH, Hooi YK, Bhanbro H, Bhatti MN, Soomro GM. Machine learning model for automated assessment of short subjective answers. International Journal of Advanced Computer Science and Applications. 2023..
[28] Hassan S, Fahmy AA, El-Ramly M. Automatic short answer scoring based on paragraph embeddings. International Journal of Advanced Computer Science and Applications. 2018 Oct 1;9(10):397-402.
[29] Rajest S, Rajan R. A new Natural Language Processing-Based Essay Grading algorithm. ResearchGate. 2023 May.
[30] Kiros R, Zhu Y, Salakhutdinov RR, Zemel R, Urtasun R, Torralba A, Fidler S. Skip-thought vectors. Advances in neural information processing systems. 2015;28.
A. Ushio, L. Espinosa-Anke, S. Schockaert and J. Camacho-Collados, "BERT is to NLP what AlexNet is to CV: Can pre-trained language models identify analogies?," arXiv:2105.04949., 2021.
[31] Gilal AR, Waqas A, Talpur BA, Abro RA, Jaafar J, Amur ZH. Question guru: an automated multiple-choice question generation system. InInternational Conference on Emerging Technologies and Intelligent Systems 2022 Sep 2 (pp. 501-514). Cham: Springer International Publishing.
[32] Sung C, Dhamecha TI, Mukhi N. Improving short answer grading using transformer-based pre-training. InArtificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20 2019 (pp. 469-481). Springer International Publishing.
[33] Lee CS, Wang MH, Wang CS, Teytaud O, Liu J, Lin SW, Hung PH. PSO-based fuzzy markup language for student learning performance evaluation and educational application. IEEE Transactions on Fuzzy Systems. 2018 Feb 28;26(5):2618-33.
[34] Hamdi M. Affirmative ant colony optimization based support vector machine for sentiment classification. Electronics. 2022 Mar 27;11(7):1051.

[35] Chen H, Heidari AA, Chen H, Wang M, Pan Z, Gandomi AH. Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. Future Generation Computer Systems. 2020 Oct 1;111:175-98.
[36] Dada E, Joseph S, Oyewola D, Fadele AA, Chiroma H, Abdulhamid SI. Application of grey wolf optimization algorithm: recent trends, issues, and possible horizons. Gazi University Journal of Science. 2022 Jun;35(2):485-504.
[37] Doewes A, Kurdhi N, Saxena A. Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring. In16th International Conference on Educational Data Mining, EDM 2023 2023 Jul 11 (pp. 103-113). International Educational Data Mining Society (IEDMS).
[38] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning. New York: springer; 2013 Jun 24.
[39] Yoon SY. Short answer grading using one-shot prompting and text similarity scoring model. arXiv preprint arXiv:2305.18638. 2023 May 29.
[40] Liang G, On BW, Jeong D, Kim HC, Choi GS. Automated essay scoring: A siamese bidirectional LSTM neural network architecture. Symmetry. 2018 Dec 1;10(12):682.
[41] Xia L, Liu J, Zhang Z. Automatic essay scoring model based on two-layer bi-directional long-short term memory network. InProceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence 2019 Dec 6 (pp. 133-137).