University of Ibadan
Journal of Science and Logics in ICT Research

A Comparative Evaluation of Embedding Techniques from Language Models for Automatic Grading of Short Answer Questions

[1]✉ Odumuyiwa V., [2]Adewoyin O., [3]Fagoroye A., [4]Fasina E., [5]Sawyerr B. and [6]Sennaike O.

[1,2,4,5,6]Department of Computer Sciences, University of Lagos, Nigeria;
[1]vodumuyiwa@unilag.edu.ng, [2]damilola_adewoyin@yahoo.com, [4]efasina@unilag.edu.ng
[5]bsawyerr@unilag.edu.ng, ,[6] osennaike@unilag.edu.ng
[3]Department of Systems Engineering, University of Lagos, Nigeria; fagoroyeayomide@gmail.com

Abstract

An automatic grading system of short answer questions on an e-learning platform can help reduce stress, save time, increase the productivity of instructors and help provide feedback to students in record time. However, the success of automatic grading of short answer questions (open-ended questions) depends on the ability of the computer to adequately capture the semantic similarity between students' answers and the reference answer provided by the examiner. This paper presents a comparative study of some embedding techniques from language models for automatic grading of short answer questions in order to address the longstanding challenge of automating the assessment of students' responses to open-ended questions. It studies five embedding techniques such as Word2vec, Bi-LSTM, BERT, SBERT, and OpenAI on four datasets (SemEval, Texas, ASAG, and MIT) to find the best method among them for Automatic Short Answer Grading (ASAG). Experiments include regression tasks and classification tasks using Mean Squared Error (MSE), Pearson Correlation, and accuracy as metrics for evaluation. The results indicate that fine-tuned BERT achieved the highest accuracy of 75% on SemEval dataset in classification tasks, while OpenAI performed better in the regression tasks with a MSE of 0.57 on the Texas dataset. The research highlights automated grading as a means to reduce instructors' workload while enhancing the quality of feedback provided to learners. Future studies will focus on extending the experiments to include both domain-specific and non-domain-specific.

Keywords: *Auto-grading, transformer models, BERT, OpenAI, natural language processing (NLP)*

1. Introduction

Since the introduction of electronic learning (e-learning), several institutions have integrated online education into their learning structure for flexibility and convenience. The emergence of Covid-19 and the threat it poses has also forced many institutions to consider online classes and examinations over physical classes and examinations. Instructors desire to perform both formative and summative evaluations for the learners on the topics taught. For such evaluations, multiple-choice questions (MCQ) or True/false questions have majorly been employed due to the ease of grading these questions. Short answer or essay questions are however effective for stimulating the students' minds [1] and encouraging creativity but they have not been widely adopted in online evaluation due to the difficulty in automatically grading them.

Automatic grading has attracted attention in the past few years because it promises fairness, saves time [2], increases the productivity of instructors [3] and reduces the negative impact of emotions and bias by the human marker in the grading process. Automatic grading can serve as an added advantage for instructors to provide several formative and summative assessments in their subjects to enhance students' learning experience. One of the advantages of automatic grading is that it standardizes the grading process and makes it easy to keep track of student information. The automated grading of MCQ or True/False questions is considered relatively simple because there is only one correct answer. Short answers and essay

Odumuyiwa V., Adewoyin O., Fagoroye A., Fasina E., Sawyerr B. and Sennaike O. (2025) A Comparative Evaluation of Embedding Techniques from Language Models for Automatic Grading of Short Answer Questions, *University of Ibadan Journal of Science and Logics in ICT Research* (UIJSLICTR), Vol. 13 No. 1, pp. 164 - 177

questions, on the other hand, are open-ended in nature and require free responses.

The difference between short answers and essays could sometimes become distorted. According to [4], a short answer question should satisfy five requirements which include the following:

a. the length of the expected answer should be between one phrase and a paragraph
b. the answers should be expressed in natural language
c. the evaluation should be based on the content of the answer and not the writing style
d. its answer should be recalled from external knowledge and not from the question
e. the level of openness should be controlled with an objective question design.

Many of the existing grading systems evaluate the students' answers based on keywords. This means that students with the most keywords have the highest scores. The use of keywords does not encourage learning and creativity as students are pushed to memorize. It is therefore important to consider the semantic relationship between words in the grading process.

Language models have proven to achieve good performance in a lot of the natural language processing (NLP) tasks such as speech recognition, machine translation, and information retrieval [5]. This paper aims to evaluate the performance of five Embedding Techniques from language models – Word2vec, Bi-LSTM, BERT, OPENAI and SBERT – on auto-grading of short answers in four different datasets to provide answers to the following research question: which embedding technique performs better on Automatic Short Answer Grading (ASAG) tasks?

The rest of this work is organized as follows: Section 2 presents the related literature, existing approaches, and their limitations. The methodology and the dataset used are discussed in section 3. Section 4 compares the performance of the language models on the datasets. Section 5 concludes the study and discusses future work.

2. Literature Review

In our contemporary times, where learning facilitation through a learner-centred delivery is considered more appropriate than the traditional teacher-centred approach, the need for qualitative assessment and provision of continuous feedback cannot be over-emphasized. One of the major challenges instructors/facilitators have in providing regular qualitative assessment to learners is the burden associated with grading especially when the number of learners is high. Automating the grading of qualitative evaluation in form of short answer questions could enhance learning through prompt feedback to learners. It could also motivate instructors to provide more regular evaluations to the learners as the burden of grading becomes shifted to the computer. The focus of this paper is on the development of technology to enhance the auto-grading of short answer questions using natural language processing.

Natural language processing (NLP) has evolved and exploded with the introduction of word2vec [6] which offered a way to represent similarities and relationships between words. During this period, recurrent neural networks (RNN) quickly became preferred because it considers the order of words and can link previous information to the current one. RNN has a shortcoming of vanishing gradient which is addressed by Long short-term memory (LSTM). LSTM learns long-term dependencies and is not limited to just one previous word. Despite the gains of LSTM, researchers continue to push the boundaries to discover better ways for the machine to capture semantics in NLP. Thus, [7] explained that the use of complex RNN with attention is not needed because attention alone is sufficient. This resulted in the introduction of Transformers which has led to the development of pre-trained systems such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT). Reimers and Gurevych [8] developed Sentence-BERT (SBERT) as a modification of the pre-trained BERT to bypass the limitation of BERT in generating sentence embeddings. Sentence-BERT is said to outperform BERT on multiple NLP tasks [8]. This evolution has influenced the choice of language models used in this study.

Automatic grading can be described as the task of assessing natural language responses to questions using computational methods [1, 4]. This assessment is used to test the student's ability to recall, articulate, and apply what has been discussed, taught, and researched. This is

unlike MCQ which majorly evaluates the recognition of correct answers [4, 9, 10, 11]. Unlike MCQ, automating the evaluation of short answer questions does not come easy and it is often faced with a lot of errors due to the inability of existing systems to properly capture the semantics of expressed answers. This is a major reason most examiners have retained the manual grading process despite its demand on time and resources [12]. Several approaches have been implemented in automatic short answer grading systems. These approaches can be summarized into handcrafted feature systems and deep learning systems [12-14]. For handcrafted feature systems, features are manually designed or extracted from the data. Extracting hand-crafted features however can be time-consuming [12, 15]. Traditional machine learning mostly involves extracting handcrafted features. The performance of a handcrafted system depends on the quality of the designed/extracted features. For deep learning systems, deep learning models are used to extract features automatically from the data hence allowing the model to learn the relationship between the scores and the answers.

2.1 Embedding Techniques

2.1.1 Word2vec

Word2vec is a two-layer neural network proposed by Google that converts text data into vectors [25]. Word2vec first generates a vocabulary of unique words from the training corpus and learns the vector representation of each word. The vectors of similar words can be grouped in vector space with Word2vec [11]. The word embedding can be obtained using two methods: Continuous Bag-of-word (CBOW) and Skip-gram model [6].

2.1.2 Bi-LSTM

Bi-LSTM is a model consisting of two LSTMs, one takes input in a forward direction and the other in a backward direction. It is effective for knowing what word precedes and follows a given word in a sentence. Bi-LSTM increases the amount of information available which improves the context available to the algorithm. LSTM network is good for storing long-term memories and was proposed to address the limitation of RNN. The key concept of LSTMs is the cell state and the gates. The cell state can be referred to as the memory of the network; it carries essential information all through the processing of the sequence. The gating mechanism in LSTM is responsible for preserving long-term dependencies in the network. The LSTM cell's operation is summarized in a few steps: it overlooks irrelevant information from the cell state, adds information to the cell state, and calculates the output. Three different gates control the flow of information in an LSTM cell. The forget gate is used to decide what information should be forgotten or kept, the update gate is used to determine which value in the cell state will be updated and the output gate selects what the next hidden state should be. The new hidden state is carried over to the next time step.

2.1.3 Transformers

The transformer is a deep learning model introduced in 2017 which uses an encoder-decoder architecture. Transformers do not process data sequentially which gives room for parallel processing, unlike RNN. The encoding and decoding components are a pack of 6 encoders and decoders respectively. The encoding component receives the input sequence and converts it into a vector. This vector holds information about the entire sequence and sends it to the decoder. The Transformer uses attention to increase the training rate of models.

2.1.3.1 Encoder

As shown in Figure 1, the encoder is divided into two sub-layers: Feedforward neural network and self-attention. Self-attention helps the encoder to focus on other words in the input sequence while encoding a specific word. The feedforward neural network processes the output encoding individually.

An embedding algorithm is used to convert each input word into a vector and positional encoding helps the transformer to know the input order of the words. These two processes take place at the bottom-most encoder.

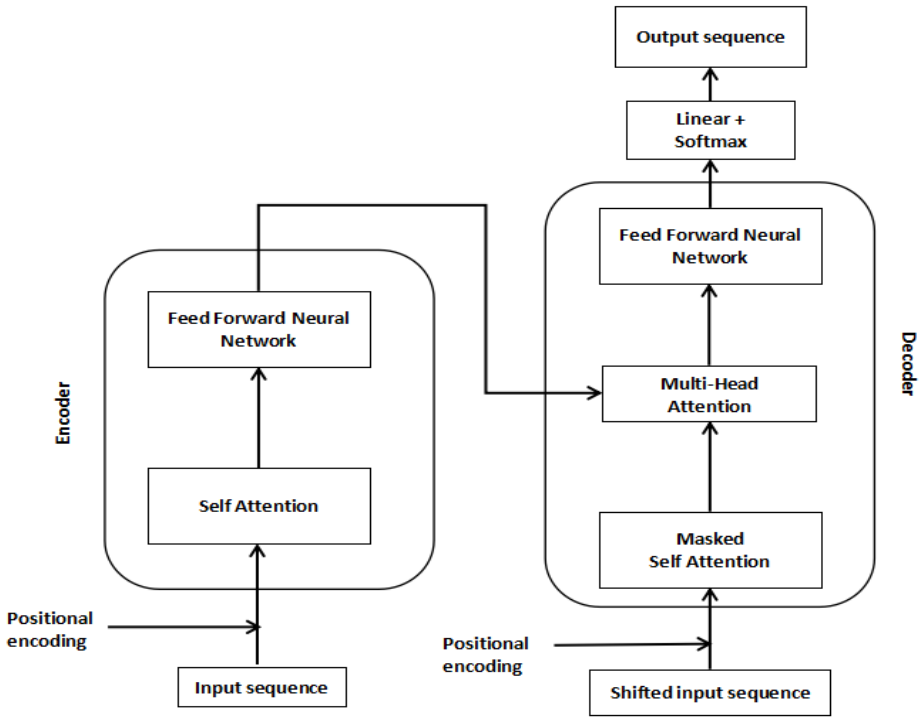

Figure 1. Transformer architecture

2.1.3.2 Decoder

Similar to the encoding layer, the decoder has the self-attention and the feedforward neural network sub-layers but also includes an additional sub-layer which retrieves essential information from the encodings generated by the encoder. The model jointly attends to information from different representation subspaces at different positions using multi-head attention. The masking on the self-attention sub-layer and the embeddings (offset by one position) ensures that the predictions for a position depend only on the known outputs at positions less than the current position [7].

A residual connection is employed around each of the sub-layers in the encoder and the decoder, followed by layer normalization [7]. The linear layer is a simple fully connected neural network which transforms the vector created by the stack of decoders into Logits while the softmax layer converts those scores into probabilities. The word associated with the score with the highest probability is the output of the time step.

2.1.4 BERT

BERT is a bidirectional model built on the transformer architecture and was released in October 2018. It is known to have achieved state-of-the-art performance in specific tasks and outperformed many models [16]. BERT uses attention to understand connections between all words in a sentence irrespective of their position.

A basic transformer consists of an encoding and decoding component but BERT uses only the encoding component to generate language representation. The output word representation from BERT takes into consideration the surrounding words. BERT is huge and expensive to train from scratch and requires high computing power. Google developed the pre-trained model which can be modified and fine-tuned for other tasks. BERT is pre-trained on masked language models and next sentence prediction tasks. Pre-training BERT gives the possibility of being fine-tuned on downstream specific tasks such as question answering, intent detection, and sentiment classification.

The input into the encoder is a sequence of tokens that are converted into vectors first and then processed in the neural network. Every word of an input sequence into BERT is converted into word embedding. BERT embedding has a fixed vocabulary of 30,000 tokens, and each token has 768 features in its embedding. For every word, the dictionary maps the string to the word Ids and the Id is used to fetch the features at the Id position in the lookup table. The input embedding in BERT is the sum of token embedding, segment embedding, and positional embedding as shown in Figure 2.

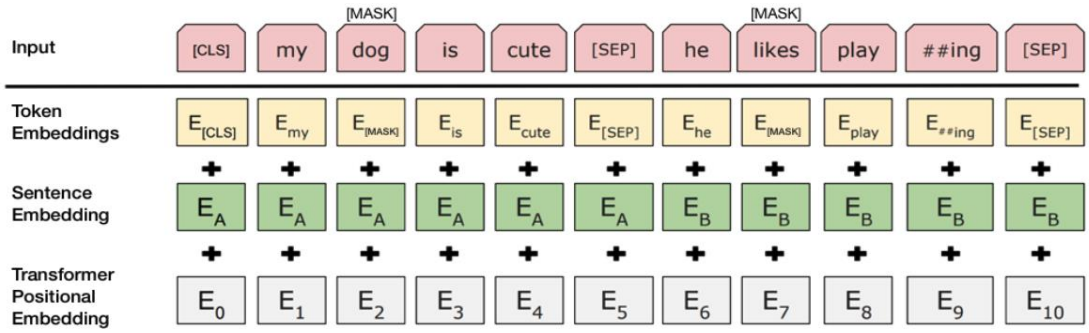

Figure 2. BERT input representation [16]

Every word goes through the process of tokenization which involves the breaking down of input text into a list of tokens present in the vocabulary. BERT handles Out of vocabulary (OOV) words by splitting those words into multiple sub-words or individual characters. For example, the word 'playing' in Figure 2 is split into 'play | ##ing'. The special token [CLS] is usually the first token of every sequence. This token is called the classification token and can be used as the aggregate of the sentence representation for classification. [SEP] is another special token that is appended at the end of each sentence to mark a separation.

Segment embedding is used to differentiate between sentences while positional embedding is added to indicate the position of each token in the sentence. The segment ids are usually a series of 0s and 1s to differentiate two sentences. The sentences passed into BERT are expected to be of the same length. Varying sentence lengths can be handled by padding or truncating to a fixed length. The maximum sentence length allowed by BERT is 512.

2.1.5 SBERT

In the BERT network, sentence embeddings are difficult to generate and it is also unsuitable for computing similarity measures. Sentence-BERT was developed to bypass this limitation. SBERT is a modification of the pre-trained BERT network. It uses Siamese and triplet network structures to generate semantically meaningful sentence embedding. SBERT was fine-tuned on natural language inference (NLI) data. It generates sentence embeddings that outperformed other state-of-the-art sentence embedding methods significantly [8]. These sentence embeddings can be compared using cosine similarity.

2.1.6 OpenAI

OpenAI is an artificial intelligence (AI) research organization based in the United States that was founded in December 2015 with the goal of developing artificial general intelligence. The group describes "safe and beneficial" AI systems as extremely autonomous entities that are more capable than humans in the majority of economically valuable tasks. The OpenAI API can be applied to tasks such as text generation, embeddings, fine-tuning, image generation, vision, text-to-speech, function calling and moderation. Embeddings are a numerical representation of text that can be used to measure the relatedness between two pieces of text. In this paper, "text-embedding-ada-002" model was used to generate sentence embeddings. It was used because it outperformed all the old embedding models on sentence similarity tasks.

Several embedding techniques have been developed and used in different approaches. Of interest to us in this study are: Word2vec, Bi-LSTM, BERT, SBERT and OpenAI.

2.2 Related Works

Researchers have proposed different methods and solutions to address the short answer questions grading task; each proposed method or solution however has its accompanying advantages and limitations. Many approaches have employed similarity measures, machine learning algorithms and deep learning architectures to predict scores.

Mohler, Bunescu and Mihalcea [17] explored the option of improving existing bag-of-words (BOW) approaches to short answer questions grading by using machine learning techniques. A set of 68 features were used to train the machine learning system to compute node-node matches, 36 of these features are based on semantic similarity of four subgraphs and the others are

lexico-syntactic features. Their approach showed that several systems appear to perform better in terms of correlation measures like Pearson while others perform better in terms of minimizing error. The support vector machine for ranking (SVMRank) system outperformed the support vector machine for regression (SVR) system in terms of correlation, however, the SVR system had a lower RMSE. The correlation reported for the BOW-only support vector machine (SVM) model for SVMRank showed an improvement upon the best BOW feature. Compared to the best BOW feature, the BOW-only SVM model for SVR reported a lower RMSE. The alignment features presented in their work are not sufficient to act as a standalone grading system but still show the possibility of an improvement in grade learning systems that consider BOW features. The correlations reported on hybrid systems showed an improvement over the BOW-only SVM system. The best Pearson coefficient of 0.518 was obtained by normalized alignment data on the hybrid system.

Another author [11] employed the use of deep learning-based models to generate paragraph embeddings and showed the effect of the choice of paragraph embedding in auto-grading tasks. Paragraph embedding was used to generate the vector representation for student and reference answers. The cosine similarity between the vectors of student and reference answers was computed and trained on a regression classifier to predict the scores. Their work evaluated four embedding models based on the sum of pre-trained word vectors and three based on trained deep learning models. Their results showed that pre-trained models achieved better results. Doc2vec trained only on sentences in the dataset achieved the best result of 0.569 and 0.797 in terms of Pearson coefficient and RMSE but raises the question: will the same result be achieved on new unseen data from the same domain of questions?

Patil and Agrawal [18] presented a hybrid Siamese network for auto-grading of short answer questions. The model consisted of two sub-models: sentence modelling and similarity measurement. The sentence modelling part used a Siamese architecture of four sub-networks to extract sentence representations. Each sub-network has a word embedding layer, a Bi-LSTM layer, and an attention layer. The similarity model used a fully connected network and logistic regression layer to grade the students' responses. Their proposed model achieved an accuracy of 76% on the SciEntsBank part of the SemEval dataset. The model however misclassified answers where: the keywords present in the reference answer were missing; and the difference in length between the student and the reference answer was large.

A Deep learning-based method with an attention mechanism was proposed by [12] to overcome the problem of low accuracy in the use of handcrafted features for automatic scoring. The model consisted of an embedding layer, a bidirectional recurrent neural network (Bi-RNN) layer, and an attention layer which output the embedding representation of the reference and response answer to the question. Bi-RNN learns the sentence embedding representation of the short answers. The output is passed into a logistic regression function to predict the score. Their approach reported a 10% increase in performance when compared to the baseline model. The only dataset used in the experiment was created by the authors. Other publicly available datasets were not used to ascertain the performance of their model.

The deep descriptive answer scoring model (D-DAS) is a sequential model that compares the performance of Simple LSTM, Deep LSTM and Bi-directional LSTM [19]. The system takes the short answer as input and converts it to glove vector representation using an embedding layer. The LSTM RNN learns from the embedding layer; the embedding vector that corresponds to the final glove vector becomes the semantic representation of the entire answer and is passed as input to the dropout layer. The final layer with a softmax activation function then predicts the score. The result showed that the model with the Bidirectional LSTM layer achieved an average accuracy of 73% on an in-house dataset.

Prabhudesai and Duong [13] presented an architecture that used a combination of feature engineering and deep learning methods to achieve a better result. Their approach compared the performance of the Texas dataset on four architectures – Simple LSTM, LSTM architecture with feature engineering, Siamese Bidirectional LSTM, and Siamese Bidirectional LSTM architecture with feature engineering. Feature engineering was considered to complement the basic Siamese bidirectional LSTM architecture. Features employed for their work include the length of the student answer,

the ratio of the length of the student answer and the reference answer, the number of words in the student answer, and the number of unique words in the student answers. Simple LSTM architecture and LSTM architecture with feature engineering achieved Pearson coefficients of 0.381 and 0.523 respectively. Siamese bidirectional LSTM architecture was also compared with Siamese bidirectional LSTM with feature engineering; Siamese bidirectional LSTM with feature engineering achieved the best Pearson coefficient of 0.655 and showed an improvement of less than 1% over Siamese bidirectional LSTM architecture. A limitation of this work is the use of handcrafted features which is time-consuming to extract and implies that an attempt to improve the model requires redesigning the features.

Tulu, Ozkaya and Orhan [20] proposed a new approach that used Manhattan LSTM (MaLSTM) and sense vectors obtained by semspace, a synset-based sense embedding method leveraging Wordnet. The LSTM architecture takes the synset representation of the student and reference answers as input which are transformed into sentence representation. Manhattan similarity was used to compute the similarity of the two representation vectors. Their model was tested on the Texas dataset and the CU-NLP dataset created from the NLP exam course in the computer engineering department of Cukurova University. A Pearson coefficient of 0.949 and RMSE of 0.040 was achieved on the Texas dataset. One shortcoming of their approach is that an increase in the number of ambiguous words and the number of words represented in the context set results in an increase in the processing time.

In recent times, transfer learning has shown considerably better performance compared to the use of older deep learning techniques because of its robustness. Transformer architecture based on a pre-trained model has yielded outstanding results across a range of NLP tasks. Transformers do not process data sequentially which means they do not need to process the input sequence one after the other. This gives room for parallel processing thereby reducing the training time. This has led to the development of pre-trained systems such as BERT [16] and Generative pre-trained transformer (GPT) [21]. The transformer makes use of attention to increase the training rate of models.

Condor, Litster and Pardos [22] in their work considered the possibility of developing ASAG models that can classify responses from out-of-training sample questions to help educators add new questions to an automatically graded assessment quickly without continuous manual grading. They considered SBERT, Word2Vec, and Bag-of-words. They used a dataset created at the Berkeley Evaluation and Assessment Research (BEAR) centre from a 2019 field test of a critical Reasoning for college Readiness (CR4CR) assessment. SBERT representation performed best with an accuracy of 0.621 when averaged across the classification methods and input combination. The authors acknowledged that the results were not promising for the generalizability of the auto-grading model to unseen questions. They emphasized the importance of finding generalizable models to reduce the time spent in creating correct human ratings.

Schlippe and Sawatzki [23] identified language as a major barrier in learning and developed a system which is based on the Multilingual BERT model to overcome this challenge. Their experiment focused on the possibility of cross-lingual ASAG which encourages students to provide answers in their native language. Zhang et al. [24] experimented with another variant of BERT, MathBERT which is adapted for mathematical content. The model was fine-tuned and scoring examples were used as input to the language model to give more context and promote generalization. The authors observed that there is still a need to develop more effective models for mathematical language and study the fairness of the system.

In the emergence of Large Language Models (LLMs), Chang and Ginter [30] investigated the feasibility of using them, specifically ChatGPT based on GPT-3.5 and GPT-4, for automatic short answer grading (ASAG) in Finnish language. The models were evaluated on 2,000 student answers across ten undergraduate courses, using both zero-shot and one-shot settings. The performance scores include Quadratic-Weighted Kappa (QWK), Tolerance-Adjusted Accuracy (TAA), and Relative Merit Consensus (RMC). The results gotten showed that GPT-4 outperforms GPT-3.5, especially in the one-shot setting, with a QWK of over 0.6. However, the models are more lenient than human graders and struggle with longer answers, indicating room for improvement before

deployment in educational settings. Also, in a study by Latif and Zhai [32], the potential of fine-tuned GPT-3.5 was explored in automatically scoring student responses to science assessment tasks. It compares the performance of GPT-3.5 with BERT model, on six tasks: two multi-label and four multi-class science education problems. Fine-tuning involved training the models on domain-specific datasets of middle and high school students' responses scored by experts. Results show GPT-3.5's performance held up significantly against BERT, with an average 9.1% better accuracy overall. This study showed that GPT-3.5 was better than BERT in handling complex, unbalanced datasets. The authors also highlighted that GPT-3.5's capacity for domain-specific fine-tuning enables it to record better accuracy and consistency towards scalable educational applications. Limitations include restricted dataset diversity and the need for broader comparisons with other emerging AI models. Ethical concerns, including bias and fairness in AI-driven assessments, are also noted for future research.

Obot et. al. [32] presented an automated system for grading essay-based examinations using machine learning and natural language processing (NLP). The study focused on addressing inconsistencies and potential bias in human grading by leveraging the Microsoft Research Paraphrase (MSRP) Corpus and datasets from the Department of Computer Science University of Uyo. The methodology involved preprocessing text (e.g., removing stop words), feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF), and applying logistic regression to predict the semantic similarity between student responses and marking schemes. The model trained on a 70:30 train-test split of the data showed a strong correlation (0.89) between predicted and human-assigned scores, with a mean relative error (MRE) of 0.59.

Del Gobbo et. al. [32] introduced GradeAid, a framework for automatic short answer grading (ASAG) in educational settings, designed to assist instructors by providing automated scoring and feedback. GradeAid uniquely combines lexical features (via TF-IDF) and semantic features (via a BERT cross-encoder) to evaluate students' answers against reference answers. The system integrates both traditional and state-of-the-art NLP techniques and applies various regression models like Support Vector Regressor (SVR) and Random Forest to predict scores. It was tested on multiple datasets, including widely-used English datasets (e.g., ASAP, SciEntBank) and an Italian dataset translated into English. It recorded a normalized root-mean-square error (NRMSE) as low as 0.25, outperforming baseline methods and many state-of-the-art systems. Results from the study also demonstrated that combining lexical and semantic features significantly improved accuracy compared to using either feature set alone.

3. Methodology

In this study, publicly available datasets for ASAG were gathered and used in the experimental design. The performance of embedding techniques - Word2vec, Bi-LSTM, BERT, SBERT and OpenAI on each of the datasets were considered.

3.1 Dataset

Four different datasets were used in this study to compare the performance of the models in predicting the scores:

1. The SemEval 2013 task 7 3-way dataset [26] was created as part of the joint student response analysis and recognizing textual challenge in the text domain. SemEval dataset holds two corpora: Beetle and SciEntBank. The ScientBank which contains student responses to questions in the science domain was used and referred to as SemEval in this study. It holds a single reference answer for each question. The answers are classified as ‘incorrect’, ‘contradictory’ or ‘correct’. The labels were assigned to scores 0 to 2 in this study. The training data of SciEntBank is used in our experiment and split into train and test data.
2. The short answer grading v2.0 dataset (Texas) [17] consists of assignments and exams administered to students taking a basic data structure course at the University of North Texas. It consists of 87 questions distributed across 10 assignments and 2 examinations. Answers were graded on a scale of 0 to 5. There are a total of 2273 responses in the dataset. The answers were graded independently by two human judges and the average of both grades is treated as the gold standard.

3. The Assisted automated short answer grading dataset (AASAG) [27] was created from an exam of a neural network course. The course was taken by graduate students at the University of Applied Sciences Bonn-Rhein-Sieg. There are a total of 646 answers which were graded using an integer scale from 0 to 2. Thirty-eight (38) students took the examination and each exam had 17 questions. The dataset contains questions, student answers, reference answers, and student grades. Additional columns for pre-processed data, word2vec embeddings, cosine similarity between the student and reference answers, and word alignment scores were also included in the dataset.
4. The MIT dataset was created from a series of exams on "Internet Technology" under the Master of Information Technology (MIT) programme at the University of Lagos. A total of 371 answers, provided in response to 5 questions, were graded using a continuous scale from 0 to 5.

3.2 Data Preprocessing

The student and reference answers were preprocessed. The pre-processing stage involves the splitting of words into tokens, removal of stop words, converting words into their root form, padding, and truncating sentences to a fixed length. The pre-processed answers were then passed as input into the language model.

3.3 Experiment

In this paper, two tasks were experimented on. These tasks are Regression and Classification task. A custom deep learning model was built to train the embeddings against the scores. This model also served as head to the BERT model used. This model incorporates a bidirectional Long Short-Term Memory (Bi-LSTM) layer for sequence processing. The `Bi-LSTM` layer enables accessing both backward and forward data, which helps increase the contextual understanding of the sequence. The `return_sequences=True` argument sends the output of every time step, which is then followed by two pooling operations: `GlobalAveragePooling1D` and `GlobalMaxPooling1D`. Average pooling involves taking an average of every feature over time while max pooling involves taking the largest value for every feature. These pooled representations are then concatenated in order to give a better representation of the features. Then, a dropout layer with a rate of 0.3 is applied after pooling, to reduce overfitting by dropping out (setting to zero) a number of input units in the training phase, with a specific probability. For classification task, softmax function is added to the output layer to produce the probabilities of the three classes. The model is then compiled with Adam optimizer and categorical cross-entropy loss function. But for regression task, RELU function is added to the output layer. The model is then compiled with Adam optimizer and mean square error loss function.

In the regression task, the embeddings of reference and student answers were extracted using the embedding techniques and grade score were used as the target column. The model was trained using embeddings of reference and student answers as the dependent variables and score as the independent variable. This experiment was performed on two datasets – Texas and MIT datasets.
For the classification task, the embeddings of reference and student answers were extracted using the embedding techniques. The model was trained using embeddings of reference and student answers as the dependent variables and the grade score as the independent variable. This experiment was performed on two datasets – ASAG and SemEval datasets.

The experiment aims to evaluate the performance of four embedding techniques – Word2vec, Bi-LSTM, BERT, SBERT and OpenAI– on ASAG datasets to ascertain which embedding technique performs best on ASAG tasks. The parameters selected for the models are shown in Table 1.

Table 1: Parameters selected for the experiment

	Word2vec	BI-LSTM	BERT	SBERT	OPENAI
Embedd-ing size	200	100	150	768	1536
Epoch	20	20	20	20	20
Batch size	32	32	32	32	32

An embedding size of 100 was used for word2vec and BI-LSTM while SBERT and BERT used the model's fixed embedding size of 768. An embedding size of 1536 was used for OpenAI. The epoch selected varies and it is based on the performance we observed during

our experiment. A batch size of 2 was selected because of the size of the datasets; they are relatively small. BERT allows a maximum sentence length of 512. We used a sentence length of 150 for BERT. The sentence length used was influenced by the task at hand and the maximum sentence length in our dataset. According to [4], the length of the answer should be between one phrase and a paragraph. We observed that using a sentence length above 100 slowed down the training process and was memory intensive. The implementation was done using Google Colab GPU which is made available by Google for computationally intensive tasks. 20% of the data was used as test data.

3.4 Evaluation Metrics

The evaluation metrics considered in this work are:

Precision - It measures how many of the positively predicted instances for a specific class are actually correct. It helps you understand how well your model is at correctly identifying each class.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

Accuracy - It provides an overall measure of the model's correctness by considering both true positives and true negatives across all classes.

$$accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + False\ Positive + False\ Negative} \quad (2)$$

Recall – It measures a model's ability to correctly identify all relevant instances of a particular class.

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

F1 score - It combines both precision and recall to provide a single, balanced measure of a model's performance. It is especially useful when dealing with imbalanced. The F1 score is the harmonic mean of precision and recall and can be interpreted as the balance between the two metrics.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

False Positive Rate – It measure the proportion of negative instances (instances that do not belong to the positive class) that are incorrectly classified as positive.

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative} \quad (5)$$

False Negative Rate – It measure the proportion of positive instances (instances that belong to a specific class) that are incorrectly classified as negative.

$$FNR = \frac{False\ Negative}{False\ Positive + False\ Negative} \quad (6)$$

Specificity - Also known as the true negative rate, measures the ability of a model to correctly identify negative instances (instances that do not belong to the positive class) among all actual negative instances.

$$specificity = \frac{True\ Negative}{False\ Positive + True\ Negative} \quad (7)$$

Quadratic Weighted Kappa (QWK) - It is a statistical metric used to evaluate the level of agreement between predicted and actual classifications in multiclass classification problems. It's an extension of Cohen's Kappa that takes into account not only the overall agreement between predictions and true labels but also the agreement's quality or "closeness" when dealing with ordinal or ordered categorical data.

$$QWK = 1 - \frac{\sum_{i,j} W_{i,j} O_{i,j}}{\sum_{i,j} W_{i,j} E_{i,j}} \quad (8)$$

Where,

k is the number of classes or categories.

$W_{i,j}$ are the quadratic weights assigned to each pair of classes (i, j).

$W_{i,j}$ is the number of instances predicted as class i and actually belonging to class j

$$E_{i,j} = \frac{Sum\ of\ the\ predicted\ class\ frequencies\ for\ class\ i \times Sum\ of\ the\ actual\ class\ frequencies\ for\ class\ j}{Total\ number\ of\ instances}$$

Matthews Correlation Coefficient (MCC) - Also known as the Phi coefficient, is a metric used to evaluate the quality of predictions in multiclass classification problems. It is a measure of the correlation between the predicted and actual classifications, accounting for both true and false positives and true and false negatives. The MCC is particularly useful when dealing with imbalanced datasets and multiclass problems.

$$MCC = \frac{(TP \cdot TN - FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad (9)$$

Mean Squared Error (MSE) - It calculates the average of the squared differences between the predicted values and the actual target values for each data point in the dataset.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (10)$$

Where,
n is the number of data points in the dataset.
y_i is the actual target value for data point i
$\hat{y}_i$ is the predicted value for data point i

Root Mean Squared Error (RMSE) - It calculates the square root of the average squared differences between predicted and actual values.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (11)$$

Pearson Correlation Coefficient (Pearson's r) - It measures the linear relationship or correlation between the predicted values and the actual target values in a regression problem. It quantifies how well the predicted values and true values align in terms of their linear association.

$$R = \frac{\sum_{i=1}^{n}(y_i - \underline{y_i})(\hat{y}_i - \underline{\hat{y}_i})}{\sqrt{\sum_{i=1}^{n}(y_i - \underline{y_i})^2 \sum_{i=1}^{n}(\hat{y}_i - \underline{\hat{y}_i})^2}} \quad (12)$$

Mean Absolute Error (MAE): It measures the average absolute difference between the predicted values and the actual target values. It quantifies the magnitude of the errors made by the model without considering their direction.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (13)$$

R-squared (R^2) score - Also known as the coefficient of determination, is a statistical metric used to evaluate the goodness of fit of a regression model. It measures the proportion of the variance in the dependent variable (the target) that is explained by the independent variables (the predictors) in the model. In other words, R2 quantifies how well the regression model captures and explains the variation in the observed data.

$$R2 = 1 - \frac{\sum_i(y_i - \hat{y}_i)^2}{\sum_i(y_i - \bar{y}_i)^2} \quad (14)$$

4. Results and Discussion

4.1 Results

The following tables show the results after conducting the experiment.

Table 2: Performance of regression task on Texas dataset while grade score was used as the target column

Model	Mean Squared Error	Root Mean Squared Error	Pearson Coefficient	Mean Absolute Error	R^2 Score
Word2Vec	0.88	0.94	0.45	0.73	0.18
Bi-LSTM	0.89	0.95	0.54	0.68	0.24
SBERT	0.79	0.89	0.61	0.61	0.33
BERT	0.65	0.81	0.71	0.61	0.45
OPENAI	0.57	0.76	0.70	0.52	0.46
BERT (finetuned)	0.51	0.71	0.76	0.48	0.57

Table 3: Performance of regression task on MIT dataset while grade score was used as the target column

Model	Mean Squared Error	Root Mean Squared Error	Pearson Coefficient	Mean Absolute Error	R^2 Score
Word2Vec	0.96	0.98	0.29	0.80	0.10
Bi-LSTM	0.76	0.87	0.54	0.60	0.27
SBERT	0.82	0.91	0.46	0.73	0.20
BERT	0.75	0.86	0.54	0.66	0.27
OPENAI	0.57	0.76	0.70	0.52	0.46
BERT (finetuned)	0.71	0.84	0.63	0.62	0.33

Table 4: Performance of classification task on ASAG dataset while grade score was used as the target column

Model	Accuracy	Precision	Recall	F1Score	Specificity	False positive rate	False Negative Rate	Kappa	Matthew's correlation coefficient
Word2Vec	0.65	0.64	0.65	0.64	0.78	0.21	0.52	0.33	0.37
Bi-LSTM	0.63	0.62	0.63	0.62	0.77	0.23	0.54	0.42	0.34
SBERT	0.67	0.68	0.67	0.66	0.80	0.20	0.49	0.51	0.43
BERT	0.68	0.67	0.68	0.67	0.81	0.19	0.48	0.39	0.42
OPENAI	0.70	0.73	0.70	0.69	0.82	0.19	0.46	0.49	0.46
BERT (finetuned)	0.69	0.68	0.68	0.68	0.82	0.18	0.47	0.56	0.44

Table 5: Performance of classification task on SemEval dataset while grade score was used as the target column

Model	Accuracy	Precision	Recall	F1Score	Specificity	False positive rate	False Negative Rate	Kappa	Matthew's correlation coefficient
Word2Vec	0.64	0.63	0.63	0.64	0.78	0.22	0.53	0.35	0.35
Bi-LSTM	0.64	0.64	0.65	0.64	0.78	0.21	0.52	0.36	0.38
SBERT	0.70	0.69	0.70	0.69	0.82	0.18	0.46	0.44	0.46
BERT	0.71	0.71	0.71	0.70	0.83	0.17	0.45	0.47	0.49
OPENAI	0.73	0.74	0.72	0.73	0.84	0.16	0.43	0.53	0.54
BERT (finetuned)	0.75	0.75	0.75	0.75	0.86	0.14	0.40	0.56	0.56

4.2 Discussion

This section analyses and discusses the results of the experiments conducted on various datasets using different embedding techniques. The experiments include regression tasks predicting teacher scores, as well as classification tasks predicting grades. The discussion is organized based on the nature of the experiments.

For the first experiment on regression task utilizing the Texas dataset, the embeddings from the fine-tuned BERT model performed better than the embeddings of all the other models as shown in Table 2. It reported an MSE of 0.51 and R2 of 0.57. The second-best model was OPENAI with MSE of 0.57 and R2 of 0.46. OPENAI also recorded the best performance in regression task using the MIT dataset as shown in Table 3. It recorded a Mean Squared Error value below the threshold of 0.57 and R2 of over 0.46. The performance of OPENAI on the MIT data set confirmed that the OPENAI embeddings captured the semantic of the sentences as it recorded the same MSE scores for both Texas and MIT datasets. However, the global performance of all the models on MIT dataset is lower than their performance on the Texas Dataset. This could be because the MIT dataset is smaller than the Texas Dataset.

The second experiment is a classification task where grade scores were taken as the target grade. According to Table 4, the embeddings from OPENAI performed better than all the other models recording an accuracy of 0.70 and a precision of 0.73. However, all the models had slightly lower scores for recall and F1 Score meaning that true positives were under-detected. This could be improved by balancing the classes. However, in case of SemEval dataset as seen in Table 5, the embeddings from the fine-tuned BERT recorded an accuracy of 0.75. It is notable to see that BERT embeddings are very effective in understanding the semantics between sentences whenever classification task is concerned.

From experiments with the datasets in this study, embeddings from fine-tuned BERT, especially for tasks involving semantic similarity performed best. This showcased the advantage of finetuning to a domain. On the other hand, embeddings from OPENAI performed better than other models because of the size of the embeddings. Furthermore, the performance differences across datasets point out the significance of the datasets' properties (such as size, length of sentences) in training and evaluating a model.

From the results gotten, it could be inferring that OpenAI and BERT (especially when fine-tuned) benefit from the use of a transformer architecture, which allows understanding contextual aspects in sentences. Word2Vec has static embeddings that are incapable of this, making them less efficient especially in grading tasks with concern to subtle answers. Also, the embeddings offered by OpenAI are trained on extremely large per-corpora collections which makes them suitable for any task. The large embedding size of OpenAI (1536) allows more complex answers to be graded more accurately than within relations of BERT.

5. Conclusion

In this paper, a comparative study of the performance of the embedding features of four language models tested on four short answer grading datasets was carried out. We showed that the embedding features of the answers from finetuning BERT performed better in terms of RMSE on the Texas dataset and accuracy on the SemEval and AASAG datasets. This study provides some information on what automatic

short answer grading entails and the benefits to both students and instructors.

Since this work was only evaluated on specific domain questions, experimenting with non-domain questions in future studies would help ascertain the possibility of achieving significant results with the same model on different domains with little or no modification. Generating a dataset specifically for this task and evaluating the models on this would help determine the consistency of the performance reported and contribute to the number of datasets available for the ASAG task.

Apart from the flexibility and convenience that online assessment provides, it is also open to academic integrity violations. One such growing threat is adversarial attack. Numerous state-of-the-art models appear to be vulnerable to adversarial attacks on various data sets. According to Manoharan and Ye [28], maintaining academic integrity is a major contributor to the difficulty in assessing students online. Filighera, Steuer and Rensing [29] discovered triggers that allow the student to pass with a threshold of 50%. One could implement protective measures to increase the robustness of the automatic grading system.

Is the short answer question ready to replace MCQ in an online examination? ASAG is still regarded as a difficult task due to the complexity of grading short answer questions. Due to the accuracy level achieved in this work, we would recommend that short answer grading systems should be considered for formative evaluation rather than summative evaluation. An accuracy of 95% and above would mean the system could be used for summative evaluation and applied to large classes. Continuous research work is needed to improve the grading efficiency of short answer grading systems.

References

[1] Suzen, N., Gorban, A.N., Levesley, J. and Mirkes, E.M., 2020. Automatic short answer grading and feedback using text mining methods. Procedia Computer Science, 169, pp.726-743.

[2] Gomaa, W.H. and Fahmy, A.A., 2019. Ans2vec: A scoring system for short answers. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 586-595). Springer, Cham.

[3] Sinha, P., Kaul, A., Bharadia, S. and Rathi, S., 2018. Answer evaluation using machine learning. In Conference Paper.

[4] Burrows, S., Gurevych, I. and Stein, B., 2015. The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education, 25(1), pp.60-117.

[5] Ghavidel, H.A., Zouaq, A. and Desmarais, M.C., 2020. Using BERT and XLNET for the Automatic Short Answer Grading Task, In Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020), 1, pp.58–67.

[6] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[8] Reimers, N. and Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[9] Sriwanna, K., 2018. Text classification for subjective scoring using K-nearest neighbors. In 2018 International Conference on Digital Arts, Media and Technology (ICDAMT) (pp. 139-142). IEEE.

[10] Pribadi, F.S., Adji, T.B., Permanasari, A.E., Mulwinda, A. and Utomo, A.B., 2017. Automatic short answer scoring using words overlapping methods. In AIP Conference Proceedings (Vol. 1818, No. 1, p. 020042). AIP Publishing LLC.

[11] Hassan, S., Fahmy, A.A. and El-Ramly, M., 2018. Automatic short answer scoring based on paragraph embeddings. International Journal of Advanced Computer Science and Applications, 9(10), pp.397-402.

[12] Gong, T. and Yao, X., 2019. An attention-based deep model for automatic short answer score. International Journal of Computer Science and Software Engineering, 8(6), pp.127-132.

[13] Prabhudesai, A. and Duong, T.N., 2019. Automatic short answer grading using siamese bidirectional LSTM based regression. In 2019 IEEE International Conference on Engineering, Technology and Education (TALE) (pp. 1-6). IEEE.

[14] Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V. and Arora, R., 2019. 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) (pp. 6071-6075).

[15] Surya, K., Gayakwad, E. and Nallakaruppan, M.K., 2019. Deep learning for short answer scoring. International Journal of Recent Technology and Engineering, 7(6), pp.1712-1715.

[16] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[17] Mohler, M., Bunescu, R. and Mihalcea, R., 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (pp. 752-762).

[18] Patil, P. and Agrawal, A., 2018. Auto Grader for Short Answer Questions.

[19] George, N., Sijimol, P.J. and Varghese, S.M., 2019. Grading descriptive answer scripts using deep learning. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(5).

[20] Tulu, C.N., Ozkaya, O. and Orhan, U., 2021. Automatic Short Answer Grading with SemSpace Sense Vectors and MaLSTM. IEEE Access, 9, pp.19270-19280.

[21] Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training.

[22] Condor, A., Litster, M. and Pardos, Z., 2021. Automatic short answer grading with SBERT on out-of-sample questions. In Proceedings of the 14th International Conference on Educational Data Mining.

[23] Schlippe, T. and Sawatzki, J., 2022. Cross-Lingual Automatic Short Answer Grading. In Artificial Intelligence in Education: Emerging Technologies, Models and Applications (pp. 117-129). Springer, Singapore.

[24] Zhang, M., Baral, S., Heffernan, N. and Lan, A., 2022. Automatic Short Math Answer Grading via In-context Meta-learning. arXiv preprint arXiv:2205.15219.

[25] Elalfi, A., Elgamal, A. and Amasha, N., 2019. Automated Essay Scoring using Word2vec and Support Vector Machine. International Journal of Computer Applications, 177(25), pp.20–29.

[26] Dzikovska, M.O., Nielsen, R.D., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I. and Dang, H.T., 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. NORTH TEXAS STATE UNIVERSITY DENTON.

[27] Kishaan, J., Muthuraja, M., Nair, D. and Plöger, P.G., 2020. Using Active Learning for Assisted Short Answer Grading. In ICML 2020 Workshop on Real World Experiment Design and Active Learning.

[28] Manoharan, S. and Ye, X., 2020. On upholding academic integrity in online examinations. In 2020 IEEE Conference on e-Learning, e-Management and e-Services (IC3e) (pp. 33-37). IEEE.

[29] Filighera, A., Steuer, T. and Rensing, C., 2020. Fooling automatic short answer grading systems. In International conference on artificial intelligence in education (pp. 177-190). Springer, Cham.

[30] Chang, L.H. and Ginter, F., 2024, March. Automatic Short Answer Grading for Finnish with ChatGPT. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 21, pp. 23173-23181).

[31] Latif, E. and Zhai, X., 2024. Fine-tuning ChatGPT for automatic scoring. Computers and Education: Artificial Intelligence, 6, p.100210.

[32] Obot, O.U., Obike, P. and James, I., 2024. Automated Marking System for Essay Questions. Journal of Engineering Research and Reports, 26(5), pp.107-126.

[33] Del Gobbo, E., Guarino, A., Cafarelli, B. and Grilli, L., 2023. GradeAid: a framework for automatic short answers grading in educational contexts—design, implementation and evaluation. Knowledge and Information Systems, 65(10), pp.4295-4334.