big data and cognitive computing

MDPI

Article

Empowering Short Answer Grading: Integrating Transformer-Based Embeddings and BI-LSTM Network

Wael H. Gomaa [1,2], Abdelrahman E. Nagib [2], Mostafa M. Saeed [2], Abdulmohsen Algarni [3] and Emad Nabil [4,5,*]

[1] Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni Suef 62511, Egypt; wahassan@msa.edu.eg
[2] Faculty of Computer Science, 6th of October Campus, MSA University, Giza 12566, Egypt; abezzeldin@msa.edu.eg (A.E.N.); mmsaeed@msa.edu.eg (M.M.S.)
[3] Faculty of Computer Science, King Khalid University, Abha 61421, Saudi Arabia; a.algarni@kku.edu.sa
[4] Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, Saudi Arabia
[5] Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt
* Correspondence: e.nabil@fci-cu.edu.eg

Abstract: Automated scoring systems have been revolutionized by natural language processing, enabling the evaluation of students' diverse answers across various academic disciplines. However, this presents a challenge as students' responses may vary significantly in terms of length, structure, and content. To tackle this challenge, this research introduces a novel automated model for short answer grading. The proposed model uses pretrained "transformer" models, specifically T5, in conjunction with a BI-LSTM architecture which is effective in processing sequential data by considering the past and future context. This research evaluated several preprocessing techniques and different hyperparameters to identify the most efficient architecture. Experiments were conducted using a standard benchmark dataset named the North Texas Dataset. This research achieved a state-of-the-art correlation value of 92.5 percent. The proposed model's accuracy has significant implications for education as it has the potential to save educators considerable time and effort, while providing a reliable and fair evaluation for students, ultimately leading to improved learning outcomes.

Keywords: automatic scoring; short answer grading; transformers; deep learning; AI in education

Citation: Gomaa, W.H.; Nagib, A.E.; Saeed, M.M.; Algarni, A.; Nabil, E. Empowering Short Answer Grading: Integrating Transformer-Based Embeddings and BI-LSTM Network. *Big Data Cogn. Comput.* **2023**, *7*, 122. https://doi.org/10.3390/bdcc7030122

Academic Editors: Tim Schlippe and Matthias Wölfel

Received: 24 April 2023
Revised: 8 June 2023
Accepted: 15 June 2023
Published: 21 June 2023

1. Introduction

The field of natural language processing (NLP) merges computer science, linguistics, and machine learning to teach computers to interpret human language similar to humans. Progress in machine learning, specifically through deep learning methods, has significantly improved NLP. NLP encompasses the following two primary categories: natural language understanding (NLU), where computers can accurately comprehend human language, and natural language generation (NLG), where computers generate natural language. NLP has a wide range of applications, encompassing various tasks such as short answer grading, essay scoring, machine translation, OCR post-correction, metadata extraction, topic detection and tracking, question answering, and chatbots. NLP is a challenging field due to the various natural languages with unique syntactic rules, which often lead to ambiguous meanings that vary depending on the context. Ambiguity is a prevalent issue in NLP and refers to words and phrases with multiple possible interpretations. In response to the COVID-19 pandemic, the use of technology in the learning process has increased, with NLP being utilized in all academic institutions to facilitate the grading of assignments, quizzes, and exams. In addition, the significance of research in this automatic short answer grading domain is anticipated to grow in the future, as the development of such methodologies will expedite the assessment of student responses through automated means, thereby eliminating the need for human evaluators. Furthermore, this approach has

the potential to address the issue of discrepancies in grading due to varying perspectives, thus enhancing the consistency and reliability of the evaluation process. Deeper learning-based methodologies, including transformer models, have been increasingly utilized in the development of ASAG systems. In our model, we utilize transformer models such as T5 and BERT in the embedding phase to convert students' answers and model answers into numerical vectors. Transformers excel at capturing contextual relationships and encoding semantic information. We move beyond older methods such as one-hot encoding and leverage the advanced representation learning capabilities of transformers to generate high-quality embedding vectors. In the training phase, we introduce a BI-LSTM as the neural network architecture for learning and predicting grades. This complements the capabilities of transformers by capturing long-term dependencies and understanding the coherence and flow of answers. We incorporate manual evaluation in training to learn from human expertise and combine it with transformer-based embeddings for improved grading. The BI-LSTM architecture, applied to the transformer-based embedding vectors, captures sequential dependencies and interprets relationships within answers. This enables our model to assess coherence and quality beyond surface-level similarities, thus, enhancing the accuracy and interpretability of short answer grading.

The following contributions were achieved:

A. The most appropriate pretrained model was identified for embedding all student answers and model answers in the North Texas data structure dataset.
B. The developed neural network was experimentally evaluated with the North Texas data structure dataset, and the most advanced results in this task for this dataset were achieved.
C. The optimal preprocessing techniques that can be utilized for this task was determined.

This paper is organized as follows: Section 2 provides a comprehensive literature review, Section 3 describes the methodologies used in each phase of the proposed model, Section 4 presents the experimental results and accompanying discussion, and Section 5 summarizes the conclusion and proposes avenues for future research.

2. Literature Review

In the preceding 10 years, many strategies have been suggested for automated brief response grading systems. Here, we present the most promising approaches that have been implemented on the North Texas dataset, which is composed of 87 questions from 10 different assignments that received an average of 30 student answers per question, totaling 2442 student answers. Moreover, for each question, a designated model answer that was the best fitting answer for each question has been provided, and the grading process involved comparing a student's response to the corresponding model solution and grading it according to the degree of similarity. For each question, there were two evaluators, and then we mainly focused on the average grade of these two evaluators, as shown in Figure 1, which shows a sample question, the model answer for this question, a student's answer, two manual evaluators' grades, and their average grade. Then, we evaluated our model predictions compared with the average grade. This method ensures a systematic and objective assessment tailored to each individual question. The concept of the text similarity technique has proven its efficiency and capability to be used to handle the ASAG task.

The authors in Ref. [1] applied a four-stage methodology to the North Texas data structure dataset. Initially, preprocessing techniques were employed, with lemmatization and lowercasing proving to be the most effective. Then, the second stage was the processing stage in which the student answer was compared with the model answer using different sting, semantic, and embedding techniques.

In the final stages, the student's answer was graded based on its similarity to a model answer, and this predicted grade was compared with the manually given grade. The process achieved a correlation score of 65.12%.

Question: What is the role of a prototype program in problem solving? **Model Answer:** To simulate the behavior of portions of the desired software product.				
	Students' answers	**Grader 1's Manual Marking**	**Grader 2's Manual Marking**	**Average Manual Mark**
Student's Answer 1	High risk problems are address in the prototype program to make sure that the program is feasible. A prototype may also be used to show a company that the software can be possibly programmed.	4	3	3.5
Student's Answer 2	To simulate portions of the desired final product with a quick and easy program that does a small specific job. It is a way to help see what the problem is and how you may solve it in the final project.	5	5	5
Student's Answer 3	To address major issues in the creation of the program. There is no way to account for all possible bugs in the program, but it is possible to prove the program is tangible.	3	2	2.5

Figure 1. North Texas dataset sample.

The authors in Ref. [2] introduced a proposed model that was mainly based on two stages, the first stage was the feature extraction phase which involved converting the model answer and the student answer into embedding and the second stage was the fine-tuning phase which was mainly based on fine-tuning the BERT pretrained model and it was augmented with a linear regression layer to predict the score based on the model answer and the student answer. The inputs, without prior embedding, underwent tokenization into word pieces, and a separator token was used to differentiate the model answer and the student answer. Finally, this approach achieved a correlation score of 78%.

The authors in Ref. [3] found that although sentence embeddings were useful for scoring student answers that were in-domain, they might not perform well for out-of-domain answers and could be influenced by non-sentential forms. To address this, the researchers proposed a novel feature encoding approach called histogram of partial similarities (HoPS) that considered token-level partial similarities, extending it to include part-of-speech tags (HoPSTags) and question-type information. By combining these features with sentence embedding-based features, the researchers achieved an improved grading performance. This approach achieved a correlation score of 57%.

In Ref. [4], the authors proposed a four-step methodology that involved preprocessing, feature extraction, training and testing, and evaluation. They used raw text and transfer learning in preprocessing, created high-dimensional vector representations of answers using pretrained embeddings, and calculated their cosine similarity in feature extraction. Models were trained on 70% of the data and tested on 30%, using various regression methods. The highest correlation score achieved was 48.5% with isotonic regression.

The study by Ref. [5], based in the UK, focused on the automatic scoring of student responses and aimed to provide feedback on the correctness or incompleteness of answers. Techniques employed included hamming similarity in Ref. [6], regression, classification, and clustering, but the specifics were not detailed. The method, tested on the North Texas dataset, achieved a correlation of 81% but did not consider semantics or synonyms.

The Ans2vec method, recommended [7], was a straightforward and effective model for evaluating succinct answers. It used a skip-thought vector model to evaluate short answers. This model, trained on extensive data, captured semantic and syntactic information of the text. Applied to the challenging North Texas dataset, it achieved a correlation of 63%.

The grading of student responses was addressed in Ref. [8] through a two-stage approach. The first stage involved using the maximum marginal relevance technique to create a reference response from the student's answer. For the second stage, the researchers proposed using a GAN longest common substring, which is an extension of the LCS that can measure the similarity between sentences of varying lengths and can determine a student's grade. The methodology was evaluated on the North Texas Dataset, and a correlation value of 0.468 was obtained.

The research presented in Ref. [9] put forward a paragraph embedding approach to predict a student's grade. This method generated vector representations of both the student's response and the reference response, and then calculated the cosine similarity between the two vectors. Word embedding techniques, such as Glove [10,11] and Fasttext, were employed to aid in the process [12]. Furthermore, sentence embedding methods, including Skip-thought and InferSent, were used to create the vector representations. The proposed strategy was evaluated using the North Texas dataset, and a correlation score of 56.9% was achieved.

In Ref. [13], the main emphasis was on vector-based techniques. To compute the similarity measures among seven different similarity models including Disco, Block distance, Jiang-Conrath, and Lesk, all stop words were initially removed from the data by the researchers. They also used the vector summing approach and sentence-level similarity metrics. The North Texas dataset was used to analyze each of these methods, and the highest correlation achieved was 58.6%.

The research described in Ref. [14] proposed a model with the following three modules: data preprocessing which included removing stop words and applying either lemmatization or stemming; similarity measures using a different method such as corpus or knowledge-based approaches [15] or corpus-based approaches [16] or Word2Vec; and a third module which was a scaling module for grading student responses. The model was evaluated on the North Texas dataset, achieving a correlation score of 55.5%. Table 1 summarizes the literature review rankings based on the correlation scores for the North Texas dataset.

Table 1. Literature review summary.

References	Year	Approach	Correlation Score
[5]	2020	Clustering and regression analysis: Apply different approaches such as calculating hamming distance, applying regression classification, and applying clustering too.	81%
[2]	2022	Embedding and transformers: Convert student and model answers using the Ans2Vec approach [7], and then fine-tune the BERT model and add a linear layer to predict the student grade.	78%
[1]	2022	Embedding and text similarity techniques: Compare student answers to model answers using various string, semantic, and embedding techniques.	65.12%
[7]	2019	Embedding: Embed the student and model answers using skip-thought vectors, and then obtain the similarity between them.	63%
[13]	2016	Embedding and text similarity techniques: Use Disco, Block distance, Jiang-Conrath, and Lesk similarity techniques between the student and model answer vectors.	58.6%

Table 1. *Cont.*

References	Year	Approach	Correlation Score
[3]	2018	Embedding and similarity techniques: Apply some new feature extraction techniques and combine them to predict a student grade.	57%
[9]	2018	Embedding and deep learning: Convert student and model answers into vectors using paragraph embedding, and then apply cosine similarity between them.	56.9%
[14]	2016	Embedding and deep learning: Apply corpus and knowledge-based similarity techniques, and then calculate the similarity using Word2Vec and Glove model.	55.0%
[4]	2020	Embedding and text similarity techniques: Use an embedding pretrained model to convert student and model answers into vectors, and then calculate the cosine similarity between them	48.5%
[8]	2018	Text similarity techniques: Use the maximum marginal relevance technique to create a reference response from a student's answer, and then use a GAN longest common substring to compute the similarity between the student and model answer.	46.8%

The following related works are mentioned as they are working on some deep learning approaches; however, the authors have tested these approaches on other datasets, not our dataset.

In Ref. [17], the authors worked on seven different comprehension datasets and proposed a strategy for assessing semi-open-ended short answer questions. The authors developed an automatic grading model for such questions by incorporating domain-general information from Wikipedia and domain-specific knowledge from graded student short answers. This integration was facilitated by using a continuous bag of words (CBOW) model to generate word vectors. These word vectors then acted as inputs for the long short-term memory (LSTM) to be passed to the classifier, and thus, to predict a student's grade.

In Ref. [18], the authors worked on the ASAP dataset and proposed a promising neural network model that was mainly based on three layers. The first layer was the word embedding layer which converted the student answers into a vector to be pushed to the BI-LSTM layer that was mainly focused on extracting the contextual features, and then the output was passed to the attention layer which was mainly used to extract the best features related to the score of each answer.

In Ref. [19], the authors introduced a deep learning model that incorporated an attention mechanism, a bidirectional RNN unit, and pretrained word embeddings for automatic short answer scoring. Two experiments were conducted to compare this model with traditional linear regression and latent semantic analysis.

Other related works also proposed very promising approaches but on datasets with different languages such as the Arabic language.

In Ref. [20], the authors suggested a system for an Arabic dataset they acquired from various schools in the Qalyubia Governorate of the Egypt Arab Republic. They started by preprocessing this data, and then used a hybrid approach to forecast utilizing this dataset. They integrated the optimization method grey wolf optimizer (GWO) with the deep learning technique LSTM. By automatically choosing the best dropout and recurring dropout rates as hyperparameters, the GWO was used to optimize the LSTM. With this strategy, the LSTM model's generalization was enhanced, overfitting was avoided, and

the ability to predict student test scores was improved. The system's ultimate goal was to enhance learning and to reduce the time and effort required of educators.

3. Methodology

As shown in Figure 2, our proposed approach consisted of four stages after inputting the North Texas dataset. The first stage is the preprocessing which involves different techniques such as applying lowercase, stemming, lemmatization, removing stop words, and removing special characters; the second stage involves embedding both the student answer and the model answer using powerful pretrained models such as a T5-XL embedding model, BERT-base, and all-distilroberta-v1; the third stage is the training of the deep neural network by experimenting with different input-layer types such as BI-LSTM and LSTM layers; the final stage is predicting the student's grade and comparing it with the actual grade that is given by the manual examiners.

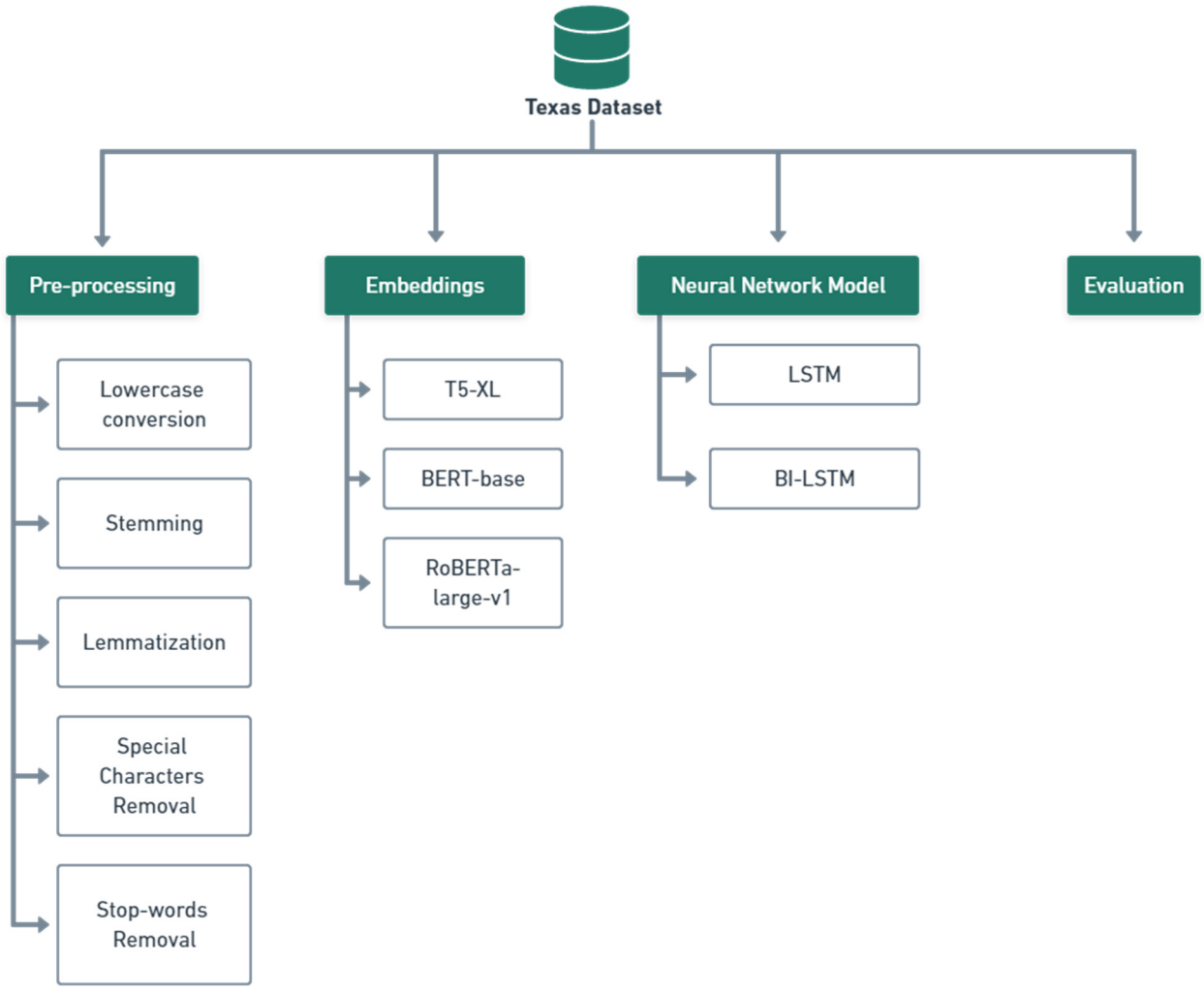

Figure 2. The general architecture of the methodology.

3.1. Preprocessing Stage

Figure 2 shows the preprocessing stage and highlights the utilization of various linguistic techniques in this task. The first strategy involves converting all phrases (including the question, reference answer, and student response) to lowercase letters, since capitalization does not hold significance in NLP. The second and third techniques employed are stemming and lemmatization. Stemming involves removing the last few characters from a word without considering its meaning, while lemmatization transforms a word into its meaningful base form. These steps are performed after splitting each phrase into its

constituent parts based on spacing. Moreover, the removal of all stop words and special characters from the phrases is also necessary.

3.2. Embedding Stage

The embedding stage, which is the second phase after preprocessing, has proven via trials how crucial it is to use a pretrained model to carry out the operation. The effectiveness of this phase relies on the computational capabilities of transformers, and a visual representation of the overall architecture of transformers can be observed in Figure 3. Different potent pretrained models such as the T5 x-large model [21] with 770 million trainable parameters, all-distilroberta-v1 with 125 million parameters, all-roberta-large-v1 with 355 million trainable parameters, and BERT-base-nli-mean-tokens with 110 million trainable parameters have been used to accomplish this sentence transformation stage. As shown in Table 2, three embedding models are compared in terms of their training corpus size, pretraining task, number of layers, and finally the total number of parameters. In addition, the table illustrates the configurations of various embedding models. It is apparent that the T5 embedding model, which is trained on 750 GB of data, surpasses the other models in terms of parameter quantity. This substantial parameter count certainly augments its performance, particularly for delicate tasks such as short answer grading and these results are shown in the Experimental Results, Section 4.2.

Among the pretrained models, the T5 model showed the best results. It was trained on a variety of supervised and unsupervised tasks [22]. T5 has a unique prefix for each task's input, allowing it to handle a variety of tasks out of the box. The teacher-learning approach was used, requiring input and target sequences for training. The input sequence is provided to the model via input ids, and the decoder input id array sends the target sequence, which begins with a start sequence token, to the decoder. The T5 model was trained on the 700 GB Colossal Clean Crawled Corpus dataset, a cleaned version of the Common Crawl dataset that included only English text. It achieved state-of-the-art performance on several NLP benchmarks and could be adapted to various downstream applications.

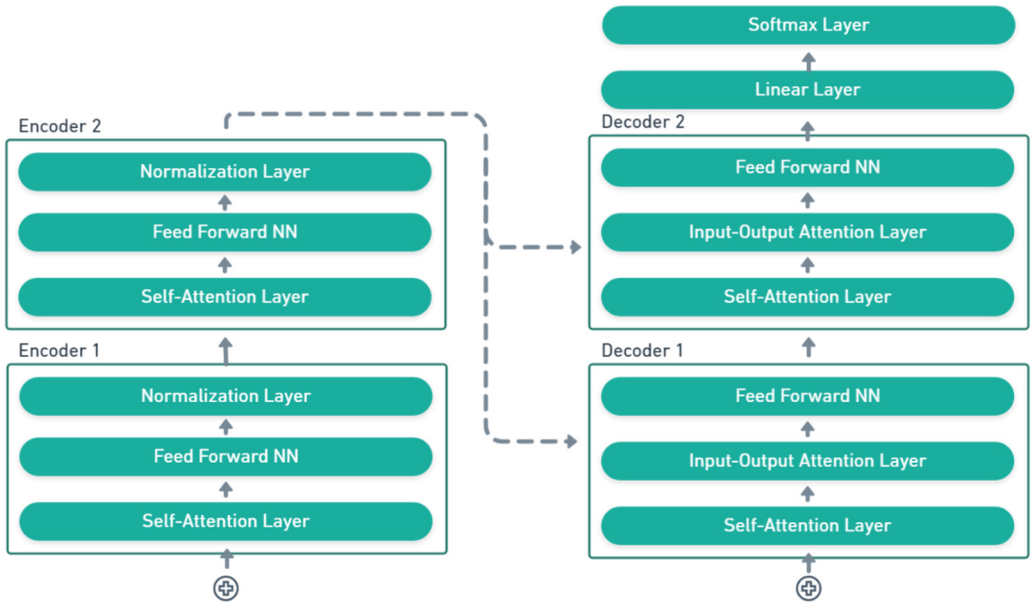

Figure 3. Transformers' general architecture.

Table 2. T5-XL, BERT-base, and Roberta-large-v1 configurations.

Model	Architecture	Training Corpus Size	Pretraining Task	Layers	Total Parameters
T5-XL	Transformer	750 GB (C4 Common Crawl)	Denoising autoencoder	Customizable (ranges from small (6 layers) to 3B (24 layers))	Customizable (ranges from 60 M to 11 B)
BERT-base	Transformer	16 GB (BooksCorpus and English Wikipedia)	Masked language model	12	125 M
Roberta-large-v1	Transformer	160 GB (Common Crawl News dataset, BooksCorpus, and English Wikipedia, etc.)	Masked language model with dynamic masking	24	355 M

3.3. Neural Network Architecture

Figure 4 displays the neural network architecture that was identified as the most successful after conducting multiple training experiments. The model's potency is due to its use of a bidirectional LSTM (BI-LSTM) [23], which is a sequence processing model consisting of two LSTMs. One of the LSTMs processes the input in a forward direction, while the other processes it in a backward direction. Layers with dropouts are added; dropout regularization is a technique used to prevent overfitting in the model. During the training process, dropout layers randomly remove a percentage of the nodes from the network, forcing the remaining nodes to learn more generalized features of the data.

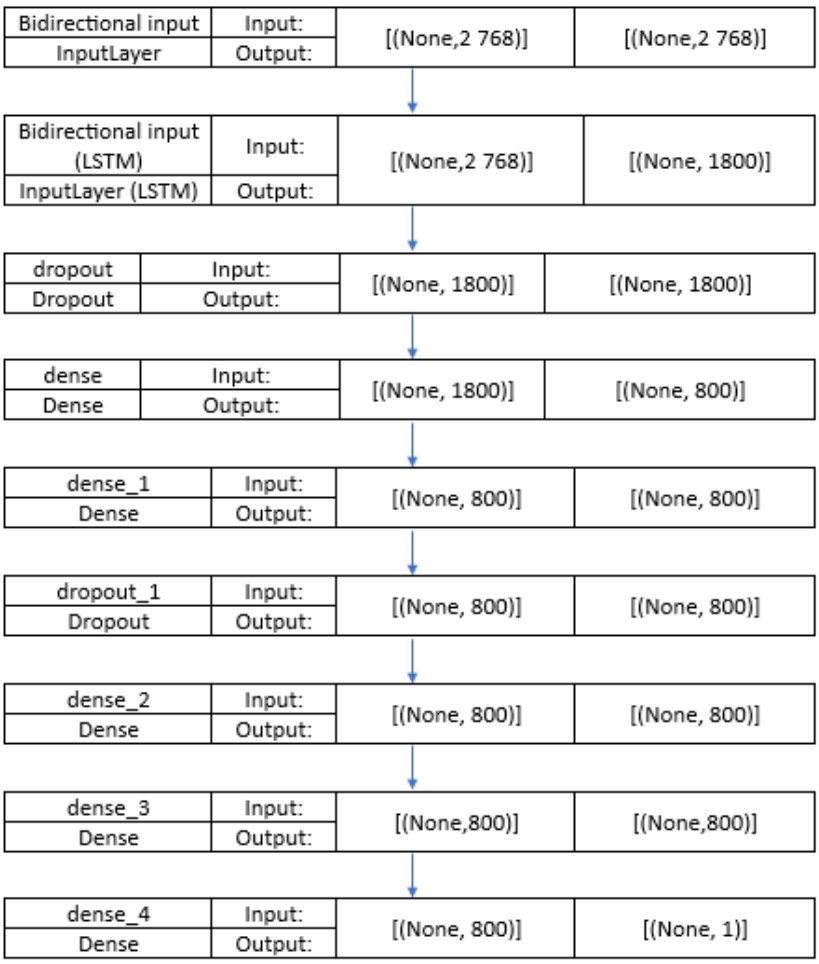

Figure 4. Best neural network architecture.

For our experiments, the rectified linear unit (ReLU) [24] activation function was utilized. Fundamentally, in neural networks, activation functions are used to add nonlinearity to the network by applying them to each neuron's output. Among all other activation functions, ReLU was our selected option because of how easy it is to use and how well it works to stop the vanishing gradient issue. In deep neural networks, a phenomenon known as the vanishing gradient problem occurs when the gradients become very small, making it challenging for the network to efficiently update its weights. The network can efficiently learn the properties of the data and can create reliable predictions by utilizing the ReLU activation function throughout all of the tests.

3.4. Predicting and Evaluating the Grade

During this phase, the main objective was to develop a deep learning model that could predict a student's grade rapidly and precisely. To assess the accuracy of the model's output, correlation evaluation metrics were employed. Then, these predicted grades were compared to the average of the two evaluators' grades in the North Texas dataset. Although several studies have proposed various methods for scaling similarity values to corresponding grades [25–27], it is important to note that grades were not scaled after each prediction in this research. This was because the evaluators' responses used decimal numbers instead of integer values.

4. Results and Discussion

This section covers the evaluation metrics, experimental outcomes, and a comparative analysis between our proposed model and previous works. Several experiments were conducted with our neural network, which utilized a pretrained model to incorporate both student and model answers, along with a BI-LSTM input layer. Additionally, various preprocessing techniques, embedding pretrained models, input layer types, and model hyperparameters were employed. The best experiment resulted in a significant increase in the correlation score and achieved a state-of-the-art score of 92.80.

4.1. Evaluation Metrics

Various evaluation metrics for short answer grading models are available based on the desired output and required analysis, as noted in [28–30]. In this research, correlation coefficients were employed to assess the correlation between automatic and manual marks. Pearson's correlation is the most commonly used method in statistics to evaluate the strength and presence of a linear relationship between two variables. In the training phase, mean squared error (MSE) was utilized for the grade that was given to each student's answer vector after comparing it with the model answer vector. However, in the testing phase, the sole focus was on Pearson's correlation because it has been used in all previous works, and the intention was to compare our results to those of other works.

4.2. Experimental Results

This subsection presents all the experiments conducted with the proposed neural network in terms of its architecture, hyperparameters, and Pearson's correlation outcomes. The best outcomes were achieved consistently without any preprocessing techniques, except for converting all student and model answers to lowercase letters. Some neural network hyperparameters were kept constant during all the experiments, since modifications to them resulted in inadequate outcomes. Table 3 displays the outcomes of all the experiments and its records are grouped using the embedding model, with the lowercase conversion being the sole preprocessing function utilized. There were also some fixed hyperparameters and, although optimizing the hyperparameters was an important step at the beginning of the research, manual fine tuning for the learning rate was implemented until we achieved the most promising one in our evaluation. We do recognize the importance of optimizing hyperparameters in machine learning and deep learning models, especially the learning rate, but our current research trajectory and our main focus were more prominently oriented

towards experimenting with diverse embedding models and determining the best one for our current task. The fixed hyperparameters are:

1. No. of layers = 8
2. No. of dropout layers = 2
3. No. of hidden dense layers = 4
4. Batch size = 3
5. Learning rate = 0.0001
6. Activation function = ReLu

Table 3. Results of all the experiments.

Model No.	Split Size	Input Layer	Val_mse	Testing Corr.
1-T5	**0.2**	**BI-LSTM**	**0.109**	**92.80%**
2-T5	0.3	BI-LSTM	0.194	85.20%
3-T5	0.4	BI-LSTM	0.307	80.00%
4-T5	0.2	LSTM	0.109	81.67%
5-T5	0.3	LSTM	0.194	74.72%
6-T5	0.4	LSTM	0.307	65.80%
7-bert base	0.2	BI-LSTM	0.183	89.10%
8-bert base	0.3	BI-LSTM	0.183	70.90%
9-bert base	0.4	BI-LSTM	0.183	69.90%
10-bert base	0.2	LSTM	0.766	71.25%
11-bert base	0.3	LSTM	0.760	57.43%
12-bert base	0.4	LSTM	0.797	57.98%
13-all-distilroberta-v1	0.2	BI-LSTM	0.177	88.20%
14-all-distilroberta-v1	0.3	BI-LSTM	0.109	88.20%
15-all-distilroberta-v1	0.2	LSTM	0.705	71.68%
16-all-distilroberta-v1	0.3	LSTM	0.760	57.43%
17-all-distilroberta-v1	0.4	LSTM	0.797	57.98%

Table 4 shows the results of all the experiments with the application of the various preprocessing techniques. The aim of the experiments was to determine whether preprocessing methods used separately or in combination affected the final results. Although stemming and lemmatization yielded encouraging results, only lowercase conversion produced the best results. These results are attributed to the effective use of embedding techniques to preserve sentence structure. In other words, removing stop words, stemming, and lemmatization may have a negative impact on the development of effective sentence vectors. The experiments were conducted with the following fixed hyperparameters:

- No. of layers = 8
- No. of dropout layers = 2
- No. of hidden dense layers = 4
- Batch size = 3
- Learning rate = 0.0001
- Activation function = ReLu
- Model user = T5
- Split Size = 0.2
- Input layer = BI-LSTM

Table 4. Results of additional experiments.

Lower Case	Lemmatization	Stemming	Remove Stop Words	Remove Special Characters	Testing Corr.
√	-	-	-	-	**92.8%**
√	√	-	-	√	92.1%
-	-	-	-	√	91.3%
√	-	-	-	√	91.2%
-	√	-	-	√	90.8%
√	√	-	-	-	90.7%
-	√	-	√	-	89.9%
√	-	√	-	√	89.5%
-	-	√	-	-	89%
-	-	√	-	√	88.5%
-	-	√	√	√	88.2%
√	-	√	√	√	87.4%
√	√	-	√	-	87.1%
-	-	-	√	-	86.8%
-	√	-	-	-	86.0%
-	-	-	-	-	86%
√	-	√	-	-	85.5%
-	√	-	√	√	85.4%
-	-	√	√	-	84.9%
-	-	-	√	√	84.6%
√	√	-	√	√	84.5%
√	-	√	√	-	84.5%
√	-	-	√	-	83.7%
√	-	-	√	√	82.1%

4.3. Comparing the Obtained Results with Previous Results

Table 5 presents a comparison between the proposed model's obtained results and the results of previous work presented in Section 2. Compared to all previous outcomes, we attained state-of-the-art performance in short answer grading. Our strategy focused on using a BI-LSTM architecture, which has been shown to be quite successful in this field. As compared with other architectures, the use of BI-LSTM in short answer grading has several benefits. First and crucially, the model's bidirectional structure enables it to take into consideration both the prior and the subsequent context of each word in the solution. As a result, the model is better able to comprehend the answer's overall meaning and is less prone to be duped by lone words or phrases.

The ability of BI-LSTM to accurately capture long-term dependencies between words in the answers is another benefit [31]. This is crucial when grading brief answers because the relationships between various components of the answer can be intricate and subtle. The model may more fully comprehend the overall coherence and importance of the response by modeling these dependencies.

Last but not least, BI-LSTM is a highly adaptable architecture that can be tailored to the exact needs of the task at hand. Depending on user requirements, it can be optimized for a variety of metrics.

Table 5. Comparative study results.

References	Year	Approach	Correlation Score
-	**2022**	**Proposed system (embedding + deep learning)**	**92.8%**
[5]	2020	Clustering and regression analysis	81%
[2]	2022	Embedding and transformers	78%
[1]	2022	Embedding and text similarity techniques	65.12%
[7]	2019	Embedding	63%
[13]	2016	Embedding and text similarity techniques	58.6%
[3]	2018	Embedding and similarity techniques	57%
[9]	2018	Embedding and deep learning	56.9%
[14]	2016	Embedding and deep learning	55.0%
[4]	2020	Embedding and text similarity techniques	48.5%
[8]	2018	Text similarity techniques	46.8%

Overall, the use of BI-LSTM in short answer grading represents a major advance in this field. By harnessing the power of this architecture, we can achieve state-of-the-art results and pave the way for further advances in this important area of NLP.

5. Conclusions and Future Work

Automated short answer grading is a critical task that requires precision, efficiency, and speed. In this study, an automated system was developed that could evaluate each student's response based on a predefined model response for each question. Our findings demonstrated that using pretrained "transformer" models, particularly T5, coupled with a lowercase preprocessing approach, outperformed other approaches in terms of accuracy, efficiency, and precision. Additionally, we found that incorporating a BI-LSTM layer into the input layers significantly improved the performance of the grading system compared to using LSTM or dense layers. Our results showed that the combined use of the T5 pretrained model and the BI-LSTM layer yielded the highest correlation value of 92.5 percent.

In terms of future work, several areas of interest could be explored further. Firstly, it may be worthwhile to investigate the effectiveness of the proposed model on different datasets from a range of domains, rather than just focusing on a data structure in computer science. Additionally, exploring the performance of the model on datasets in different languages, such as Arabic, could yield interesting insights. Another potential avenue for future research could be to investigate more advanced embedding techniques to enhance the model's accuracy. In addition, automatic tuning for the learning rate may be very promising future work that could help in the generalization of our proposed model to be as effective as possible for this task on any dataset. Finally, one area of particular interest could be the essay scoring task, which presents a more complex challenge than the short answer grading task and may require further exploration.

Author Contributions: Conceptualization, W.H.G., A.E.N., M.M.S., A.A. and E.N.; methodology, W.H.G., A.E.N., M.M.S., A.A. and E.N.; software, W.H.G., A.E.N. and M.M.S.; writing—original draft preparation, W.H.G., A.E.N., M.M.S., A.A. and E.N.; writing—review and editing, W.H.G., A.E.N., M.M.S., A.A. and E.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research is financially supported by the Deanship of Scientific Research at King Khalid University under research grant number (R.G.P.2/549/44).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saeed, M.M.; Gomaa, W.H. An Ensemble-Based Model to Improve the Accuracy of Automatic Short Answer Grading. In Proceedings of the 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 8–9 May 2022; pp. 337–342.
2. Sawatzki, J.; Schlippe, T.; Benner-Wickner, M. Deep Learning Techniques for Automatic Short Answer Grading: Predicting Scores for English and German Answers. In Proceedings of the 2021 2nd International Conference on Artificial Intelligence in Education Technology, Dali, China, 18–20 June 2021.
3. Saha, S.; Dhamecha, T.I.; Marvaniya, S.; Sindhgatta, R.; Sengupta, B. Sentence Level or Token Level Features for Automatic Short Answer Grading?: Use Both. In Proceedings of the International Conference on Artificial Intelligence in Education, London, UK, 23–30 June 2018.
4. Gaddipati, S.K.; Nair, D.; Plöger, P.G. Comparative Evaluation of Pretrained Transfer Learning Models on Automatic Short Answer Grading. *arXiv* **2020**, arXiv:2009.01303.
5. Süzen, N.; Gorban, A.N.; Levesley, J.; Mirkes, E.M. Automatic short answer grading and feedback using text mining methods. *Procedia Comput. Sci.* **2020**, *169*, 726–743. [CrossRef]
6. Bookstein, A.; Kulyukin, V.A.; Raita, T. Generalized hamming distance. *Inf. Retr.* **2002**, *5*, 353–375. [CrossRef]
7. Gomaa, W.H.; Fahmy, A.A. Ans2vec: A scoring system for short answers. In Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019), Cairo, Egypt, 28–30 March 2019; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 586–595.
8. Pribadi, F.S.; Permanasari, A.E.; Adji, T.B. Short answer scoring system using automatic reference answer generation and geometric average normalized-longest common subsequence (gan-lcs). *Educ. Inf. Technol.* **2018**, *23*, 2855–2866. [CrossRef]
9. Hassan, S.; Fahmy, A.A.; El-Ramly, M. Automatic short answer scoring based on paragraph embeddings. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 397–402. [CrossRef]
10. Hindocha, E.; Yazhiny, V.; Arunkumar, A.; Boobalan, P. Short-text Semantic Similarity using GloVe word embedding. *Int. Res. J. Eng. Technol.* **2019**, *6*, 553–558.
11. Sánchez Rodríguez, I. Text Similarity by Using GloVe Word Vector Representations. 2017. Available online: https://riunet.upv.es/handle/10251/90045 (accessed on 5 January 2021).
12. Kenter, T.; De Rijke, M. Short text similarity with word embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 1411–1420.
13. Roy, S.; Dandapat, S.; Nagesh, A.; Narahari, Y. Wisdom of students: A consistent automatic short answer grading technique. In Proceedings of the 13th International Conference on Natural Language Processing, Varanasi, India, 17–20 December 2016; pp. 178–187.
14. Magooda, A.E.; Zahran, M.; Rashwan, M.; Raafat, H.; Fayek, M. Vector based techniques for short answer grading. In Proceedings of the Twenty-Ninth International Flairs Conference, Key Largo, FL, USA, 16–18 May 2016.
15. Lofi, C. Measuring semantic similarity and relatedness with distributional and knowledge-based approaches. *Inf. Media Technol.* **2015**, *10*, 493–501.
16. Bonthu, S.; Rama Sree, S.; Krishna Prasad, M.H.M. Automated Short Answer Grading Using Deep Learning: A Survey. In *Machine Learning and Knowledge Extraction*; Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E., Eds.; CD-MAKE 2021; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12844. [CrossRef]
17. Zhang, L.; Huang, Y.; Yang, X.; Yu, S.; Zhuang, F. An automatic short-answer grading model for semi-open-ended questions. *Interact. Learn. Environ.* **2022**, *30*, 177–190. [CrossRef]
18. Xia, L.; Guan, M.; Liu, J.; Cao, X.; Luo, D. Attention-Based Bidirectional Long Short-Term Memory Neural Network for Short Answer Scoring. In *Machine Learning and Intelligent Communications, Proceedings of the 5th International Conference, MLICOM 2020, Shenzhen, China, 26–27 September 2020*; Proceedings 5; Springer International Publishing: Berlin/Heidelberg, Germany, 2021. [CrossRef]
19. Gong, T.; Yao, X. An Attention-based Deep Model for Automatic Short Answer Score. *Int. J. Comput. Sci. Softw. Eng.* **2019**, *8*, 127–132.
20. Salam, M.A.; El-Fatah, M.A.; Hassan, N.F. Automatic grading for Arabic short answer questions using optimized deep learning model. *PLoS ONE* **2022**, *17*, e0272269. [CrossRef]
21. Ni, J.; Abrego, G.H.; Constant, N.; Ma, J.; Hall, K.; Cer, D.; Yang, Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv* **2021**, arXiv:2108.08877.
22. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
23. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
24. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
25. Gomaa, W.H.; Fahmy, A.A. Short answer grading using string similarity and corpus-based similarity. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2012**. [CrossRef]

26. Gomaa, W.H.; Fahmy, A.A. Arabic short answer scoring with effective feedback for students. *Int. J. Comput. Appl.* **2014**, *86*, 35–41.
27. Gomaa, W.H.; Fahmy, A.A. Automatic scoring for answers to Arabic test questions. *Comput. Speech Lang.* **2014**, *28*, 833–857. [CrossRef]
28. Shah, N.; Pareek, J. Automatic Evaluation of Free Text Answers: A Review. In *Advancements in Smart Computing and Information Security, Proceedings of the First International Conference, ASCIS 2022, Rajkot, India, 24–26 November 2022*; Revised Selected Papers, Part II; Springer: Cham, Switzerland, 2023; pp. 232–249.
29. Susanti, M.N.I.; Ramadhan, A.; Warnars, H.L.H.S. Automatic essay exam scoring system: A systematic literature review. *Procedia Comput. Sci.* **2023**, *216*, 531–538. [CrossRef] [PubMed]
30. Schlippe, T.; Stierstorfer, Q.; Koppel, M.T.; Libbrecht, P. Explainability in Automatic Short Answer Grading. In *Artificial Intelligence in Education Technologies: New Development and Innovative Practices, Proceedings of the 2022 3rd International Conference on Artificial Intelligence in Education Technology, Birmingham, UK, 21–23 October 2022*; Springer: Singapore, 2023; pp. 69–87.
31. Firoz, N.; Beresteneva, O.G.; Vladimirovich, A.S.; Tahsin, M.S.; Tafannum, F. Automated Text-based Depression Detection using Hybrid ConvLSTM and Bi-LSTM Model. In Proceedings of the 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2–4 February 2023; pp. 734–740.