Check for updates

Received: 28 September 2023 | Revised: 14 February 2024 | Accepted: 23 April 2024

DOI: 10.1111/ejed.12684

ORIGINAL ARTICLE

WILEY

Coherence-based automatic short answer scoring using sentence embedding

Dadi Ramesh[1,2] | **Suresh Kumar Sanampudi**[3]

[1]School of Computer Science and Artficial Intelligence, SR University, Warangal, India

[2]Research Scholar in JNTUH, Hyderabad, India

[3]Department of Information Technology, JNTUH College of Engineering Jagitial, Jagtial, India

Correspondence
Dadi Ramesh, School of Computer Science and Artficial Intelligence, SR University, Warangal, India.
Email: d.ramesh@sru.edu.in

Abstract

Automatic essay scoring (AES) is an essential educational application in natural language processing. This automated process will alleviate the burden by increasing the reliability and consistency of the assessment. With the advances in text embedding libraries and neural network models, AES systems achieved good results in terms of accuracy. However, the actual goals still need to be attained, like embedding essays into vectors with cohesion and coherence, and providing student feedback is still challenging. In this paper, we proposed coherence-based embedding of an essay into vectors using sentence-Bidirectional Encoder Representation for Transformers. We trained these vectors on Long short-term memory and bidirectional long short-term memory to capture sentence connectivity with other sentences' semantics. We used two datasets: standard ASAP Kaggle and a domain-specific dataset with almost 2500 responses from 650 students. Our model performed well on both datasets, with an average quadratic weighted kappa score of 0.76. Furthermore, we achieved good results compared to other prescribed models, and we also tested our model on adversarial responses of both datasets and observed decent outcomes.

KEYWORDS
adversarial responses, Bi-LSTM, essay scoring, sentence embedding, sentence-BERT

Abbreviations: AES, automated essay scoring; BERT, Bidirectional Encoder Representation for Transformers; Bi-LSTM, bidirectional long short-term memory; CNN, Convolutional Neural Network; LSTM, long short-term memory; OS, operating system; QWK, quadratic weighted kappa; RNN, Recurrent Neural Network; USE, Universal Sentence Encoder.

 © 2024 John Wiley & Sons Ltd.

1 | INTRODUCTION

Automatic essay scoring (AES) systems streamline the evaluation of student responses to prompts, ensuring consistency while minimizing the human effort. Recent research has predominantly focused on ASE systems, as outlined by Ramesh and Sanampudi (2022). Categorizing these approaches into four classes based on the amalgamation of manually and automatically extracted features, researchers have employed machine learning and neural network models to train these systems. Early iterations such as those by Page (1967), Ajay et al. (1973), Burstein (2003), Leacock and Chodorow (2003), Adamson et al. (2014), and Cummins et al. (2016) primarily relied on manually extracted features like bag of words, term frequency, inverse document frequency, word count, sentence count, and sentence length. These statistical features were trained on machine learning models such as regression and support vector machines; these approaches attempted to establish relationships between essays and labels using manually extracted statistical features. However, they fell short in capturing the semantics and content of essays.

In the second category of approaches, Sultan et al. (2016), Cozma et al (2018), Darwish and Mohamed (2020), and Süzen et al. (2020) extracted content-based features with Word2vec Mikolov et al. (2013), and Glove Jeffrey Pennington et al. (2014). However, again, they used a machine learning model for assessment. The machine learning model considers the feature vectors independently, and they do not connect the words to capture the semantics of the essay.

The third and fourth categories of approaches would have extracted features manually, automatically, and trained different types of neural network models. Like Taghipour and Ng (2016), Dong and Zhang (2016), Riordan et al. (2019), Mathias and Bhattacharyya (2018), Wang et al. (2018), Dasgupta et al. (2018), Kumar et al. (2019), Zhu and Sun (2020), and Wang et al. (2023) used pre-trained natural language processing models like word2vec, glove to extract features. Then, they used neural networks Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and a combination of CNN and RNN to fine-tune the essays. These approaches performed well in terms of quadratic weighted kappa (QWK) score. However, the word-level feature extracting methods cannot handle polysemous words, and they miss sentence semantics and connectivity from an essay. Chen and Li (2023) implemented cross prompt scoring system to map the features of essays. Andersen et al. (2023) proposed AES model that can extracts proficiency score from text and evaluates the essay. Moreover, no AES model has proved the model's robustness by testing adversarial responses for consistency.

From Horbach and Zesch (2019), Riordan et al. (2019), Ding et al. (2020), and Kumar et al. (2020) proved that the black box type of models is prone to adversarial responses and its challenging task to handle irrelevant, repeated sentences type of responses. So, we need an AES system to handle adversarial responses and evaluate essays based on content.

1.1 | Contribution

1. We developed an Automated Short answer scoring system based on sentence-level embedding to capture sequential features.
2. Our model evaluated on two datasets to prove our model's robustness.
3. We proved the persuasion of our approach via empirical evaluation of diverse adversarial responses.
4. We compared the results of word and sentence embedding models with different test cases.

1.2 | Organization

The rest of the paper is organized as follows. Section 2 discusses the related work about text embeddings and deep learning models used for AES systems and their challenges. Section 3.1 presents the proposed method of the AES

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

system on different datasets and sentence embeddings. Section 3.2 discuss the sentence embedding methods with vector dimension, and Section 3.3 discusses the implemented models, their architectures, and the hyperparameters used during training. Section 4.1 discusses the experimental results compared with other models and presents the model's performance on adversarial responses. Finally, Section 5 discusses the conclusion and future work.

2 | RELATED WORK

AES is the task of assessing student responses (short or essay); it is the most challenging task in NLP. The main job of the AES system is to extract features from essays with cohesion and coherence and train a neural network model to fine-tune the features and assign a final score. The early systems like Ajay et al. (1973), Burstein (2003), Foltz et al. (1999), Rudner and Liang (2002), Adamson et al. (2014), Cummins et al. (2016). Extracted handcrafted features from the essay assigned score.

With advances in NLP and neural networks, the feature extraction for AES also changed from manual to automatic extraction. The first automatic features extraction method is word2ve, a pre-trained model that can extract word-level context into a vector. Like Sakaguchi et al. (2015), Mathias and Bhattacharyya (2018), Taghipour and Ng (2016), Kumar et al. (2019), Xia et al. (2019) and Zhu and Sun (2020). Wang et al. (2018) used a reinforcement model to train word vectors. However, word-level context encoding fails on polysemous words and does not handle adversarial responses. These models assign a score when a student submits an irrelevant response with few related words.

Rather than training word context vectors only on the long short-term memory (LSTM) model, researchers like Dong and Zhang (2016) and Dong et al. (2017). Riordan et al. (2019) and Mathias and Bhattacharyya (2018) Wang et al. (2023) added an extra CNN layer on top of the LSTM or bidirectional long short-term memory (Bi-LSTM) to form sentences. The CNN layers will add a fixed number of words to form sentences. However, with this approach, the model's accuracy has been increased, the actual sentences have diverged, and actual word connectivity is missing. Still, do not handle irrelevant responses like one-word, random-word responses. Kumar et al. (2019) implemented a domain-based AES system based on word embedding.

In recent researchers embedded essays directly sentence by sentence using Universal Sentence Encoder (USE) and sentence-Bidirectional Encoder Representation for Transformers (BERT) like Rodriguez et al. (2019), Lun et al. (2020), Yang et al. (2020), Song et al. (2020), Ormerod et al. (2021), Doewes and Pechenizkiy (2021). Mayfield and Black (2020)used BERT for essay embedding, but they used word level embedding instead of setence, and trained an LSTM model for fine-tuning. USE and sentence-BERT capture coherence and cohesion from an essay, and the LSTM model will tune sentence connectivity and coherence to assign a score. Fernandez et al. (2022)used BERT for reading comprehension and trained separately on prompt, text, and student response for assessment. Wang et al. (2022) used BERT and LSTM and performed well in terms of accuracy. Chen and Li (2023) implemented cross prompt method wit unlabeled samples to map the features of essay with target essay. However, no model showed the model's robustness and how the system handles adversarial responses.

Table 1 illustrates the possible combinations of text embedding and machine and deep learning models used for essay scoring. Table 2 shows the text embedding technique and vector dimension they create; out of all sentence-BERT will give a low dimension vector for sentence. The best combination is sentence embedding and recurrent neural network because sentence embedding will capture coherence from an essay and can easily handle polysemous words, which was a deficiency in word embeddings.

3 | METHODOLOGY

We introduce a novel approach that employs sentence-based text embedding to capture coherence, employing separate training of LSTM and Bidirectional LSTM (Bi-LSTM) models. Our methodology encompasses two

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 Comparison of machine learning and neural network models and features extraction methods.

	BoW/TF-IDF	Word2vec/glove (word embedding)	USE (sentence embedding)
Regression models/ classification models	The system implemented with Bow features and regression or classification algorithms will have low cohesion and coherence	The system implemented with Word2vec features and regression or classification algorithms will have low to medium cohesion and coherence	Sentence-based encoding will capture coherence from sentence, but with regression models sentence to sentence to connectivity will miss
Neural networks (LSTM)	The system implemented with BoW features and neural network models will have low cohesion and coherence	The system implemented with Word2vec features and neural network model (LSTM) will have medium to high cohesion and coherence	Sentence-based embedding, with neural network (LSTM) will capture cohesion and coherence. But USE embeds essay into 512 dimensions

Abbreviations: BoW, bag of words; LSTM, Long short-term memory; TF-IDF, Term Frequency –Inverse Document Frequency; USE, Universal Sentence Encoder.

TABLE 2 Comparison of the available text embedding methods.

Embedding model	Vector dimension	Essay dimension	Remarks
Word2Vec	32 dimensions per word	32 96* number of words per each sentence	Word-level embedding
ELMO	1024 dimensions per word	1024*96* number of words per each sentence	Word-level embedding
XP2			Word-level embedding
USE	512 dimensions per sentence	96*512	High-dimension vectors for sentence
Sentence-BERT	128 dimensions per sentence	96*128	Low-dimension vector for each sentence

Abbreviations: BERT, Bidirectional Encoder Representation for Transformers; USE, Universal Sentence Encoder.

datasets: a standard ASAP dataset and a domain-specific dataset comprising 2300 responses from 600 students. Furthermore, we rigorously tested our model against various adversarial responses to assess its robustness. The intricate architecture of our proposed AES system, integrating sentence-BERT (Devlin et al., 2019) and Bi-LSTM, is delineated in Figure 1.

3.1 | Dataset

We used the ASAP Kaggle dataset widely used in AES systems. ASAP dataset comprises 12,978 essays of 8th to 10th-grade standard students across eight different prompts. Each prompt consists of 1500 or more essays evaluated by two raters. Prompts 3, 4, 5, 6 are source-dependent essays, and the remaining are others. A detailed description of the essay dataset is presented in Table 3.

Furthermore, we created a novel dataset within the Operating Systems (OS) domain to evaluate the performance of AES systems on domain-specific essays. To accomplish this, we crafted five fundamental questions as assignments about OS, a core subject within computer science, distributed among Bachelor of Technology students across various engineering colleges. From these efforts, we amassed a total of 2981 responses from

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

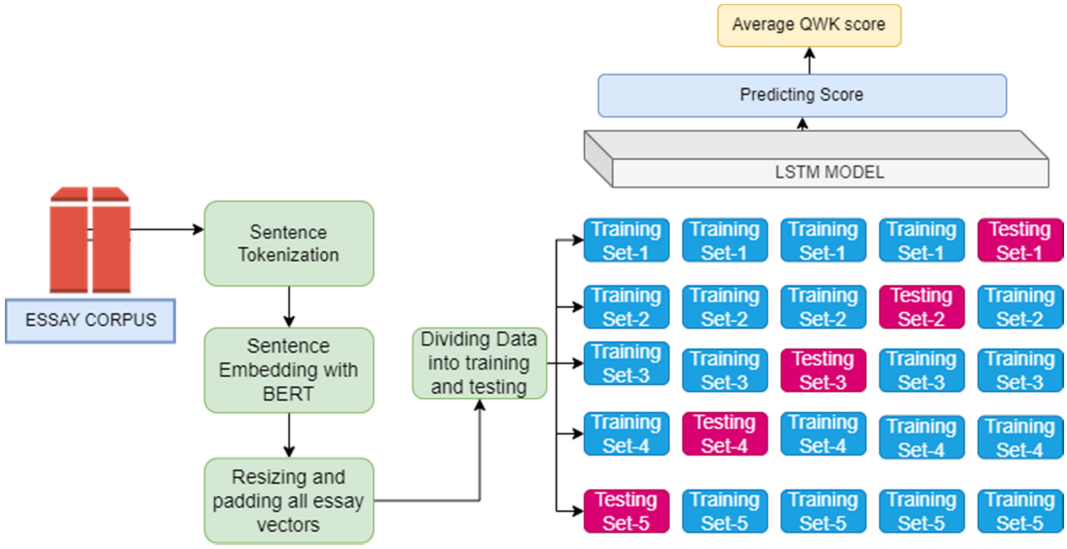

FIGURE 1 Architecture of an AES system. AES, automated essay scoring.

TABLE 3 Kaggle ASAP dataset for essay scoring.

Essay set	No. of essays	Average length of essays	Rating range
1	1783	350	2–12
2	1800	350	1–6
3	1726	150	0–3
4	1772	150	0–3
5	1805	150	0–4
6	1800	150	0–4
7	1569	250	0–30
8	723	650	0–60

students, diligently filtering out any repeated or duplicate submissions. Subsequently, we refined the dataset to include 2390 unique responses obtained from 626 distinct students. These include one-word, adversarial, revilement, and silly responses.

To ensure the quality and reliability of the dataset, two subject matter experts were engaged to manually evaluate the responses, assigning scores on a scale ranging from 0 to 5. The evaluators' details are outlined in Table 4, with scores ranging from a minimum of 0 to a maximum of 5. The inter-rater agreement was quantified using the QWK score, yielding a commendable value of 0.842, as depicted in Figure 2. A detailed description of the OS dataset is demonstrated in Table 5.

3.2 | Sentence embedding

In natural language processing, achieving contextual and semantic text representation in vector form poses a significant challenge. Traditional embedding techniques such as word2vec and GloVe independently convert text into word vectors, disregarding the surrounding words and their contexts. Additionally, these methods struggle

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 Raters information.

Raters	Teaching experience(years)	Number times thought OS
1	11	6
2	8	4

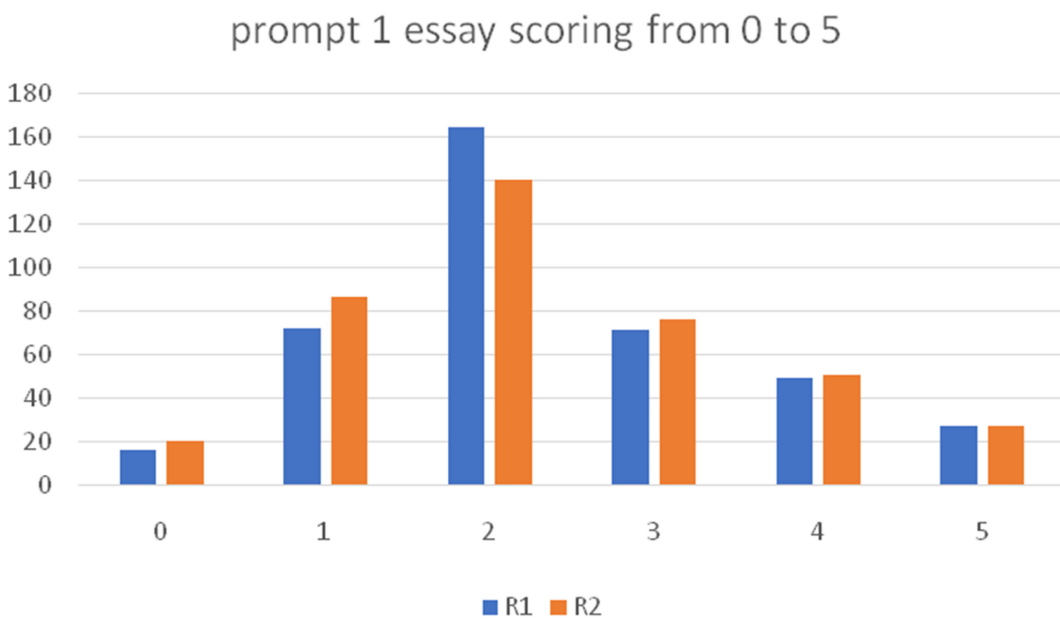

FIGURE 2 OS dataset prompt-1, reviewers score (R1-rater 1 and R2-rater 2) the agreement between Rater-1 and Rater-2.

TABLE 5 Operating system (OS) dataset.

Prompt-id	Prompt	Prompt-wise number of essays	Prompt-wise maximum number of sentences	Rating range (min to max)
1	Explain about OS?	516	23	0–5
2	Explain the advantages of a multiprocessor system?	596	21	
3	Explain how OS handles multiple tasks at a time?	312	19	
4	Difference between single processor and multiprocessor OS?	513	19	
5	Explain different scheduling algorithm?	453	15	

with handling polysemous words effectively. While sentence embedding techniques exist, they typically involve converting word vectors into sentence vectors by averaging all word embeddings.

In our model, we employ Sentence-BERT to convert essays into vectors. Sentence-BERT dynamically generates vectors with context and semantics, allowing for reconstructing the original sentence from the vector representation. Initially, we preprocess the essays by removing special symbols (e.g., '@' and '#') and tokenizing them

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

into sentences. Across the ASAP and OS datasets, we observe that the maximum number of sentences per essay is 96 and 23, respectively. We then use a pre-trained transformer model, Sentence-BERT, to embed each sentence into a 128-dimensional vector.

Consequently, for an essay from the ASAP dataset, we obtain 96 * 128-dimensional vectors, while for an essay from the OS dataset, we obtain 23 * 128-dimensional vectors. Finally, to ensure uniformity in dimensionality, we pad all essays into 96 * 128 and 23 * 128 vectors, with 96 and 23 representing the maximum number of sentences in the ASAP and OS datasets, respectively. Table 6 illustrates the sentence vectors extracted from essays in both datasets, further elucidating the transformation process.

3.3 | Model implementation

To ensure that the implemented model comprehensively assesses essays, it must extract coherence, cohesion, and linguistic features from the text to generate a final score. To achieve this, we embedded all essays sentence-wise, retaining stop words to capture coherence effectively. Subsequently, we trained separate LSTM and Bi-LSTM models on the sentence vectors. Additionally, we adopted a CNN+LSTM architecture inspired by Taghipour and Ng (2016), leveraging CNN to capture N-gram features before feeding them into an LSTM unit for further processing.

3.3.1 | LSTM/Bi-LSTM

LSTM and Bi-LSTMare recurrent neural network architectures designed to handle sequential data by leveraging their memory cells. These memory cells consist of four crucial components: an input gate, a forget gate, an output gate, and a context gate. Working in concert, these gates process information and maintain necessary long-term dependencies for future utilization of features, transmitting this information to the input gate of the subsequent cell. In Figure 3, the process of context generation within an LSTM unit is depicted.

Bi-LSTM traverses sentence vectors in both forward and backward directions, facilitating a comprehensive capture of coherence within the text. This approach entails storing context information from each sentence and connecting it to the subsequent sentence. By amalgamating this contextual information, Bi-LSTM enables

TABLE 6 ASAP dataset, OS dataset essay vector after padding.

dataset	Sample Essay embedded vector by BERT	dimension
ASAP	[[0.01377844 -0.09247074 0.01014971 ... -0.01349053 -0.04146808 0.05626552] [-0.01370333 -0.0240192 -0.03880018 ... -0.05234249 -0.06115109 0.05296136] [-0.05529318 -0.02587196 -0.00212097 ... 0.02701825 0.02506788 0.00300164] ... [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.]]	96 * 128
OS	[[-0.04040171 0.00535519 -0.02015072 ... -0.07707731 -0.07179338 0.05336216] [-0.00677465 0.01085155 0.03253689 ... -0.07136418 -0.05266594 -0.01695863] [0.03762686 0.02985795 -0.05078415 ... -0.05361476 -0.02895073 0.04693419] ... [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.]]	23*128

Abbreviation: BERT, Bidirectional Encoder Representation for Transformers.

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

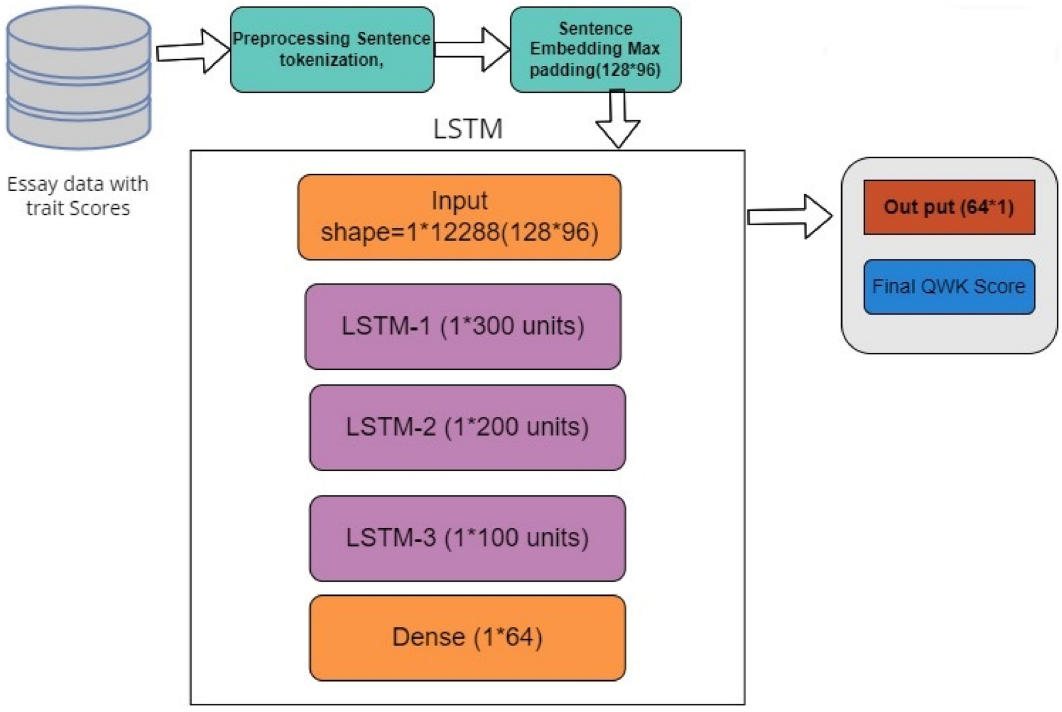

FIGURE 3 Working of the proposed LSTM model on sentence vectors of each essay. LSTM, long short-term memory.

valuable insights that allow all sentences to collaborate in summarizing and predicting the final essay score effectively.

$$H(\text{forward}) = \sigma\left(w_1x_1 + w_2x_2 + w_3x_3 \ldots \ldots w_nx_n\right) + b \tag{1}$$

$$H(\text{backward}) = \sigma\left(w_1x_1 + w_2x_2 + w_3x_3 \ldots \ldots w_nx_n\right) + b \tag{2}$$

where $w_1, w_2, w_3 \ldots \ldots$ are the weights, b is the bias, σ is the activation function, y is the output, and n is the number of sentences in the essay.

3.4 | Training an LSTM model

First, we tokenized all the essays into sentence vectors using sentence BERT, then padded all the essays into the max-size essay, i.e., 96 * 128 like in Figure 4. Then we converted all vectors to 3-dimensional vectors to train on a neural network.

In LSTM, we stacked five layers of LSTM; each unit has an input gate, output gate, and context gate. We used an RMSprop optimizer to reduce the mean square error like Dong et al. (2017), drop rate as 0.5, assign initial learning rate as 0.001, and activate function as rectified linear unit. The hyperparameters of our models are shown in Table 7.

In the training phase, we used 5-fold cross-validation, like Taghipour and Ng (2016), to split the essay vectors into training, testing, and validation in a 70:15:15 ratio for both datasets. To fix the hypermeters, we trained our model for 10, 15, 20, 35 epochs and fixed the hyperparameters. We use QWK as an evaluation metric in our models to find the agreement between the human and system rater, which has been widely used for AES Taghipour and Ng (2016), Wang et al. (2018), Tay et al. (2018), Mathias and Bhattacharyya (2018), Wang et al., 2022 In each

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

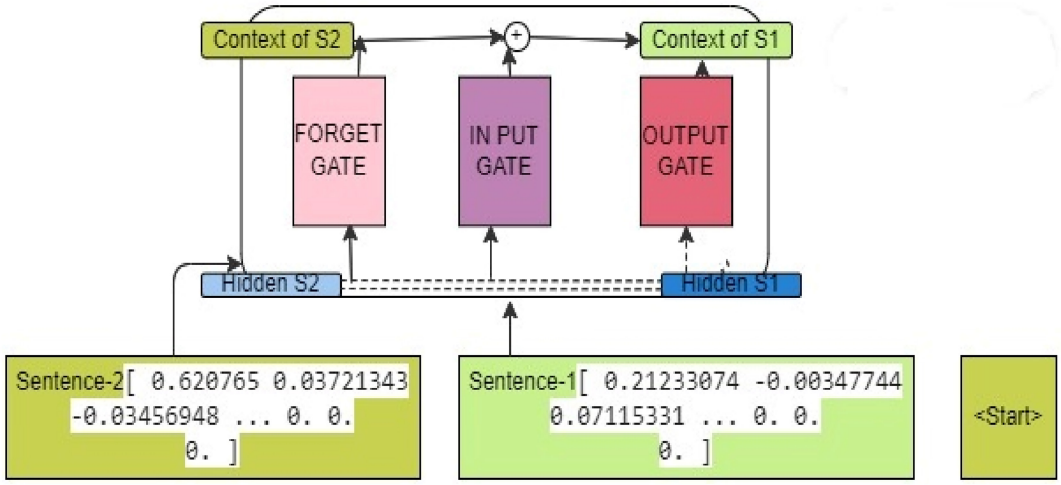

FIGURE 4 Context generation from sentences in LSTM unit. LSTM, long short-term memory.

TABLE 7 Neural network model hyper parameters and values.

Layer	Parameter	Value
Embedding	Sentence embedding (BERT)	128 size vectors for each sentence
CNN layer	Input size	(1, 96, 128), (1, 23, 128)
LSTM layers	No. of layers	5
LSTM units	LSTM units	300
Hidden	Hidden units	200, 100
Drop out	Dropout rate	0.4
	Recurrent drop out	0.5
Others	Epochs	35
	Batch size	32
	Learning rate	0.001
	Optimizer	Adam

Abbreviations: BERT, Bidirectional Encoder Representation for Transformers; CNN, Convolutional Neural Network; LSTM, long short-term memory.

fold, we calculated the QWK score. Finally, we use the model that achieves the best performance on training data to predict the test data. Figure 4 illustrates the training and validation loss of the proposed model and it portrays that our proposed model is neither overfitted nor under fitted.

We used the same hyperparameters and 5-fold cross-validation to train sentence-LSTM and sentence-Bi-LSTM on the OS dataset. OS dataset input dimension is 23*128 for LSTM and Bi-LSTM, 23 is the maximum number of sentences, and 128 is the sentence vector.

4 | RESULTS AND DISCUSSION

The experiment results we obtained based on the ASAP dataset and OS dataset for the AES system are shown in Figures 5 and 6; we observed that our proposed model performance on both datasets is best fitted. We trained

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

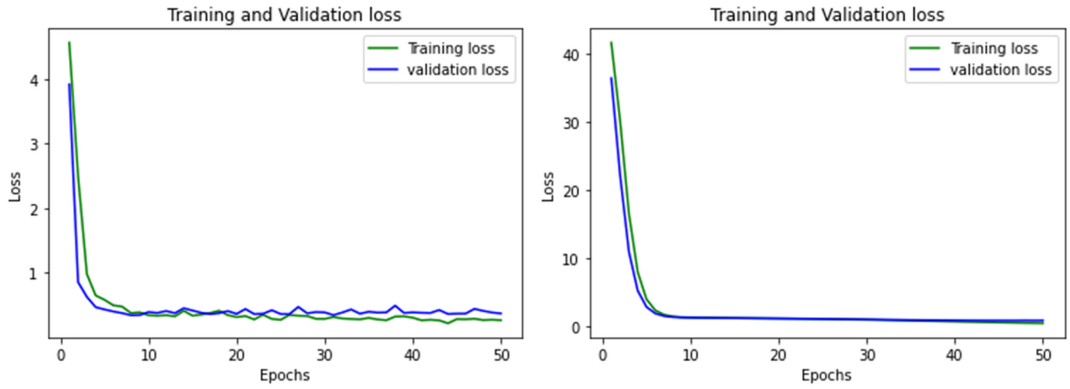

FIGURE 5 Comparison of prompt-wise training and validation loss of the OS dataset model.

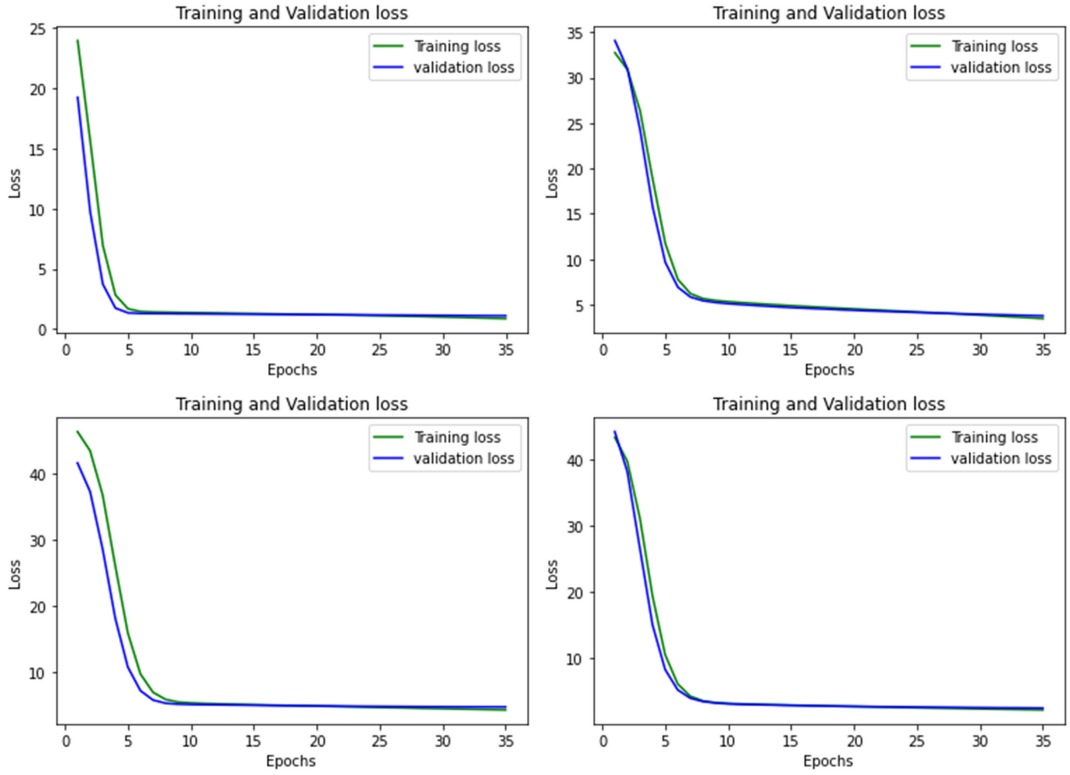

FIGURE 6 Comparison prompt-wise training and validation loss of the ASAP dataset model.

the model prompt-wise and calculated training and validation loss prompt-wise. Figure 7 illustrates the prompt-wise comparison of actual and predicted scores on test data. From Figure 7 it is observed that in score 1 and 5 the agreement between human generated and system generated score little bit difference, but for score 2, 3 and 4 it is prediction correctly.

The results show that our proposed models outperformed and equally performed with other models. The comparison of all baseline models on the ASAP dataset and our proposed models on average QWK score is

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

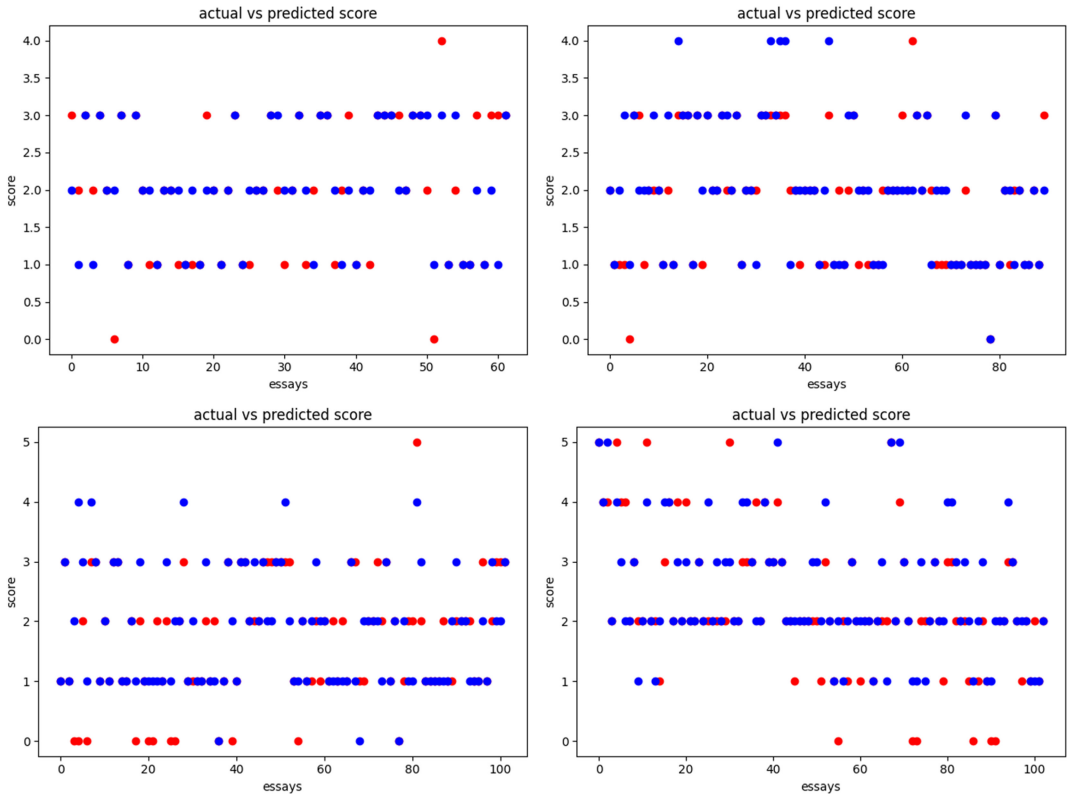

FIGURE 7 Prompt-wise comparison of actual and predicted scores red colour represents actual score and blue colour represents predicted score on test data.

shown in Table 8. We found that sentence embedding-LSTM and sentence embedding-Bi-LSTM models performed well compared to other models and were consistent with the human rater score. Furthermore, it observed that Sentence Embedding-LSTM and Bi-LSTM performed better than models like Muangkammuen and Fukumoto (2020), Agrawal and Agrawal (2018), LSTM-MOT model of Taghipour and Ng (2016), and CNN+ LSTM integrating deep learning model (2021). Though SKIPFLOW-LSTM (2017), TSLF-ALL Liu et al. (2019), and CNN-LSTM Attention Dong et al. (2017). Models performed equally with sentence embedding-LSTM and Bi-LSTM. However, these integrated and word embedding models did not capture sentence coherence; because of neural networks, the QWK score was high.

However, like other models, our model consistently performed on source-dependent essay traits like prompt 3, 4, 5, 6; these essay traits' rating range is between 1 and 6. Furthermore, based on the QWK score, the performance of our models and other baseline models on persuasive, narrative, and expository essay traits is the same and a little bit high. However, the performance was reduced when we used the CNN layer on the Sentence Embedding-LSTM, Bi-LSTM neural networks model; with this, we conclude that when we split essays or sentences into tokens, the coherence and cohesion were missing. Though Ormerod et al. (2021) and Wang et al. (2022) used BERT for text embedding; they tokenized the essay into words. Moreover, they were embedded through BERT and performed well but could not capture coherence from an essay.

The models trained on the ASAP dataset were also used on the OS dataset with the same hyperparameters, and it is performed with an average QWK score of 0.746 and 0.751 by Bi-LSTM and LSTM models. Table 9 illustrates the prompt-wise QWK score and average QWK score of the word embedding, Sentence Embedding-LSTM, and Sentence Embedding-Bi-LSTM model on the OS dataset.

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 8 Prompt-wise QWK score with LSTM and Bi-LSTM on the ASAP dataset.

Models	1	2	3	4	5	6	7	8	Avg	Remarks
Rater1 to Rater 2 agreement	0.721	0.812	0.769	0.851	0.753	0.776	0.720	0.627	0.754	-NA-
PEG, Page (1967)	0.82	0.72	0.75	0.82	0.83	0.81	0.84	0.73	0.79	Hand crafted features, simple linear regression model
Multi-task (word & sentence sentiment) Muangkammuen and Fukumoto (2020)	0.803	0.658	0.664	0.772	0.799	0.816	0.787	0.644	0.743	Word level embedding
EASE	0.76	0.61	0.62	0.74	0.78	0.78	0.73	0.62	0.71	Hand crafted features
LSTM-MoT model, Taghipour and Ng (2016)	0.775	0.687	0.683	0.795	0.818	0.813	0.805	0.594	0.746	Ensemble method on word vectors
SKIPFLOW-LSTM model, Tay et al. (2018)	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.764	Ensemble method on word vectors
RL1, Wang et al. (2018)	0.766	0.659	0.688	0.778	0.805	0.791	0.760	0.545	0.724	Word level embedding
CNN + LSTM integrating deep learning model (2021)	0.87	0.64	0.63	0.83	0.86	0.85	0.79	0.53	0.73	Ensemble method on word vectors
Dong et al. (2017) (LSTM-CNN attention)	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764	Ensemble method on word vectors
R2 BERT	0.817	0.719	0.698	0.845	0.841	0.847	0.839	0.744	0.794	Word level embedding and used regression
ALBERT, Ormerod et al. (2021)	0.807	0.671	0.672	0.813	0.802	0.816	0.826	0.700	0.766	Sentence embedding
Tran-BERT-MS-ML-R, Wang et al. (2022)	0.834	0.716	0.714	0.812	0.813	0.836	0.839	0.766	0.791	Token (word) level embedding, each word with 510 lengths after embedding.
Sentence Embedding-LSTM[a]	0.811	0.698	0.702	0.816	0.821	0.815	0.795	0.675	0.766	Sentence level embedding
Sentence Embedding-Bi-LSTM[a]	0.794	0.691	0.694	0.795	0.811	0.807	0.789	0.634	0.752	Sentence level embedding
Sentence Embedding-CNN-LSTM[a]	0.773	0.689	0.661	0.791	0.805	0.793	0.784	0.626	0.740	Sentence level embedding, added CNN on top of LSTM to embed n-grams
Sentence Embedding-CNN-Bi-LSTM[a]	0.753	0.682	0.668	0.788	0.805	0.788	0.757	0.589	0.728	Sentence level embedding, added CNN on top of Bi-LSTM to embed n-grams

Abbreviations: BERT, Bidirectional Encoder Representation for Transformers; Bi-LSTM, bidirectional long short-term memory; CNN, Convolutional Neural Network; LSTM, long short-term memory; QWK, quadratic weighted kappa. a-proposed models

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 9 Prompt-wise QWK score of word embedding and sentence embedding models on the ASAP and OS datasets.

	Word embedding model		Sentence embedding -LSTM	
PROMPT	OS dataset	ASAP dataset	OS dataset	ASAP dataset
1	0.745	0.834	0.766	0.811
2	0.738	0.716	0.759	0.698
3	0.634	0.714	0.696	0.702
4	0.722	0.812	0.748	0.816
5	0.742	0.813	0.776	0.821
6	–	0.836	–	0.815
7	–	0.839	–	0.795
8	–	0.766	–	0.675

Abbreviations: LSTM, long short-term memory; QWK, quadratic weighted kappa.

TABLE 10 Testing and comparing results of the proposed model and word embedding model on adversarial responses.

Test case	Adversarial responses	Actual score (human rater score)	Predicted score of proposed models (sentence embedding with LSTM)	Predicted score of word embedding model (Word2vec + LSTM)
1	Essay from Prompt −1 which is relevant to the response	9	9.5	7.7
2	One sentence response for prompt-1	2	2.2	1.5
3	Essay from Prompt-1 which is relevant to the response	7	7.1	5.2
4	Prompt as response	1	1.02	2.1
5	Irrelevance response to the prompt-1 (where few words are matching to the prompt)	0	0.89	2.90
6	Response with repeated sentences (where 50% of the answer is correct)	5	4.82	7.6
7	One word response	0	0	0.4
8	Response with 50% relevant answer and remaining 50% is irrelevant	6	5.88	7.2

Abbreviation: LSTM, long short-term memory.

4.1 | Testing on adversarial responses

To assess how well models handle adversarial responses and to determine whether essays are being evaluated based on their semantic content. We designed eight test cases that cover various scenarios, including irrelevant responses, relevant responses, using the prompt as a response, and repeating sentences. Tables 10 compare the actual and predicted scores generated by word embedding (word2vec) and sentence embedding models for all eight test cases.

The results of test cases 1, 2, and 3 show a substantial difference between the actual scores and the scores predicted by the word embedding model. Conversely, the sentence embedding model performed admirably in these scenarios.

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 11 Testing adversarial responses on the OS dataset.

Test case	Adversarial response type	Adversarial response	Prompt	Actual score	Predicted score
1	100% wrong answer for the given prompt, not related	It is a good interaction between user and the computer	Explain the advantages of a multiprocessor system?	0	0.2
2	One word answer, that is relevant to the prompt	Intermediate to user and hard ware	Explain the advantages of a multiprocessor system?	1	0.8
3	The given response is not related to the prompt, but words individually are relevant to the prompt, and not as sentences	A system can be defined as a collection of elements that interact and interconnect by a predefined set of rules, resulting in a cohesive and integrated entity. This system is not isolated; it exists within an environment that surrounds and influences it	Explain how operating system (OS) handles multiple tasks at a time?	0	0.6
4	Some part of the student response is relevant and some part is not relevant to the prompt	The OS allocates tasks to a CPU and executes them sequentially, utilizing different priority mechanisms. These priorities are determined by scheduling algorithms, of which there are two main types: preemptive and non-preemptive	Explain how OS handles multiple tasks at a time?	2	2.6
5	Though the given response is 50% relevant to the prompt but remaining 50% response is repeated	The OS allocates tasks to a CPU and executes them sequentially, utilizing different priority mechanisms. These priorities are determined by scheduling algorithms, of which there are two main types: preemptive and non-preemptive. The OS allocates tasks to a CPU and executes them sequentially, utilizing different priority mechanisms. These priorities are determined by scheduling algorithms, of which there are two main types: preemptive and non-preemptive	Explain how OS handles multiple tasks at a time?	4	5.1

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

In test cases 5 and 6, we observed that the word embedding model struggled when faced with irrelevant responses that contained a few matching words with the essay content. In contrast, when it came to sentence-repeated responses, such as test cases 5 and 6, our model demonstrated a remarkable ability to grasp the semantic nuances of the essay. Notably, it did not penalize or consider duplicate sentences when determining the final score.

For instance, Table 10 reveals that test cases 1 and 3 correspond to irrelevant responses, test case 2 involves a one-sentence response, test case 4 is a 50% relevant and 50% irrelevant response, and test case 5 involves repeated sentences. Our model consistently performed well in all of these instances, showcasing its robustness in handling diverse and challenging essay evaluation scenarios.

We observed that the word embedding model is underperformed and does not capture essay semantics. In contrast, Sentence Embedding with LSTM performed well on irrelevant responses, given a final score based on relevance (Table 11).

We strongly argue that our model captures coherence and cohesion from essays while evaluating and assigning a score. We used sentence-to-sentence embedding and trained neural networks to capture sequence-to-sequence patterns from the essay. However, our model's average QWK score is greater than or equal to other baseline models, but our model assesses essays based on coherence and cohesion. Moreover, our model performance is consistent in semantics and relevance while testing adversarial responses.

5 | CONCLUSION

In this paper, we proposed and implemented a novel approach for an AES system with a combination of sentence embedding with an LSTM and Bi-LSTM. All the models are trained and tested on the Kaggle ASAP and OS datasets. In this approach, we embedded the essay sentence by sentence after preprocessing to capture sequence-to-sequence coherence patterns and trained on recurrent neural networks. Specifically, we compared our models: Sentence embedding-LSTM and Sentence Embedding-Bi-LSTM with baseline models of AES; it has been observed that Sentence Embedding-LSTM and Bi-LSTM performed well among all models. Moreover, our proposed models outperformed other baseline models without missing coherence. Some of the model's performance is high in terms of QWK score, but so far, all models have used word-based embedding.

We tested our proposed models on adversarial responses and compared the actual and predicted scores. While testing on adversarial responses, our model assigned scores for essays based on coherence and cohesion; when we were given irrelevant responses and repeated sentence essays, our model captured content and assigned scores. From the results, it's proved that our models are performing consistently. Furthermore, we also trained our models on different datasets like OS and observed consistent performance.

In the future, we will continue our study on trait-based AES systems and test the model on more adversarial responses to test the robustness of the model. To handle Out of Vocabulary (OOV) words, we are creating a separate corpus related to the OS dataset domain to handle OOV words.

AUTHOR CONTRIBUTIONS

All authors equally contributed to this work and approved the final manuscript.

ACKNOWLEDGEMENTS

We thank SR University and JNTU college of Engineering Jagitial, students, and faculty for collecting the Essay dataset.

FUNDING INFORMATION

Not applicable.

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest.

DATA AVAILABILITY STATEMENT

The raw datasets which were used for proposed approach can be found at ASAP data (https://www.kaggle.com/c/asap-sas/), and Operating System dataset at (https://github.com/RAMESHDADI/OS-data_1-set-for-AES).

ORCID

Dadi Ramesh https://orcid.org/0000-0002-3967-8914

REFERENCES

Adamson, A., Lamb, A., & Ma, R. (2014). Automated essay grading.

Agrawal, A., & Agrawal, S. (2018). Debunking neural essay scoring.

Ajay, H. B., Tillett, P. I., & Page, E. B. (1973). Analysis of essays by computer (AEC-II) (No. 8-0102). Washington, DC: U.S. Department of Health, Education, and Welfare, Office of Education, National Center for Educational Research and Development.

Andersen, M. R., Kabel, K., Bremholm, J., Bundsgaard, J., & Hansen, L. K. (2023). Automatic proficiency scoring for early-stage writing. *Computers and Education: Artificial Intelligence*, *5*, 100168.

Burstein, J. (2003). The E-rater® scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis & J. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 113–121). Lawrence Erlbaum Associates Publishers.

Chen, Y., & Li, X. (2023). PMAES: Prompt-mapping contrastive learning for cross-prompt automated essay scoring. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics* (Vol. *1*, pp. 1489–1503). ACL.

Cozma, M., Butnaru, A. M., & Ionescu, R. T. (2018). Automated essay scoring with string kernels and word embeddings. *arXiv preprint arXiv:1804.07954*.

Cummins, R., Zhang, M., & Briscoe, T. (2016). Constrained multi-task learning for automated essay scoring. ACL.

Darwish, S. M., & Mohamed, S. K. (2020). Automated essay evaluation based on fusion of fuzzy ontology and latent semantic analysis. In A. Hassanien, A. Azar, T. Gaber, R. F. Bhatnagar, & M. Tolba (Eds.), *The international conference on advanced machine learning technologies and applications*, *volume 921 of Advances in intelligent Systems and computing* (pp. 566–575). Springer.

Dasgupta, T., Naskar, A., Dey, L., & Saha, R. (2018). Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th workshop on natural language processing techniques for educational applications*, *Association for Computational Linguistics* (pp. 93–102). Association for Computational Linguistics.

Devlin, J., Chang, M.-. W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the Association for Computational Linguistics: Human language technologies* (pp. 4171–4186). Association for Computational Linguistics.

Ding, Y., Riordan, B., Horbach, A., Cahill, A., & Zesch, T. (2020). Don't take "nswvtnvakgxpm" for an answer–the surprising vulnerability of automatic content scoring systems to adversarial input. In *Proceedings of the 28th international conference on computational linguistics* (pp. 882–892). Association for Computational Linguistics.

Doewes, A., & Pechenizkiy, M. (2021). *On the limitations of human-computer agreement in automated essay scoring*. International Educational Data Mining Society.

Dong, F., & Zhang, Y. (2016). Automatic features for essay scoring-an empirical study. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (Vol. *435*, pp. 1072–1077). Association for Computational Linguistics.

Dong, F., Zhang, Y., & Yang, J. (2017). Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)* (pp. 153–162). Associaion for Computational Linguistics.

Fernandez, N., Ghosh, A., Liu, N., Wang, Z., Choffin, B., Baraniuk, R., & Lan, A. (2022). Automated Scoring for Reading Comprehension via In-context BERT Tuning. *arXiv preprint arXiv:2205.09864*.

Horbach, A., & Zesch, T. (2019). The influence of variance in learner answers on automatic content scoring. *Frontiers in Education*, *4*, 28. https://doi.org/10.3389/feduc.2019.00028

Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2019). Get it scored using autosas—An automated system for scoring short answers. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. *33*, pp. 9662–9669).

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Kumar, Y., Bhatia, M., Kabra, A., Li, J. J., Jin, D., & Shah, R. R. (2020). Calling out bluff: Attacking the robustness of automatic scoring systems with simple adversarial testing. *arXiv preprint arXiv:2007.06796*.

Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, *37*(4), 389–405.

Liu, J., Xu, Y., & Zhu, Y. (2019). Automated essay scoring based on two-stage learning. *arXiv preprint arXiv:1901.07744*.

Lun, J., Zhu, J., Tang, Y., & Yang, M. (2020). Multiple data augmentation strategies for improving performance on automatic short answer scoring. *Proceedings of the AAAI Conference on Artifcial Intelligence*, *34*(9), 13389–13396.

Mathias, S., & Bhattacharyya, P. (2018). ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *Proceedings of the eleventh international conference on language resources and evaluation*. LREC.

Mayfield, E., & Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring? In *Proceedings of the fifteenth workshop on innovative use of NLP for building educational applications* (pp. 151–162). Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Muangkammuen, P., & Fukumoto, F. (2020). Multi-task learning for automated essay scoring with sentiment analysis. In *Proceedings of the 1st conference of the Asia-Pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing: Student research workshop* (pp. 116–123). Association for Computational Linguistics.

Ormerod, C. M., Malhotra, A., & Jafari, A. (2021). Automated essay scoring using efficient transformer-based language models. arXiv preprint arXiv:2102.13136.

Page, E. B. (1967). Statistical and linguistic strategies in the computer grading of essays. In *COLING 1967 Volume 1: Conference internationale sur le traitement automatique des langues*. ACL.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation.

Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: A systematic literature review. *ArtifIntell Rev*, *55*, 2495–2527. https://doi.org/10.1007/s10462-021-10068-2

Riordan, B., Flor, M., & Pugh, R. (2019). How to account for mispellings: Quantifying the benefit of character representations in neural content scoring models. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications* (pp. 116–126). Association for Computational Linguistics.

Rodriguez, P. U., Jafari, A., & Ormerod, C. M. (2019). Language models and automated essay scoring. *arXiv preprint arXiv:1909.09482*.

Rudner, L. M., & Liang, T. (2002). Automated essay scoring using Bayes' theorem. *The Journal of Technology, Learning and Assessment*, *1*(2), 1–22.

Süzen, N., Gorban, A. N., Levesley, J., & Mirkes, E. M. (2020). Automatic short answer grading and feedback using text mining methods. *Procedia Computer Science*, *169*, 726–743.

Sakaguchi, K., Heilman, M., & Madnani, N. (2015). Efective feature integration for automated short answer scoring. In *Proceedings of the 2015 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 1049–1054).

Song, W., Zhang, K., Fu, R., Liu, L., Liu, T., & Cheng, M. (2020). Multi-stage pre-training for automated Chinese essay scoring. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)* (pp. 6723–6733).

Sultan, M. A., Salazar, C., & Sumner, T. (2016). Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: Human language technologies* (pp. 1070–1075).

Taghipour, K., & Ng, H. T. (2016). A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1882–1891).

Tay, Y., Phan, M., Tuan, L. A., & Hui, S. C. (2018). SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. *32*).

Wang, J., Chen, J., Ou, X., Han, Q., & Tang, Z. (2023). Multi-level feature fusion for automated essay scoring.

Wang, Y., Wang, C., Li, R., & Lin, H. (2022). On the use of BERT for automated essay scoring: Joint learning of multi-scale essay representation. *arXiv preprint arXiv:2205.03835*.

Wang, Z., Liu, J., & Dong, R. (2018). Intelligent auto-grading system. In *2018 5th IEEE international conference on cloud computing and intelligence systems (CCIS)* (pp. 430–435). IEEE.

Xia, L., Liu, J., & Zhang, Z. (2019). Automatic essay scoring model based on two-layer Bi-directional LongShort term memory network. In *Proceedings of the 2019 3rd international conference on computer science and Artifcial intelligence* (pp. 133–137).

Yang, R., Cao, J., Wen, Z., Wu, Y., & He, X. (2020). Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 1560–1569).

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Zhu, W., & Sun, Y. (2020). Automated essay scoring system using multi-model machine learning. In *CS & IT Conference Proceedings* (Vol. *10*, No. 12). CS & IT Conference Proceedings.

How to cite this article: Ramesh, D., & Sanampudi, S. K. (2024). Coherence-based automatic short answer scoring using sentence embedding. *European Journal of Education*, *00*, e12684. https://doi.org/10.1111/ejed.12684

14653435, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/ejed.12684 by Ege Universitesi, Wiley Online Library on [27/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License