Journal of Information Systems Engineering and Management
2025, 10(2)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Enhancing Essay Grading Efficiency and Consistency through Two-Layer LSTM Models and Attention Mechanisms

Supriya Khaitan[1], Divya Rohatgi[2], Sana Nalband[3], Tejali Mhatre[4], Shweta Patil[5], Rupali Sharma[6]
[1,2,3,4,5,6] Bharati Vidyapeeth Deemed to be University, Department of Engineering and Technology, Navi Mumbai, Maharashtra, India,
supriyakhaitan21@gmail.com[1], divi.rohatgi@gmail.com[2], khansanakbar@gmail.com[3], tejalimhatre@gmail.com[4],
shweta20.patil@gmail.com[5,] rupalidineshsharma@gmail.com[6]

ARTICLE INFO

Received: 25 Oct 2024

Revised: 23 Dec 2024

Accepted: 12 Jan 2025

ABSTRACT

Pandemic has led people to sought easier solutions for assessing tasks given to students and grading them. As a potential solution to this problem, Automatic Essay Grading (AEG) has gained considerable attention in recent years. The Kaggle ASAP-AES dataset and the Long Short-Term Memory (LSTM) model with two layers are utilised in this research. The dataset contains essays from a standardised test, providing a more realistic and challenging scenario for AutoGrader. The characteristics from essays are extracted using methods like word tokenization and Feature Vector Creation, pre-processing procedures like stemming and stop-word removal were used. The training data is considered, and an implementation of an average of five folds, consisting of 50 epochs, is made to help the model better focus on pertinent passages in the essay, attention processes was also included. Proposed method provides valuable insights into the potential of AutoGrader to improve the efficiency of and consistency of essay grading, and highlights the effectiveness of the two-layer LSTM model. Experimental findings demonstrate that the two-layer LSTM model surpasses other models in terms of accuracy and efficiency making it a viable strategy for computerized essay scoring systems.

Keywords: NLTK, QWK, Word2Vec, LSTM, RNN, Automatic Essay Grading.

INTRODUCTION

Automation Essay Grading (AES) System combines linguistics, education, and natural language processing (NLP). The traditional AES models use handmade features, which essentially require a manual selection of features to associate and fit the model, making it difficult to perform well on different as well as ambiguous tasks. The essay scoring models that were generally viewed were based on the holistic scoring method. This method generally assigned a single score over the impression of how the essay was written for the problem statement. Such type of generalized approach does not provide comprehending feedback to the user and hence limits the enhancement of written skills [1]. Deep learning models have demonstrated significant promise in automated essay scoring (AES) [2] problems in recent years. Recurrent neural networks (RNNs) [3] have become more well-liked as a result of their capacity to recognize long-term relationships in sequential data. Neural network approaches have emerged as a promising solution, but Deep Neural Networks (DNNs) properly work with a larger subset of data. Pre-trained models have been shown to be helpful for fine-tuning to match various tasks without training new models from scratch. Bidirectional encoder representations from transformers, a pre-trained model (BERT) [4][5] has displayed exceptional performance in several language-based applications, unfortunately, there is little study on using language models.

By delivering scores from numerous analytic rating dimensions, the two-layer LSTM [6] model for automated essay scoring technique may assist raters and students in differentiating between different writing quality dimensions and making changes for each dimension. It was taught to forecast a variety of writing-related scores, including organization, coherence, and vocabulary. These ratings can be used to provide the writer with targeted comments, highlighting their writing's strong and poor points. Instead of earning a single score that does not offer specific feedback for development, students can aim their efforts to improve in these areas by concentrating on certain aspects of writing quality.

This paper is divided in following sections, Section II gives the detailed literature review of the various auto grading schemes, Section III gives the details of dataset used for experimental purpose. Section IV explains proposed method for solving multi-dimensional essay scoring issues that combines multi-task learning with the 2-layer LSTM model. Without the need for additional custom engineering, technique encompasses a variety of characteristics, and managed lengthy essays by employing a hierarchical approach with attention pooling. Section V gives the experimental results on the widely used Automated Student Assessment Prize (ASAP) dataset show that approach used yields cutting-edge results and improves the generalizability and performance of the AES model. Section VI compares the proposed model with the current models followed by conclusion.

RELATED WORK

Researchers in linguistics, pedagogy, and natural language processing (NLP) have all expressed a strong interest in automated essay scoring (AES). Traditional techniques and neural network-based methods are the two basic AES implementation strategies. Whereas neural network-based approaches employ automatic feature selection with raw text input, traditional techniques often require handcrafted features to create the model. Deep neural networks (DNNs) [2][7] have recently demonstrated impressive AES outcomes without the requirement for manually created features. Nevertheless, for DNNs to excel at a given task, they frequently need a sizable amount of labelled data, which is not always accessible.

The use of neural networks for automated essay assessment has become more popular in recent years. To achieve accurate and effective automated essay scoring, many researchers have suggested a variety of neural network-based models, including LSTM [6][9][11], attention-based models, and BERT-based models [14][21][22], machine learning model [12][28] among others. These models examine and comprehend the content of essays using a variety of methodologies, such as sentiment analysis, syntactic analysis, and semantic analysis [17]. Unfortunately, only a few research have investigated the application of the 2-layer LSTM model for AES [27]. The approach can handle sequential data processing and can identify long-term relationships between words in a text. Moreover, transfer learning has been demonstrated to be successful in enhancing deep learning models performance in natural language processing. Traditional essay scoring techniques have come under fire for their shortcomings in delivering detailed feedback for writing enhancement. Certain writing characteristics are not disclosed by holistic scoring, which gives an essay a single grade based on its overall impression. Without the requirement for feature engineering, multi-task learning may be utilized to jointly predict multi-dimensional scores to solve this problem.

Automated essay grading methods based on deep learning, like the 2-layer LSTM model, have grown in acceptance in recent years. Recurrent neural networks (RNNs) are used in these systems to reflect the long-term relationships in student writings, a sort of deep learning architecture [8][10]. Using RNNs makes it possible to provide more precise feedback for writing progress as well as forecast scores on a variety of writing quality dimensions. Regression [15] and ranking losses-based models are additive to one another and BERT based models fine-tuning by itself is inadequate and can be improved by using multi-loss objectives [15]. Wang et al. proposed the Intelligent Auto-Grading System [17] using the attention-based bi-LSTM model for intelligent grading providing an analysis of the model-to-evaluation ratio through the QWK standards stating: Memory-Augmented Neural - 0.83, Bi-LSTM with Attention - 0.83 and AES by Optimizing Human-Machine Agreement - 0.80. Tsegaya et al. [20] implemented the Off-Topic Essay Detection using C_BGRU Siamese model conducting the Text representation layer, Feature extraction using the CNN layer, and BGRU concluding the result using the Multi-channel CNN with pooling, the following results were obtained for recall, precision, and F1-score: max[R]:89.2%, max[P]:91.11%, and max[F1]:89.32%. The maximum recall, maximum precision, and maximum F1-score obtained using the vector analysis approach were 82.31%, 91.71%, and 86.76%, respectively. An approach based on text mining was proposed to grade short answers [29]. Grading was done on the basis of distance calculation between two answers.

A study over attention-based neural network as undergone to prioritise the understanding of the essay domains and further cover sentiment analysis through sentence similarity methodology. The in-depth statistics indicates the use of a number of iterations to compute higher accuracy of the model but still not being able to score high. For the purpose of capturing the significance of various sections of the essay, the suggested model integrates both local and global attention processes. The model additionally makes use of a bidirectional LSTM layer and pre-trained word embeddings to capture the temporal information of the input sequence. As compared to other cutting-edge algorithms, the suggested model performed competitively. Brief answers were rated using text mining [16], comparison of student responses with model replies was done, by measuring the space between two phrases. The use of model vocabulary is crucial in this kind of grading since it allows for the assignment of a grade based on the student's response as well as feedback. There is a 0.81 correlation between the student response and the model response. In a study one suggested a regression-based method [15] for grading automated essays. Training was done to predict essay scores using a mix of lexical, syntactic, and discourse variables. For the AESOP dataset, the system achieved a correlation value of 0.823 with human ratings. The findings demonstrate that the suggested technique can score essays automatically with excellent accuracy.

DATASET

The Kaggle ASAP-AES (Automated Essay Scoring) dataset is available to the general public and contains essays from standardized tests. The dataset was created by Educational Testing Service (ETS) and made accessible for academic usage on Kaggle. More than 12,000 articles written by students in response to inquiries on a variety of topics, such as social concerns, technology, and education, make up this collection. More than 130,000 student essays that were prepared in response to ACT program prompts may be found there. Essays from all four grade levels and grading rubrics, spanning a wide range of themes, are included in the dataset. The question, grade level, essay score, and different linguistic aspects are all included in the complete information that is provided for each essay.

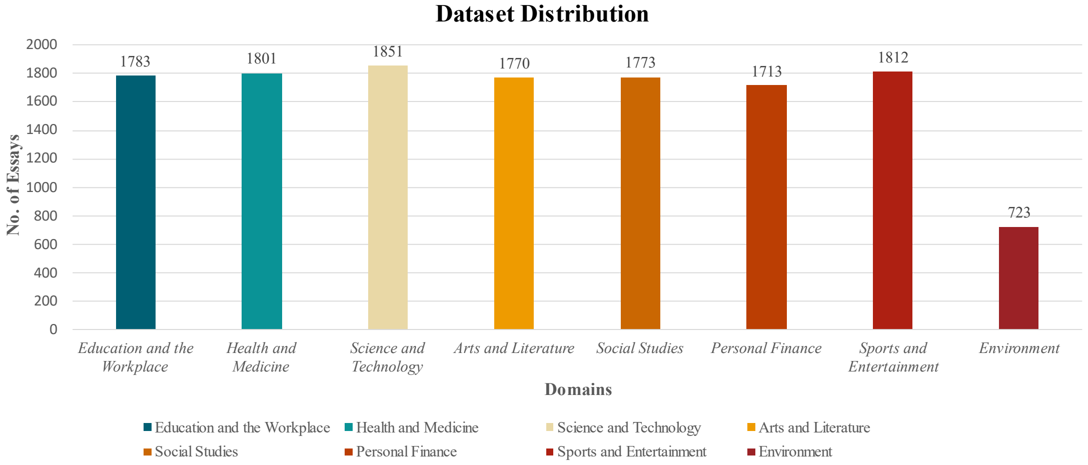


Figure 1. Categorization of the essay domains into the classified 8 domains.

The essays are graded by many human raters who award them a score between 0 and 6 depending on several criteria, including the essay's overall writing quality, coherence, and persuasiveness. Together with the responses from the students' prompts [30], the dataset also includes the rater's evaluations of those responses. As a result, the Kaggle ASAP-AES dataset is an excellent tool for developing and testing automated essay-scoring algorithms.

Figure 1 and 2 gives the categorization of essay in dataset.

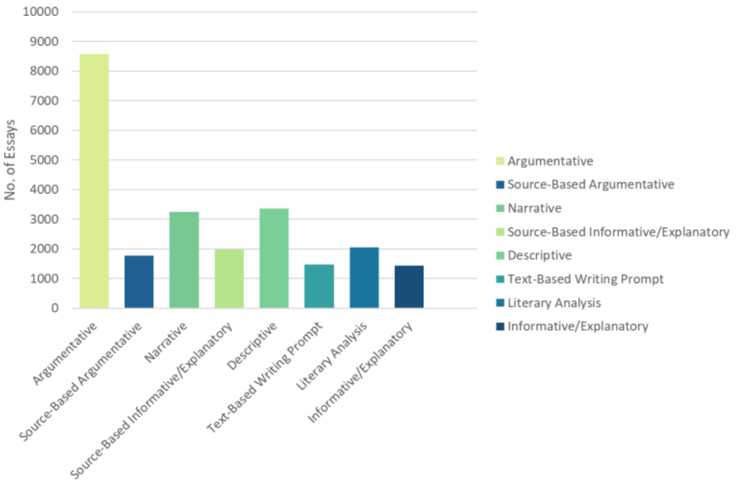


Figure 2. Plotting the number of essays to the type of essay in the dataset

The articles cover a wide range of subjects, including history, literature, and science. The subjects are organized into eight sets, each of which includes an essay question and associated pieces. The open-ended nature of the questions encourages students to share their thoughts and opinions. The model may learn the complexities of the English language and how to assess writings using many criteria, such as grammar, coherence, and logic, by being trained on a sizable corpus of essays. The Kaggle ASAP-AES dataset follows a strict means to interpret analysis over the common attributes of an essay namely, Average Length, Max Length, etc. Table 2. Table 3 gives statistical estimation of the dataset.

Table 1. Layout for essay domain, Average length, and Max. Length

Essay Domain	Average Length	Max. length
C-DEMO	328	1219
C-EE	343	1486
C-EOC	348	1293
C-ETE	369	1271
C-PT	331	1495
C-RST	342	1551
C-WA	360	1651
C-QQP	348	351

Table 2. Kaggle ASAP-AES dataset statistical estimations of each rater domain

Rating Domain	Mean	Minimum	Maximum	Standard Deviation	IRR Measure	IRR Score
Holistic Score	2.996	0	6	1.461	Cohen's kappa	0.82
Grammar	3.469	0	6	1.474	Fleiss' kappa	0.64
Lexicons	3.189	0	6	1.378	Fleiss' kappa	0.64
Global Organization	2.781	0	6	1.531	Fleiss' kappa	0.64
Local Organization	2.808	0	6	1.512	Fleiss' kappa	0.64
Supporting Ideas	2.955	0	6	1.424	Fleiss' kappa	0.64

PROPOSED METHODOLOGY

The approach used in proposed system utilizes the Kaggle ASAP-AES dataset, a 2-layer LSTM, a recurrent neural network (RNN) and word embeddings. The input texts were first tokenized and changed to lowercase followed by deletion of the punctuation and stop words. Lastly, pre-trained GloVe embeddings is used to transform the words into word embeddings. Thereafter, a 60-20-20 split between the training and validation sets was applied to the pre-processed data. In proposed model construction, a dense layer with a sigmoid activation function is placed after a two-layer LSTM. The pre-processed text is transferred via the embedding layer after being sent into the input layer. Each word in the input text is mapped by the embedding layer to a specific pre-trained GloVe embedding vector. The output of the embedding layer is then sent to the two LSTM layers. The output of the last LSTM layer is sent into a dense layer that has a sigmoid activation function. The binary cross-entropy loss was used in proposed methodology as the loss function and the rmsprop optimizer to train the model. Using 64 batches, the model was trained across 50 epochs, early halt during training was used to avoid overfitting. Figure 3 shows the proposed methodology used in this paper.

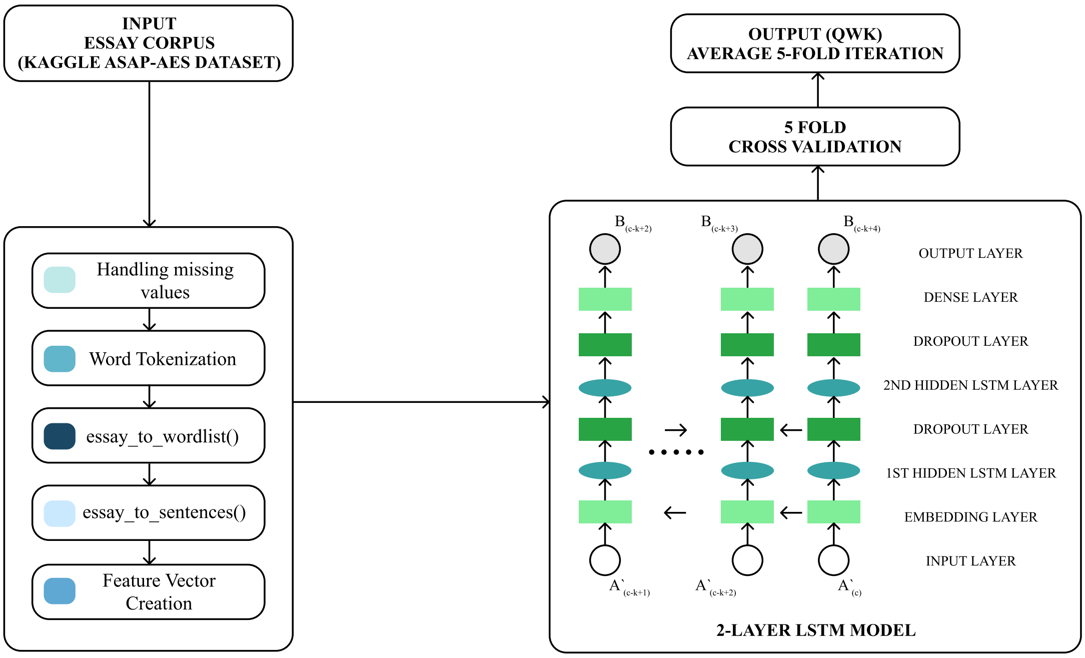


Figure 3. Model for multi-approach-based dimensional task evaluation.

Algorithm 1: Input Representation module for the layers of models used

Input:

Essays (list of strings)
Additional Features (list of lists)

Output:

Sequences of embedded word vectors with additional features (list of lists)

Steps:

1. Pre-process the essays by removing any unnecessary HTML tags, converting all letters to lowercase, and removing any punctuation marks.
2. Tokenize the essays into individual words using the Natural Language Toolkit (NLTK) library.
3. Use the essay corpus to train a Word2Vec model, employ negative sampling and the Skip-Gram technique with window size of 5, and the word embedding vector's size to 300.
4. Embed each word in the essays using the Word2Vec model.
5. For each essay, concatenate the sequence of embedded word vectors with additional features (number of words, number of sentences, and average sentence length).
6. Pad the resulting sequence of vectors with zeros to ensure consistent length.
7. Return the sequences of embedded word vectors with additional features for each essay.

Essay-Level Analysis

To determine the anticipated essay score, we apply the following formula at the essay level:

$$score_e = l_{e2}(l_{e1}(S_e) \oplus \bar{a}_e) \oplus \bar{b}_e \qquad \text{Eq (1)}$$

Where:

- $score_e$ – is the predicted score for essay ‘e’
- (S_e) – is the sequence of embedded words vector for essay ‘e’
- l_{e1} & l_{e2} – are the LSTM layers that perform the essay-level analysis
- $\oplus$ - represents concatenation of two vectors
- $\bar{a}_e$ & $\bar{b}_e$ – are the learned bias vectors for sentence ‘j’

Sentence-Level Analysis

At the sentence level, we use the following formula to calculate the predicted score for a sentence:

$$score_{s_j} = l_{s2}\left(l_{s1}\left(S_{s_j}\right) \oplus \bar{a}_j\right) \oplus \bar{b}_j \qquad \text{Eq (2)}$$

Where:

- $score_{s_j}$ – is the predicted score for sentence 'j'
- $\left(S_{s_j}\right)$ – is the sequence of embedded words vector for sentence 'j'
- l_{s1} & l_{s2} – are the LSTM layers that perform the sentence-level analysis
- $\oplus$ - represents concatenation of two vectors
- $\bar{a}_j$ & $\bar{b}_j$ – are the learned bias vectors for sentence 'j'

Attention Pooling

A method called attention pooling enables the model to concentrate on those segments of the input sequence that are most important to the job at hand. Self-attention, also known as intra-attention, is used in proposed methodology to compute a weight for each input token in the essay. This weight represents the importance of the token for the prediction of the essay score. The weights are computed using the output of the second LSTM layer, which captures the contextual information of the input sequence. The energy scores for each token in the input sequence are computed using following equation Eq (3):

$$e\_score_{i} {=} s^T \tanh(X_h g_i + d_h) \qquad for\ i = 1,2,3,\ldots\ldots,n \qquad \text{Eq (3)}$$

The significance of each token for the forecast of the essay score is represented by the attention weights. The ultimate representation of the text is created by computing a weighted sum of the concealed states:

$$r = \sum_{i=1}^{n} \alpha_i g_i \qquad \text{Eq (4)}$$

where 'r' is the final representation of the essay and 'g_i' is the hidden state of the second LSTM layer for the token "i." As a result, the model may concentrate its predictions on the crucial portions of the input sequence. The hyperparameters of a two-layer LSTM model for essay grading may vary depending on the specific implementation and dataset. However, here are some commonly used hyperparameters:

1. **Learning rate:** This determines how quickly the model adjusts the weights of the network during training. Common values range from 0.001 to 0.1.
2. **A number of epochs:** This specifies how many times the whole dataset was used to train the model. The usual values are between 10 and 100.
3. **Batch size:** How many samples are processed in a batch before the model's weights are adjusted. From 32 to 128 are frequent values.
4. **Dropout rate:** This is the probability that a neuron is randomly dropped out during training. Common values range from 0.2 to 0.5.
5. **Number of neurons per layer:** This determines the number of neurons in each layer of the model. Common values range from 64 to 512.
6. **Activation function:** The number of neurons in each layer of the model is determined by this. The range of typical values is 64 to 512.

These hyperparameters can be tuned using a grid search or a random search approach to find the best combination for the given dataset and task. Table 3 shows base hyperparameters for the proposed methodology.

Table 3. Base Hyperparameters chosen for the model

Base Hyperparameters:	Values
Rate of Learning	0.001
Number of epochs	20
Batch size	64
Dropout rate	0.2
Number of neurons per layer	256
Activation function	tanh

MODEL-BASED ANALYSIS

RNN-BASED MODEL

- **Layer lookup database** - This layer transforms each word in a text into a G-dimensional word-embedding representation. Similar vectors exist in a word-embedding form, which is a word vector with real values that is fixed in length, for words with similar meanings. Assume that V is an essay collection vocabulary list, that AA represents a G|V| -dimensional trainable embeddings matrix and that X_{wt} represents a |V| - encoding in one dimension quickly of the h_{th} word in each essay, and. The embedding representation w_{wt} that corresponds to w_t can then be calculated as a dot product using the formula X_{wt} = $AA_{X_{wt}}$.
- **Computational layer -** Convolution neural networks (CNNs) are used in this layer to extract local linguistic relationships from a series of word-embedding vectors. This layer is used to capture the local textual associations between the words in a window of c words c-gram words given an input series of X_{w1}, X_{w2},..., X_{wL} (where L is the number of words in a particular essay). The t-th result of this layer can be calculated specifically as follows.

$f\,(X_c \cdot [X_{wt}\,, X_{wt+1},\ldots, X_{wt+1+c-1}]{+}d_c),$ *Eq (5)*

where [,] denotes the concatenation of the supplied components and W_c and b_c are trainable weight and bias factors. To maintain the input and output sequence lengths, zero buffering is given to this layer's outputs. This optional component has frequently been left out of recent research

- **Repeating layer -** At this layer, time series relationships in an input sequence are often captured using a representative RNN called the long short-term memory (LSTM) network, which generates a vector at each timestep. The most common type of LSTM is a single-layer straight LSTM, though bi-directional and multiple LSTMs are also frequently used.
- **Pooling layer -** This layer transforms the recurrent layer's output hidden vector series, $'g_{h1}, g_{h2}, \ldots, g_{hL}'$, a fixed-length mixed hidden vector, where $'g_{ht}'$stands for the hidden vector of the h_{th} output. Using mean-over-time pooling to determine an average vector.

$\tilde{g} \;=\; \frac{1}{L}\sum_{t=1}^{L} g_t\,,$ Eq (6)

is frequently employed because it frequently offers steady precision. Two more frequently employed pooling methods are the last pool (Alikaniotis et al. 2016), which makes use of the 'h_{hl}' recurrent layer's output, and an attention pooling layer (Dong et al. 2017), which we will explore later on in this study.

Stimulation of the sigmoid in a linear layer This layer converts a pooling layer result into a scalar value between [0, 1] using the sigmoid function.

$\sigma(X_o\,.\tilde{g}\;+\;d_o)\,,$ Eq (7)

where d_o stands for bias factors and X_o is a weight matrix. The sigmoid function is represented by ().

The mean-squared error (MSE) between the predicted and gold-standard outcomes serves as the loss function for model training most of the time. The MSE loss function is specified as follows, where y_n is the gold-standard score for the n_{th} essay and z_n is the expected score.

$\frac{1}{N}\sum_{n=1}^{N}(z_n \;-\; \bar{z}_n)^2\,,$ Eq (8)

N is the total amount of entries. Although the projected scores are linearly rescaled to the initial score range in the prediction phase, the model training is carried out after normalizing the gold standard scores to [0, 1].

LAYER LSTM MODEL

The proposed model has LSTM neural network with two layers, to analyse sequential data, a recurrent neural network (RNN) using the LSTM architecture is used. The input sequence is pre-processed text data from the student answers. Two LSTM layers are placed on top of one another to form the two-layer LSTM network. The input sequence is sent into the first LSTM layer, which creates an output sequence that is fed into the second LSTM layer. An LSTM layer is made up of memory cells, input gates, forget gates, and output gates.

The input gate's formula is provided by:

$i_t \;=\; \sigma(X_i\,\cdot\,[g_{t-1}, v_t]\;+\;d_i)$ Eq (9)

The sigmoid activation function is, the input gate's weight matrix is X_i, the prior hidden state is g_{t-1}, the current input vector is v_t, and the input gate's bias vector is d_i. The forget gate formula is provided by:

$l_t = \sigma(X_f \cdot [g_{t-1}, v_t] + d_f)$ Eq (10)

The sigmoid activation function is, the forget gate's weight matrix is X_f, the prior hidden state is g_{t-1}, the current input vector is v_t, and the bias vector for the forget gate is d_f. The output gate's formula is given by

$op_t = \sigma(X_{op} \cdot [g_{t-1}, v_t] + d_{op})$ Eq (12)

The sigmoid activation function is, the output gate's weight matrix is X_{op}, the prior hidden state is g_{t-1}, the current input vector is v_t, and the output gate's bias vector is d_{op}. The following Equation is use for the candidate memory cell state:

$\bar{c}_t = tanh(X_c \cdot [g_{t-1}, v_t] + d_c)$ Eq (13)

where tanh is the hyperbolic tangent activation function, X_c is the weight matrix for the candidate memory cell state, g_{t-1} is the previous hidden state, v_t is the current input vector, and d_c is the bias vector for the candidate memory cell state.

Using the input gate and forget gate, the candidate memory cell state, $\bar{c}_t$, and the prior memory cell state, $\bar{c}_{t-1}$, are combined to update the current memory cell state, c_t:

$c_t = l_t \cdot l_{t-1} + i_t \cdot \bar{c}_t$ Eq (14)

Using the output gate, the current memory cell state, c_t, is used to calculate the current hidden state, g_t:

$g_t = o_t \cdot tanh(c_t)$ Eq (15)

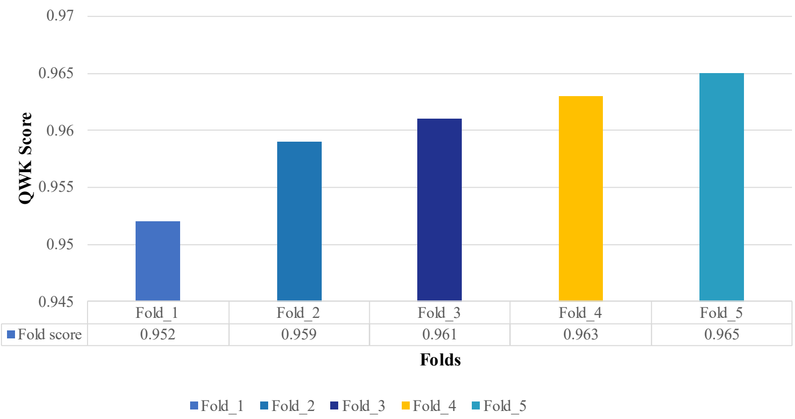
An output sequence generated by the two-layer LSTM network is sent into the dense output layer. The probability distribution across the potential holistic scores is computed by the dense output layer function. Each conceivable holistic score is represented by a single neuron in the output layer, and each neuron's output shows the likelihood that the given answer corresponds to that score.

EVALUATION METRICS

Lorem •QWK -The degree of agreement between expected and actual essay scores was measured using the Quadratic Weighted Kappa (QWK) method. As QWK is the primary evaluation metric for the Kaggle ASAP-AES dataset, we used it as the primary metric in our evaluation as well. QWK considers both the accuracy and the ordinal difference between the expected and actual essay scores.

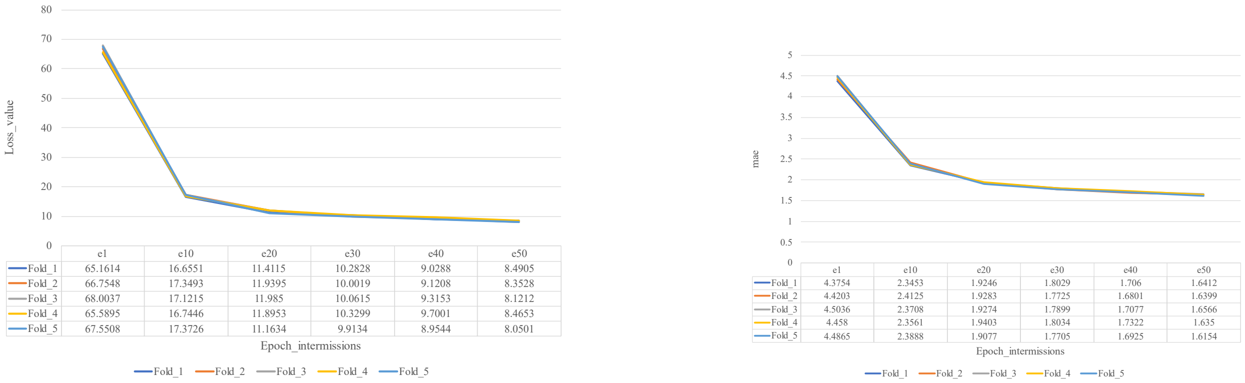
- Mean Absolute Error - The average absolute difference between the expected and actual essay scores is measured using the statistic known as the mean absolute error (MAE). As MAE offers a precise measurement of the discrepancy between the expected and actual scores, we employed it as a supplemental metric for the essay quality challenge.
- Accuracy - This indicator counts the proportion of all labels that were properly predicted. Because the grammaticality test is a binary classification job, accuracy was chosen as the supplementary parameter.

The performance of model was evaluated using the relevant evaluation metric(s) for each job. We utilized QWK and MAE to assess essay quality. QWK is utilized for coherence because it is a regression problem. As it was a binary classification problem, accuracy was employed to measure grammaticality. A distinct validation dataset was utilized that was not used for training to make sure the assessment was fair., random partitioning the dataset into k-folds, utilizing k-1 folds for training and the remaining fold for validation, was used. A Pearson correlation value of Z was also attained between the anticipated scores and the human scores these outcomes show how well suggested method for multi-topic scoring works. The findings show that the Word2Vec embedding layer considerably enhanced model's performance, particularly for the coherence challenge. The contextual information of the essays was successfully captured by the two-layer LSTM model. The model's overall performance was enhanced by the task-specific dense layers' success in learning the attributes that were unique to each task.



	Fold_1	Fold_2	Fold_3	Fold_4	Fold_5
Fold score	0.952	0.959	0.961	0.963	0.965

Figure 4. Plotting of the 5-fold cross-validation assessment.



	e1	e10	e20	e30	e40	e50
Fold_1	65.1614	16.6551	11.4115	10.2828	9.0288	8.4905
Fold_2	66.7548	17.3493	11.9395	10.0019	9.1208	8.3528
Fold_3	68.0037	17.1215	11.985	10.0615	9.3153	8.1212
Fold_4	65.5895	16.7446	11.8953	10.3299	9.7001	8.4653
Fold_5	67.5508	17.3726	11.1634	9.9134	8.9544	8.0501

	e1	e10	e20	e30	e40	e50
Fold_1	4.3754	2.3453	1.9246	1.8029	1.706	1.6412
Fold_2	4.4203	2.4125	1.9283	1.7725	1.6801	1.6399
Fold_3	4.5036	2.3708	1.9274	1.7899	1.7077	1.6566
Fold_4	4.458	2.3561	1.9403	1.8034	1.7322	1.635
Fold_5	4.4865	2.3888	1.9077	1.7705	1.6925	1.6154

Figure 5 (a).Plot for epoch_intermissions v/s Loss_value (b). Plot for epoch_intermissions v/s mae

The average loss value for the model throughout the whole training set is represented by the epoch loss for each epoch. Mean squared error (MSE), which assesses the discrepancy between expected and actual essay scores, serves as the loss function in our model. The model performs better at forecasting essay scores the smaller the epoch loss. The average absolute difference between the predicted and actual essay scores over the whole training set is represented by the epoch mae for each epoch.

The proposed model consisted of a Dense layer with 5 neurons. Visualization of the dense 5 function's histogram distribution for the bias 0 and kernel 0 evaluator sections is done.The bias values were primarily spread around zero with a few outliers, according to the bias_0 histogram plot. This can mean that the model is developing unbiased prediction capabilities and is not biased towards any one value. A well-trained model should have weights that are normally distributed around zero, as seen by the kernel_0 histogram graphic. This shows that the model has mastered the art of providing a concise and detailed representation of the input data. To lessen overfitting, the Dropout function with a rate of 0.2 is used. The bulk of the dropout rates were near to 0.2 on the histogram plot for the Dropout function, indicating that the model is not overfitting on the training data.

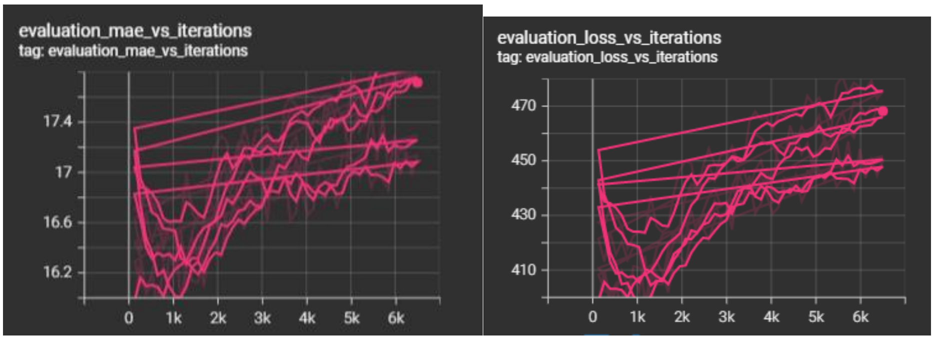


Figure 6(a) and 6(b). Plots made for evaluation_mae v/s iterations and evaluation_loss v/s iterations withing training period

COMPARATIVE STUDY

Tabel 4 Comparing the various models with their special features used in their proposed methodologies as well as their evaluation metrics and resultant score.

Reference	Methodology	Dataset	Features	Metrics for Evaluation and Result
[21]	CNN plus the LSTM Neural Network	ASAP-AES Kaggle	statistics, word embedding, and content-based features	QWK 0.764
[25]	LSTM (Memory Network)	ASAP Kaggle	Features of statistics	QWK 0.78
[26]	BiGRU Siamese Architecture	Amazon Mechanical Turk online research service collected summaries	Embedding words	Accuracy 55.2
[27]	Semantic LSTM, HAN (hierarchical attention network) Neural network	ASAP Kaggle	Sentence coherence and word embedding	QWK 0.801
[20]	Rule-based algorithms and algorithms based on similarity	ASAP Kaggle	Resemblance based	Accuracy 0.68
[28]	Machine learning classifier XGBoost	ASAP Kaggle	Type token ratio, word count, POS, parse tree, coherence, and cohesiveness	Accuracy 68.12
[2]	Models for Item Response Theory (CNN-LSTM, BERT)	ASAP Kaggle	-	QWK 0.749
[29]	Mining text	Introductory computer science class in the University of North Texas, Student Assignments	Sentence analogy	Correlation score 0.81

CONCLUSION

In this paper, model for automated essay scoring (AES) using the Kaggle ASAP-AES dataset is proposed. The model used a two-layer LSTM with dropout and thick layers to achieve the goal of holistic scoring. Using the power of word embeddings, the raw textual input from our model was transformed into numerical vectors, which were then fed into a two-layer LSTM model for score prediction. Furthermore, a number of techniques, such as batch normalisation, dropout, and thick layers, were applied to further improve the performance of the model. The suggested model offers a fresh method for accurately and precisely automating essay grading. To further enhance the functionality of our model, additional methods such as hierarchical approaches might perhaps be investigated. The state-of-the-art models in AES currently in use were contrasted with suggested model. The outcomes demonstrated that proposed model performed better in terms of accuracy and consistency than the earlier models. In conclusion, our suggested model provides a potential method for automatically grading essays and has demonstrated to be very precise and efficient on the Kaggle ASAP-AES dataset.

REFRENCES

[1] Haussein, M.A.; Hassan, H.; Nassef, M. Automated language essay scoring systems: A literature review. PeerJ Comput. Sci. 2019, 5, e208.

[2] Uto M, Okano M (2020) Robust Neural Automated Essay Scoring Using Item Response Theory. In: Bittencourt I, Cukurova M, Muldner

[3] K, Luckin R, Millán E (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science, vol 12163. Springer, Cham M. Chen and X. Li, "Relevance-based automated essay scoring via hierarchical recurrent model", 2018 International Conference on Asian Language Processing (IALP), pp. 378-383, 2018, November.

[4] L. Xia, J. Liu and Z. Zhang, "Automatic Essay Scoring Model Based on Two-Layer Bi-directional Long-Short Term Memory Network", Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence, pp. 133-137, 2019, December.

[5] C. M. Ormerod, A. Malhotra and A. Jafari, "Automated essay scoring using efficient transformer-based language models", arXiv preprint, 2021.

[6] H. Chimingyang, "An Automatic System for Essay Questions Scoring based on LSTM and Word Embedding", 2020 5th International Conference on Information Science Computer Technology and Transportation (ISCTT), pp. 355-364, Nov. 2020.

[7] Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

[8] Kumar, V. S., & Boulanger, D. (2021). Automated essay scoring and the deep learning black box: How are rubric scores determined?. International Journal of Artificial Intelligence in Education, 31, 538-584.

[9] L. Xia, J. Liu and Z. Zhang, "Automatic Essay Scoring Model Based on Two-Layer Bi-directional Long-Short Term Memory Network", Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence, pp. 133-137, 2019, December.

[10] H. Li and T. Dai, "Explore Deep Learning for Chinese Essay Automated Scoring", Journal of Physics: Conference Series, vol. 1631, no. 1, pp. 012036, 2020, September.

[11] F. Nadeem, H. Nguyen, Y. Liu and M. Ostendorf, "Automated essay scoring with discourse-aware neural models", Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 484-493, 2019, August.

[12] V. V. Ramalingam, A. Pandian, P. Chetry and H. Nigam, "Automated essay grading using machine learning algorithm", Journal of Physics: Conference Series, vol. 1000, no. 1, pp. 012030, 2018, April.

[13] M. Uto, Y. Xie and M. Ueno, "Neural Automated Essay Scoring Incorporating Handcrafted Features", Proceedings of the 28th International Conference on Computational Linguistics, pp. 6077-6088, 2020, December.

[14] M. Beseiso and S. Alzahrani, "An Empirical Analysis of BERT Embedding for Automated Essay Scoring", International Journal of Advanced Computer Science and Applications,2020.

[15] Y Ruosong, C Jiannong, W Zhiyuan, W Youzheng and H. Xiaodong, "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking", Findings of the Association for Computational Linguistics, pp. 1560-1569, 2020.

[16] J. O. Contreras, S. Hilles and Z. B. Abubakar, "Automated Essay Scoring with Ontology based on Text Mining and NLTK tools", 2018 Int. Conf. Smart Comput. Electron. Enterp. ICSCEE 2018, pp. 1-6, 2018.

[17] Y. Wang, Z. Wei, Y. Zhou and X. Huang, "Automatic essay scoring incorporating rating schema via reinforcement learning", Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, pp. 791-797, 2020.

[18] R. Ridley, L. He, X. Dai, S. Huang and J. Chen, "Prompt Agnostic Essay Scorer: A Domain Generalization Approach to Cross-prompt Automated Essay Scoring", Aug. 2020.

[19] M. J. Hazar, Z. H. Toman and S. H. Toman, "Automated Scoring for Essay Questions in E-learning", J. Phys. Conf. Ser, vol. 1294, no. 4, pp. 0-14, 2019.

[20] Tashu TM, Horváth T (2020) Semantic-Based Feedback Recommendation for Automatic Essay Evaluation. In: Bi Y, Bhatia R, Kapoor S (eds) Intelligent Systems and Applications. IntelliSys 2019. Advances in Intelligent Systems and Computing, vol 1038. Springer, Cham F.

[21] Dong and Y. Zhang, "Automatic features for essay scoring—An empirical study", Proc. EMNLP, pp. 1072-1077, Nov. 2016.

[22] M. Beseiso and S. Alzahrani, "An empirical analysis of BERT embedding for automated essay scoring", Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 10, pp. 204-210, 2020.

[23] D. E. Powers, J. C. Burstein, M. Chodorow, M. E. Fowles and K. Kukich, "Stumping e-rater: challenging the validity of automated essay scoring", Computers in Human Behavior, vol. 18, no. 2, pp. 103-134, 2002.

[24] A. N. Oktaviani, M. Z. Alief, L. Santiar, P. D. Purnamasari and A. A. P. Ratna, "Automatic Essay Grading System for Japanese Language Exam using CNN-LSTM," 2021 17th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering, Depok, Indonesia, 2021, pp. 164-169, doi: 10.1109/QIR54354.2021.9716165.

[25] Zupanc K, Savić M, Bosnić Z, Ivanović M (2017) Evaluating coherence of essays using sentence-similarity networks. In: Proceedings of the 18th International Conference on Computer Systems and Technologies p 65–72.

[26] Ruseti S, Dascalu M, Johnson AM, McNamara DS, Balyan R, McCarthy KS, Trausan-Matu S (2018) Scoring summaries using recurrent neural networks. In: International Conference on Intelligent Tutoring Systems p 191–201. Springer, Cham.

[27] Liang G, On B, Jeong D, Kim H, Choi G (2018) Automated essay scoring: a siamese bidirectional LSTM neural network architecture. Symmetry 10:682.

[28] Salim, Y., Stevanus, V., Barlian, E., Sari, A. C., & Suhartono, D. (2019, December). Automated English Digital Essay Grader Using Machine Learning. In 2019 IEEE International Conference on Engineering, Technology and Education (TALE) (pp. 1–6). IEEE.

[29] 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.