Migration Letters

Volume: 20, No: S13(2023), pp. 393-406

ISSN: 1741-8984 (Print) ISSN: 1741-8992 (Online)
www.migrationletters.com

Deep Learning Based Automatic Answer Scoring Through Bi-Directional LSTM

T.S Adharsh[1*] and M.K Jeyakumar[2]

Abstract

Due to population expansion and the increasing importance of education, it is becoming increasingly difficult for assessors to evaluate the correctness and relevance of the responses provided by students. The LSTM model was initially used to build the answer-scoring system. The Bi-LSTM model has been designed with callbacks to acquire the student answer scoring system due to the LSTM's limitations for optimal scoring. The proposed system has been implemented using the ASAP Short Answer Scoring dataset. The results show that the system developed using Bi-LSTM displays better performance than LSTM.

Keywords: *Answer Scoring, Answer Evaluation, Deep Learning, LSTM, Bi-directional LSTM, Word Embedding, TF-IDF.*

Introduction

The number of solutions utilised for automated answer grading has grown significantly during the past few years. According to Valenti et al., some of them featured lexical and syntax analysis as well as grammar verification of the student's response [1]. Responses from pupils are highly focused and concise in nature. It has been determined whether a student's response corresponds to a certain word or phrase in the rubric text using regular expressions, text templates, or patterns. When determining a student's capacity for learning, the assessment is crucial. Most automated evaluation systems primarily cater to multiple-choice questions, making it challenging to assess short and essay responses. The education sector has become increasingly reliant on machine learning and Natural Language Processing (NLP) as a result of the growing popularity of online educational platforms and computer-based testing. However, traditional evaluation techniques like pattern matching and language processing using simple programming languages are no longer practical [2]. The issue here is that there are multiple possible answers to the same question from students, each with a unique justification. Therefore, we must assess each response to the question.

Automated Essay Scoring (AES) refers to the process of utilizing computers to automatically assess student responses and assign scores or grades according to relevant criteria. Essay qualities are several characteristics of the essay that can help explain the grade that was given to it. According to Persing and Ng [3], some examples of essay characteristics include content (how much material is included in the essay), organisation (how well the essay is structured), style (how well the essay is written), prompt adherence (how much the essay stays on topic for the essay prompt), etc. Instead of focusing on the

[1*] Department of Computer Applications, Noorul Islam Centre for Higher Education, Kumaracoil, Kanyakumari(dist), TamilNadu, India. adharshts@gmail.com
[2] Department of Computer Applications, Noorul Islam Centre for Higher Education, Kumaracoil, Kanyakumari(dist), TamilNadu, India. drjeyakumarmk@gmail.com

significance of essay qualities in the overall essay score, the majority of research on the topic of AEG is directed at assessing the essay holistically.

We are developing a method that may be quick, accurate, and time-efficient when giving feedback and grading performance. In comparison to the ratings provided by two human raters, the scores generated by our system would be far more detail-oriented [4]. Additionally, computerised scoring has various flaws. Hamp-Lyons' methodology made clear the absence of interpersonal communication as well as the manner in which the essayist rated the contributions. Similarly, Page argued that due to their programmed nature and absence of human emotions, computers are unable to assess essays in the manner human raters do. As a result, they cannot understand the context. Construct objections are still another point of contention [5]. That instance, a machine can prioritise minor characteristics when grading something or giving a user a score, emphasising conventional aspects rather than unconventional ones.

One of the significant research projects [6] conducted an evaluation of an AES (Automated Essay Scoring) system. Various techniques were combined in this system, including the histogram intersection string kernel, v-Support Vector Regression (v-SVR), and word embeddings using a bag of super-word embeddings. To measure the similarity between two strings, the number of shared character n-grams, as calculated by the string kernels, was utilized. The AES models were trained on the ASAP essay datasets and tested with and without transfer learning across essay datasets [7]. Transfer learning is especially beneficial when there is limited labelled data, as it allows the system to leverage knowledge from one task and apply it to a related task, resulting in improved performance. As a result, the model used in the second task is built using the knowledge from the first.

A second study of great relevance delved into the application of transfer learning to minimize the requirement for extensive prompt-specific training datasets, utilizing ASAP datasets [8]. The proposed AES model encompassed two components: one for predicting essay rank and another for predicting the overall score. Following the training of the AES model on distinct versions of the two articles, a variance vector was produced. As a result, the program was able to make a prediction regarding which of the two essays would exhibit superior quality. Using the ranking data, a straightforward linear regression was then used to construct the holistic scores. The method increased the performance of the suggested technique and decreased the amount of data that AES systems needed, which made it more competitive.

This paper outlines the creation of an automated system for scoring essay answers. The document is divided into multiple sections. Section 2 comprises a literature review, Section 3 elaborates on the methodology, Section 4 showcases the implementation results, and lastly, Section 5 draws the paper to a conclusion.

Literature Review

In this section, a comprehensive exploration of AES systems is provided, delving into the latest developments in automated essay scoring. The performance of these systems has been extensively assessed and documented through the measurement of average agreement levels and quadratic weighted kappa in the majority of published research. These metrics are utilized to quantify and clarify the extent of agreement between automated graders and human graders.

Ramachandran, et al [9] offered a novel strategy that makes use of word-order graphs to spot key trends in top-scoring student responses and human-provided rubric texts. The method makes use of semantic metrics to identify clusters of similar words that can serve as substitute responses. The testbeds used for their methodology include the Kaggle Short Answer dataset (ASAP-SAS, 2012) and a short answer dataset given by Mohler et al. [10]. Their approach entails grouping words or phrases that a human assessor would anticipate finding in top responses into meaningful categories. These semantic groups are then merged

to create patterns that help identify the crucial ideas or vocabulary that define strong student responses.

English essays are assessed automatically through the utilization of diverse machine learning methods like Latent Semantic Analysis (LSA), Generalized LSA, Bilingual Evaluation Understudy, and Maximum Entropy. Enhancing the effectiveness of these techniques is possible by incorporating an ontology, which represents a concept map of the domain's knowledge. Using ontologies in the review process makes it more comprehensive because it enables the examination of keywords, synonyms, the right language usage, and the extent of concept coverage. The aforementioned strategies are applied in the study of Devi, et al [11] both with and without the use of an ontology, and they are tested using a common set of input data made up of computer science-related technical responses. The design and development of the computer graphics domain ontology.

Feng, et al [12] created a network with MTA-LSTMs or multi-topic aware short-term memory. They keep an innovative multi-topic coverage vector in this model, which updates sequentially while decoding and learning the weight of each topic. To direct the generator, this vector is then given to an attention model. Aside from that, they also automatically create 55,000 question-and-answer pairs, 305,000 essay paragraphs, and two paragraph-level Chinese essay corpora.

Pribadi, et al [13] directed their attention towards developing an automatic scoring system for short answers. While some automated scoring techniques used for lengthy answers showed promising results in grading student responses, the information retrieval approach is commonly employed in automatic long answer systems to compare students' answers with reference texts. However, automatic short answer scoring still faces challenges due to the brevity of each response, consisting of only a few words. One to three sentences make up each response. The evaluation of a brief description with a small word count requires unique attention, particularly during the weighing procedure. Due to the weighting procedure's constraints and the word's extremely low frequency, using the frequency model is not an option. This study explores various approaches that utilize overlapping methods to evaluate the similarity between answers given by references and students. The findings suggest that the Cosine Coefficient method surpasses both the Dice and Jaccard Coefficient methods in effectively measuring the degree of resemblance.

Kumar, et al [14] suggested and elucidated the development of AutoSAS, a system for SAS. AutoSAS can be trained to accurately evaluate a prompt if it is given a question and the corresponding graded samples. This study encompasses essential components necessary for constructing our proposed model, encompassing lexical diversity, Word2Vec, prompt, and content overlap. Additionally, it presents a novel approach to recognize the factors that impact the efficiency of answer scoring. To evaluate the model's performance, we employ the readily accessible public dataset known as the Automated Student Assessment Prize Short Answer Scoring (ASAP-SAS) dataset.

For enhancing the effectiveness of automatic short answer scoring, multiple data augmentation strategies, or MDA-ASAS, were suggested by Lun, et al [15]. Back-translation, using the right response as a reference answer, and content swapping are just a few of the data augmentation procedures that are included in the MDA-ASAS language representation learning framework. We argue that external knowledge significantly affects the ASAS process. On the other hand, studies have shown that the Bidirectional Encoder Representations from Transformers (BERT) model is proficient at assimilating external knowledge through its ability to learn semantic, grammatical, and other characteristics from vast amounts of unsupervised data. This capability empowers BERT to enhance various natural language processing tasks.

Hussein, et al [16] established a methodology that improves a neural-based AES model's baseline accuracy and validity concerning evaluating and scoring attributes. In order to provide trait-specific adaptive feedback, we modify the model and provide a technique

based on essay trait prediction. In the automatic essay scoring project, we extensively explored various deep-learning models and conducted multiple studies to extract specific insights from these models. Based on the results, the LSTM-based system outperformed the control group in Quadratic Weighted Kappa (QWK) by 4.6%. Additionally, including the prediction of trait scores significantly improved the overall accuracy of score predictions.

Ludwig et al. [17] conducted a comparative study that involved analyzing the performance of a logistic regression model using the Bag-of-Words (BOW) technique and a transformer-based approach. The study aimed to analyze the performance differences between the two models. For this investigation, they used a dataset of 2088 emails that were categorized manually as courteous or not for a problem-solving task. Interestingly, both transformer models considered in the study outperformed the logistic regression model, even without any hyperparameter adjustment for the regression-based approach.

Beseiso et al. [19] introduced an innovative transformer-based neural network model intending to improve Automated Essay Scoring (AES) performance. The model leveraged Kaggle's ASAP dataset and incorporated the powerful RoBERTa language model. The primary objective of their research was to tackle the issue of essay coherency, a crucial aspect often overlooked by conventional essay scoring methods, which include classic NLP pipelines, deep learning-based techniques, or their combination. Instead of following conventional approaches, the authors ingeniously combined a Bi-LSTM model with a pre-trained RoBERTa language model, successfully addressing the coherency problem in essays.

Proposed Methodology

The education sector is increasingly prioritizing Automated Essay Scoring (AES) to reduce the burden of manual grading and provide learners with immediate feedback. Machine learning-powered Natural Language Processing (NLP) has shown remarkable success in text classification and AES. Figure 1 illustrates the architecture of the suggested framework for automatic answer scoring, which consists of four stages: data exploration, feature extraction, model training, and model evaluation. The subsequent sections delve into a detailed discussion of each of these processes.

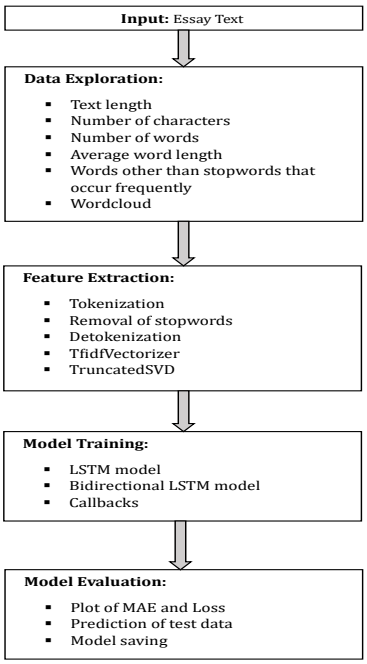

Figure 1: Architecture of the proposed system

Data Exploration

The input data for the proposed automatic essay answer scoring system is the detailed essay, which is entered into the data exploration phase. Data exploration involves investigating data, generating novel ideas, testing hypotheses, verifying assumptions, and identifying concealed patterns. During the data exploration phase, the following activities are undertaken:

- Text length
- Number of characters
- Number of words
- Average word length
- Words other than stopwords that occur frequently
- Wordcloud

The input data is an essay that needs to be evaluated or scored, in which the text length is measured initially. Consequently, the essay scoring system incorporates various criteria customized for writing, encompassing elements such as text structure (e.g., text length, sentence length, and paragraph length), cohesion (covering local, global, and situational cohesion), lexical sophistication (encompassing word frequency, age of acquisition, word hypernymy, and word meaningfulness), keyword usage, part of speech tags (adjectives, adverbs, cardinal numbers), syntactic complexity, and rhetorical attributes.

The input essay is evaluated based on the number of characters and words it contains. The mean clausal length focuses on sub-clausal complexity within phrases, whereas the average sentence length assesses the potential multi-clausal complexity achieved through subordination, coordination, and modification, among other forms. The average word length is measured which comprises the number of characters.

Stopwords are commonly occurring words in various natural language texts or sections of texts that contribute little meaningful information to the overall context they appear in. Therefore, words other than stopwords that occur frequently are taken into account. Then the wordcloud has been generated with the resulting words of the essay from the result we have achieved from the previous operation. Wordclouds are an excellent way to visualise text data. Each word's size and colour in the wordcloud represent its frequency or significance. In essence, a word-cloud is a visual representation that shows the frequency of terms in a corpus of text. The font size of a word is inversely correlated with its frequency in the corpus. As a result, word clouds are an excellent analytical and visualisation tool for perusing text corpora. This results in the data exploration phase of the automatic essay scoring system.

Feature Extraction

Once the data exploration phase of the operation is completed, then the features are extracted from the resultant of data exploration through the following steps:

- Tokenization
- Removal of stopwords
- Detokenization
- TfidfVectorizer
- TruncatedSVD

Tokenization is the process of dividing a written essay into tokens, which are very little pieces of text. Words, word fragments, or simple characters like punctuation can all be considered tokens. It is a challenging and one of the most fundamental NLP tasks. The embeddings provide meaning to tokens by converting a solitary integer into a high-dimensional vector. These embedding vectors can be calibrated through either unsupervised training tasks or token-to-token cooccurrence data using a neural network. Alternatively, a supervised approach involves comparing each token's meaning to the results of a particular

NLP task in order to fine-tune the embedding vectors. Tokenization is essential to future exploration. It is challenging to extract higher-level information from a document without first identifying the tokens. In the situation being studied, the white space was used as a delimiter in this operation.

To reduce the extensive word count that may arise in a sentence, it is common practice to eliminate stop words or apply stemming techniques. Stemming focuses solely on the word's root, disregarding its conjugations, declensions, or plurals. Stop words, the most common but meaningless words in any language, are usually disregarded by NLP tasks. These words are removed from natural language data (text) either before or after processing. Therefore, eliminating stopwords might improve unstructured text's signal-to-noise ratio and boost the statistical significance of phrases that could be crucial for a certain task. During this stage, common terms like "a," "the," "is," "was," "got," and "have" that don't contribute to the response's major theme were removed. As a result, the identified sentences are processed once again to separate the recognised stopwords. The remaining words are employed as keywords to discriminate between good and bad essays.

These irreversible operations necessitate that detokenization, a procedure to recover the original raw input from the tokenized sequence, be language-dependent. When detokenizing most European languages, for instance, the primitive tokens are typically separated by whitespace. When a soft token is fed into the input of a detokenizer or the input of a network that predicts the next token, the input embedding is calculated using a weighted average of the embeddings from a codebook based on the probability values. This shows that the soft token embedding has covered an interpolable continuous space, which may more accurately represent the visual output, particularly when it is continuous. Due to the continuous nature of the soft token, it is also possible to introduce an auxiliary loss that learns the task output from beginning to end, from the output of the detokenizer to the input of the task-solver. Thus, this stage outputs the decoding of the tokens and is given into the vectorizer.

Vectorization is a method used to compute text similarities. In the field of text mining and information retrieval, Term Frequency-Inverse Document Frequency (TF-IDF) is a numeric metric employed to extract features. It gauges the importance of a word within a corpus or a set of documents. This approach is frequently employed to assign weights to words during text summarization and categorization, thereby preventing word filtering. TF-IDF values are typically inversely related to a word's frequency in a document but are adjusted by the term's frequency in the entire corpus, ensuring that more common words don't dominate the analysis. The term "frequency term" denotes the actual frequency of a term in a document. To generate low-dimension word embedding vectors, the TF-IDF output is further processed using truncated Singular Value Decomposition (SVD). Truncated SVD selects the largest singular values, effectively reducing the dimensionality for scoring purposes.

Model Training

In this work, we train the model with LSTM and then with bidirectional LSTM to generate the scoring for the input essay and further, the callback function has been employed. As mentioned earlier, the initial implementation of Long Short-Term Memory (LSTM) involved representing its structural diagram in Figure 2. The LSTM operates on a feature-extracted sequence denoted as x=(x_1, x_2, … x_n), where 'n' represents the length of the input sequence. The core structure of the LSTM consists of three control gates that regulate the activation vector c of a memory cell. The first gate is known as the forget gate, responsible for determining how much of the previous cell state (c_(t-1)) should be retained for the current cell state (c_t). The second gate is the input gate, which decides how much of the input (x_t) should be integrated into the current cell state (c_t). Lastly, the third gate is the output gate, which governs the extent to which the current cell state (c_t) influences

the production of the current output value (h_t) of the LSTM network. A completely linked layer of three gates has a vector as its input and a real number in [0,1] as its output.

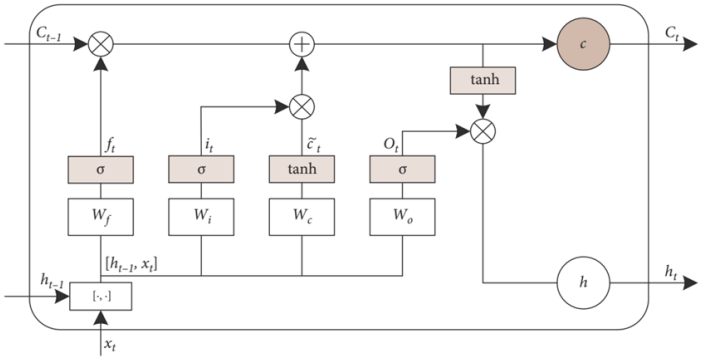

Figure 2: Structure of the long short-term memory

Input gates: $i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$

Forget gates: $f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$

Output gates: $o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$

Cell states: $f_t * h_{t-1} + i_t * \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c)$

Cell outputs: $h_t = o_t * \tanh(c_t)$

The architecture employed in this study is illustrated in Figure 3, utilizing a single-layer LSTM with a logistic sigmoid function (σ) along with word vectors (x_t), hidden states (h_t), weight matrix terms represented by (W), and bias vectors for the three gates denoted by (b). Additionally, a single dropout and dense layer are incorporated into the model.

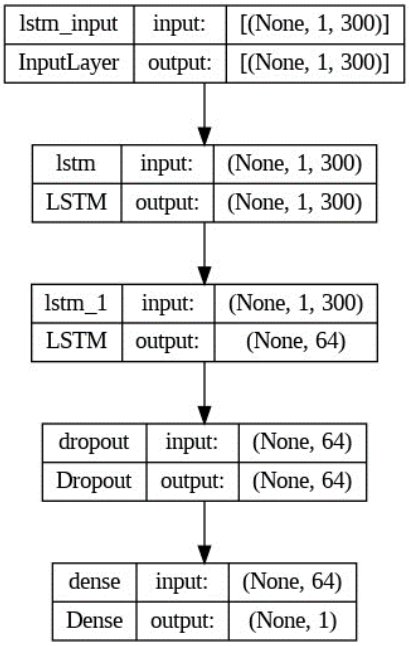

Figure 3: Architecture of LSTM structure

In addition, the bidirectional LSTM (Bi-LSTM) has been implemented for the essay scoring system. In this instance, current information is related to future information and is dependent on past information. Only past data was processed by the unidirectional LSTM, which sometimes loses the true meaning of a sentence. To integrate two distinct hidden LSTM layers with opposing orientations into one output, the bi-LSTM was developed. The output layer can use related data from both the prior and subsequent context ought to this structure. Bidirectional networks have shown notable superiority over unidirectional ones. Their advantage lies in processing sequences both forward and backwards, allowing them

to capture information from the past and future by feeding words from left to right and right to left, respectively. The forward hidden sequence $\overrightarrow{h_t} = (\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots \overrightarrow{h_n})$ of a bidirectional LSTM considers the input in ascending order and the backward hidden layer $\overleftarrow{h_t} = (\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots \overleftarrow{h_n})$, which considers the input in descending order for the input sequence $x = (x_1, x_2, \ldots x_n)$. Following is an illustration of how the two network directions behave independently up until the final layer, where their outputs are concatenated as $y_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$:

$$\overrightarrow{h_t} = \sigma\left(W_{\vec{h}x}x_t + W_{\vec{h}\vec{h}}\overrightarrow{h}_{t-1} + b_{\vec{h}}\right)$$
$$\overleftarrow{h_t} = \sigma\left(W_{\overleftarrow{h}x}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right)$$
$$y_t = W_{y\vec{h}}\overrightarrow{h_t} + W_{y\overleftarrow{h}}\overleftarrow{h_t} + b_y$$

The hidden layer's output sequence is represented as y_t=(y_1, y_2, ..., y_n). The present setup employs a Bi-LSTM model consisting of 19 layers, three dropout layers, and seven dense layers. Figure 4 visually illustrates the architecture of this Bi-LSTM model. During the iteration process, the system utilizes a callback function.

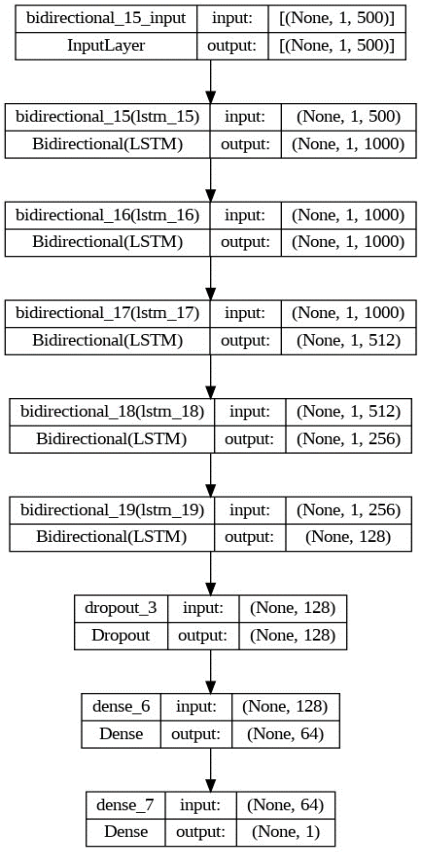

Figure 4: Architecture of the Bidirectional LSTM

During the training process, individuals commonly assess the student model using a development set at various checkpoints. At the same time, they record the loss curve to keep track of the student model's performance. The callback function receives the student model and the current training step as its two arguments. Adding callbacks for each function result is an established programming change known as the continuation-passing style. The distiller stores the student model and then calls the callback function at each checkpoint step, which is specified by the num_train_epochs and the ckpt_frequency.

Result and Discussion

The experiments were conducted using the ASAP-SAS dataset, a publicly available dataset from a past Kaggle competition focused on autonomous scoring of short answers. To gather

this dataset, graded tests for students in Grade 10 were scanned together with the corresponding answers, and the scanned answers were then converted to text using OCR software.

The proposed automatic answer scoring system has been implemented with the essay input obtained from the ASAP-SAS dataset. The system started with the data exploration phase. The sample input essay has been presented in Figure 5.

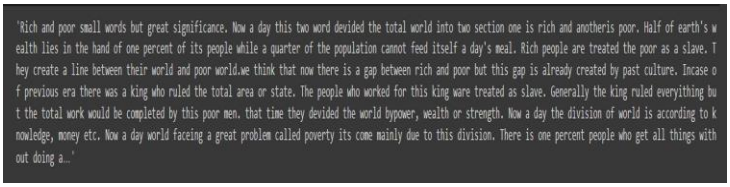

Figure 5: Sample input essay

The data exploration process commences by examining the length of the text, and subsequently, the average word count in each sentence is calculated. This information is then depicted graphically in Figure 6. Consequently, the average word length in each sentence has been measured and plotted in Figure 7. Subsequently, the number of characters in each sentence has been graphically plotted in Figure 8.

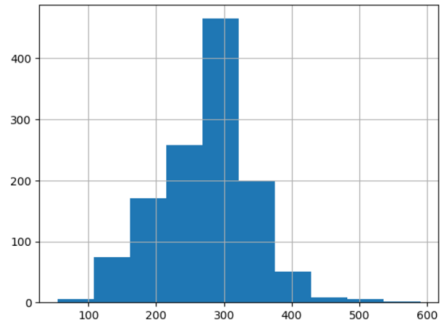

Figure 6: Average number of words in the sentence

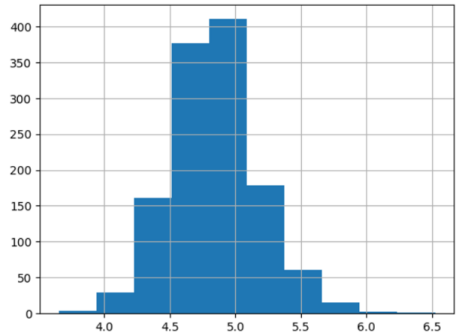

Figure 7: Average word length in each sentence

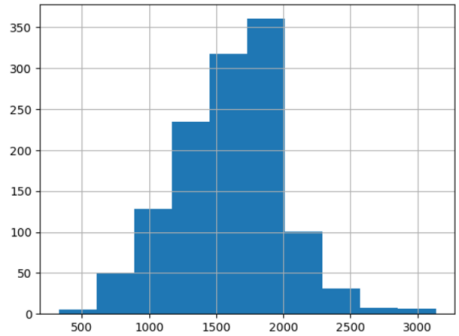

Figure 8: Number of characters present in each sentence

A word cloud is a visualization technique used to represent text data, where words are displayed according to their occurrence or significance within the text. It provides an excellent means to analyze text data by presenting tags or words, with their significance indicated by their frequency of occurrence. In such a way, the generated word cloud for the given input essay has been presented in Figure 9. Furthermore, after the tokenization problem with the removal of stop words, the frequently used words have been represented in Figure 10.

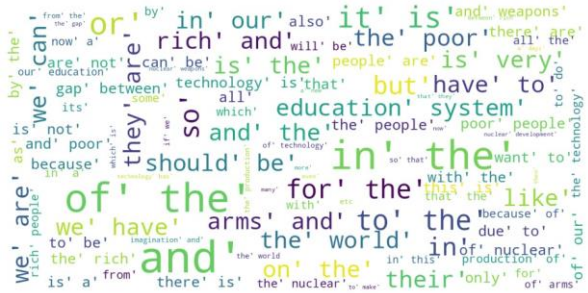

Figure 9: Word cloud

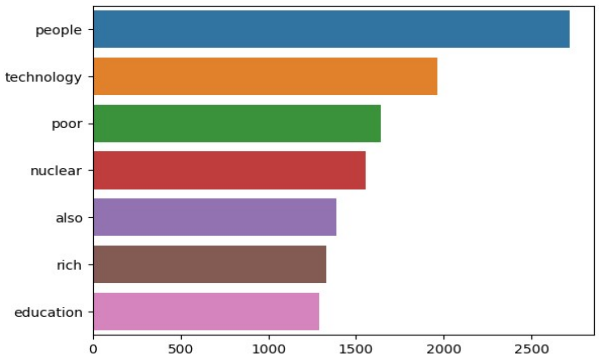

Figure 10: Statistics of frequently used words

Once the data exploration and feature extraction phases had been completed the model training has been done in two ways. Initially, the LSTM model was trained with the parameters mentioned in Figure 11 having 814705 trainable parameters. The model was executed iteratively 150 times. The resulting Mean Absolute Error (MAE) and loss values are graphically plotted in Figure 12 and Figure 13 respectively. However, it results in more fluctuations in the waveform which represents the poor result of scoring.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 300)	721200
lstm_1 (LSTM)	(None, 64)	93440
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65

Total params: 814,705
Trainable params: 814,705
Non-trainable params: 0

Figure 11: Parameters of the LSTM model

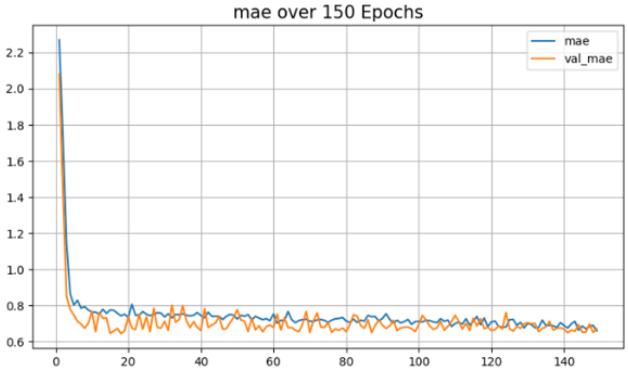

Figure 12: Mean absolute error of the LSTM system

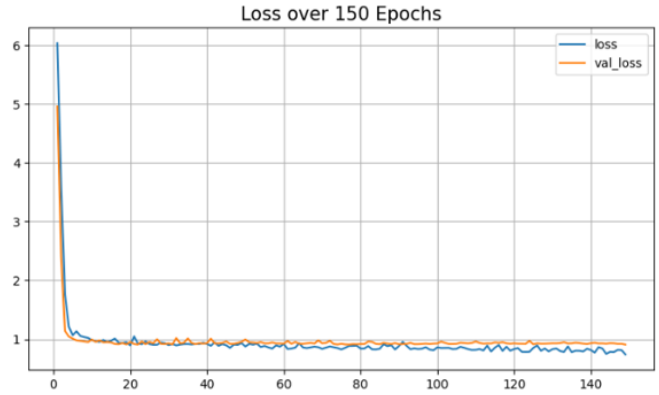

Figure 13: Loss value of LSTM system

Since the LSTM model results in poor MAE and loss values, the scoring system has been implemented with the proposed bidirectional LSTM model having 19 layers. The training parameters of the Bi-LSTM model with 13411393 parameters have been presented in Figure 14. The predicted score for the input essay through the Bi-LSTM model has been presented in Figure 15. In addition, the evaluator rating for the input test essay through the proposed system is presented in Figure 16.

```
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional_35 (Bidirecti  (None, 1, 1000)          4004000
 onal)

 bidirectional_36 (Bidirecti  (None, 1, 1000)          6004000
 onal)

 bidirectional_37 (Bidirecti  (None, 1, 512)           2574336
 onal)

 bidirectional_38 (Bidirecti  (None, 1, 256)           656384
 onal)

 bidirectional_39 (Bidirecti  (None, 128)              164352
 onal)

 dropout_7 (Dropout)         (None, 128)               0

 dense_14 (Dense)            (None, 64)                8256

 dense_15 (Dense)            (None, 1)                 65

=================================================================
Total params: 13,411,393
Trainable params: 13,411,393
Non-trainable params: 0
_________________________________________________________________
```

Figure 14: Parameters of Bidirectional LSTM model

	Unnamed: 0	promptId	uniqueId	essay	predicted_score
0	0	1	1_315	Curriculum has been adopted in many schools. T...	3.0
1	1	1	1_214	I strongly agree with the statement , The tig...	3.0
2	2	1	1_196	Imagination and creativity is the most importa...	3.0
3	3	1	1_178	In our eduction system leaves no room for imag...	2.0
4	4	1	1_201	I will agree at some what extend, because if w...	3.0
...	...	...	...	...	...
300	300	5	5_146	Earth is a creation of God and everything that...	3.0
301	301	5	5_65	production of arms and weapons in this present...	3.0
302	302	5	5_151	Race to become more powerful can destroy the e...	3.0
303	303	5	5_404	In its attempt to harness the power of the ato...	3.0
304	304	5	5_360	Racein the production of arms and weapons in t...	2.0

305 rows × 5 columns

Figure 15: Predicted score for the test essay

	Unnamed: 0	promptId	uniqueId	essay	evaluator_rating
0	0	1	1_323	At present age, our education system is not go...	3.0
1	1	1	1_238	I am agree the tightly defined curriculum of o...	4.0
2	2	1	1_212	I strongly agree with the statement that tight...	2.0
3	3	1	1_117	Our education system is nice quitely but i dis...	2.0
4	4	1	1_229	i am totally agree with the statement that tig...	3.0
5	5	1	1_226	I am totally disappointed(Not Agree) with our ...	2.0
6	6	1	1_99	the education system is very rough it does not...	3.0
7	7	1	1_240	I agreed with the tightly defined curriculum o...	2.5
8	8	1	1_127	Now a days, education system is very important...	2.0
9	9	1	1_176	In the current scenario,education plays a vita...	4.0

Figure 16: Evaluator rating

The mean absolute error and loss values are estimated through 26 epochs and 150 epochs on the Bi-LSTM model. The MAE and loss values for 26 epochs has been illustrated in figure 17 and 18 respectively. And this results in the best metric values such as a mean absolute value of 0.6708 and validation loss of 0.3503. Similarly, the MAE and validation loss values through 150 epochs have been plotted in Figures 19 and 20 respectively.

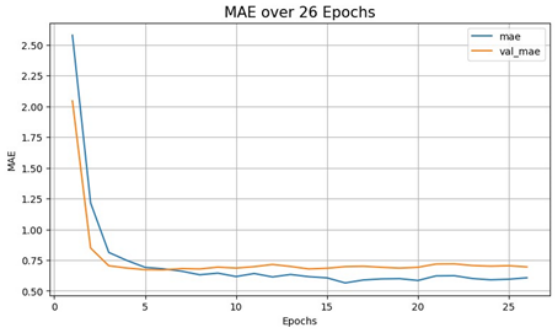

Figure 17: MAE over 26 epochs on the Bi-LSTM model

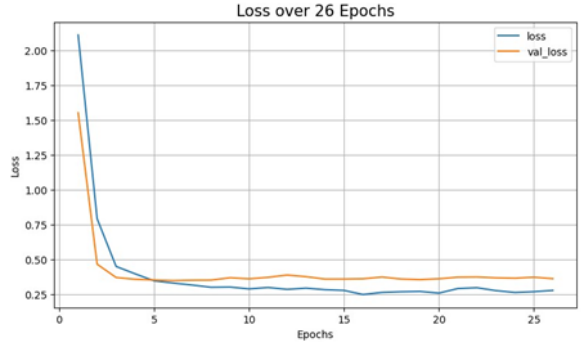

Figure 18: Loss over 26 epochs on the Bi-LSTM model

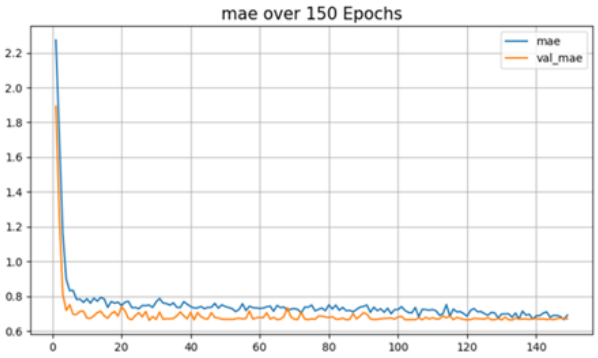

Figure 19: MAE over 150 epochs on the Bi-LSTM model

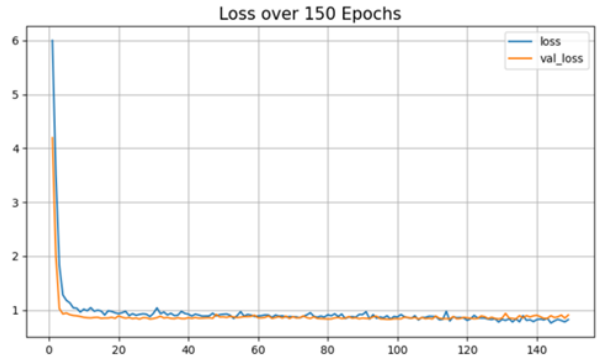

Figure 20: Loss over 150 epochs on the Bi-LSTM model

Conclusion

This research explores the viability of employing automated scoring methods for assessing the calibre of student essays. The answer-scoring system has been initially implemented through the LSTM model. Due to the limitation in LSTM for optimal scoring, the Bi-LSTM model has been developed with callbacks to obtain the student answer scoring system. The system has been implemented in the ASAP short answer scoring dataset. The Bi-LSTM model results in better AME and validation loss at 0.6708 and 0.3503 respectively. In such a way, it shows overwhelmed performance on answer scoring for essays.

References

Valenti, Salvatore, Francesca Neri, and Alessandro Cucchiarelli. "An overview of current research on automated essay grading." *Journal of Information Technology Education: Research* 2, no. 1 (2003): 319-330.

Chen, Hongbo, and Ben He. "Automated essay scoring by maximizing human-machine agreement." In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1741-1752. 2013.

Persing, Isaac, and Vincent Ng. "Modeling prompt adherence in student essays." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1534-1543. 2014.

Gierl, Mark J., Syed Latifi, Hollis Lai, André-Philippe Boulais, and André De Champlain. "Automated essay scoring and the future of educational assessment in medical education." *Medical education* 48, no. 10 (2014): 950-962.

Ramalingam, V. V., A. Pandian, Prateek Chetry, and Himanshu Nigam. "Automated essay grading using a machine learning algorithm." In *Journal of Physics: Conference Series*, vol. 1000, p. 012030. IOP Publishing, 2018.

Cozma, Mădălina, Andrei M. Butnaru, and Radu Tudor Ionescu. "Automated essay scoring with string kernels and word embeddings." *arXiv preprint arXiv:1804.07954* (2018).

Cozma, Mădălina, Andrei M. Butnaru, and Radu Tudor Ionescu. "Automated essay scoring with string kernels and word embeddings." *arXiv e-prints* (2018): arXiv-1804.

Cummins, Ronan, Meng Zhang, and Ted Briscoe. "Constrained multi-task learning for automated essay scoring." Association for Computational Linguistics, 2016.

Ramachandran, Lakshmi, Jian Cheng, and Peter Foltz. "Identifying patterns for short answer scoring using graph-based lexico-semantic text matching." In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 97-106. 2015.

Mohler, Michael, Razvan Bunescu, and Rada Mihalcea. "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments." In *Proceedings of the 49th annual meeting of the Association for computational linguistics: Human language technologies*, pp. 752-762. 2011.

Devi, M. Syamala, and Himani Mittal. "Machine learning techniques with ontology for subjective answer evaluation." *arXiv preprint arXiv:1605.02442* (2016).

Feng, Xiaocheng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. "Topic-to-essay generation with neural networks." In *IJCAI*, pp. 4078-4084. 2018.

Pribadi, Feddy Setio, Teguh Bharata Adji, Adhistya Erna Permanasari, Anggraini Mulwinda, and Aryo Baskoro Utomo. "Automatic short answer scoring using words overlapping methods." In *AIP Conference Proceedings*, vol. 1818, no. 1. AIP Publishing, 2017.

Kumar, Yaman, Swati Aggarwal, Debanjan Mahata, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. "Get it scored using autosas—an automated system for scoring short answers." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, pp. 9662-9669. 2019.

Lun, Jiaqi, Jia Zhu, Yong Tang, and Min Yang. "Multiple data augmentation strategies for improving performance on automatic short answer scoring." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 09, pp. 13389-13396. 2020.

Hussein, Mohamed A., Hesham A. Hassan, and Mohammad Nassef. "A trait-based deep learning automated essay scoring system with adaptive feedback." *International Journal of Advanced Computer Science and Applications* 11, no. 5 (2020).

Ludwig, Sabrina, Christian Mayer, Christopher Hansen, Kerstin Eilers, and Steffen Brandt. "Automated essay scoring using transformer models." *Psych* 3, no. 4 (2021): 897-915.

Ahmed, Abbirah, Arash Joorabchi, and Martin J. Hayes. "On Deep Learning Approaches to Automated Assessment: Strategies for Short Answer Grading." *CSEDU (2)* (2022): 85-94.

Beseiso, Majdi, Omar A. Alzubi, and Hasan Rashaideh. "A novel automated essay scoring approach for reliable higher educational assessments." *Journal of Computing in Higher Education* 33 (2021): 727-746.