Encoder Decoder Approach To Automated Essay Scoring For Deeper Semantic Analysis

Priyatam Naravajhula[1], **Sreedeep Rayavarapu**[1], and **Srujana Inturi**[1]

[1]CBIT, Hyderabad-500078, India
{priyatamnaravajhula,sreedeep}@gmail.com, isrujana_cse@cbit.ac.in

Abstract

Descriptive answers have always played a major role in education of children. They are representative of student's grasp on knowledge and presentation skills. Manual evaluation of essay answers is a arduous process to human evaluators owing to limited numbers of evaluators and an out of proportional number of essays to be graded hence leading to an inefficient or an inaccurate score. It can be concluded that due to the major shift in paradigm of learning from traditional classroom education to online education engendered by COVID-19 pandemic that future assessment of education shall be online, making the solution of automatic essay scorer not only relevant, but of paramount importance. We explore several neural architecture models for the task of automated essay scoring system. Results and Experimental analysis exhibit that our model based on recurrent encoder-decoder provides for a deeper semantic analysis hence, outperforming a strong baseline in terms of quadratic weighted kappa score.

1 Introduction

The exponential advancement of deep learning in the past decade has seen its applications in a wide range of fields from molecular biology to quantum physics. This flexible nature of deep learning and neural architectures is the reason why we have seen its application to a wide array of issues in natural language processing. Automated essay scoring is one such problem which aims to find a relation between the essay written and the score assigned so that given an unseen essay, we can predict the score as accurately as possible. Essay writing forms important aspect in the academic assessment of the student, grading these essays is a laborious task therefore most of the educational organizations like Educational Testing Service (ETS) employ automated essay scorers to evaluate essays. The major pitfalls of these systems stem for the reason that they use hand crafted features to score the essay. The continuous space representations and non-linearity of neural network have provided a great potential in natural language processing. BERT and GPT-3, neural architectures developed by Google and OpenAI respectively achieve state of the art performance in NLP tasks such as word prediction, question-answering and neural machine translation.

Researchers have applied convolutional neural networks(CNNs), recurrent neural networks (RNNs), attention mechanisms (16) and a permutation of ensembles to the task of automated essay scoring. In this paper we present our encoder-decoder model that learns the relation between the essay and the score assigned by performing a deeper semantic analysis than the current existing models. By applying self-attention and non-linear layers at both encoder level and decoder level, our model is able to effectively capture the information at word level and sentence level respectively, required for scoring. We show that our model performs significantly better than our baseline neural net and finds patterns between words for a better semantic analysis.

The rest of the paper is divided into section 2 which deals with related work, section 3 which gives an idea about the task of automated essay scoring .In section 4 we present our model and all its intricacies. Section 5 gives an idea of training , section 6 deals with our experimental setup and lastly we present our results and discussion in section 7, followed by conclusion and references

2 Related Work

Some of the earliest systems of AES were dependent on handcrafted features and feature engineering. Page(1986) developed an AES tool called Project Essay Grade(PEG) by using only linguistic

Proceedings of the 18th International Conference on Natural Language Processing, pages 399–407
Silchar, India. December 16 - 19, 2021.

surface features.A well-known early example of automated essay scorer is E-Rater (Jill Burestein) (7) that employed more traditional techniques of natural language processing. The same project was released under version 2 in the year 2004 which utilizes a new set of features to represent characteristics related to organization and development, lexical complexity ,etc. All these methods shared a common regression equations for essay assessment, therefor share a common limitation of being dependent on feature engineering.

The introduction of neural networks eliminated the need for handcrafted features .Alikaniotis et al. (2016)(1) and, Taghipour and Ng, (2016)(12) presented scoring models based on LSTM. These formed some of the early examples of application of deep learning in automated essay scoring process. Particularly Taghipour and Ng, (2016)(12) presented a method to extract word level semantics by applying 1D convolution over vectors. The major limitation of the paper being usage of one-hot representations that do not extract relations as effectively as word embedding does. The usage of single layer LSTM also does not provide effective semantic relation analysis. Interestingly, Dong and Zhang,(2016)(13) presented a model involving two CNN's. In the recent years, we have seen fascinating neural architectures applied to automated essay scoring systems. Zhang and Litman,(2018)(9) proposed a novel co-attention based model that deals with source article for scoring the essay,with major limitation of not being scalable to all type of essays. Jiawei Liu et al., (2019)(14) presented a two-stage learning approach leveraging both handcrafted features and neural networks to calculate three different scores and giving a final score based on those. Siamese Neural architecture was introduced by Liang G et al,(2018)(8) where Bidirectional LSTM was used in a Siamese fashion to predict scores. In this paper, we aim to provide an end-to-end system that predicts a holistic score of the essay while ensuring that the network captures the semantic relations. Excited by the performance of encoder-decoder models in applications of NLP such as machine translation, we adopted this neural architecture for automated essay scoring system

3 Model

Our model is inspired by the neural architecture presented by Dong et al.,(2017)(3). The model presented by Dong et al; is divided into three sections:Intially, A convolution layer and attention was used to capture sentence representations. thereafter, LSTM with attention pooling for document representation was utilised. At the end, sigmoid layer was utilised for mark prediction. We have introduced a decoder layer into the network architecture, influenced by the performance of recurrent encoder-decoder layers presented by Robert Susik, (2020)(11).By doing so, our model extracts meaningful semantic relationship between sentences in the essay written by the student. Our model consists of ten layers with four layers forming the encoder architecture and the remaining six forming decoder architecture. Figure 1 depicts the architecture of our network.

3.1 Encoder Architecture

Encoder architecture consists of 4 layers:word embedding layer, convolutional layer, word level attention and an encoder LSTM layer.

3.1.1 Word Embedding Layer

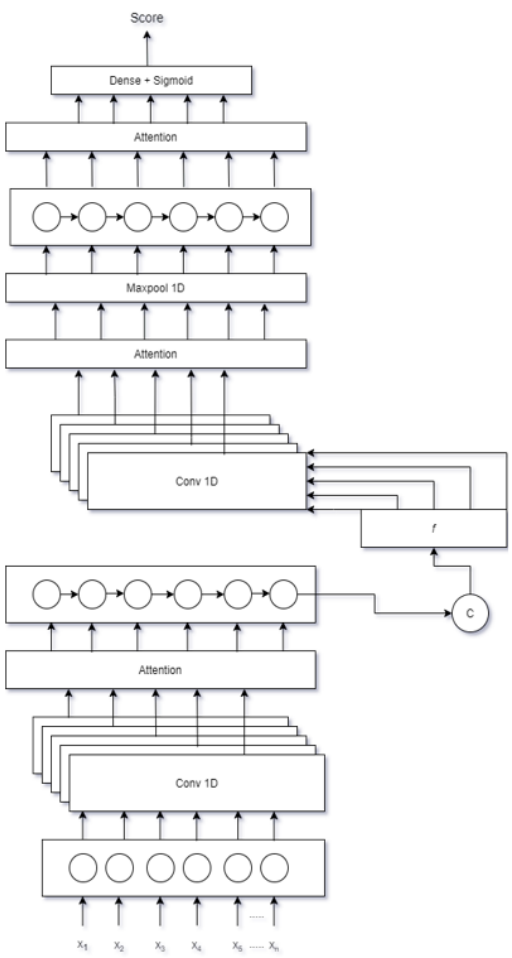
Word embeddings are used to map a word to a specific dimensional vector. We have used Glove embeddings (Pennington et al., 2014)(6) to obtain word embeddings.This particular embeddings were developed by training on six billion words from two sources It has around four hundred thousand uncased vocabulary items. The embeddings in the proposed model is restricted to fifty dimensions .The output of this layer is a matrix of dimension $L_E = R^{S \times W \times d_L}$,where S, W, dL are the number of sentences of the essay, length of the essay and embedding size .A dropout layer is applied after the embedding layer to control overfit.

3.1.2 Encoder Convolutional Layer

A 1-D convolution is performed in this layer over word representations to fetch isolated represntations in each sentence. For each word w_i in sentence, we perform a convolution:

$$k_i = a([w_i : w_{(i+l-1)}].fil_c + bs_c) \qquad (1)$$

where a is a non-linear activation function, l is the kernel size, fil_c is the filter matrix and bs_c is the bias vector. The outputs for this layer are $C_E = R^{S \times f_e \times n_C}$, where S, f_e, n_C are count of sentences in the essay, filtered lengths of sentences of the essay and number of filters used in convolutional layer .



Decoder	Score Predicted
	Output Dense Layer
	Final self-Attention pooling
	Decoder LSTM layer
	1D Max-pooling
	Sentence level attention pooling
	Decoder 1D Convolution
Encoder	Encoder LSTM layer
	Word Level Attention pooling
	Encoder 1D convolution
	Embedding Layer
	Word Inputs

Figure 1: Neural Architecture

3.1.3 Word Level Self- Attention Pooling Layer

Attention is applied following the convolutional layer , as presented in Dong et al,(2017) (3) to capture sentence representations. The attention mechanism is defined by following equations

$$mk_i = \tanh(W_m x_i + bs_m) \tag{2}$$

$$vk_i = \frac{e^{W_v mk_i}}{\sum e^{W_v mk_j}} \tag{3}$$

$$D = \sum vk_i a_i \tag{4}$$

Where W_m, w_v, bs_m are weight matrix, weight vector , bias vector respectively. mk_i, vk_i are attention weight and attention vector for a_i. The outputs for this layer are $A_E = R^{D \times n_C}$

3.1.4 Encoder Sentence Level LSTM

This layer receives the input from provious attention layer and forms the basis of first context extraction. LSTM is a modified version of recurrent units that overcome the problem of vanishing gradients effectively. (Hochreiter and Schmidhbur, 1997)(5). The power of LSTM comes from the fact that it can control the flow of information for a better sentence representation by leveraging three gates that are used to preserve or forget the information required for capturing the context of sentence representation. The output of this layer is interpreted in a manner where contextual information C is interpreted as sequence C',

$$C = \sum_{i=0}^{p_c - 1} c_i \tag{5}$$

$$C' = \sum_{i=0}^{p_c/\alpha - 1} \sum_{j=0}^{\alpha - 1} c_{i\alpha + j} \tag{6}$$

Where $\alpha = p_c / n_x, p_c, n_x$being size of context(size of output of this layer) and size of input to this layer. The output of dimension $C = R^{S \times n_H}$, where n_H is the number of hidden states

3.2 Decoder Architecture

Decoder architecture consists of 1D convolutional layer, decoder LSTM layer, a self-attention layer

and an output linear sigmoid layer.

3.2.1 Decoder Convolutional Layer

A convolutional layer is added right before the decoder to extract meaningful representations from the context C' and to also restrict the number of output channels and are is derived as follows:

$$C" = \sum_{p} \sum_{l} {C'}_{p,l} W_{p,l} \tag{7}$$

Where p,l are the number of kernels and length of kernel size.

3.2.2 Sentence level Attention Layer

Self-attention layer as described in section 4.1.3 is applied over 1D convolution layer. The output x of this layer is proved as input max-pool layer.

3.2.3 Decoder LSTM layer

The final context extracted from previous convolutional layer is given to this layer for capturing the semantic relations by using input output and forget gates.This layer serves as a modeling layer to construct the final sentence representation

3.2.4 Decoder Self-Attention

After obtaining the intermediate states of LSTM, a final layer of attention pooling is applied to learn the final text representation. The equations presented in 4.1.3 are also applicable here. The output O is the final text representation.

3.2.5 Linear Layer

After obtaining the final sentence representations O, a linear layer with sigmoid activation is used to predict the final output.

$$y = sigmoid(W_o O + b_o) \tag{8}$$

Where Wo and bo are weight and bias vectors.

4 Training

Automated Essay scoring is the process of evaluating the essays written by students for a particular prompt without any human intervention. Their performance is assessed by comparing the scores generated to the human-assigned gold standard scores. Rest of this section deals with the data utilized for training and the evaluation metric chosen for comparing the performance of AES systems

Table 1: ASAP Dataset Statistics

Prompt	Avg Length	Score
1	350	2–12
2	350	1–6
3	150	0–3
4	150	0–3
5	150	0–4
6	150	0–4
7	250	0–30
8	650	0–60

4.1 Data

The data that we used for our training is the one published by Hewlett Foundation for the 2012 competition titled 'Automated Student Assessment Prize' on Kaggle .The dataset consists of 8 prompts with three different types of essays: persuasive, source-dependent and narrative. The essays have different score ranges, being scored on average by three raters across two domains. The statistics of the dataset are given in Table 1

4.2 Evaluation Metric

The scores generated by AES systems need to be compared to ratings assigned by human-annotators. While there are many correlation metrics such as Pearson's correlation, Spearman's correlation, we have chosen Quadratic Weighted Kappa(QWK) score to be our evaluation metric. The main reason of this choice is because this metric is useful when it's necessary to evaluate the possible impact of random selection in computation of standard accuracy. (Giuseppe Bonaccorso,2017)(4) IN QWK, a weighted matrix is calculated as follows

$$W(x, y) = \frac{(x - y)^2}{(U - 1)^2} \tag{9}$$

Where x and y are the reference ratings and hypothesis rating respectively. U is the number of possible ratings. A matrix P is calculated where P(i,j) denotes number of essays that received a rating x from human annotators and rating y from AES. An expected count matrix K is constructed as cross vectors of two(reference and hypothesis) ratings. After normalization of K such that sum of elements of K and P are same, QWK is calculated as follows

$$\kappa = 1 - \frac{\sum_{x,y} W(x, y) P(x, y)}{\sum_{x,y} W(x, y) K(x, y)} \tag{10}$$

In our experiments, we compare QWK scores of our model to chosen baseline and performed paired t-test analysis to test the improvement obtained

4.3 Loss

MSE(Mean Square Error) calculates the average value of difference between gold standard scores y_i^* and prediction scores y_i. MSE is applied ubiquitously to regression tasks. Hence we have decided to adopt this loss function for our AES system. The following equation defines MSE, given N is the total number of samples.

$$MSE(y, y^*) = \frac{1}{N} \sum_{(i=1)^N} (y_i - y_i^*)^2 \qquad (11)$$

4.4 Optimization

In this paper, we adopted Adam optimizer (Ba et al., 2017) owing to its efficiency. Learning rate is set to 0.001, momentum to 0.9 for training our whole model. We have set Dropout rate to 0.5 to prevent overfitting.

5 Experiments

We have designed our experiments to test three hypotheses:

H1: The proposed model will perform equally or surpass baseline model on ASAP essay corpora in holistic score prediction.

H2:The proposed model will perform equally or surpass as the non-neural network baselines.

H3:Our model will have a better or at least equal semantic attention score as our baseline model.

Text preprocessing is done using NLTK , vocabulary size is restricted to 4000 consisting of most frequent words and all other words are treated as unknowns. The scores are scaled to range [0,1].(Taghipour and Ng, 2016)(12) For model training and the prediction assessment ,the predicted scores are converted back into original score ranges during model evaluation. We have divided the dataset into five folds to perform 5-fold cross validation and average QWK score across five folds on test set is reported.. In each fold,60% of data are used for training and the rest of data is equally divided between development testing.Table 2 gives a summary of hyperparameters used for training the models, taken from Dong et al.,2017(3) Best model was evaluated on development set after the completion of each epoch, this process was repeated for 100 epochs. The

Table 2: Hyperparameters

Hyperparameter	Value
Embedding dimension	50
CNN-kernel size	5
CNN-number of kernels	100
LSTM-Hidden units	100
GRU-Hidden units	100
Dropout Rate	0.5
Batch-size	100
Learning Rate	0.001
Momentum	0.9
Epochs	100

We have conducted the experiments in following software environment: Ubuntu, Python 3.7, Keras 2.4.0 using Tensorflow 2.4.1 backend. The baseline chosen for our paper is the model presented in (Dong et al, 2017). An attention based recurrent-convolutional network, where the word embedding are given to a convolutional layer to extract sentence representations. The extracted sentence representations are given as input to LSTM layer to extract semantic context. An attention layer is used for final representations. We have trained our model on this architecture and reported the QWK scores obtained. For non-neural baselines, we report the results of SVR and BLRR presented in Phandi et al, (2015)(10).They extract features such as length, prompt, and Bag of Words to classify using SVR and BLRR classifiers

6 Results and Discussion

Examining H1 hypothesis, results in Table 3 support this hypothesis. Encoder decoder model with architecture of LSTM+LSTM yields higher performance than our baseline.

The QWK scores obtained for encoder-decoder model have been shown to be significantly better on performing a paired t-test ($p < 0.05$). The reason of this high performance can be attributed to a finer text representations obtained by the architecture of encoder-decoder model and the usage of attention mechanisms at both word and sentence levels. It is interesting to note that the architecture GRU+GRU does not perform as well as its counterpart LSTM.

As we examine H2 hypothesis, QWK scores from Table 3 provide evidence to support this hypothesis, the proposed architecture outperforms or performs equally well across all non-neural archi-

Table 3: QWK scores for various architectures and baselines. The scores with statistical significant improvement (p¡0.05) are marked with "*".The highest scores for a prompt are marked in bold. Note: In system layers for decoder row, The operand before '+' is recurrent layer of encoder and the operand after is recurrent layer in decoder layer. The same notation is followed throughout the paper

ID	Architecture	System-layers	Prompts								
			1	2	3	4	5	6	7	8	Avg QWK
1	With Decoder +Attn	LSTM+LSTM	**0.808**	**0.648**	**0.686**	0.761	**0.811**	**0.823**	0.786	**0.702**	**0.753***
		GRU+GRU	0.784	0.620	0.612	**0.771**	0.757	0.791	**0.802**	0.675	0.722
2	Encoder(Baseline)+Attn	CNN +LSTM	0.796	0.644	0.593	0.752	0.761	0.782	0.762	0.699	0.719
3	Non-Neural	EASE (SVR)	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
		EASE (BLRR)	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705

tectures. One contributing factor is that, the final representation in neural architectures contains more semantic information than information encoded in hand-crafted information.

Table 4: QWK score of variants of proposed model. The scores with statistical significant improvement (p <0.05) are marked with '*'

I.D	System Layers	Avg.QWK
1	LSTM+GRU	**0.747***
2	GRU+LSTM	0.726
3	LSTM+BILSTM	**0.743***

Apart from the models that are reported in Table 3, we have also experimented with possible permutations of encoder-decoder architecture, as shown in Table 4.A combination of GRU and LSTM in the last layer of encoder and decoder respectively, was trained across all the prompts. Results from Table 4 show that the architecture having LSTM as final layer of encoder and GRU in the decoder(LSTM+GRU)preforms significantly better than our baseline model($p<0.05$). It is observed from Table 3 and Table 4 that final layer of encoder model is having a significant impact on the performance of the whole model. The usage of LSTM in final layer of encoder is giving significant improvement in performance than GRU. This is attributed to inefficient sentence representation of entire essay by GRU hence leading to ineffective context construction in decoder layer. We also used a bidirectional LSTM in decoder, in which the sequence of words are processed in both directions. The results of these architectures are summarized in Table 4.

Table 5 supports H3 hypothesis. In Table 5, we enlist the heatmaps of attention scores assigned by models to every word in the essay and report the average attention score. The observations are made on an essay response to prompt 5, which has been assigned a gold-standard score of 3(highest) and a predicted score of 3. The darkness of red is proportional to the attention assigned to that particular word. Prompt 5 asked the students to write about the mood created by Narciso Rodriguez in his memoir. Examining the architecture (LSTM+LSTM) closely, we can see that certain words like culinary, family, memoir are getting the highest attention while words like better, good, grateful ,love receive attention better than rest of the words. The overall average attention score of this model is higher than our baseline model, which assigns same attention to most of the words in the essay. Looking at the next architecture, (GRU+GRU) the average attention score is higher than the proposed architecture as it assigns higher attention to words better, traditions. The highest attention score is obtained by architecture (LSTM+BLSTM) that utilizes a bidirectional LSTM in the decoder layer. It assigns high attention to important words and it is also interesting to note that the model assigns high score to word collocations, some of the words like gratitude ,grateful ;culinary ,cooking received highest attention .Figure 2 depicts loss graphs for all the architectures proposed. The graphs are plotted for prompt 5, utilizing mean squared error as loss functions. The graphs show variation of loss function with 100 epochs. Figure 2 (a) shows how loss varies over 100 epochs for architecture LSTM+LSTM, while the overall trend is decreasing, intermittent pulses indicate the presence of varied samples that the model is trying to learn. The presence of LSTM in the encoder layer is giving a similar curve as observed in Figure 2 (a),(c),(e); plotted over architectures: LSTM+LSTM,LSTM+GRU,LSTM+BILSTM. Figure 2 (b),(d) plotted across architectures: GRU+GRU,GRU+LSTM, depicts a steady decrease in loss followed by a sharp convergence.

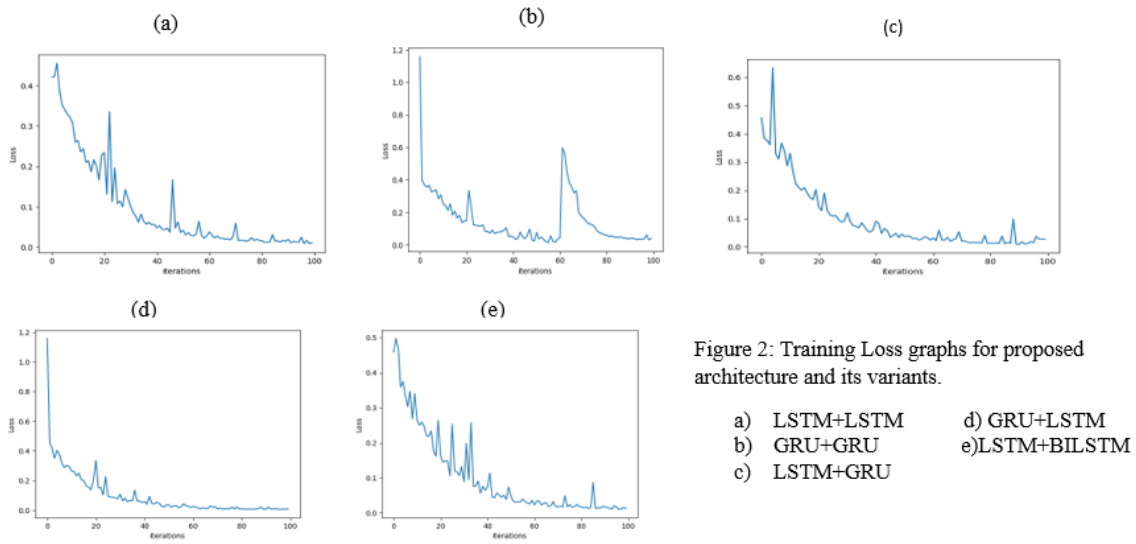


Figure 2: Training loss graphs for various architectures

Table 5: Accuracy Score and attention visualizations for prompt 5 response. The darkness of red is proportional to attention value assigned.

Architecture	System-layers	Essay	Attention
3*With Decoder	LSTM+LSTM	The mood created by the author in this memoir is gra titude Narciso Rodriguez , grateful for way his cuban parents brought him up when they had so little to be gin with @ CAPS thing that are traditions family pass ed on . One of would be there rich culinary skills and a love cooking mother father came country give bett er life even though it meant	0.492
	GRU+GRU	The mood created by the author in this memoir is gra titude Narciso Rodriguez , grateful for way his cuban parents brought him up when they had so little to be gin with @ CAPS thing that are traditions family pass ed on . One of would be there rich culinary skills and a love cooking mother father came country give bett er life even though it meant	0.501
	LSTM+BILSTM	The mood created by the author in this memoir is gra titude Narciso Rodriguez , grateful for way his cuban parents brought him up when they had so little to be gin with @ CAPS thing that are traditions family pass ed on . One of would be there rich culinary skills and a love cooking mother father came country give bett er life even though it meant	0.533
Encoder (Baseline)	CNN +LSTM	The mood created by the author in this memoir is gra titude Narciso Rodriguez , grateful for way his cuban parents brought him up when they had so little to be gin with @ CAPS thing that are traditions family pass ed on . One of would be there rich culinary skills and a love cooking mother father came country give bett er life even though it meant	0.353

6 provides a comparison between the proposed Encoder-Decoder model and the state of the art BERT model. The table provides QWK scores of BERT model taken from Rodriguez et al,**??**. The proposed model performs equally or well than BERT model in all the eight prompts.

Table 6: Comparison of QWK scores between proposed Encoder-Decoder model and BERT

Prompt	Encoder-Decoder (LSTM+LSTM)	BERT
1	0.808	0.792
2	0.648	0.679
3	0.686	0.715
4	0.761	0.801
5	0.811	0.805
6	0.823	0.805
7	0.786	0.785
8	0.702	0.595

7 Conclusion

In this paper, we have proposed a recurrent based encoder-decoder model to address the problem of automated essay scoring that outperforms the state-of -the art attention based models. The proposed model employed a decoder layer and attention mechanism to recognize germane words and sentences. Our model produces better sentence representations hence leading to a deeper semantic analysis than state of the art models. Empirical results on ASAP dataset report outperformance of our model to strong established baselines in terms of quadratic weighted Kappa score. The future scope of this to make the task of essay scoring prompt agnostic and extend beyond English language.

References

[1] Dimitrios Alikaniotis, Helen Yannakoudakis and Marek Rei. 2016. Automatic text scoring using neural networks. In Proceedings of 54th annual meeting of association for Computational Linguistics(Volume 1: Long Papers), (pp. 715-725).

[2] Diederik P. Kingma and Jimmy Ba 2017. Adam: A Method for Stochastic Optimization. Computing Research Repository. arXiv:1412.6980

[3] Fei Dong, Yue Zhang and Jie Yang. 2017. Attention baes recuurent convolutional neural networks for automatic essay scoring. 21st Conference on Computational Natural Language Learning, (pp. 153-162).

[4] Giuseppe Bonaccorso. 2017. Machine Learning Algorithms: A Reference Guide to Popular Algorithms for Data Science and Machine Learning. Packt Publishing

[5] Hochreiter Sepp and Jurgen Scmidhuber (1997). Long short-term memory. Neural computation, 1735–1780.

[6] Jeffery Pennington, Richard Socher, and Christopher D. Manning .2014. Glove:Global Vectors for word representation. Emperical Methods in Natural Language processing(EMNLP), (pp. 1532-1543)

[7] Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris.. 1998. Automated Scoring using Hybrid feature identification technique. 17th-International Conference on Computational Linguistics-Volume 1 (pp. 206-210). Association for Computational Linguistics.

[8] Guoxi Liang, Byung-Won , Dongwon Jeong , Hyun-Chul Kim and Gyu Sang Choi .2018. Automated Essay Scoring: A Siamese Bidirectional LSTM Neural Network Architecture. Symmetry.

[9] Zhang Haoran, Litman Diane 2018. Co-Attention Based Neural Network for Source-Dependent Essay Scoring. Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 399–409). Association for Computational Linguistics.

[10] Peter Phandi, Kian Ming A Chai,abd Hwee Tou Ng 2015. Flexible domain adaptation for automated essay scoring using correlated Linear Regression. In Proceedings of the 2015 Conference On Empirical Methods in Natural Language Processing, (pp. 431-439).

[11] Robert Susik. 2020. Recurrent autoencoder with sequence-aware encoding. Computing Research Repository. arXiv:2009.07349

[12] Taghipour Kaveh and Hwee Tou Ng. .2016. A Neural Approach to Automated Essay Scoring. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing(EMNLP) (pp. 1882–1891).: Association for Computational Linguistics

[13] Dong Fei, and Yue Zhang. 2016. Automatic features for essay scoring-an empirical study. 2016 Confernece on Emprical Methods in Natural Language Processing, (pp. 1072-1077)

[14] Zhu Jiawei Liu, Yang Xu and Yaguang Zhu (2019). Automated Essay Scoring based on Two-Stage Learning. Computing Research Repository. arXiv:1901.07744

[15] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N. and Kaiser, undefinedukasz and Polosukhin, Illia.(2017)Attention is All You Need

[16] Pedro Uria Rodriguez, Amir Jafari, and Christopher M Ormerod. 2019. Language models and automated essay scoring. arXiv preprint arXiv:1909.09482.