Relevance-Based Automated Essay Scoring via Hierarchical Recurrent Model

Minping Chen[1]
[1]*School of Information Science and Technology*
Guangdong University of Foreign Studies
Guangzhou, China
minpingchen@126.com

Xia Li[2,1]✉
[2]*Key Laboratory of Language Engineering and Computing*
Guangdong University of Foreign Studies
Guangzhou, China
xiali@mail.gdufs.edu.cn

***Abstract*—In recent years, neural network models have been used in automated essay scoring task and achieved good performance. However, few studies investigated using the prompt information into the neural network. We know that there is a close relevance between the essay content and the topic. Therefore, the relevance between the essay and the topic can aid to represent the relationship between the essay and its score. That is to say, the degree of relevance between the high score essay and the topic will be higher while the low score essay is less similar to the topic. Inspired by this idea, we propose to use the similarity of the essay and the topic as auxiliary information which can be concatenated into the final representation of the essay. We first use a hierarchical recurrent neural network combined with attention mechanism to learn the content representation of the essay and the topic on sentence-level and document-level. Then, we multiply the essay representation and the topic representation to get a similarity representation between them. In the end, we concatenate the similarity representation into the essay's representation to get a final representation of the essay. We tested our model on ASAP dataset and the experimental results show that our model outperformed the existing state-of-the-art models.**

Keywords-Topic information; Automated essay scoring; Hierarchical Recurrent Neural Networks.

I. INTRODUCTION

Automated Essay Scoring (AES) is a task of grading essays automatically. Traditional Automated Essay Scoring methods are based on rich features (e.g., surface features and content features) and have achieved good results, especially for scenes that require feedback.

In recent years, with the success of neural networks in various tasks, neural network-based models have also achieved good results in AES tasks. In this paper, we focus on the task of scoring essays using neural networks.

Previous neural network-based models mainly focused on the representation of essay content [1-4]. The main difference between them is that they use different neural networks and different levels for essay learning. Alikaniotis et al. [1] and Taghipour and Ng [2] use LSTMs [5] to learn the text representation from document-level. They treat an essay as a single sequence and input it into the model. Dong and Zhang [3] use a two-level of CNN structure to capture the semantic representation of essays. In their subsequent work [4], CNN is used to model sentences and LSTM is used to model documents. They proposed an architecture to model sentences and document separately and use the document-level outputs as the final representation of the essay. Some of models also proposed a neural network architecture to capture the coherence of the essay [6]. Tay et al. [6] adopt a tensor layer to model the relationship between each pair of outputs of LSTM hidden units, aiming to capture the coherence of the essay. These models all have achieved promising results on ASAP dataset.

However, to our best knowledge, few studies have investigated the topic information of the essay. Zhang and Litman [7] is one of the exceptions. They proposed a co-attention based neural network for source-dependent essay scoring to learn which sentences in the essay are mentioned in the source article, which mainly aims to evaluate students' ability to find and use evidence from a source article to write a response. Therefore, the purpose of their work is different from ours. The goal of our model is to use topic information as auxiliary scoring criterion based on the essay content. We know that there is a close relevance between the essay content and the topic. A human rater will give higher scores to those essays related to the topic, and lower scores for those essays that are not relevant to the topic. Therefore, we propose to use the correlation between essay and topic to be used as auxiliary representation for essay scoring.

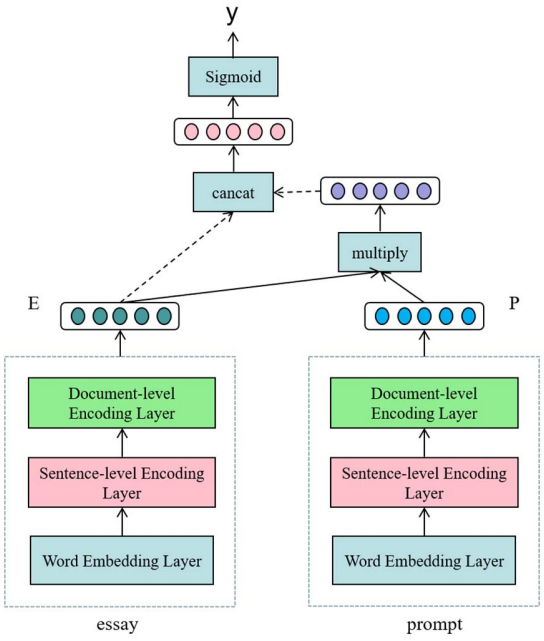

Figure 1. The structure of our model.

Based on this idea, in this paper, we propose to use the similarity between the essay and the topic as auxiliary new

✉ corresponding author: xiali@mail.gdufs.edu.cn

information to be concatenated into the final representation of the essay. Firstly, we propose a hierarchical recurrent neural network combined with attention mechanism to learn the content representation of the essay and the topic respectively on sentence-level and document-level. Then, we multiply the content representation of the essay and the content representation of the topic to get their similarity representation. In the end, we concatenate this similarity representation into the content representation of the essay to get the final representation of the essay. The architecture of our model is illustrated in Fig.1.

We use the proposed model to learn the text representation of the essay and the prompt respectively and then perform an element-wise multiplication on the two vectors to get the similarity between the essay and the prompt. In the end, we concatenate the essay representation, which contains the content information of the essay and the similarity vector as the final representation for an essay. The architecture of sentence-level encoding layer is shown in Fig.2, which is the same as the architecture of document-level encoding Layer.

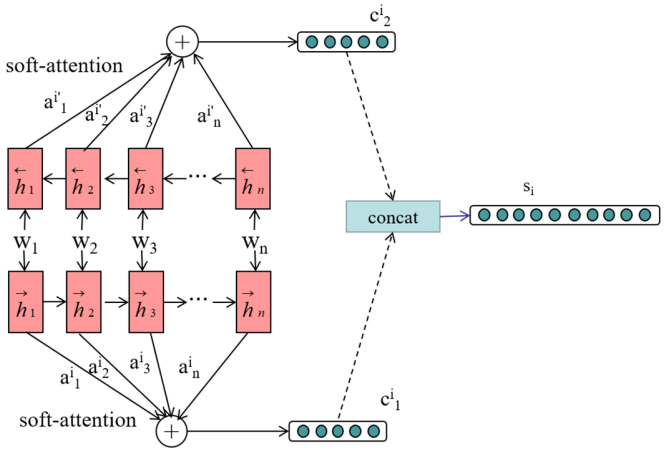

Figure 2. The architecture of Sentence-level Encoding Layer, where w_i means the i-th word in a sentence, $\overrightarrow{h}$ and $\overleftarrow{h}$ denote the hidden state of the two directional LSTMs, a represents the attention weight, c_1 and c_2 represent the weighted sum of the two directional LSTM hidden outputs respectively, and s denotes the final sentence representation.

We use Bi-LSTM to model sentences and documents respectively. One of the advantages of using Bi-LSTM is that training the LSTM in a unidirectional manner might lose some important information of the words sequence and Bi-LSTM can alleviate this problem to some extent. Another advantage is that training LSTM in a bidirectional manner can provide more context information for the word encoding and sentence encoding at each timestep. In our model, we use the attention mechanism [8-9] on the outputs of two directional Bi-LSTM hidden layer units to learn the contribution of each element in the sequence.

The main contributions of our work include:

1)We integrate the topic information into the representation of the essay to improve the performance of scoring model. To our knowledge, no previous work has investigated using the representation of the prompt aid to improving the representation of the essay. We will show that our model outperforms the state-of-the-art methods on the ASAP dataset.

2)We propose to use Bi-LSTM for both sentence-level encoding and document-level encoding aiming to alleviate the loss of content information and learn a better representation based on the context information captured by Bi-LSTM.

II. Model

Different from previous work, we integrate the representation of topic into the content of the essay to better represent the relationship between essay and score. First, we use Bi-LSTM to learn the representation of essays on sentence-level and document-level respectively. Then, we use the soft-attention mechanism to learn the contribution of each word in a sentence and each sentence in an essay. In this section, we will describe our model in detail.

A. Sentence-level Encoding

1) Look-Up Layer

We first segment the essay and the prompt into sentences and the words of every sentence are inputted into a look-up layer to obtain a dense representation. The max length of every sentence is set to 30. We perform a padding operation on sentences less than 30 in length, and divide sentences with lengths longer than 30 into short sentences. We use Stanford GloVe 50-dimensional word-embeddings [10] as pretrained word embeddings in our model. All the embeddings will be fine-tuned during training.

2) Sentence-level Encoding Layer

After obtaining a dense representation of each word in a given sentence $\{w_1, w_2, \dots, w_n\}$, where n is the length of the sentence, we feed them into a Bi-LSTM layer, in which the sequence pass two LSTMs independently in two directions (from left to right and from right to left). Then we use soft-attention mechanism to learn the contribution weights of each word in the sentence. As the hidden outputs at the same timestep of two LSTMs contain context information of the sentence in both directions, we perform two attention operations on the LSTM hidden unit outputs of both directions separately, denoting as $(\overrightarrow{h}_1, \overrightarrow{h}_2, \dots, \overrightarrow{h}_n)$ and $(\overleftarrow{h}_1, \overleftarrow{h}_2, \dots \overleftarrow{h}_n)$. The attention is calculated as shown in equation (1) ~ (3):

$$o_i = tanh(W_a \cdot h_i + b) \quad (1)$$

$$\alpha_i = softmax(u_i \cdot o_i) \quad (2)$$

$$c = \sum_{i=1}^{n} a_i h_i \quad (3)$$

Where W_a, u_i are trainable weights, b is the bias, and a_i is the attention weight. The attention operation is performed on the outputs of two LSTMs $(\overrightarrow{h}_1, \overrightarrow{h}_2, \dots, \overrightarrow{h}_n)$ and $(\overleftarrow{h}_1, \overleftarrow{h}_2, \dots \overleftarrow{h}_n)$ generating two vectors c_1 and c_2. We concatenate the two vectors as the final representation of the sentence s.

B. Document-level Encoding

Suppose the learned representations of sentences are expressed as $\{s_1, s_2, \dots, s_m\}$, where m is the number of sentences in an essay. In order to learn the contribution of each sentence to the final score on the document-level, we use another Bi-LSTM and the same soft-attention mechanism to learn the distributed representation of the essay at the document level.

As shown in Fig. 1, we use the proposed model to learn the representation of the essay and the prompt respectively, denoting as E and P. Then we perform element-wise multiplication on E and P to get the relevance vector $S_{E,P}$ between the essay and the topic. We show the operation as equation (4), where ° means element-wise multiplication.

$$S_{E,P} = E \circ P \tag{4}$$

We expect that $S_{E,P}$ contains some information about the relationship between the essay and the topic, which is helpful for scoring an essay. Therefore, we concatenate the representation of the essay E and the representation of the relevance between the essay and topic $S_{E,P}$ as the final representation of the essay.

C. Model Training

We input the final representation of the essay into a sigmoid layer to get the score of an essay. The equation is described in equation (5).

$$\hat{y} = sigmoid(W[E, S_{E,P}] + b) \tag{5}$$

Where W is trainable weight, b is the bias, and $\hat{y}$ is the predicted score. Following previous work, we use the Mean Square Error (MSE) as the loss function, which is show as equation (6). In equation (6), y is the essay's score rated by human rater and $\hat{y}$ is the score predicted by the model, N is the number of the training essays.

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{6}$$

III. Experiments

A. Datasets

We use ASAP[1] dataset in our experiments. ASAP dataset is widely used in existing work which consists of 12,978 essays with eight different prompts (essay topics) written by students from grades 7-10. The detailed description of ASAP dataset is shown in Table 1.

TABLE I. DESCRIPTION OF ASAP DATASET.

Prompt	#Essay	Avg Len.	Score Range	Score Median
1	1783	350	2-12	8
2	1800	350	0-6	3
3	1726	150	0-3	1
4	1772	150	0-3	1
5	1805	150	0-4	2
6	1800	150	0-4	2
7	1569	250	0-30	16
8	723	650	0-60	30

Following previous studies [2, 4], we split each prompt into 60% as training data, 20% as development data and 20% as test data. We use 5-fold cross-validation in our experiments.

[1] https://www.kaggle.com/c/asap-aes/data

B. Evaluation Metric

We use Quadratic Weighted Kappa (QWK) as the evaluation metric in our experiments. QWK is widely used in many previous studies[1-4,6-7]. It is defined as equation (7).

$$k = 1 - \frac{\sum W_{ij} O_{ij}}{\sum W_{ij} E_{ij}} \tag{7}$$

Where O_{ij} is the number of essays that receive a rating i by the human rater and a rating j by the AES system, and the matrix E is the outer product of vectors of human ratings and system ratings. We performed a normalization operation on the matrix E such that the elements of the matrix E and the elements of the matrix O are the same. The quadratic-weight matrix W_{ij} is defined as equation (8), where i and j are the human rating and the system rating respectively, N is the number of the essays.

$$W_{ij} = \frac{(i-j)^2}{(N-1)^2} \tag{8}$$

Following Taghipour and Ng [2], Dong and Zhang [4], we performed one-tailed t-test to determine the statistical significance of improvements.

C. Experiment Setup

In order to make our results more comparable, we use the same preprocessing as in Taghipour and Ng [2] and Dong and Zhang [4]:

1)NLTK is used to tokenize each essay;

2)All words are lowercased;

3)The score is normalized within the range of (0, 1).

During the model evaluation phase, the score is converted back into an integer within the original score range to facilitate the calculation of the QWK value. We select 4000 words with the highest frequency from the training data as the vocabulary and treat all other words as unknown words. The hyper-parameters of our model are showed in Table 2.

TABLE II. PARAMETERS OF OUR MODEL.

Layer	Parameters	Value
Look-up	Word embedding dim	50
Sentence-level Encoding	Bi-LSTM Hidden units	50
Document-level Encoding	Bi-LSTM Hidden units	50
Dropout	Dropout rate	0.5
Others	Optimization	ADAM
	epoch	50
	Batch size	10
	Initial learning rate	0.001
	Max_sentence_length	30

We use ADAM [11] as our optimizer and the initial learning rate is set to 0.001. The model is trained for 50 epochs in each prompt and evaluation is performed after each training epoch. We use the model which achieves the best performance on the development data to predict on the test data.

D. Experimental Results

In this section, we will introduce the baselines we used in our experiments and present the results of our model on ASAP dataset.

1) Baselines

We use four state-of-the-art models as the baselines in our experiments.

CNN-CNN-MoT model. (Dong and Zhang [3]). The model uses two layers of CNN, one of which is performed to learn the sentence representation and the other is stacked above, followed average pooling layer to get the text representation.

LSTM-MoT model. (Taghipour and Ng [2]). The model takes all words of the essay as input to a LSTM, and then averages all the LSTM hidden layer outputs as the final essay representation.

LSTM-CNN-attention model (Dong and Zhang [4]). The model uses CNN to learn the sentence representation, in which soft-attention mechanism is performed to learn the n-grams weights in sentences and then the sentence representations are inputted to a LSTM layer, after that soft-attention is performed to obtain the final representation of the essay.

SKIPFLOW-LSTM model (Tay et al. [6]). The model adopts a tensor layer to model the relationship between each pair of outputs of LSTM hidden units, which aims to capture textual coherence.

We name our model described in Section 2 as Topic-BiLSTM-attention. We also use the Topic-BiLSTM-MoT model and BiLSTM-attention as another two baselines. The former performs an average pooling on the LSTM hidden units output and this is the only difference from Topic-BiLSTM-attention model. The latter only models the essay with the proposed model without using topic information compared with Topic-BiLSTM-attention model.

2) Results on ASAP Dataset.

The experimental results are shown in Table 3 (*means statistical significance). As we can see, the average QWK of our model on ASAP dataset is 0.773, which outperforms all existing state-of-the-art models. In particularly, the average QWK of our model is 3.9% higher than CNN-CNN-MoT model, 2.7% higher than LSTM-MoT model, and 0.9% higher than LSTM-CNN-attention and SKIPFLOW-LSTM. The average QWK of another two baselines Topic-BiLSTM-MoT and BiLSTM-attention is 0.760 and 0.766 respectively.

E. Discussion

In this section we will do some analysis from several aspects to verify the effectiveness of our model.

1) Topic information:

We use BiLSTM-attention model which does not employ the topic information to see whether our model has captured some semantic relation between the essay and prompt which can provide some helpful assistance for grading an essay. As shown in Table 3, Topic-BiLSTM-attention model outperforms BiLSTM-attention model for 0.7% on average QWK score. Although the improvement is not very large, it seems that our model learned some semantic relationship between the essay and the prompt which can be one of the criterion to evaluate the goodness of an essay. Besides, even lacks of the topic information, the BiLSTM-attention model still outperforms the state-of-the-art models LSTM-CNN-attention and SKIPFLOW-LSTM, which proves the competitiveness of BiLSTM-attention model.

2) Attention and Average pooling.

The Topic-BiLSTM-attention model outperforms Topic-BiLSTM-MoT model for 1.3% on QWK, which proves the effectiveness of attention mechanism. One of the reasons for this improvement is that the average pooling treats words and sentences equally, which is different from human raters, who can distinguish the contribution of each word and sentence. On the contrary, the attention mechanism can learn which part in a sequence is more important for scoring and assign a weight for every element, which makes the essay representation is more suitable for scoring.

3) Human rating and Neural networks rating:

We also compared the human performance with that of neural networks models. The QWK results of human raters on ASAP dataset is shown in Table 3. Most neural network models outperformed human performance except CNN-CNN-MoT and LSTM-MoT. Our model achieves 1.9% improvement on QWK compared with human raters. For five prompts of total eight prompts, the QWK of our model is higher than that of human raters. The comparison with human performance indicates that our model is actually appropriate for AES task.

TABLE III. QWK RESULTS ON ASAP DATASET.

Models	Prompts								
	1	2	3	4	5	6	7	8	Avg.
Human1 – Human2	0.721	0.812	0.769	0.851	0.753	0.776	0.720	0.627	0.754
CNN-CNN-MoT	0.805	0.613	0.662	0.778	0.800	0.809	0.758	0.644	0.734
LSTM-MoT	0.775	0.687	0.683	0.795	0.818	0.813	0.805	0.594	0.746
LSTM-CNN-attention	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
SKIPFLOW-LSTM	**0.832**	0.684	**0.695**	0.788	0.815	0.810	0.800	0.697	0.764
Topic-BiLSTM-MoT	0.819	0.682	0.683	0.811	0.811	0.810	0.799	0.669	0.760
BiLSTM-attention	0.822	0.681	0.689	**0.820**	**0.819**	0.816	0.802	0.678	0.766
Tpioc-BiLSTM-attention	0.827	**0.696**	0.691	0.816	0.811	**0.823**	**0.809**	**0.707**	**0.773***

IV. CASE STUDY

In order to know how our model learns to capture the relationship between the essay and prompt and how the text representations of the essay and prompt are formed from sentences, we take an essay from prompt7 and its prompt content to analyze the attention weights of theirs sentences separately, as shown in Table 4 and Table 5.

The topic of prompt7 is to write about patience. We list all sentences in the essay and prompt in order and each sentence is associated with its attention weight. We only analyze one of the two-directional attentions because the other one works in the same way for scoring. The sums of the attention weights of sentences in the essay and prompt are both 1.

In Table 5, we can see that the first sentence which indicates the required writing content for writers directly has the biggest attention weight. Actually, the differences of attention weights between five sentences in the prompt are not very large as they are all important for students to understanding the requirements before writing an essay. However, in Table 4, it is obvious to see that the differences of attention weights between all the sentences in the essay are more significant. This essay wrote a story about patience. The first several sentences of it described how the story happed, of which the attention weights are relatively small. On the contrary, we can see that the last four sentences have larger attention weights as they are all written to express something about patience which make more contributions to the quality of the essay. Besides, they are all more relevant with the prompt content, which indicates that our model can capture some semantic relation between the essay and prompt to some extent. It also proves that attention mechanism can learn the key sentences in the essay and prompt. As the essay representation is generated from the weighted sum of the sentence representation, it can prove that the attention weight can reflect the importance of each sentence for scoring as we expect.

TABLE IV. SENTENCE ATTENTION WEIGHTS IN AN ESSAY FROM PROMPT7.

No.	Sentences	Attention weights
1	There have been times in my life where I had to be patient.	0.04363
2	My friends and I that day had decided to go to @LOCATION1 @CAPS1.	0.05155
3	I love roller coasters, but I hate the lines that come with.	0.06716
4	When we got there we had to wait a half an hour to get in.	0.07252
5	When we made the decision to go on the @CAPS2 ride first.	0.08020
6	Next, we had to wait another @NUM1 minutes to get on the ride.	0.09182
7	Other rides we want on too had a long wait.	0.08242
8	So throughout the day there was quite a bit of waiting.	0.08968
9	Even when we got food we had to wait with all this waiting it **took a lot of patience**.	0.10115
10	My friend and I had to be **understanding and tolerant of the waiting and long lines.**	0.10116
11	By the end of the day all of us had gotten used to **using our patience**.	0.11400
12	That trip to @LOCATION1 @CAPS3 taught me to be **a more patient person**.	0.10471

TABLE V. SENTENCE ATTENTION WEIGHTS FROM TOPIC OF PROMPT7.

No.	Sentences	Attention weights
1	Write about **patience.**	0.20048
2	Being **patient** means that you are **understanding and tolerant**.	0.20030
3	**A patient person** experience difficulties without complaining.	0.20016
4	Do only one of the following: write a story about a time when you were **patient** OR write a story about a time when someone you know was **patient**	0.19952
5	OR write a story in your own way about **patience.**	0.19952

V. RELATED WORK

Traditional studies in AES are mainly based on feature engineering. These studies use manual features such as bag-of-words, part-of-speech tags, length of essays and parsing trees. The existing AES methods can be divided into two categories: supervised methods and unsupervised methods. In the case of supervised methods, previous studies [12-18] used traditional machine learning algorithms for AES task based on classification or regression. The unsupervised methods mainly use ranking methods [19-20]. All these studies require time-expensive manual work.

In recent years, neural network-based models have been used in the task of AES, which can avoid handcrafted features. Alikaniotis et al. [1] proposed a two-layer bidirectional LSTM model to learn the representation of essays and take the last hidden state as the final essay representation. They train score-specific word embeddings (SSWEs) to represent the words. Taghipour and Ng [2] treated an essay as a sequence and feed it into a layer of LSTM and takes the average of the hidden outputs of LSTMs as text representation. Dong and Zhang [3] adopted a two-layer CNN model to learn sentence-level representation and document-level representation respectively. In their subsequent work [4], they used a CNN layer and a LSTM layer to model sentences and documents separately. Attention mechanism was used in both of these two layers. Zhao et al. [21] employed a memory network model, in which an ungraded essay was assigned a score by computing the relevance between each selected essay as grading criteria specified in memory and the ungraded essay. Tay et al. [6] developed a SKIPFLOW-LSTM model for AES task. They employed a tensor layer to capture the relationship between snapshots of an essay based on the LSTM hidden states pairs.

Although existing neural network-based studies have achieved good results in AES tasks, they are mainly focused on extracting semantic and coherent information from essays. Few previous neural network-based studies have investigated using topic information as auxiliary scoring criterion. One of these studies is Zhang and Litman [7]. They proposed a model based on Dong and Zhang [4], in which they replace the attention pooling layer for text representation with a bi-directional attention flow layer and an additional modeling layer.

Inspired by the fact that human raters usually care about the relevance between the essay and the prompt when scoring an essay, we employ the topic information to learn the relevance between the essay and the topic as auxiliary scoring criterion in

addition to the essay content criterion. Experimental results on ASAP dataset show that our model achieves the best results compared with all the baselines in this paper and can be applied to AES tasks.

VI. Conclusion

In this paper, we propose a hierarchical recurrent model with attention mechanism to model sentences and documents separately. We use two layers of Bi-LSTM to learn sentence representation and document representation hierarchically, in which attention mechanism is used to learn the importance of each word for one sentence and each sentence for a whole document. We use the proposed model to learn a representation of the essay (essay vector) and a representation of the topic (topic vector) respectively and then perform an element-wise multiplication over the two vectors to get a representation of the relationship between the essay and the topic. In the end, we concatenate the representation of the essay and the representation of the relationship as the final representation of the essay. The results on ASAP dataset show that our model outperforms the current state-of-the-art model.

Acknowledgment

This work is supported by the National Science Foundation of China (61402119) and Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation("Climbing Program" Special Funds).

References

[1] D. Alikaniotis, H.Yannakoudakis, M. Rei, "Automatic Text Scoring Using Neural Networks," in proceeding of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 715-725, 2016.

[2] T. Kaveh, T. Hwee, "A neural approach to automated essay scoring," in proceeding of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1882-1891, 2016.

[3] F. Dong, Y. Zhang. "Automatic Features for Essay Scoring," in proceeding of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 968-974, 2016.

[4] F. Dong, Y. Zhang, J. Yang. "Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring," in proceeding of the 21st Conference on Computational Natural Language Learning, pp. 153-162, 2017.

[5] S. Hochreiter, J.S chmidhuber. "Long short-term memory," Neural computation, 9(8), pp.1735-1780, 1997.

[6] Y. Tay, M. Phan, L. Tuan, S. Hui. "SkipFlow: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring," arXiv preprint arXiv: 1711. 04981, 2017.

[7] H. Zhang, D. Litman. "Co-Attention Based Neural Network for Source-Dependent Essay Scoring," in proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 399-409, 2018.

[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio. "Show, attend and tell: neural image caption generation with visual attention," in proceedings of the 32nd International Conference on Machine Learning , pp. 77-81, 2015.

[9] [J. Li, M. T Luong, D. Jurafsky. "A Hierarchical Neural Autoencoder for Paragraphs and Documents," in proceedings of the 53th Meeting of the Association for Computational Linguistics, pp. 1106-1115, 2015.

[10] J. Pennington, R. Socher, C. Manning. "Glove: Global Vectors for Word Representation," in proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532-1543, 2014.

[11] D. P. Kingma, and Ba, J. "Adam: A method for stochastic optimization," CoRR abs/1412.6980, 2014.

[12] Page. Ellis Batten. "Computer Grading of Student Prose, Using Modern Concepts and Software," Journal of Experimental Education, 62(2): 127-142, 1994.

[13] K. L Thomas, W. F Peter, L. Darrell. "An introduction to latent semantic analysis," Discourse processes 25(2-3):259–284, 1998.

[14] T. K. Landauer, P. W. Foltz, D. Laham. "An introduction to latent semantic analysis," Discourse processes 25(2-3):259–284, 1998.

[15] P. W. Foltz, D. Laham, T. K. Landauer. "Automated essay scoring: Applications to educational technology," in proceedings of EdMedia, volume 99, pp. 40–64, 1999.

[16] L. S. Larkey. "Automatic essay grading using text categorization techniques," in proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 90–95, 1998.

[17] L. M. Rudner. "Automated essay scoring using Bayes' theorem," National Council on Measurement in Education New Orleans La, 1(2):3-21, 2002.

[18] Y. Attali, J. Burstein, Y. Attali, J. Burstein. "Automated essay scoring with e-rater R V. 2.0," ETS Research Report Series, pp. 1–21, 2004.

[19] P. Phandi, K. M. A. Chai, H. T. Ng. "Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression," in proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing(EMNLP), pp. 431-439, 2015.

[20] H. Yannakoudakis, B. Medlock, B. Medloc. "A new dataset and method for automatically grading ESOL texts," in proceedings of the 49th Meeting of the Association for Computational Linguistics, pp. 180–189, 2011.

[21] S. Zhao, Y. Zhang, X. Xiong, A. Botelho, N. Heffernan. "A Memory-Augmented Neural Model for Automated Grading," in proceedings of the 4th ACM Conference on Learning at Scale, pp. 189–192 ,2017.