Coherence-based Automated Essay Scoring using Self-Attention

Xia Li[1,2*], Minping Chen[2], Jianyun Nie[3], Zhenxing Liu[2], Ziheng Feng[2], Yingdan Cai [2]

[1] Key Laboratory of Language Engineering and Computing
Guangdong university of foreign studies, Guangzhou, China
shelly_lx@126.com
[2] School of Information Science and Technology/School of Cyber Security
Guangdong university of foreign studies, Guangzhou, China
{minpingchen,liuzhenxingw,zihengfeng,ldchoy}@126.com
[3] Department of Computer Science and Operations Research
University of montreal, Montreal, Canada
nie@iro.umontreal.ca

Abstract. Automated essay scoring aims to score an essay automatically without any human assistance. Traditional methods heavily rely on manual feature engineering, making it expensive to extract the features. Some recent studies used neural-network-based scoring models to avoid feature engineering. Most of them used CNN or RNN to learn the representation of the essay. Although these models can cope with relationships between words within a short distance, they are limited in capturing long-distance relationships across sentences. In particular, it is difficult to assess the coherence of the essay, which is an essential criterion in essay scoring. In this paper, we use self-attention to capture useful long-distance relationships between words so as to estimate a coherence score. We tested our model on two datasets (ASAP and a new non-native speaker dataset). In both cases, our model outperforms the existing state-of-the-art models.

Keywords: Self-Attention; Automated Essay Scoring; Neural Networks.

1 Introduction

Traditional Automated Essay Scoring (AES) methods are based on manually determined features, which require much manual work. In contrast, the recent neural-network-based models [1-5] can automatically extract features to avoid feature engineering work. It turns out that neural-network-based methods can achieve better performance than the traditional methods and human raters [2, 4].

Previous neural network models for AES use word embedding of essay's words as input and use CNN and LSTM to capture the content information and local relationship between the words within sentences. However, the captured relationships remain local (within sentences), and no global relationships among the words across sentences and paragraphs can be obtained.

In human essay scoring, we observe that global relationships play an important role. When rating an essay, human raters not only consider the goodness of an essay's content, but also pay much attention to the structure of the essay. In particular, the *coherence* between different parts of the essay is an important rating criterion. A good essay should contain related parts with strongly connected words, while a bad essay may contain parts that are unrelated.

For example, the following text fragments (a) and (b) are all grammatically correct:

(a) *I was born in **Glasgow**. **Glasgow** is the largest city in Scotland.*
(b) *I was born in **Glasgow**. It is very nice in Scotland.*

However, the first fragment reads better than the second one because sentences in the first fragment are more connected and it is more coherent. A higher coherence implies that the sentences read more smoothly [6-7]. A human rater would rate the first segment higher than the second one according to the coherence criterion. For a whole essay, the principle is the same. We expect a good essay to be coherent between different parts.

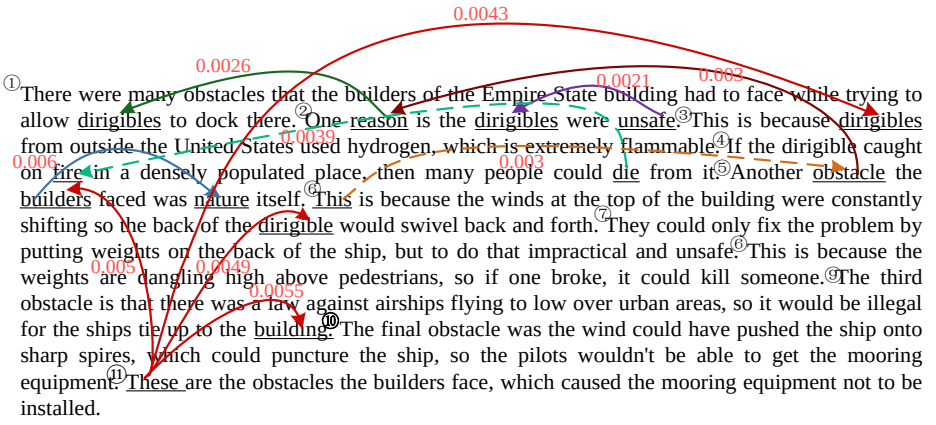

Fig. 1. Coherences learned in a 4-score essay from prompt 6. In this figure, we show that the coherences of the essay are important for judging of good essay for human raters. We can see that the word "reason" in sentence ② has a high relationship with the word "dirigibles" in sentence ①, which represent the causal relationship between sentence ① and ②. We can also see that the word "These" in sentence ⑪ has four high relationships with "dirigibles" in sentence ③, "builders" in sentence ⑤, "dirigible" in sentence ⑥ and "building" in sentence ⑨. These relationships can help and improve the essay's readability which gives the ground of high score for human raters.

Most previous studies on AES did not take into account the coherence aspect. Tay et al. [5] is one of the few exceptions. It incorporated the relationship between two LSTM hidden unit outputs as neural coherence features. These relationships were used as auxiliary neural coherence features to predict essay scores. However, these neural

coherence features were captured only between words within windows of fixed size. The relationships among words are thus limited to local relationships. The model was unable to capture the relationships between distant words and thus to estimate the global coherence.

In general, it is known that LSTM and CNN have difficulty to capture long-distance dependencies. To solve the problem, recent work [8] used self-attention instead, which is shown to be a mechanism capable of capturing long-distance relationships between words in a sequence. Inspired by this work, we propose to use self-attention to learn the relationships between words in the whole essay. The words in relation can be from the same sentence or from different sentences. Fig.1 shows some of the relationships recognized between words by self-attention[1]. We use different colored lines with arrow to indicate the relationships of different words. The values beside the lines represent the attention weights of the relationship between the two words. We can see that self-attention mechanism can learn these explicit relationships between words at long distance. In this figure, we can also notice that the connected words are also semantically related (based on their contents).

Based on these observations, our intuition used in this paper is that an essay with strong connections between words has a high coherence, and thus should be rated high. This coherence criterion is combined with the usual criterion on content. It may not be possible to simply sum up all the connections as a coherence measure. However, the notion of coherence may be much more complex. Therefore, we use a more sophisticated mechanism to aggregate the connections into a rating score. We observe that there is a natural sequence between the connections among words: a connection usually connects a word with some previous words in the sentence or in other sentences. Based on this observation, we use a LSTM layer stacked on the output of self-attention to learn the whole score of the essay's coherence based on the recognized connections.

The architecture of our model is illustrated in Fig.2. Multi-head self-attention is first applied to the words (embeddings) and their positions are used to recognize the connections between words. Each head is intended to capture one type of relationship. Then LSTM is used to aggregate the outputs of multiple heads into a rating score.

An important difference between this approach and the previous ones using LSTM is that our LSTM works on relations between words rather than their representations (contents). Therefore, our LSTM will also account for the connections among words, in addition to what the words represent, which is the common focus of previous LSTM-based approaches. The aggregated output of the LSTM will then encode the global coherence of the essay.

To our knowledge, no prior work has investigated using coherence based on self-attention for AES. We will show in our experiments that our model outperforms the state-of-the-art methods. The main contributions of our paper include:

1) We propose a model based on self-attention mechanism to capture the relationships between different parts of the essay, and we show that the method is appropriate for AES.

[1] The essay is from prompt 6 of ASAP dataset - https://www.kaggle.com/c/asap-aes/data. We only show some of the strong relationships for clarity.

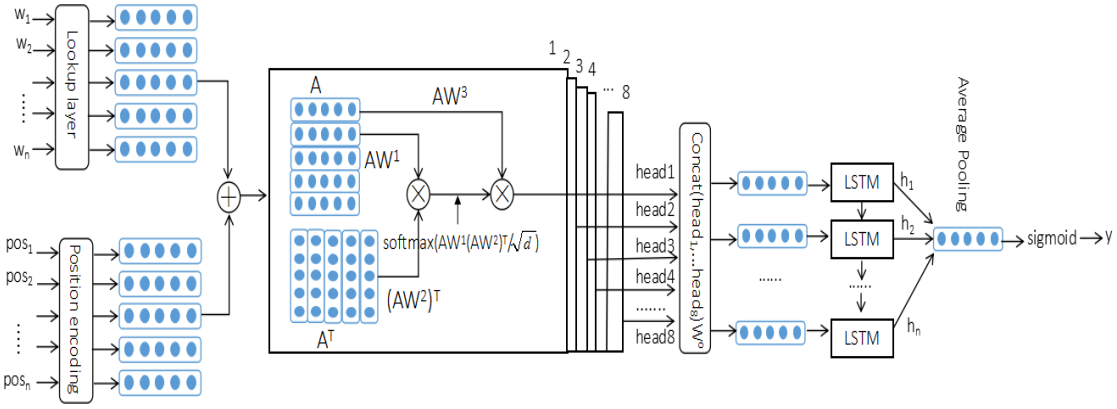

Fig. 2. Structure of our model.

2) We tested our approach on two sets of essays, one from native English speakers and another from English learners. In both cases, we show that our model outperforms the existing approaches.

2 Coherence-based Scoring Model

Different from previous work, we use multi-head self-attention to learn the relationships between different words in the essay and obtain a relationship-based essay representation. Then, we use LSTM to learn the essay's overall score based on the output of multi-head self-attention. In this section, we will describe our model in detail.

2.1 Words and Positions Encoding

Look-Up Layer. First, the words of an essay are inputted into the look-up layer to obtain a dense representation. We use the length of the longest essay in each topic as the length of all the essays in the topic and use the padding operation when the length of other essays is shorter. We use the Stanford's open source 50-dimensional GloVe word embedding [9] as our word representations. The word embedding will be fine-tuned during training.

Position Encoding. Position encoding intends to mark the position of each word in the essay, so that a word can be connected to the word around it. Following [8], we encode the absolute position information of the words to obtain a position embedding with the same dimension as the word embedding (i.e. 50). This embedding is obtained by equation (1) and (2), where *pos* is the position of the word, *i* is the *i*-th dimension in the position embedding, and *d* is the dimension of embedding.

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \tag{1}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (2)$$

PE defines a sinusoidal function, which could allow the model to attend relative positions [8]. Through this operation, each word position is represented by a vector with values between 0~1 and has a relation with the word's position. We add the word embedding and the position embedding to form the input matrix to self-attention in the same way as in [8]. This produces the input matrix *A* used for self-attention networks.

2.2 Getting Relationships with Self-Attention

After obtaining a dense representation of each word and position, we use self-attention to capture the relationships between each pair of words across the essay. In our model, we use 8 heads of self-attention that work in parallel. A head of self-attention is intended to capture one type of relationship.

Vaswani et al. [8] used self-attention for machine translation. Self-attention is based on key-value network, in which key memories (*K*) encode the information to calculate the attention for a given input (*Q*), while the value memories (*V*) encode the corresponding context information and the distribution for the output (next translation word). Vaswani et al. [8] defined the following scaled-dot-product attention for their purpose:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (3)$$

where QK^T determines where the attention should be paid to, $\frac{1}{\sqrt{d}}$ is a scaling factor (d is the dimension of the embedding) to counteract the fact that the softmax function is pushed into regions where it has extremely small gradients when d is large. Similar mechanism has been used for question answering – to determine the best answer for a given question (*Q*).

In our case, the situation is different: we intend to estimate the connections between words in the same essay. This can be basically estimated through AA^T, which tells us how different words are connected. The value network is also based on the same representation space as *A*. However, instead of using *A*, we use 3 different linear projections of *A* as in [8]: AW^1, AW^2 and AW^3. Attention is thus determined by equation (4).

$$Attention(AW^1, AW^2, AW^3) = softmax\left(\frac{AW^1(AW^2)^T}{\sqrt{d}}\right)AW^3 \quad (4)$$

We also use 8 heads of self-attention in parallel. The 8 heads are concatenated at the end, which is projection with another weight matrix W^0. The multi-head attention mechanism is then defined as follows:

$$head_i = Attention(AW_i{}^1, AW_i{}^2, AW_i{}^3) \quad (5)$$

$$MultiHead(A) = Concat(head_1, \ldots, head_h)W^0 \quad (6)$$

where h is the number of attention heads (i.e. 8) and W^0, W^1, W^2, W^3 are trainable parameters.

2.3 Estimating global score with LSTM

The output of multi-head attention should be used according to the task. For machine translation, it is used to help the decoding, i.e. determining the next translation word. In our task, we want to obtain a global rating for an essay. As we noticed, the connections between words in an essay are sequential: a word is usually connected to the words before it. Therefore, we use LSTM [10-11] to cope with the sequential nature of the connections. LSTMs have been generally used to compose a document (text) representation in most previous models [2-4]. The main difference of our case is that we deal with relationship-based representation.

Assuming that an essay consists of m words $w_1, w_2, \dots, w_m$. After the operation of self-attention, the output is m vectors $w'_1, w'_2, \dots, w'_m$, each vector w'_i representing the relationship encoding between the word w_i and other words w_j $(j = 1,2, \dots, m)$. We consider these representations as a sequence and use LSTM to produce the internal states for the sequence at each timestep t: (h_1, h_2, ..., h_t, …, h_n).

Then, we use an average pooling to get the final representation of essay S. Here average pooling is used because we believe that all the connections between different sentences and different words are equally important. Finally, the final essay score is obtained through a fully-connected layer with a nonlinear activation function. The activation function is showed in equation (7).

$$\hat{y} = sigmoid(WS + b) \tag{7}$$

Where W is the weight matrix, b is the bias, and $\hat{y}$ is a predicted score. As in previous work, we use the Mean Square Error (MSE) as a loss function, as in equation (8), where y is the essay's score rated by human and $\hat{y}$ is the score predicted by the model.

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{8}$$

3 Experiments

3.1 Datasets

In this paper, we use two datasets in our experiments. The first dataset is ASAP which is widely used in the existing work. The ASAP contains 12,978 essays in eight different prompts (essay topics) which are written by students from grades 7-10. Another dataset was collected from the College Entrance English Examination (CEEE dataset) of Guangdong province in China. CEEE contains 3,958 essays written by Chinese English-learners who are senior students. There is only one prompt in CEEE, which asks to write an essay based on four given pictures. Some detailed description of these two datasets is showed in Table 1. Following previous studies [3-4], we split each prompt into 60% training data, 20% development data and 20% testing data. We use 5-fold cross-validation in our evaluations.

Table 1. Details of ASAP and CEEE datasets.

Data	Prompt	#Essay	Avg Len.	Score Range	Score Median
ASAP	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
CEEE	1	3958	145	0-25	13

3.2 Evaluation Metric

We use Quadratic Weighted Kappa (QWK) as the evaluation metric in our experiments. This metric is widely used in many previous studies. It is defined in equation (9).

$$k = 1 - \frac{\sum W_{ij} O_{ij}}{\sum W_{ij} E_{ij}} \tag{9}$$

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. Matrix E needs to be normalized such that the sum of elements in E and the sum of elements in O are the same. The quadratic-weight matrix W_{ij} is defined in equation (10), where *i* and *j* are the human rating and the system rating respectively, and *N* is the number of the essays.

$$W_{ij} = \frac{(i-j)^2}{(N-1)^2} \tag{10}$$

Following Taghipour and Ng [3], Dong and Zhang [4], we performed one-tailed t-test to determine the statistical significance of improvements.

3.3 Experiment Setup

To make our results comparable, we use the same preprocessing as in Taghipour and Ng (2016) and Dong and Zhang (2017) [3-4]: NLTK is used to tokenize each essay, all the words are lowercased, and the score is normalized within the range of [0,1]. During the model evaluation phase, the score is converted back to 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.

We use RMSprop [12] 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 retain the model that produces the best performance on the development set as the final model to be used on testing data.

Table 2. Parameters of our model.

Layer	Parameters	Value
Look-up	Word embedding dim	50
Self-Attention	Number of heads	8
	Size per head	16
	Number of layers	1
LSTM	Hidden units	100
Dropout	Dropout rate	0.5
Others	Optimization	RMSprop
	epoch	50
	Batch size	10
	Initial learning rate	0.001

3.4 Experimental Results and Discussion

In this section, we will introduce the baselines and present the results of our model on the two datasets.

Baselines. We use three state-of-the-art models as our baselines in the experiments:

- **LSTM-MoT** model (Taghipour and Ng, 2016 [3]): The model inputs all words of the essay into the LSTM model and uses the average of all hidden layer outputs of the LSTM model as the final representation of the essay.
- **LSTM-CNN-attention** model (Dong and Zhang, 2017 [4]): The LSTM-CNN-attention model [4] uses the soft-attention mechanism [13-14] to learn the n-grams weights of the sentences and then inputs these sentence representations to a LSTM layer. The model uses soft-attention to all the hidden outputs of the LSTM layer to obtain the final representation of the essay.
- **SKIPFLOW-LSTM** model (Tay et al., 2018 [5]): The SKIPFLOW-LSTM model adopts a tensor layer to model the relationship between each pair of hidden unit output of the LSTM, which aims to capture the textual coherence.

Our model described in Section 2 is named **Self-attention-LSTM**. In addition, we also tested the **Self-attention-MoT** model. Instead of stacking an LSTM operation on the relationship representations, the model simply performs an average pooling and uses the average pooled result as the final representation of the essay.

Results on ASAP Dataset. The experimental results are showed in Table 3. As we can see, our model has achieved better performance than other state-of-the-art methods, except for prompt 4. The average QWK of our Self-attention-LSTM model on ASAP data is 3.0% higher than LSTM-MoT and 1.2% higher than LSTM-CNN-attention and

SKIPFLOW-LSTM. These differences are statistically significant. On prompt 4, the performance of our model is close to that of LSTM-CNN-attention.

Table 3. QWK results on Kaggle data set. * means statistical significance.

Models	Prompts								
	1	2	3	4	5	6	7	8	Avg.
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
Self-attention-MoT	0.830	0.683	0.667	0.796	0.810	0.812	0.801	0.703	**0.763**
Self-attention-LSTM	**0.834**	**0.692**	**0.700**	0.811	**0.819**	**0.822**	**0.816**	**0.713**	**0.776***

We also do the experiments on ASAP dataset using only self-attention without LSTM layer (Self-attention-MoT), in which the relationship-based representation are directly fed into an average pooling layer to get the final representation of the essay. In this case the average QWK on ASAP dataset 0.763 is slightly lower than the best baseline methods, which indicates that self-attention can learn useful relationship-based representation of the essay to some extend, despite the simplicity of the model. This difference with Self-attention-LSTM, although not very large, reflects the gain brought by LSTM to aggregate the sequential information about the relationships between words.

Results on CEEE Dataset. The CEEE dataset is different from ASAP in which the essays are written by English learners. We expect that sentences are less fluent in CEEE than in ASAP. The results are showed in Table 4. We can see that Self-attention-LSTM method still achieves the highest QWK value 0.731, which is 2.2% higher than LSTM-MoT and 0.6% higher than LSTM-CNN-attention. The differences are again statistically significant with $p < 0.05$. This result confirms that our model can be used for scoring different types of essays, whether by native English speakers or by English learners.

Table 4. QWK results on CEEE data set. SKIPFLOW-LSTM (Tensor) has not been tested on this dataset, so the result is missing. * means statistical significance.

Models	**CEEE**
LSTM-MoT	0.709
LSTM-CNN-attention	0.725
SKIPFLOW-LSTM (Tensor)	-
Self-attention-MoT	0.729
Self-attention-LSTM	**0.731***

Discussion. The experiments demonstrate that the model we propose can produce superior performance than the existing state-of-the-art models. The key difference between our model and previous models lies in the use of relationships between words across the essay so that scoring relies on coherence. Compared to the previous models that capture only local relationships between words (mainly within sentences), the self-attention mechanism is capable of detecting relationships between words at any positions in the essay. Such global relationships can better reflect an essay's coherence. Our experimental results have confirmed that the coherence criterion is an important factor in essay scoring.

4 Related Work

Many of traditional AES models are devoted to the design of features. These works including supervised methods based on machine learning algorithms [15-21] and unsupervised methods based on ranking methods [22-23]. All these previous methods are based on handcrafted features, which require significant amount of manual works.

In recent year, neural networks have been used in AES task [1-5]. Compared to the traditional approaches, neural approaches do not need any handcrafted features. Alikaniotis et al. [1] train a score-specific word embeddings (SSWEs) to represent words. They use a two-layer bidirectional LSTM and take the last hidden state as the final representation of essays. Taghipour and Ng [3] input the essay's words into a layer of LSTM, taking the average of all hidden states of LSTM as the essay's representation. Dong and Zhang [2] proposed a hierarchical CNN model for sentence-level and text-level representation by processing text into sentences. In their subsequent work [4], they used the soft-attention mechanism to learn the n-grams weights in sentences and then use these sentence representations as input to a LSTM layer to obtain the final representation of the quality of essays. Zhao et al. [24] use memory network model for AES task. The model predicts a score for an ungraded essay by computing the relevance between the ungraded essay and each selected essay as grading criteria specified in memory.

These previous works use the word embedding as input and can be seemed as content-based representations for the next LSTM layer. But relationships of words and sentences from different parts of the essay are important for evaluating the quality of an essay.

Tay et al. [5] propose a SKIPFLOW-LSTM model for AES task, they adopt a tensor layer to model the relationship between each two LSTM hidden unit outputs as neural coherence features to represent and approximate textual coherence. Although SKIPFLOW-LSTM model can capture the relationship between snapshots of essay, these snapshots are continued words of sequence with a fixed size. Their model still can't capture the relationships between different parts of the essay, for example, the first sentence and the last sentence.

Inspired by works using self-attention on different tasks [8, 25-30], we propose to use self-attention to learn the relationships between words from different parts of the essay. And we use a LSTM layer to capture the overall coherence of the essay based on these relationship-level representations.

5 Conclusion

In this paper, we propose an automated essay scoring model based on relationship-representations via self-attention and LSTM framework. Firstly, we add word embeddings of the essay and position encoding matrices as first input to self-attention layer. And then, we use self-attention mechanism to learn the relationship between words from different parts of the essay. Before self-attention, we use content-based representation as input and obtain the relationship-based representation of the essay as output. In the end, we use a LSTM layer to capture the overall coherence of the essay through averaging all the hidden unit output of LSTM as the final representation of the quality of essays. The results on Kaggle data set and CEEE data set show that our model outperforms the current state-of-the-art model.

Acknowledgement. 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. Alikaniotis, D., Yannakoudakis, H., Rei, M.: Automatic Text Scoring Using Neural Networks. arXiv preprint arXiv: 1606.04289 (2016)
2. Dong F, Zhang Y.: Automatic Features for Essay Scoring. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 968-974 (2016)
3. Kaveh, T., Hwee, T.: A neural approach to automated essay scoring. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1882−1891 (2016)
4. Dong F, Zhang Y, Yang, J.: Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In: Proceedings of the 21st Conference on Computational Natural Language Learning(CoNLL), pp. 153-162 (2017)
5. Tay, Y., Phan, M., Tuan, L., Hui, S.: SkipFlow: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring. arXiv preprint arXiv: 1711. 04981 (2017)
6. Halliday, M. A. K., Hasan, R.: Cohesion in English. Longman (1976)
7. Danielle S. M., Kintsch, W.: Learning from texts: effects of prior knowledge and text coherence. Discourse Processes. 22(3), 247-288 (1996)
8. Ashish V., Noam S., Niki P., Jakob U., Llion J., Aidan N., Łukasz, K., Illia P.: Attention is all you need. In: Neural Information Processing Systems (NIPS), pp. 6000 – 6100 (2017)
9. Pennington, J., Socher, R, Manning, C.: Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543 (2014)
10. Hochreiter S, Schmidhuber J.: Long short-term memory. Neural Computation. 9(8), 1735-1780 (1997)
11. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: Proceedings of International Conference on International Conference on Machine Learning (ICML), pp. 1310-1318 (2013)
12. Dauphin, Y. N., Vries, H. D, Bengio Y.: Equilibrated adaptive learning rates for non-convex optimization. In: Proceedings of International Conference on Neural Information Processing Systems (NIPS), pp. 1504-1512 (2015)

13. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 77 – 81 (2015)
14. Li J., Luong M. T., Jurafsky D.: A Hierarchical Neural Autoencoder for Paragraphs and Documents. arXiv preprint arXiv: 1506.01057 (2015)
15. Ellis Batten Page.: Computer Grading of Student Prose, Using Modern Concepts and Software. Journal of Experimental Education. 62(2): 127-142 (1994)
16. Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. Discourse processes 25(2-3):259–284 (1998)
17. Landauer, T. K., Foltz, P. W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes. 25(2-3): 259-284 (1998)
18. Foltz, P. W., Laham D., Landauer T. K.: Automated Essay Scoring: Applications to Educational Technology. In: Proceedings of EdMedia, pp. 40–64 (1999)
19. Larkey, L. S.: 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)
20. Rudner, L. M.: Automated essay scoring using Bayes' theorem. National Council on Measurement in Education New Orleans La. 1(2):3-21 (2002)
21. Attali Y, Burstein J. Attali, Y., Burstein, J.: Automated essay scoring with e-rater R V. 2.0. ETS Research Report Series, pp. 1–21 (2004)
22. Phandi, P., Chai, K. M. A., Ng, H. T.: 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)
23. Yannakoudakis, H., Medlock, B., Medloc, B.: A new dataset and method for automatically grading ESOL texts In: Proceedings of the 49th Meeting of the Association for Computational Linguistics (ACL), pp. 180−189 (2011)
24. Zhao, S., Zhang, Y., Xiong, X., Botelho, A., Heffernan, N.: A Memory-Augmented Neural Model for Automated Grading. In: Proceedings of the Fourth ACM Conference on Learning at Scale (L@S), pp. 189−192 (2017)
25. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. arXiv preprint arXiv: 1601.06733 (2016)
26. Parikh, A. P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing(EMNLP), pp. 2249–2255 (2016)
27. Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv preprint arXiv: 1703. 03130 (2017)
28. Shen, T., Zhou, T., L, G., Jiang, J., Pan, S., Zhang, C.: Disan: Directional self-attention network for rnn/cnn-free language understanding. arXiv preprint arXiv: 1709. 04696 (2017)
29. Tan, Z., Wang, M., Xie, J., Chen, Y., Shi, X.: Deep semantic role labeling with self-attention. arXiv preprint arXiv: 1712. 01586 (2017)
30. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv: 1705.04304 (2017).