Exploring Relevance and Coherence for Automated Text Scoring using Multi-task Learning

Yupin Yang
College of Computer Science
Chongqing University
Chongqing, China
yyp@cqu.edu.cn

Jiang Zhong
College of Computer Science
Chongqing University
Chongqing, China
zhongjiang@cqu.edu.cn

Chen Wang
College of Computer Science
Chongqing University
Chongqing, China
chenwang@cqu.edu.cn

Qing Li
School of computer science
Northwestern Polytechnical University
Xi’an, China
qingli@nwpu.edu.cn

***Abstract*—With the explosive growth of the information on the Internet, the evaluation of the quality and credibility of web content has become more important than ever before. In this work, we focus on the quality assessment of texts. Recently, various methods have been proposed for the automated text scoring task and obtained competitive results. However, few studies have focused on both relevance and coherence, which are two important factors in evaluating text quality. To improve the scoring task, we propose two auxiliary tasks using negative sampling and integrate them into a multi-task learning framework. The first auxiliary task is relevance modeling and the other one is coherence modeling. We evaluate our model on the Automated Student Assessment Prize (ASAP) dataset. Experimental results show that our model achieves higher Quadratic Weighted Kappa (QWK) scores with an improvement of 1.5% on average.**

***Keywords*—automated essay scoring, multi-task learning, natural language processing**

I. Introduction

Web2.0 accelerates the transition to Web3.0, and global data storage presents explosive growth. The Internet is flooded with all kinds of information. Organizations with poor website quality or inefficient service may establish a bad image and weaken the status of the organization. Therefore, it is necessary to develop effective Web content quality control. Research shows that if visitors find the site pleasant, they are more likely to visit the site again. Accordingly, we explore utilizing neural network models to assess the quality of texts.

Automated text scoring (ATS) aims to predict scores related to the quality of a text. Evaluating texts is time-consuming, and different evaluators may grade different scores to the same text. In this case, the computer-based automatic text scoring system can effectively overcome the inadequacies of manual scoring [1]. Typically, researchers use a combination of Natural Language Processing (NLP) and machine learning to perform this task.

Existing ATS models can be divided into two types: feature engineering-based and neural networks-based methods. The approach based on feature engineering uses handcrafted features (e.g., text length) to score texts. For example, the Enhanced AI Scoring Engine (EASE) is a typical model that has been shown to work well [2]. These models are highly interpretable but require additional engineering. To address the issues, the neural networks-based approach automatically extracts features (e.g., lexical features) from texts for the final grading. Recently, these approaches have achieved high performance. For example, Taghipour and Ng [3] innovatively used Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) to learn text features. Dong et al. [4] adopted Recurrent Convolutional Neural Network (RCNN) with an attention mechanism to model this task and grade the texts automatically.

The automated text scoring task is usually evaluated on the ASAP dataset. However, few studies have focused on the relevance of essays and prompts as well as the coherence of sentences in the essay. Note that the prompt refers to the topic of the essay, which usually consists of reading materials and task descriptions. Relevance is how an essay fits the topic and coherence is what makes multi-sentences of text logical and syntactical. To capture the relevance between the essay content and the prompt, Chen et al. [5] incorporated the similarity between the essay and the prompt into the final representation to grade the text. To further obtain the relevance between each sentence and the source materials, Zhang et al. [6] introduced the co-attention based neural network to model the similarity between them. Besides, the coherence of sentences in the essay is also one of the important criteria for scoring. To our knowledge, a well-written essay is more coherent than a random combination of sentences. Based on this point, Mesgar et al. [7] introduced a local coherence model to obtain the flow of content that semantically connects adjacent sentences in the essay. Li et al. [8] employed the self-attention mechanism to learn the relationship between long-distance words in the essay

DOI reference number: 10.18293/SEKE2022-024

to estimate coherence scores.

In this work, we propose a multi-task learning framework for automated text scoring. In our framework, two auxiliary tasks are introduced including relevance modeling task and coherence modeling task. The relevance modeling task aims to enhance the ability to extract prompt-specific features. Specifically, we mix all the essays under different prompts together and feed them into the model, and then predict which prompt the essay belongs to. The coherence modeling task aims at enhancing the ability to capture the discourse coherence of essays. In our model, this task can be regarded as a binary classification task. We randomly select three consecutive sentences as a group and make some modifications to the group-based data to construct negative samples, and then predict whether the essay is coherent or not. Finally, we integrate these two auxiliary tasks with the scoring task into a multi-task learning framework for final scoring.

To verify the effectiveness of our multi-task learning framework, we conduct experiments on the ASAP dataset. Experimental results show that our method achieves better performance than previous methods, which demonstrates that our proposed method is effective for automated text scoring. Our work also shows that auxiliary tasks can enhance the performance of the BERT model on downstream tasks.

II. Related Work

The discussion of related work is divided into two subsections: ATS-related and MTL-related. In our model, we apply multi-task learning to the task of automated text scoring. There is a long history of automated essay scoring and multi-task learning, and in this chapter, we concisely review some common methods.

A. Automated Essay Scoring

Automated essay scoring is an important application of natural language processing (NLP) in education. Previous methods were mainly based on feature engineering, in which the ATS task was considered as a classification or regression problem. In the case of the former, the classifier directly outputs the label that represents the score. In the latter case, the output is in the range of the golden score. The feature-engineering methods require handcrafted features, which include some statistical features such as essay length, number of spelling errors, etc. These approaches include e-rater [9], PEG [10], and EASE [2]. In the PEG, more than thirty writing quality factors are considered. Besides, Cozma [11] combined string kernels and word embeddings to capture text features, namely the bag-of-super-word-embeddings (BOSWE). Dascalu et al. [12] implemented an automated essay scoring system for Dutch by integrating features such as lexical and semantics features.

To avoid the need for feature engineering, researchers begin to explore the application of neural networks to the automated essay scoring task. Taghipour and Ng [3] innovatively combined CNN and Long Short-Term Memory (LSTM) to learn text representations for final scoring and obtained competitive results. Dong et al. [4] adopted ConvNet and LSTM to learn sentence representation and text representation respectively. Mesgar et al. [7] employed the RNN layer for the words in the sentence to integrate contextual information. Yang et al. [13] introduced the BERT model to learn text representations.

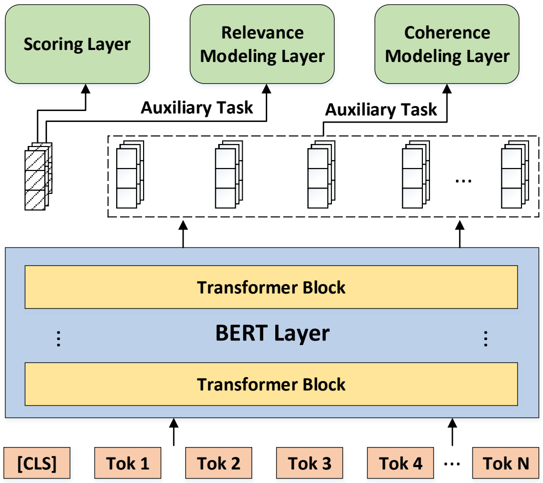

Figure. 1. An overview of our multi-task learning architecture. It first adopts BERT as the shared encoder. Then, three task-specific layers are connected behind the encoder, sharing the text representation learned from the BERT layer.

B. Multi-task Learning

Multi-task learning was firstly proposed in 1994 by Caruana [14] to improve the generalization ability of the model. It is an inductive transfer mechanism that shares parameter information between multiple tasks [15]. Compared with single-task learning, multi-task learning refers to learning multiple tasks simultaneously, which can achieve better performance. In multi-task learning, the main task and auxiliary tasks learn from each other and jointly enhance the generalization ability [16]. For auxiliary tasks, a basic assumption is that auxiliary tasks should be related to the main task and can promote the learning of the main task.

Multi-task learning has been widely used across applications of machine learning, from natural language processing [17] and speech recognition [18] to computer vision [19]. Liu et al. [20] introduced Two-Stage Learning Framework for ATS where semantic score, coherence score, and prompt-relevant score are computed at the first stage and they are combined with handcrafted features in the second stage. Nadeem et al. [21] used natural language inference and discourse marker prediction as auxiliary tasks for capturing discourse characteristics of essays.

III. Method

In this section, we demonstrate the main steps of our proposed model. It consists of a shared encoder and three task-specific layers including the scoring task, relevance modeling task, and coherence modeling task. The overview of our proposed model is shown in Fig.1. In the following subsections, more details of each module will be introduced.

A. Shared Encoder

Large pre-trained language models (e.g., BERT [22], GPT [23], and XLNet [24]) have shown the remarkable ability of representation and generalization in many tasks. These pre-trained language models achieve great success in learning text representations with deep semantics. In our framework, we choose BERT as the shared encoder to better capture the semantics of the given essay.

BERT is trained on enormous corpora with more than 3000M words. It has two target tasks, including the masked language model and next sentence prediction. Many NLP downstream tasks, such as sentence classification and question answering, have gained benefits by utilizing pre-trained BERT to learn text representation. Specifically, we adopt RoBERTa [25] as the encoder to get better performance. The self-attention mechanism [26] is the key to the success of BERT, in which a sequence calculates the word weights with itself. The attention process is defined as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where Q, K, and V are the matrix transformed from the input. d_k denotes the hidden dimension of matrix K.

Given an input essay $E = \{w_1, w_2, ..., w_N\}$, where N is the length of the essay, we add a special token [CLS] at the beginning of the sequence and get a new sequence $E' = \{[CLS], w_1, w_2, ..., w_N\}$. Then, we send the sequence E' into a pre-trained BERT and take the hidden state of the [CLS] token as the text representation:

$$H = BERT(E') \tag{2}$$

$$R = H_{[CLS]} \tag{3}$$

where H denotes the final hidden state sequence of BERT, and R is the text representation.

B. The Scoring Task

The main task of our method is the scoring task, which aims to predict a score for each essay. Following other ATS methods [3], [20], [21], [27], we utilize a dense layer to compute the score with the text representation R as:

$$p_s = sigmoid(W_s R + b_s) \tag{4}$$

where W_s is the weight matrix, b_s is the bias initialized with the mean score of the essays, and sigmoid is the activation function to normalize the calculated score into [0,1]. In this part, we select mean square error (MSE) as the loss function:

$$L_S = \frac{1}{m}\sum_{i=1}^{m}(y_s - p_s)^2 \tag{5}$$

where y_s is the true target, and p_s is the prediction result for the scoring task.

C. Relevance Modeling Task

To our knowledge, there is a close relevance between the essay content and the topics. ATS systems may give a high score to an unrelated but well-written essay. However, a human rater will give higher scores to those essays related to the topics, and lower scores to those essays that are not relevant to the topics. To exploit the prompt-specific knowledge, we design this auxiliary task named relevance modeling. Since high-scoring essays always stick to the prompt, we first mix up the top 40% essays of all prompts, and their labels are the prompt they belong to. After that, we feed the latent text representation R learned from BERT into a dense layer to predict the prompt:

$$p_{rm} = softmax(W_{rm} R + b_{rm}) \tag{6}$$

where W_{rm} is the weight matrix, b_{rm} is the bias, and softmax is the activation function for the multi-classifier. In this module, we optimize this auxiliary task with the cross-entropy loss as:

$$L_{RM} = -\sum y_{rm} * log(p_{rm}) \tag{7}$$

where y_{rm} is the true target, and p_{rm} is the prediction result.

D. Coherence Modeling Task

Language learners may have learned to make both meaningful and grammatical sentences but do not know how to organize the sentences together to construct a good essay. Coherence is what makes a multi-sentence text meaningful, both logically and syntactically. In this work, our coherence modeling task is used to better capture the discourse coherence of essays.

In this task, we regard the coherence modeling task as a binary classification task. The vanilla essay is with the label ”1” while the others are with the label ”0”. To construct negative samples, we perform three operations on the essays. Note that each negative sample is prompt-related but incoherent. For an input essay, we randomly select three consecutive sentences as a group. There are three ways to construct negative samples: 1) delete operation—removing the selected group; 2) replace operation—replacing the selected group with a new group from another essay of the same topic; 3) inserting operation—appending a group from another essay under the same prompt to the beginning or end of the selected group. For example, given an essay E consisting of k sentences $E = \{s_1, ..., s_{i-1}, s_i, s_{i+1}, s_{i+2}, s_{i+3}, ..., s_k\}$, the one with delete operation is formed as $E^* = \{s_1, ..., s_{i-1}, s_{i+3}, ..., s_k\}$. The one with replace operation is denoted as $E^* = \{s_1, ..., s_{i-1}, s_j^{'}, s_{j+1}^{'}, s_{j+2}^{'}, s_{i+3}, ..., s_k\}$. The one with inserting operation can be $E^* = \{s_1, ..., s_{i-1}, s_i^{'}, s_{i+1}^{'}, s_{i+2}^{'}, s_i, s_{i+1}, s_{i+2}, s_{i+3}, ..., s_k\}$ or $E^* = \{s_1, ..., s_{i-1}, s_i, s_{i+1}, s_{i+2}, s_j^{'}, s_{j+1}^{'}, s_{j+2}^{'}, s_{i+3}, ..., s_k\}$. After that, we send negative samples and positive samples into the model for training. Moreover, to reduce the deviation caused by imbalanced samples, we mix the positive sample and the negative sample evenly. Our approach is based on the

TABLE I
STATISTICS OF THE ASAP DATASET.

Prompt	Essay Type	#Essays	Avg length	Scores
1	Argumentative	1,783	350	2-12
2	Argumentative	1,800	350	1-6
3	Source-Dependent	1,726	150	0-3
4	Source-Dependent	1,772	150	0–3
5	Source-Dependent	1,805	150	0–4
6	Source-Dependent	1,800	150	0–4
7	Narrative	1,569	250	0–30
8	Narrative	723	650	0–60

assumption verified in Lin et al.'s work [28] that the original article is always more coherent than the changed one.

In this module, we feed the hidden states H obtained from BERT into a Bi-LSTM network to model the semantic relationships among sentences:

$$h_t = \text{Bi-LSTM}(h_{t-1}, H_i) \tag{8}$$

where H_i is the i-th output of the BERT layer and h_t is the hidden state of the Bi-LSTM at time t. We concatenate the forward and backward output together and obtain the last hidden state h_N. Then, a fully connected layer is adopted to predict whether the essay is coherent or not:

$$p_{cm} = sigmoid(W_{cm}h_N + b_{cm}) \tag{9}$$

where W_{cm} is the weight matrix, b_{cm} is the bias, and sigmoid denotes the activation function for the binary classification task. We then train this task with the binary cross-entropy loss as:

$$L_{CM} = -y_{cm} * log(p_{cm}) - (1 - y_{cm}) * log(1 - p_{cm}) \tag{10}$$

where y_{cm} is the true target and p_{cm} is the prediction result.

While training, we alternatively optimize the scoring task with L_S in Equation (5), the relevance modeling task with L_{RM} in Equation (7), and the coherence modeling task with L_{CM} in Equation (10). The 'mix ratio' of the three tasks is set as $\lambda_S : \lambda_{RM} : \lambda_{CM} = 0.6 : 0.2 : 0.2$.

IV. EXPERIMENT

In this section, we introduce the ASAP dataset and experiment settings firstly. Then the evaluation metric is illustrated. In addition, baseline models, results of the experiment, and analyses are displayed.

A. Experiment Settings

We use a common dataset for the ATS task, which is from a Kaggle competition. There are 8 prompts of different genres, and the number of essays in the dataset is 12976. In Table I, we list some statistics of the ASAP dataset.

We implement our model using Pytorch and the BERT comes from HuggingFace [29]. Since the average length of the essay in prompt 8 is 650, we truncate the essays with the max length of 512 words. For the BERT model, we use the uncased $BERT_{base}$ model with 12 layers, 768 hidden units, and 12 heads. We use the pre-trained parameters and fine-tune the parameters with the learning rate set to 1e-5. Following previous works, we also utilize 5-fold cross-validation to evaluate our model with a 60/20/20 split for train, validation, and test sets.

B. Evaluation Metric

QWK is the official evaluation metric in the ASAP competition, which measures the agreement between ratings assigned by humans and ratings predicted by ATS systems. Following previous works, we adopt QWK as the evaluation metric. The quadratic weight matrix is calculated as follows:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{11}$$

where i and j are gold scores and calculated scores respectively. N is the number of possible ratings. The QWK value is defined as:

$$\kappa = 1 - \frac{\sum W_{i,j}O_{i,j}}{W_{i,j}E_{i,j}} \tag{12}$$

Where O is the observed score matrix and E is the expected score matrix. $O_{i,j}$ denotes the number of essays that receive rating i by human rater and j by ATS system. E is calculated as the outer product of histogram vectors of the two (reference and hypothesis) ratings.

C. Baselines

In this section, we introduce several baseline models. Enhanced AI Scoring Engine (EASE) is a statistical model based on hand-crafted features such as length-based features and part-of-speech tags. After feature extraction, support vector regression (SVR) and bayesian linear ridge regression (BLRR) are used to build the model [2]. Cozma et al. [11] proposed HISK+BOSWE, which combined string kernels and word embeddings to extract text features on both low-level character n-gram features and high-level semantic features. Wang et al. [30] proposed RL1, which is a reinforcement learning framework incorporating quadratic weighted kappa as guidance to optimize the scoring system. Taghipour and Ng [3] proposed to assemble CNN and LSTM. Dong et al. [4] introduced hierarchical neural networks with attention mechanisms to learn the representation of essays. Tay et al. [31] proposed SKIPFLOW LSTM, where there is a mechanism to simulate the relationship between hidden states in the LSTM network during reading, so as to learn the characteristics of text coherence. Yang et al. [13] proposed R^2BERT that utilized a pre-trained language model to get the scores and used mean square error loss and the batch-wise ListNet loss with dynamic weights to constrain the scores simultaneously.

https://www.kaggle.com/c/asap-aes/data
https://github.com/huggingface/transformers

https://github.com/edx/ease

TABLE II

QWK scores of different methods on the ASAP dataset (* denotes statistical model). In this table, RM denotes the relevance modeling task and CM denotes the coherence modeling task.

Methods	Prompt1	Prompt2	Prompt3	Prompt4	Prompt5	Prompt6	Prompt7	Prompt8	Average
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
HISK+BOSWE*	0.845	0.729	0.684	0.829	0.833	0.830	0.804	0.729	0.785
RNN	0.687	0.633	0.552	0.744	0.744	0.757	0.743	0.553	0.675
RL1	0.766	0.659	0.688	0.778	0.805	0.791	0.760	0.545	0.724
LSTM	0.780	0.697	0.683	0.787	0.795	0.767	0.758	0.651	0.740
CNN+LSTM	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
LSTM-CNN-att	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
SKIPFLOW	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.765
R^2BERT	0.817	0.719	0.698	0.845	0.841	0.847	0.839	0.744	0.794
BERT	0.815	0.720	0.730	0.814	0.820	0.824	0.833	0.730	0.786
BERT+CM	0.838	0.731	0.733	0.816	0.826	0.845	0.836	0.742	0.796
BERT+RM	0.831	0.728	0.741	0.838	0.834	0.853	0.829	0.733	0.798
BERT+RM+CM	0.842	0.733	0.746	0.842	0.836	0.857	0.842	0.747	0.806

D. Results

In this part, the performance of the baselines and our method on the ASAP dataset are analyzed in detail. Table II shows the QWK scores of different methods on each prompt.

In general, neural network-based methods have achieved better results than statistical-based methods. Even so, the statistical model HISK+BOSWE gain a better QWK score on Prompt 1, reaching 0.845. To some degree, it shows that when the handcrafted features can adequately represent the information in the original text, better results can be obtained. We notice that EASE performs better than RNN which also shows well-designed handcrafted features are more effective than simple neural networks. Among the methods based on neural networks, the performance of RNN and LSTM is not as good as R^2BERT. This is because that there are hundreds of words making it difficult to learn long-term dependencies. Meanwhile, we observe that CNN+LSTM, SKIPFLOW, and LSTM-CNN-att outperform LSTM models, which means that the ensemble model can make up for the shortage of simple neural networks. Additionally, BERT based model outperforms all other neural models on the average QWK score, which indicates the pre-trained language model does well in capturing deep semantic features.

Compared with other baseline models, the average QWK scores on eight prompts show that our model achieves the best results. Our model outperforms R^2BERT by 1.5% in the average QWK score. The result demonstrates that through the multi-task learning framework, our model can capture more coherence and relevance information for the final score evaluation. On Prompt 4 and 5, R^2BERT achieves higher results, indicating that combining complementary objectives via dynamic weights can effectively enhance the performance of the scoring system. In particular, BERT with auxiliary tasks outperforms R^2BERT on each prompt except Prompt 4 and 5, which shows the effectiveness of the auxiliary tasks and the success in improving the performance of BERT on downstream tasks. Overall, BERT+RM+CM gains a higher average QWK score compared with the aforementioned neural models as well as the latest statistical model HISK+BOSWE.

E. Discussion

To further verify the effectiveness of our proposed auxiliary tasks, we conduct ablation experiments with different settings. The relevant results are illustrated in Table II.

We can see that both the auxiliary tasks improve the automated essay scoring performance remarkably. Compared with the baseline BERT, employing the coherence modeling task (BERT+CM) yields a result of 0.796 in averaged QWK score, which brings a 1.3% improvement. Meanwhile, employing the relevance modeling task (BERT+RM) individually outperforms the baseline by 1.5%. The two tasks behave differently on different prompts. The RM task performs better on Prompt 3 to Prompt 6, while the CM task does on the others. The results may be due to differences in the genre and guidelines of the essay. For example, In Prompt 5, students were asked to describe the mood created by the author in the memoir and use the relevant information in the source material to support the answer. Therefore, for Prompt 5, a high-scoring essay is expected to contain specific information about the memoir, and the details of the memoir mentioned in the written essay are more important than the coherence of the sentence. For Prompts 7 and 8, the type of essay is narrative. The guidelines for these two prompts require human raters to give the highest score to essays that are coherent and engage the reader's attention through telling a story. Accordingly, the CM task shows better performance as it captures the sequence of semantic changes. When we integrate the two auxiliary tasks together (BERT+RM+CM), the performance further improves to 0.806 in the QWK score on average. It is obvious that these two auxiliary tasks have brought great benefits to the scoring task.

V. Conclusion

In this work, we introduce an approach based on two auxiliary tasks to assess the quality of texts. We integrate the auxiliary tasks into a multi-task learning framework to benefit the scoring task. To verify the effectiveness of our proposed method, we compare our model with several methods on the ASAP dataset. Experimental results show that our model outperforms previous methods. Our work also shows that auxiliary tasks can enhance the performance of the BERT model for downstream tasks. For future work, we plan to explore more dimensions for the automated text scoring task.

Acknowledgment

The authors acknowledge National Natural Science Foundation of China (Grant No: 62176029), the Key Research Program of Chongqing Science and Technology Bureau (cstc2020jscx-msxmX0149), and Graduate Research and Innovation Foundation of Chongqing, China (Grant No.CYS21061). This work is also supported by the National Natural Science Foundation of China under Grant 62102316, in part by the NWPU Development Strategy Research Fund Project Grant 2022FZY16. This work is also supported by National Natural Science Foundation of China under Grant No. 62002226 and Zhejiang Provincial Natural Science Foundation of China under Grant No.LHQ20F020001.

References

[1] J. G. Borade and L. D. Netak, "Automated grading of essays: A review," in *International Conference on Intelligent Human Computer Interaction*. Springer, 2020, pp. 238–249.

[2] P. Phandi, K. M. A. Chai, and 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*, 2015, pp. 431–439.

[3] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1882–1891.

[4] F. Dong, Y. Zhang, and J. Yang, "Attention-based recurrent convolutional neural network for automatic essay scoring," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 2017, pp. 153–162.

[5] M. Chen and X. Li, "Relevance-based automated essay scoring via hierarchical recurrent model," in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 378–383.

[6] H. Zhang and D. Litman, "Co-attention based neural network for source-dependent essay scoring," *arXiv preprint arXiv:1908.01993*, 2019.

[7] M. Mesgar and M. Strube, "A neural local coherence model for text quality assessment," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4328–4339.

[8] X. Li, M. Chen, J. Nie, Z. Liu, Z. Feng, and Y. Cai, "Coherence-based automated essay scoring using self-attention," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 2018, pp. 386–397.

[9] M. Chodorow and J. Burstein, "Beyond essay length: evaluating e-rater's performance on tofel essays," *ETS Research Report Series*, vol. 2004, no. 1, pp. i–38, 2004.

[10] M. D. Shermis and J. C. Burstein, *Automated essay scoring: A cross-disciplinary perspective*. Routledge, 2003.

[11] M. Cozma, A. Butnaru, and R. T. Ionescu, "Automated essay scoring with string kernels and word embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 503–509.

[12] M. Dascalu, W. Westera, S. Ruseti, S. Trausan-Matu, and H. Kurvers, "Readerbench learns dutch: building a comprehensive automated essay scoring system for dutch language," in *International Conference on Artificial Intelligence in Education*. Springer, 2017, pp. 52–63.

[13] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "Enhancing automated essay scoring performance via cohesion measurement and combination of regression and ranking," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 1560–1569.

[14] R. Caruana, "Learning many related tasks at the same time with backpropagation," in *Advances in neural information processing systems*, 1995, pp. 657–664.

[15] ——, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[16] S. Thrun and J. O'Sullivan, "Discovering structure in multiple learning tasks: The tc algorithm," in *ICML*, vol. 96, 1996, pp. 489–497.

[17] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[18] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8599–8603.

[19] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[20] J. Liu, Y. Xu, and Y. Zhu, "Automated essay scoring based on two-stage learning," *arXiv preprint arXiv:1901.07744*, 2019.

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

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[23] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[27] Y. Farag, H. Yannakoudakis, and T. Briscoe, "Neural automated essay scoring and coherence modeling for adversarially crafted input," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 263–271.

[28] Z. Lin, H. T. Ng, and M.-Y. Kan, "Automatically evaluating text coherence using discourse relations," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 997–1006.

[29] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[30] Y. Wang, Z. Wei, Y. Zhou, and X.-J. Huang, "Automatic essay scoring incorporating rating schema via reinforcement learning," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 791–797.

[31] Y. Tay, M. Phan, L. A. Tuan, and S. C. Hui, "Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.