Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/52/20241321

Effectiveness of finetuning pretrained BERT and deBERTa for automatic essay scoring

Wentao Zhong

College of Computer Science and Cyber Security Oxford Brookes College, Chengdu University of Technology, Chengdu, Sichuan, 610059, China

zhong.wentao@student.zy.cdut.edu.cn

Abstract. With the growing importance of summary writing skills in the educational system and the inherent complexity of manual assessment, there is an urgent need for automated summary scoring solutions. Pre-trained models are popular nowadays, such as Bidirectional Encoder Representations from Transformers (BERT) and Decoding enhanced BERT with disentangled attention (deBERTa). The performance of direct use with trained models on specific tasks still needs to be improved. This paper focuses on the impact on the performance of summary scoring systems after adding linear and dropout layers to these pre-trained models for feature extraction and dimensionality reduction operations. The paper details the optimization for the particular task of summary scoring automation after using the pre-trained models. This paper focuses on adding linear and dropout layers to perform feature extraction and dimensionality reduction operations. The aim is to make the model more adaptable to this specific educational task. Ultimately, it is hoped that these studies will enhance the pedagogical toolkit for educators and enrich the academic experience for students.

Keywords: automatic essay scoring, BERT, transformer.

1. Introduction

Summary Writing plays an important role in developing reading comprehension, critical thinking, and improving writing skills [1]. This ability is especially critical for second language learners and students with learning disabilities. In traditional academic settings, the evaluation of written summaries relies heavily on manual judgment. As online education platforms continue to grow putting pressure on education practitioners. The ratio of online students to teachers has risen significantly. Digital education is increasingly becoming the norm. The shortcomings of traditional assessment methods have also become apparent. There is therefore an urgent need to rethink grading methods and seek innovative, scalable, and efficient solutions for assessing student-written summaries.

Traditional scoring systems often involve multiple raters. And the scoring criteria are often inconsistent among them. This may lead to unnecessary subjective bias and jeopardize the objectivity of academic evaluation [2,3]. It has become essential to create automatic summary grading systems in order to solve this issue. By providing standardized and consistent ratings, these systems not only ensure fairness for students, but also reduce the workload of educational practitioners and create a more efficient and fairer environment for academic evaluation [4].

DOI: 10.54254/2755-2721/52/20241321

Currently, most automated evaluation systems remain challenging to correctly assess essay answers. For this reason, a robust automatic essay scoring (AES) system is proposed. For this specific task. The performance of large language models such as Bidirectional Encoder Representations from Transformers (BERT) and Decoding enhanced BERT with disentangled attention (DeBERTa) is improved by enhancing feature extraction and noise reduction operations [5,6]. Specifically, linear and dropout layers are added to these models for better feature extraction and dimension reduction, to improving the effectiveness of the original models [7,8].

In this paper, the AES dataset from the Kaggle website is used with the aim of scoring summaries based on the given contextual information and questions by means of a context score and a wording score, taking into account the important information in the sentence part. To achieve this goal, BERT and DeBERTa models are used on the dataset to automatically extract features and reduce dimensionality by adding linear and dropout layers. Absolute location information is also considered to score summaries more accurately. Next, we discuss and test the performance of the BERT and DeBERTa models with the addition of Dropout and Linear layers to the AES task, as well as the performance of the original model, and present experimental results.

In Section 2, the approach is described. In Section 3, the experiment report and result comparison are illustrated. Finally, Section 4 will provide a summary of this paper's key findings and conclusions.

2. Method

2.1. BERT

BERT comes from the Transformer's bidirectional encoder representation and represents a paradigm shift in Natural Language Processing(NLP). Unlike the original Transformer model, BERT uses only the encoder mechanism. This design reduces the number of parameters and improves computational efficiency, making it suitable for tasks that are either tagged by sequence or categorized by sequence, such as text categorization and sentiment analysis. For these, the encoder mechanism is sufficient to map the input text into a high-dimensional space that captures the contextual relationships between words. In addition, the design of the encoder allows parallel processing of all input tokens, accelerating training and inference.

Masked Language Modeling (MLM) is an approach used in BERT pre-training that enables BERT to learn the contextual representation of text by randomly masking some of the tokens in the input text and allowing the model to predict the masked tokens.

Next Sentence Prediction (NSP) is trained to predict whether a sentence logically follows another sentence, so as to understand the relationship between sentences.

For example, each sample consists of two sentences A and B. There are two cases: the first case is that sentence B is indeed the next sentence of sentence A, and the sample is labeled as being next. The second case is that sentence B is not the next sentence of sentence A. Sentence B is some other random sentence in the corpus, and the sample is labeled as not next. In the sample set, there is a 50/50 split between the two cases.

The bi-directional training of BERT and these specific pre-training tasks enable it to learn rich contextual representations that can be very effective for a variety of downstream tasks such as summarizing ratings.

2.2. DeBERTa

DeBERTa is an evolution of the original BERT model. It provides key advances in both architecture and functionality to address specific limitations and improve natural language understanding. DeBERTa introduces the mechanism of disentangled attention to better capture different types of dependencies between tokens in a sentence.

DeBERTa uses a vector of content and a vector of position to represent each word. In addition, DeBERTa calculates the attentional weights between words according to their content and relative position by using a disentangle matrix.

DOI: 10.54254/2755-2721/52/20241321

This allows the model to focus on aspects of context and relationships more efficiently. The disentangled attention is computed as:

$$Attention = Softmax(\frac{Q_{content}K_{content}^{T}+Q_{position}K_{position}^{T}}{\sqrt{d_k}}) \quad (1)$$

where Q denotes query, K denotes key and V denotes value matrix, d_k is the dimensionality of the keys.

This is especially important for tasks involving the generation of coherent text. While there is no single formula that captures all decoding enhancements, they aim to optimize decoding losses as well as raw masking language modeling losses.

Meanwhile, DeBERTa uses cross-layer parameter sharing and can be represented as:

$$h_{l,i} = h_{l-1,i} + FFN(MultiHeadAttention(h_{l-1,i}, h_{l-1})) \quad (2)$$

Allow information to flow more freely between layers. This helps to improve the dissemination of contextual information through the model.

Compared to standard BERT, DeBERTa has improved performance on a variety of benchmarking and natural language understanding tasks. It is also capable of being able to handle a wide range of natural language understanding tasks without major modifications to the specific task. Its enhancements are aimed at improving the overall quality of the representations for various tasks. These improvements stem from its disentangled attention, decoding enhancements, and other architectural changes. As for AES systems, DeBERTa-based systems can take into account more word position information, which would be more effective in scoring summaries in terms of wording or syntactic dimensions outside the context.

2.3. Feature Extraction and Dimension Reduction

Many pre-trained language models, usually on large-scale texts, have been pre-trained. These pre-trained models, however, may not be suitable for specific tasks. Such as text categorization or regression. Therefore, the model is tuned to perform specific tasks by adding additional layers after the model. For example, in this paper the addition of linear and dropout layers are used for feature extraction and dimensionality reduction. They can map the high-dimensional BERT and deBERTa hidden states to a lower dimensional representation space and enable the model to perform better in capturing the relevant features of the task. The use of dropout layers helps mitigate overfitting problems. This flexibility allows the use of pre-trained models in different application domains and fine-tuning to obtain better performance than the original model. The main layers added in this experiment are described next [9].

2.3.1. Linear Layer. Linear Layer, also named as Fully Connected Layer or Dense Layer, is mainly used for linear transformation and feature mapping. The output of the linear layer can be considered as a transformation or mapping of the input features. By adjusting the weight matrix and bias terms, the Linear Layer can learn the relationship between different features to generate a new feature representation. The linear layer can also usually map the input features from one dimension to another mapping to another dimension. This is particularly useful for tuning the dimensionality of the feature representation of a neural network. The linear layer multiplies the input tensor with the weight matrix and adds a bias term to achieve a linear transformation. The formula is expressed as:

$$output = input \times W + b \quad (3)$$

where W is a learnable parameter of the linear layer that determines how the input features are linearly combined. The bias term is a class-learned offset that helps the model fit the data.

DOI: 10.54254/2755-2721/52/20241321

2.3.2. ReLU Layer. ReLU is fully known as Rectified Linear Unit. it is usually used in the hidden layer of neural networks to map non-linear properties so that the network can learn complex patterns and features better. It is capable of modeling complex linear relationships. And it can effectively mitigate the problem of gradient vanishing. In return propagation when the input is positive, the gradient of ReLU is 1. The hard spiked gradient information can be effectively propagated back to the earlier layers of the network, which helps to speed up the training process. Also when the ReLU output is non-zero for positive inputs and 0 for negative inputs. this means that the ReLU activation neurons are fine and only activate when the input is greater than 0. This helps to reduce the redundancy of the neural network. This helps to reduce the redundancy of the neural network. Also the calculation of ReLU is very simple with the following formula:

$$ReLU(x) = \max(0, x) \tag{4}$$

Only one max function operation is designed here, so the computation is fast.

2.3.3. Dropout. Dropout is a technique used to regularize deep neural networks, which mainly refers to mitigating the problem of overfitting by randomly show gassing a portion of neurons during the training process of a neural network. This random dropout operation is achieved by turning off neurons with a certain probability during the training iterations. Usually, the probability will be this only between 0.2 and 0.5 to set the output to zero. Dropout does not come since any individual neuron during training, thus increasing the diversity, robustness, and generalization ability of the model. It effectively avoids the network from overfitting the training data and improves the adaptability to new data. Suppose the output of the neuron input to the Dropout layer is x with probability p. Then the Dropout layer operates as follows:

$$output = \begin{cases} x & with\ probability\ 1-p \\ 0 & with\ probability\ p \end{cases} \tag{5}$$

For each neuron in each training iteration, its output is retained with a probability of 1-p, and its output is set to zero with a probability of p. The model is designed to be more stable by using a dropout during training.

Reduce the complexity of the model by using Dropout during training and then turn dropout off during the inference or testing phase to obtain more stable predictions.

3. Experiments

3.1. Experimental Setup

BERT and DeBERTa were used as the base models for training in the experiments. The BERT and DeBERTa models with added linear and dropout layers are also used for training. The training dataset used is from kaggle. The CommonLit "Evaluate Student summaries" competition. The evaluation metric is Mean Squared Error (MSE).

The hardware and software used for the experiments were Pytorch 2.0, python 3.9, NVIDIA RTX 4090 GPU, and I9-13900k CPU.

For each setting, the maximum training calendar time was set to 100 and an early stopping mechanism was applied. We terminated the training process when the evaluation metrics stopped improving within 6 calendar elements. Optimization of the model was done by adding a linear and dropout layer at the bottom of the base model with a Dropout rate of 0.5. and using ReLU as the activation function.

3.2. Dataset

The dataset used in this study was obtained from the CommonLit "Evaluate Student summaries" competition, hosted by Kaggle [10]. Each sample in the dataset consists of summary text written by

DOI: 10.54254/2755-2721/52/20241321

students in response to a specific prompt. That is, the article to be summarized. Along with the summary text, two scores are provided for evaluation: a content score and a wording score. Content and Wording. Most of the texts in the dataset are within 512 words, which is consistent with the maximum input length of the BERT model. To ensure stratified sampling between different cues, the dataset is split into two parts where the training is 80% and validation is 20%, the number of samples are shown in Table 1.

Table 1. Number of training examples and validation examples.

	Training	Validation
Number	5732	1433

3.3. Evaluation Metrics

The Mean Squared Error (MSE) was used as an evaluation metric for the model. The MSE was defined as the average of the squared differences between the predicted and actual values of the mean predicted and actual values.

Given a dataset containing n observations. Assuming that y_i is the true value of the i-th observation and $\hat{y}_i$ is the value predicted by the model, MSE is defined as:

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

When the MSE is 0, it means that the prediction is perfect. The smaller the value of MSE, the better the predictive ability of the model. The MSE is positive because each term is squared. The MSE gives higher weight to larger errors because the error term is squared. For the two-label regression, the average MSE of the content and wording scores was used as the measure.

3.4. Comparison of Modeling Results With and Without Linear and Dropout Layers

In this section, two different pre-trained models are used to simultaneously compare their effectiveness in terms of predicting content and wording scores in their original state versus with the addition of linear and dropout layers. By doing so, the aim is to see if the linear and dropout layers affect the regression performance for a given task in different frameworks. The performances are shown in Table 2.

Two pre-trained models from HuggingFace are presented here:

BERT-base-cased: A version of BERT that retains case information, trained on English text with differentiation between uppercase and lowercase letters. The results are shown in Figure 1 and Figure 2 respectively. Vocabulary size: 28,996, Hidden state size: 768.

deBERTa-v3-base: A base version of DeBERTa, enhancing BERT architecture with disentangled attention mechanisms. The results are shown in Figure 3 and Figure 4 respectively. Vocabulary size: 32,768, Hidden state size: 768.

DOI: 10.54254/2755-2721/52/20241321

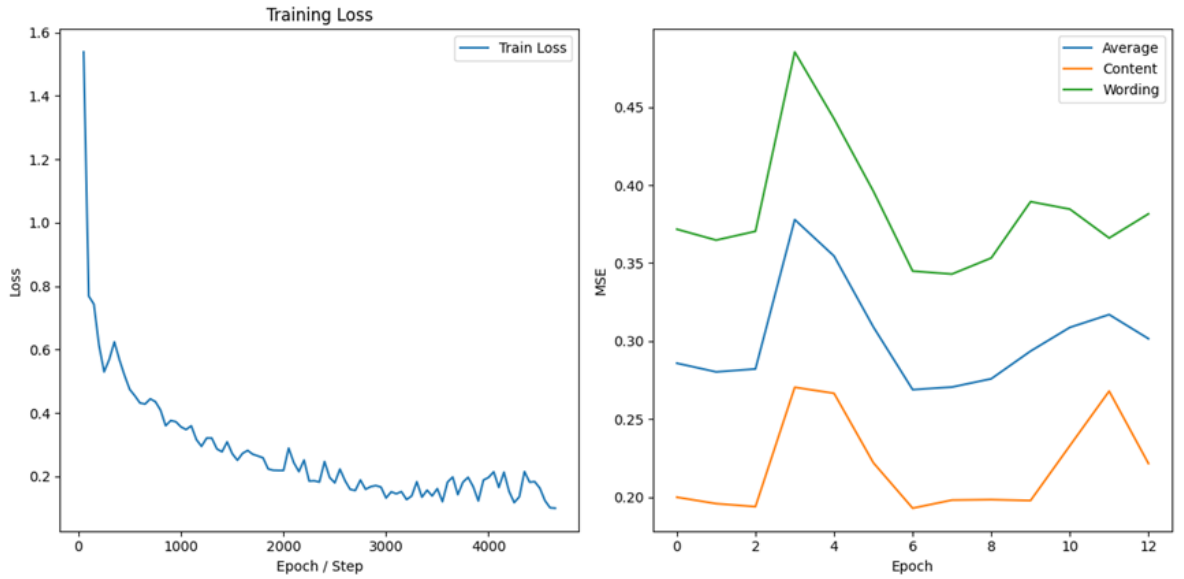

Figure 1. Performance of BERT-base-cased.

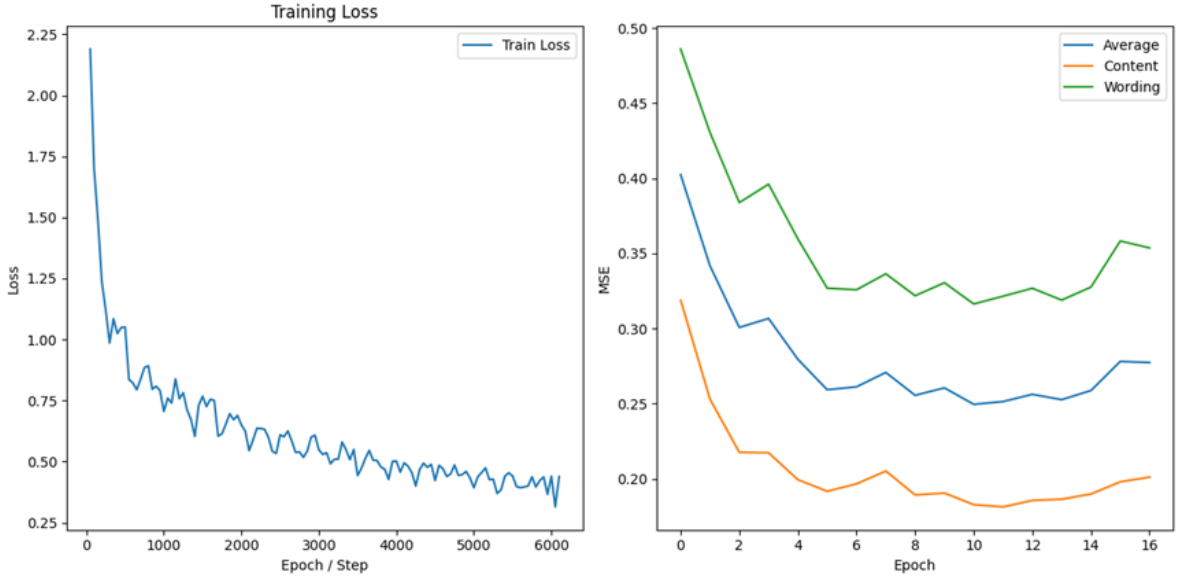

Figure 2. Performance of BERT-base-cased with Linear and dropout layers.

DOI: 10.54254/2755-2721/52/20241321

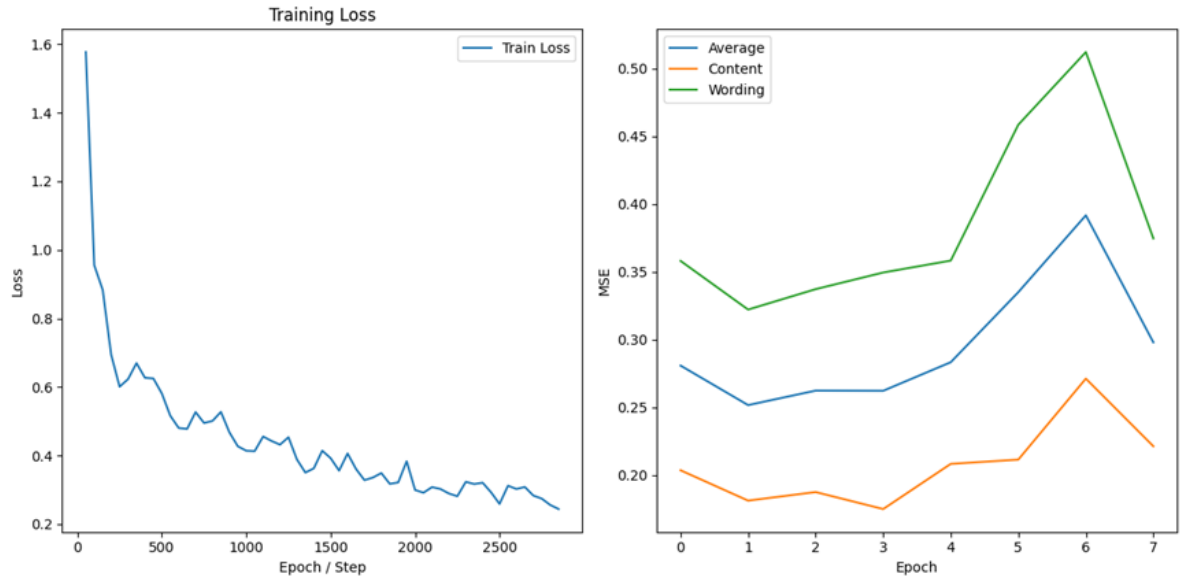

Figure 3. Performance of deBERTta-V3-base.

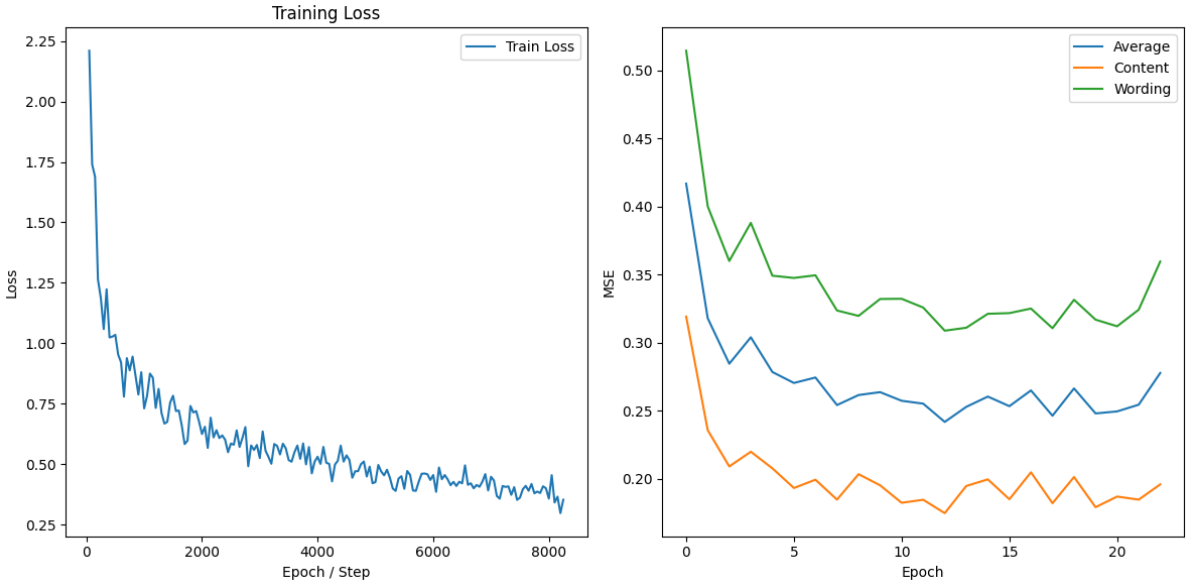

Figure 4. Performance of deBERTa-v3-base with Linear and dropout layers.

Table 2. MSE of four models.

	Content	Wording	Average
bert-base-cased	0.2215	0.3815	0.3015
deberta-v3-base	0.2212	0.3745	0.2979
bert-base-cased with Linear and dropout	0.1864	0.3189	0.2527
deberta-v3-base with Linear and dropout	0.1790	0.3168	0.2479

DOI: 10.54254/2755-2721/52/20241321

3.5. Analysis

Based on the above results, we can observe that the addition of the linear and dropout layers positively affects the performance of deberta-v3-base, especially the best performance in terms of Mean Squared Error (MSE) for content and wording prediction. This indicates that the linear and dropout layers play a good role in feature extraction and dimensionality reduction. On the other hand, for the bert-base-cased model, the addition of linear and dropout layers significantly outperforms the unmodified version. In addition, the DeBERTa model with a distractor mechanism is compared to the BERT model, and although there is a slight difference, the difference is not enough to clearly show that one of the models outperforms the other. Although both models have similar hidden state sizes, DeBERTa may be responsible for the better performance in terms of vocabulary size and enhanced architecture. Indeed, in this case, DeBERTa seems to be the best choice for content and wording prediction, although with the addition of the linear and dropout layers, it also outperforms the BERT model with the same modifications. However, it is important to note that these results may be affected differently by variations in hyperparameters and additional preprocessing, providing interesting directions for future research.

4. Conclusion

The experiments in this study found that after adding linear and dropout layers for in the pre-trained model, the model has a positive impact on the summarization grading task. After performing feature extraction and noise reduction operations on the pre-trained models like BERT and deBERTa, both have positive impact on the MSE values compared to for the modified version. The adjusted deberta-v3-base has an MSE value of 0.2479, which is better than most of the RNN-based methods used in the CommonLit "Evaluate Student summaries" competition. Also when comparing the different base models, deBERTa was found to be better than BERT due to its distraction mechanism. However, there are still many aspects to be further investigated, different preprocessing schemes, tuning of hyperparameters, use of different linear layers and other advanced model architectures can be explored in more depth in terms of the potential of AES. In addition, the dataset provides four hints, such as how to include long texts of more than 5000 words in the grading.

References

[1] Ahn, S. (2022). Developing Summary Writing Abilities of Korean EFL University Students through Teaching Summarizing Skills. English Teaching, 77(2), 25-43.

[2] Susanti, M. N. I., Ramadhan, A., & Warnars, H. L. H. S. (2023). Automatic essay exam scoring system: A systematic literature review. Procedia Computer Science, 216, 531-538.

[3] Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. Artificial Intelligence Review, 55(3), 2495-2527.

[4] Hussein, M. A., Hassan, H., & Nassef, M. (2019). Automated language essay scoring systems: A literature review. PeerJ Computer Science, 5, e208.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.

[7] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune bert for text classification?. In Chinese Computational Linguistics: 18th China National Conference, 194-206.

[8] Mayfield, E., & Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring?. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, 151-162.

[9] Yang, R., Cao, J., Wen, Z., Wu, Y., & He, X. (2020,). Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In Findings of the Association for Computational Linguistics, 1560-1569.

DOI: 10.54254/2755-2721/52/20241321

[10] CommonLit - Evaluate Student Summaries, URL: https://www.kaggle.com/competitions/commonlit-evaluate-student-summaries. Last accessed: 2023/10/13