On the Use of BERT for Automated Essay Scoring: Joint Learning of Multi-Scale Essay Representation

Yongjie Wang[1] **Chuan Wang**[1] **Ruobing Li**[1] **Hui Lin**[1,2]

[1]LAIX Inc.
[2]Shanghai Key Laboratory of Artificial Intelligence in Learning and Cognitive Science

{yongjie.wang, chuan.wang, ruobing.li, hui.lin}@liulishuo.com

Abstract

In recent years, pre-trained models have become dominant in most natural language processing (NLP) tasks. However, in the area of Automated Essay Scoring (AES), pre-trained models such as BERT have not been properly used to outperform other deep learning models such as LSTM. In this paper, we introduce a novel multi-scale essay representation for BERT that can be jointly learned. We also employ multiple losses and transfer learning from out-of-domain essays to further improve the performance. Experiment results show that our approach derives much benefit from joint learning of multi-scale essay representation and obtains almost the state-of-the-art result among all deep learning models in the ASAP[1] task. Our multi-scale essay representation also generalizes well to CommonLit Readability Prize (CRP[2]) data set, which suggests that the novel text representation proposed in this paper may be a new and effective choice for long-text tasks.

1 Introduction

AES is a valuable task, which can promote the development of automated assessment and help teachers reduce the heavy burden of assessment. With the rise of online education in recent years, more and more researchers begin to pay attention to this field.

AES systems typically consist of two modules, which are essay representation and essay scoring modules. The essay representation module extracts features to represent an essay and the essay scoring module rates the essay with the extracted features.

When a teacher rates an essay, the scores are often affected by multiple signals from different granularity levels, such as token level, sentence level, paragraph level and etc. For example, the features may include the numbers of words, the essay structure, the master degree of vocabulary and syntactic complexity, etc. These features come from different scales of the essay. This inspires us to extract multi-scale features from the essays which represent multi-level characteristics of the essays.

Most of the deep neural networks AES systems use LSTM or CNN. Some researchers (Uto et al., 2020; Rodriguez et al., 2019; Mayfield and Black, 2020) attempt to use BERT (Devlin et al., 2019) in their AES systems but fail to outperform other deep neural networks methods (Dong et al., 2017; Tay et al., 2018). We believe previous approaches using BERT for AES suffer from at least three limitations. First, the pre-trained models are usually trained on sentence-level, but fail to learn enough knowledge of essays. Second, the AES training data is usually quite limited for direct fine-tuning of the pre-trained models in order to learn better representation of essays. Last but not least, mean squared error (MSE) is commonly used in the AES task as the loss function. However, the distribution of the sample population and the sorting properties between samples are also important issues to be considered when designing the loss functions as they imitate the psychological process of teachers rating essays. Different optimizations can also bring diversity to the final overall score distribution and contribute to the effectiveness of ensemble learning.

To address the aforementioned issues and limitations, we introduce joint learning of multi-scale essay representation into the AES task with BERT, which outperforms the state-of-the-art deep learning models based on LSTM (Dong et al., 2017; Tay et al., 2018). We propose to explicitly model more effective representations by extracting multi-scale features as well as leveraging the knowledge learned from numerous sentence data. As the training data is limited, we also employ transfer learn-

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

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3416 - 3425

ing from out-of-domain essays which is inspired by (Song et al., 2020). To introduce the diversity of essay scoring distribution, we combine two other loss functions with MSE. When training our model with multiple losses and transfer learning using R-Drop (Liang et al., 2021), we almost achieve the state-of-the-art result among all deep learning models. The source code of prediction module with a trained model for ASAP's prompt 8 is publicly available[3].

In summary, the contribution of this work is as follows:

- We propose a novel essay scoring approach to jointly learn multi-scale essay representation with BERT, which significantly improve the result compared to traditionally using pre-trained language models.
- Our method shows significant advantages in long text tasks and obtains almost the state-of-the-art result among all deep learning models in the ASAP task.
- We introduce two new loss functions which are inspired by the mental process of teacher rating essays, and employ transfer learning from out-of-domain essays with R-Drop (Liang et al., 2021), which further improves the performance for rating essays.

2 Related Work

The dominant approaches in AES can be grouped into three categories: traditional AES, deep neural networks AES and pre-training AES.

- **Traditional AES** usually uses regression or ranking systems with complicated handcrafted features to rate an essay (Larkey, 1998; Rudner and Liang, 2002; Attali and Burstein, 2006; Yannakoudakis et al., 2011; Chen and He, 2013; Phandi et al., 2015; Cozma et al., 2018). These handcrafted features are based on the prior knowledge of linguists. Therefore they can achieve good performance even with small amounts of data.
- **Deep Neural Networks AES** has made great progress and achieved comparable results with traditional AES recently (Taghipour and Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017; Alikaniotis et al., 2016; Wang et al., 2018; Tay et al., 2018; Farag et al., 2018; Song et al., 2020; Ridley et al., 2021; Muangkammuen and Fukumoto, 2020; Mathias et al., 2020). While the handcrafted features are complicated to implement and careful manual design makes these features less portable, deep neural networks such as LSTM or CNN can automatically discover and learn complex features of essays, which makes AES an end-to-end task. Saving much time to design features, deep neural networks can transfer well among different AES tasks. By combining traditional and deep neural network approaches, AES can even obtain a better result, which benefits from both representations (Jin et al., 2018; Dasgupta et al., 2018; Uto et al., 2020). However, ensemble way still needs handcrafted features which cost numerous energy of researchers.
- **Pre-training AES** uses the pre-trained language model as the initial essay representation module and fine-tune the model on the essay training set. Though the pre-trained methods have achieved the state-of-the-art performance in most NLP tasks, most of them (Uto et al., 2020; Rodriguez et al., 2019; Mayfield and Black, 2020) fail to show an advantage over other deep learning methods (Dong et al., 2017; Tay et al., 2018) in AES task. As far as we know, the work from Cao et al. (2020) and Yang et al. (2020) are the only two pre-training approaches which surpass the other deep learning methods. Their improvement mainly comes from the training optimization. Cao et al. (2020) employ two self-supervised tasks and domain adversarial training, while Yang et al. (2020) combine regression and ranking to train their model.

3 Approach

3.1 Task Formulation

The AES task is defined as following:

Given an essay with n words $X = \{x_i\}_{i=1}^{n}$, we need to output one score y as a result of measuring the level of this essay.

Quadratic weighted Kappa (QWK) (Cohen, 1968) metric is commonly used to evaluate AES systems by researchers, which measures the agreement between the scoring results of two raters.

[3] https://github.com/lingochamp/Multi-Scale-BERT-AES

3.2 Multi-scale Essay Representation

We obtain the multi-scale essay representation from three scales: token-scale, segment-scale and document-scale.

Token-scale and Document-scale Input We apply one pre-trained BERT (Devlin et al., 2019) model for token-scale and document-scale essay representations. The BERT tokenizer is used to split the essay into a token sequence $T_1 = [t_1, t_2, ...t_n]$, where t_i is the ith token and n is the number of the tokens in the essay. The **token** we mentioned in this paper all refer to WordPiece, which is obtained by the subword tokenization algorithm used for BERT. We construct a new sequence T_2 from T_1 as following. L is set to 510, which is the max sequence length supported by BERT except the token $[CLS]$ and $[SEP]$.

$$T_2 = \begin{cases} \text{[CLS]}+[t_1, t_2, .., t_L]+\text{[SEP]} & n > L \\ \text{[CLS]}+T_1+\text{[SEP]} & n = L \\ \text{[CLS]}+T_1+\text{[PAD]}*(L-n)+\text{[SEP]} & n < L \end{cases}$$

The final input representation are the sum of the token embeddings, the segmentation embeddings and the position embeddings. A detailed description can be found in the work of BERT (Devlin et al., 2019).

Document-scale The document-scale representation is obtained by the $[CLS]$ output of the BERT model. As the $[CLS]$ output aggregates the whole sequence representation, it attempts to extract the essay information from the most global granularity.

Token-scale As the BERT model is pre-trained by Masked Language Modeling (Devlin et al., 2019), the sequence outputs can capture the context information to represent each token. An essay often consists of hundreds of tokens, thus RNN is not the proper choice to combine all the token information due to the gradients vanishing problem. Instead, we utilize a max-pooling operation to all the sequence outputs and obtain the combined token-scale essay representation. Specifically, the max-pooling layer generates a d-dimensional vector $W = [w_1, w_2, ..., w_j, ..., w_d]$ and the element w_j is computed as below:

$$w_j = max\{h_{1,j}, h_{2,j}, ..., h_{n,j}\}$$

where d is the hidden size of the BERT model. As we use the pre-trained BERT model **bert-base-uncased**[4], the hidden size d is 768. All the n sequence outputs of the BERT model are annotated as $[h_1, h_2, ..., h_i, ..., h_n]$, where h_i is a d-dimensional vector $[h_{i,1}, h_{i,2}, ..., h_{i,d}]$ representing the ith sequence output, and $h_{i,j}$ is the jth element in h_i.

Segment-scale Assuming the segment-scale value set is $K = [k_1, k_2, ...k_i, ..., k_S]$, where S is the number of segment scales we want to explore, and k_i is the ith segment-scale in K. Given a token sequence $T_1 = [t_1, t_2, ...t_n]$ for an essay, we obtain the segment-scale essay representation corresponding to scale k_i as follows:

1. We define n_p as the maximum number of tokens corresponding to each essay prompt p. We truncate the token sequence to n_p tokens if the essay length is longer than n_p, otherwise we pad $[PAD]$ to the sequence to reach the length n_p.

2. Divide the token sequence into $m = \lceil n_p/k_i \rceil$ segments and each segment is of length k_i except for the last segment, which is similar to the work of (Mulyar et al., 2019).

3. Input each of the m segment tokens into the BERT model, and get m segment representation vectors from the $[CLS]$ output.

4. Use an LSTM model to process the sequence of m segment representations, followed by attention pooling operation on the hidden states of the LSTM output to obtain the segment-scale essay representation corresponding to scale k_i.

The LSTM cell units process the sequence of segment representations and generate the hidden states as follows:

$$\begin{aligned} i_t &= \sigma(Q_i \cdot s_t + U_i \cdot h_{t-1} + b_i) \\ f_t &= \sigma(Q_f \cdot s_t + U_f \cdot h_{t-1} + b_f) \\ \hat{c}_t &= tanh(Q_c \cdot s_t + U_c \cdot h_{t-1} + b_c) \\ c_t &= i_t \circ \hat{c}_t + f_t \circ c_{t-1} \\ o_t &= \sigma(Q_o \cdot s_t + U_o \cdot h_{t-1} + b_o) \\ h_t &= o_t \circ tanh(c_t) \end{aligned}$$

where s_t is the tth segment representation from BERT $[CLS]$ output and h_t is the tth hidden state generated from LSTM. Q_i, Q_f, Q_c, Q_o, U_i, U_f, U_c and U_o are weight matrices, and b_i, b_f, b_c, and b_o are bias vectors.

The attention pooling operation we use is similar to the work of (Dong et al., 2017), which is defined as follows:

$$\begin{aligned} \hat{\alpha}_t &= tanh(Q_a \cdot h_t + b_a) \\ \alpha_t &= \frac{e^{q_a \cdot \hat{\alpha}_t}}{\sum_j e^{q_a \cdot \hat{\alpha}_j}} \\ o &= \textstyle\sum_t \alpha_t \cdot h_t \end{aligned}$$

[4] https://huggingface.co/bert-base-uncased

o is the segment-scale essay representation corresponding to the scale k_i. α_t is the attention weight for hidden state h_t. Q_a, b_a, q_a are the weight matrix, bias and weight vector respectively.

3.3 Model Architecture

The model architecture is depicted in Figure 1.

We apply one BERT model to obtain the document-scale and token-scale essay representation. The concatenation of them is input into a dense regression layer which predicts the score corresponding to the document-scale and token-scale. For each segment-scale k with number of segments m, we apply another BERT model to get m CLS outputs, and apply an $LSTM$ model followed by an attention layer to get the segment-scale representation. We input the segment-scale representation into another dense regression layer to get the score corresponding to segment-scale k. The final score is obtained by adding the scores of all S segment-scales and the score of the document-scale and token-scale, which is illustrated as below:

$$\begin{aligned} y &= \textstyle\sum_k y_k + y_{doc,tok} \\ y_k &= \hat{W}_{seg} \cdot o_k + b_{seg} \\ y_{doc,tok} &= \hat{W}_{doc,tok} \cdot H_{doc,tok} + b_{doc,tok} \\ H_{doc,tok} &= w_{doc} \bigoplus W \end{aligned}$$

y_k is the predicted score corresponding to segment-scale k. $y_{doc,tok}$ is the predicted score corresponding to the document-scale and token-scale. $\hat{W}_{seg}$ and b_{seg} are weight matrix and bias for segment-scale respectively. $W_{doc,tok}$ and $b_{doc,tok}$ are weight matrix and bias for document and token-scales, o_k is the segment-scale essay representation with the scale k. w_{doc} is the document-scale essay representation. W is the token-scale essay representation. $H_{doc,tok}$ is the concatenation of document-scale and token-scale essay representations.

3.4 Loss Function

We use three loss functions to train the model.

MSE measures the average value of square errors between predicted scores and labels, which is defined as below:

$$MSE(y, \hat{y}) = \tfrac{1}{N} \textstyle\sum_i (y_i - \hat{y}_i)^2$$

where y_i and $\hat{y}_i$ are the predicted score and the label for the ith essay respectively, N is the number of the essays.

Similarity (SIM) measures whether two vectors are similar or dissimilar by using cosine function. A teacher takes into account the overall level distribution of all the students when rating an essay. Following such intuition, we introduce the SIM loss to the AES task. In each training step, we take the predicted scores of the essays in the batch as the predicted vector y, and the labels as the label vector $\hat{y}$. The SIM loss awards the similar vector pairs to make the model think more about the correlation among the batch of essays. The SIM loss is defined as below:

$$\begin{aligned} SIM(y, \hat{y}) &= 1 - cos(y, \hat{y}) \\ y &= [y_1, y_2, ..., y_N] \\ \hat{y} &= [\hat{y}_1, \hat{y}_2, ..., \hat{y}_N] \end{aligned}$$

where y_i and $\hat{y}_i$ are the predicted score and label for the ith essay respectively, N is the number of the essays.

Margin Ranking (MR) measures the ranking orders for each essay pair in the batch. We intuitively introduce MR loss because the sorting property between essays is a key factor to scoring. For each batch of essays, we first enumerate all the essay pairs, and then compute the MR loss as follows. The MR loss attempts to make the model penalize wrong order.

$$MR(y, \hat{y}) = \tfrac{1}{\hat{N}} \textstyle\sum_{i,j} max(0, -r_{i,j}(y_i - y_j) + b)$$

$$r_{i,j} = \begin{cases} 1 & \hat{y}_i > \hat{y}_j \\ \text{-1} & \hat{y}_i < \hat{y}_j \\ \text{-}sgn(y_i - y_j) & \hat{y}_i = \hat{y}_j \end{cases}$$

y_i and $\hat{y}_i$ are the predicted score and label for the ith essay respectively. $\hat{N}$ is the number of the essay pairs. b is a hyper parameter, which is set to 0 in our experiment. For each sample pair (i, j), when the label $\hat{y}_i$ is larger than $\hat{y}_j$, the predicted result y_i should be larger than y_j, otherwise, the pair contributes $y_j - y_i$ to the loss. When $\hat{y}_i$ is equal to $\hat{y}_j$, the loss is actually $|y_i - y_j|$.

The combined loss is described as below:

$Loss_{total}(y, \hat{y}) = \alpha MSE(y, \hat{y}) + \beta MR(y, \hat{y}) + \gamma SIM(y, \hat{y})$.

α, β, γ are weight parameters which are tuned according to the performance on develop set.

4 Experiment

4.1 Data and Evaluation

ASAP data set is widely used in the AES task, which contains eight different prompts. A detailed description can be seen in Table 1. For each prompt, the WordPiece length indicates the smallest number which is bigger than the length of 90% of the

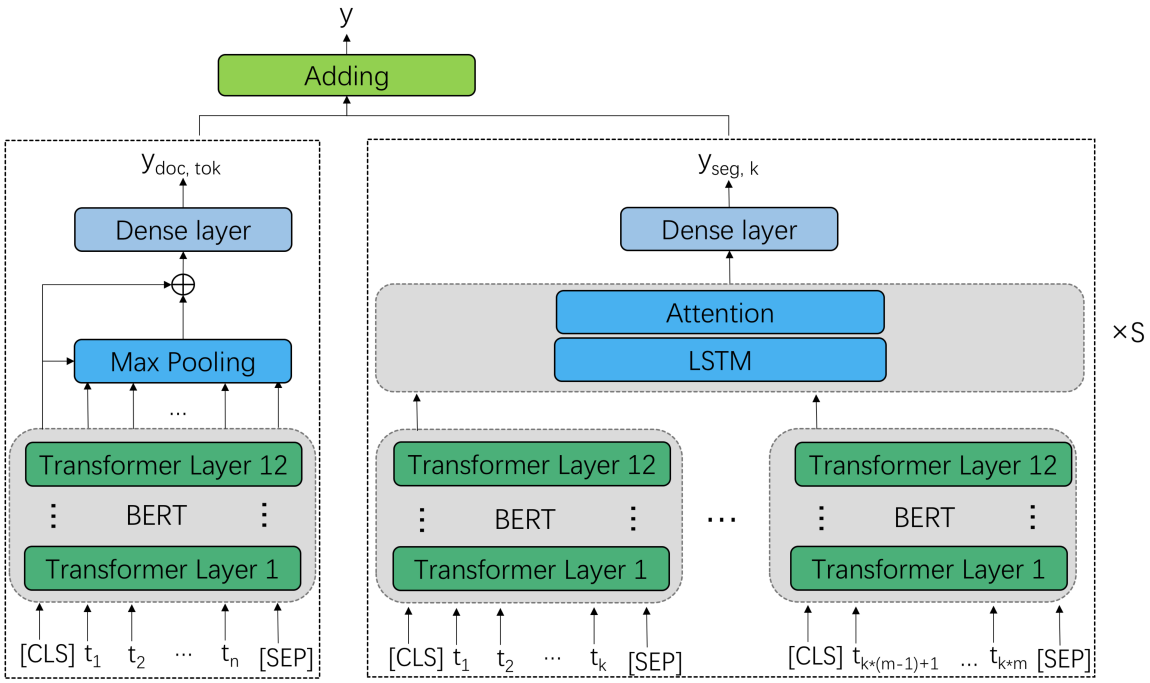

Figure 1: The proposed automated essay scoring architecture based on multi-scale essay representation. The left part illustrates the document-scale and token-scale essay representation and scoring module, and the right part illustrates S segment-scale essay representations and scoring modules.

essays in terms of WordPiece number. We evaluate the scoring performance using QWK on ASAP data set, which is the official metric in the ASAP competition. Following previous work, we adopt 5-fold cross validation with 60/20/20 split for train, develop and test sets.

CRP data set provides 2834 excerpts from several time periods and reading ease scores which range from -3.68 to 1.72. The average length of the excerpts is 175 and the WordPiece length is 252. We also use 5-fold cross validation with 60/20/20 split for train, develop and test sets on CRP data set. As the RMSE metric is used in the CRP competition, we also use it to evaluate our system in ease score prediction task.

Prompt	Essays	Avg length	Score Range	WordPiece length
1	1783	350	2-12	649
2	1800	350	1-6	704
3	1726	150	0-3	219
4	1772	150	0-3	203
5	1805	150	0-4	258
6	1800	150	0-4	289
7	1569	250	0-30	371
8	723	650	0-60	1077

Table 1: Statistics of ASAP data set.

4.2 Baseline

The baseline models for comparison are described as follows.

EASE [5] is the best open-source system that participated in the ASAP competition and ranked the third place among 154 participants. EASE uses regression techniques with handcrafted features. Results of EASE with the settings of Support Vector Regression (SVR) and Bayesian Linear Ridge Regression (BLRR) are reported in (Phandi et al., 2015).

CNN+RNN Various deep neural networks based on CNN and RNN for AES are studied by (Taghipour and Ng, 2016). They combine CNN ensembles and LSTM ensembles over 10 runs and get the best result in their experiment.

Hierarchical LSTM-CNN-Attention (Dong et al., 2017) builds a hierarchical sentence-document model, which uses CNN to encode sentences and LSTM to encode texts. The attention mechanism is used to automatically determine the relative weights of words and sentences in generating sentence representations and text representations respectively. They obtain the state-of-the-art result among all neural models without pre-training.

[5] http://github.com/edx/ease

SKIPFLOW (Tay et al., 2018) proposes to use SKIPFLOW mechanism to model the relationships between snapshots of the hidden representations of an LSTM. The work of (Tay et al., 2018) also obtains the state-of-the-art result among all neural models without pre-training.

Dilated LSTM with Reinforcement Learning (Wang et al., 2018) proposes a method using a dilated LSTM network in a reinforcement learning framework. They attempt to directly optimize the model using the QWK metric which considers the rating schema.

HA-LSTM+SST+DAT and BERT+SST+DAT (Cao et al., 2020) propose to use two self-supervised tasks and a domain adversarial training technique to optimize their training, which is the first work to use pre-trained language model to outperform LSTM based methods. They experiment with both hierarchical LSTM model and BERT in their work, which are $HA{-}LSTM{+}SST{+}DAT$ and $BERT + SST + DAT$ respectively.

BERT^2 (Yang et al., 2020) combines regression and ranking to fine-tune BERT model which also outperforms LSTM based methods and even obtains the new state-of-the-art.

4.3 Settings

To compare with the baseline models and further study the effectiveness of multi-scale essay representations, losses and transfer learning, we conduct the following experiments.

Multi-scale Models. These models are optimized with MSE loss, and **BERT-DOC** represents essays with document-scale features based on BERT. **BERT-TOK** represents essays with token-scale features based on BERT. **BERT-DOC-TOK** represents essays with both document-scale and token-scale features based on BERT. **BERT-DOC-TOK-SEG** represents essays with document-scale, token-scale, and multiple segment-scale features based on BERT. Longformer (Beltagy et al., 2020) is an extension for transformers with an attention mechanism that scales linearly with sequence length, making it easy to process long documents. We conduct experiments to show that our multi-scale features also works with Longformer and can further improve the performance in long text tasks. **Longformer-DOC-TOK-SEG** uses document-scale, token-scale, and multiple segment-scale features to represent essays, but based on Longformer instead of BERT. **Longformer-DOC** represents essays with document-scale features based on Longformer.

Models with Transfer Learning. To transfer learn from the out-of-domain essays [6], we additionally employ a pre-training stage, which is similar to the work of (Song et al., 2020). In this stage, we scale all the labels of essays from out-of-domain data into range 0-1 and pre-train the model on them with MSE loss. After the pre-training stage, we continue to fine-tune the model on in-domain essays. **Tran-BERT-MS** has the same modules as **BERT-DOC-TOK-SEG** with pre-training on out-of-domain data. **MS** means multiple scale features.

Models with Multiple Losses. Based on **Tran-BERT-MS** model, we explore the performance of adding multiple loss functions. **Tran-BERT-MS-ML** additionally employs MR loss and SIM loss. **ML** means multiple losses. **Tran-BERT-MS-ML-R** incorporates R-Drop strategy (Liang et al., 2021) in training based on **Tran-BERT-MS-ML** model.

For the proposed model architecture which is depicted in Figure 1, the BERT model in the left part are shared by the document-scale and token-scale essay representations, and the other BERT model in the right part are shared by all segment-scale essay representations. We use the "bert-base-uncased" which includes 12 transformer layers and the hidden size is 768. In the training stage, we freeze all the layers in the BERT models except the last layer, which is more task related than other layers. The Longformer model used in our work is "longformer-base-4096". For the MR loss, we set b to 0. The weights α, β and γ are tuned according to the performance on develop set. We use Adam optimizer (Kingma and Ba, 2015) to fine-tune model parameters in an end-to-end fashion with learning rate of 6e-5, $\beta 1$=0.9, $\beta 2$=0.999, $L2$ weight decay of 0.005. The coefficient weight α in R-Drop is 9. We set the batch size to 32. We use dropout in the training stage and the drop rate is set to 0.1. We train all the models for 80 epochs, and select the best model according the performance on the develop set. We use a greedy search method to find the best combination of segment scales, which is shown in detail in Appendix A. Following (Cao et al., 2020), we perform the significance test for our models.

[6] For each prompt, we use all the essays from other prompts in ASAP data set.

ID	Models	P1	P2	P3	P4	P5	P6	P7	P8	Average
1	EASE(SVR) (Phandi et al., 2015)	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
2	EASE(BLRR) (Phandi et al., 2015)	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705
3	CNN(10 runs) + LSTM(10 runs) (Taghipour and Ng, 2016)	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
4	Hierarchical LSTM-CNN-Attention (Dong et al., 2017)	0.822	0.682	0.672	0.814*	0.803	0.811	0.801	0.705	0.764
5	SKIPFLOW LSTM(Bilinear) (Tay et al., 2018)	0.830	0.678	0.677	0.778	0.795	0.807	0.790	0.670	0.753
6	SKIPFLOW LSTM(Tensor) (Tay et al., 2018)	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.764
7	Dilated LSTM With RL (Wang et al., 2018)	0.776	0.659	0.688	0.778	0.805	0.791	0.760	0.545	0.724
8	HA-LSTM+SST+DAT (Cao et al., 2020)	**0.836**	**0.730**	**0.732**	0.822	0.835	0.832*	0.821	0.718	0.790
9	BERT+SST+DAT (Cao et al., 2020)	0.824	0.699	**0.726**	**0.859**	0.822	0.828	**0.840**	0.726	0.791*
10	R^2BERT (Yang et al., 2020)	0.817	0.719	0.698	0.845	**0.841**	**0.847**	0.839	0.744	**0.794***
11	**BERT-DOC-TOK-SEG**	**0.836**	0.695	0.700	0.815	0.812	0.816	0.838	0.744	0.782
12	**Tran-BERT-MS-ML-R**	0.834	0.716	0.714	0.812	0.813	0.836	0.839	**0.766**	0.791*

Table 2: Experiment results of all models in terms of QWK on ASAP. The name of our implemented models are in bold. The bold number is the best performance for each prompt. The best 3 average QWK are annotated with *.

ID	Models	P1	P2	P8	Average
8	HA-LSTM+SST+DAT	0.836	0.730	0.718	0.761
9	HA-BERT+SST+DAT	0.824	0.699	0.726	0.750
10	R^2BERT	0.817	0.719	0.744	0.760
12	**Tran-BERT-MS-ML-R**	0.834	0.716	0.766	0.772

Table 3: Experiment results of our model and the state-of-the-art models on ASAP long essays (WordPiece length are longer than 510). The name of our implemented model is in bold.

4.4 Results

Table 2 shows the performance of baseline models and our proposed models with joint learning of multi-scale essay representation. Table 3 shows the results of our model and the state-of-the-art models on essays in prompt 1, 2 and 8, whose WordPiece length are longer than 510. We summarize some findings from the experiment results.

- Our model 12 almost obtains the published state-of-the-art for neural approaches. For the prompts 1,2 and 8, whose WordPiece length are longer than 510, we improve the result from 0.761 to 0.772. As Longformer is good at encoding long text, we also use it to encode essays of prompt 1, 2 and 8 directly but the performance is poor compared to the methods in Table 3. The results demonstrate the effectiveness of the proposed framework for encoding and scoring essays. We further re-implement $BERT^2$ proposed by (Yang et al., 2020), and our implementation of $BERT^2$ is not as well-performing as the published result. Though (Uto et al., 2020) obtain a much better result(QWK 0.801), our method performs much better than their system with only neural features(QWK 0.730), which demonstrates the strong essay encoding ability of our neural approach.
- Compared to the models 4 and 6, our model 11 uses multi-scale features to encode essays instead of LSTM based models, and we use the same regression loss to optimize the model. Our model simply changes the representation way and significantly improves the result from 0.764 to 0.782, which demonstrates the strong encoding ability armed by multi-scale representation for long text. Before that, the conventional way of using BERT can not surpass the performance of models 4 and 6.

4.5 Further analysis

Multi-scale Representation We further analyze the effectiveness of employing each scale essay representation to the joint learning process.

Models	Average QWK
BERT-DOC	0.760
BERT-TOK	0.764
BERT-DOC-TOK	0.768
BERT-DOC-TOK-SEG	0.782

Table 4: Performance of different feature scale models on ASAP data set.

Models	RMSE
BERT-DOC	0.742
BERT-TOK	0.760
BERT-DOC-TOK	0.691
BERT-DOC-TOK-SEG	0.607

Table 5: Performance of different feature scale models on CRP data set. The evaluation metric is RMSE. Lower numbers are better.

Table 4 and Table 5 show the performance of our models to represent essays on different feature scales, which are trained with MSE loss and without transfer learning. Table 4 shows the performance on ASAP data set while Table 5 shows the performance on CRP data set. The improvement of

BERT-DOC-TOK-SEG over BERT-DOC, BERT-TOK, BERT-DOC-TOK are significant (p<0.0001) on CRP data set, and are significant (p<0.0001) in most cases on ASAP data set. Results on both table indicate the similar findings.

- Combining the features from document-scale and token-scale, BERT-DOC-TOK outperforms the models BERT-DOC and BERT-TOK, which only use one scale features. This demonstrates that our proposed framework can benefit from multi-scale essay representation even with only two scales.
- By additionally incorporating multiple segment-scale features, BERT-DOC-TOK-SEG performs much better than BERT-DOC-TOK. This demonstrates the effectiveness and generalization ability of our multi-scale essay representation on multiple tasks.

Models	Average QWK
Longformer-DOC	0.746
Longformer-DOC-TOK-SEG	0.771

Table 6: Performance of multi-scale Longformer models on ASAP data set.

Reasons for Effectiveness of Multi-scale Representation Though the experiment shows the effectiveness of multi-scale representation, we further explore the reason. We could doubt that the effectiveness comes from supporting long sequences, not the multi-scale itself. As Longformer is good at dealing with long texts, we compare the results between Longformer-DOC and Longformer-DOC-TOK-SEG. The results of the significance test show that the improvement of Longformer-DOC-TOK-SEG over Longformer-DOC are significant (p<0.0001) in most cases. Performance of the two models are shown in Table 6, and we get the following findings.

- Though Longformer-DOC supports long sequences encoding, it performs poor, which indicates us that supporting long sequence ability is not enough for a good essay scoring system.
- Longformer-DOC-TOK-SEG outperforms Longformer-DOC significantly, which indicates the effectiveness of our model comes from encoding essays by multi-scale features, not only comes from the ability to deal with long texts.

These results are consistent with our intuition that our approach takes into account different level features of essays and predict the scores more accurately. We consider it caused by that multi-scale features are not effectively constructed in the representation layer of pre-trained model due to the lack of data for fine-tuning in the AES task. Therefore, we need to explicitly model the multi-scale information of the essay data and combine it with the powerful linguistic knowledge of pre-trained model.

Models	Average
BERT-DOC-TOK-SEG	0.782
Tran-BERT-MS	0.788
Tran-BERT-MS-ML	0.790
Tran-BERT-MS-ML-R	0.791

Table 7: Experiment results for transfer learning with multiple loss functions and R-Drop .

Transfer Learning with Multiple Losses and R-Drop We further explore the effectiveness of pre-training with adding multiple loss functions and employing R-Drop. As is shown in table 7, by incorporating the pre-training stage which learns the knowledge from out-of-domain data, Tran-BERT-MS model improves the result from 0.782 to 0.788 compared to BERT-DOC-TOK-SEG model. The model Tran-BERT-MS-ML which jointly learns with multiple loss functions further improves the performance from 0.788 to 0.790. We consider it due to the reason that MR brings ranking information and SIM takes into account the overall score distribution information. Diverse losses bring different but positive influence on the optimization direction and act as an ensembler. By employing R-Drop, Tran-BERT-MS-ML-R improves the QWK slightly, which comes from the fact that R-Drop plays a regularization role.

5 Conclusion and Future Work

In this paper, we propose a novel multi-scale essay representation approach based on pre-trained language model, and employ multiple losses and transfer learning for AES task. We almost obtain the state-of-the-art result among deep learning models. In addition, we show multi-scale representation has a significant advantage when dealing with long texts.

One of the future directions could be exploring soft multi-scale representation. Introducing linguistic knowledge to segment at a more reasonable scale may bring further improvement.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 715–725.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v.2. *The Journal of Technology, Learning, and Assessment*, 4(3).

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. In *arXiv: Computation and Language*.

Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-adaptive neural automated essay scoring. In *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information*, pages 1011–1020.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752.

J Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213–220.

Mădălina Cozma, Andrei M. Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Tirthankar Dasgupta, Abir Naskar, Lipika Dey, and Rupsa Saha. 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 93–102.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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*, pages 4171–4186.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring – an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. 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 Technolog*, pages 263–271.

Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. Tdnn: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1088–1097.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations*.

Leah S. Larkey. 1998. Automatic essay grading using text categorization techniques. In *SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieva*, pages 90–95.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*, pages 10890–10905.

Sandeep Mathias, Rudra Murthy, Diptesh Kanojia, Abhijit Mishra, and Pushpak Bhattacharyya. 2020. Happy are those who grade without seeing: A multi-task learning approach to grade essays using gaze behaviour. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 858–872.

Elijah Mayfield and Alan W Black. 2020. Should you fine-tune bert for automated essay scoring? In *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162.

Panitan Muangkammuen and Fumiyo Fukumoto. 2020. Multi-task learning for automated essay scoring with sentiment analysis. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 116–123.

Andriy Mulyar, Elliot Schumacher, Masoud Rouhizadeh, and Mark Dredze. 2019. Phenotyping of clinical notes with improved document classification models using contextualized neural

language models. In *33rd Conference on Neural Information Processing Systems (NeurIPS)*.

Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439.

Robert Ridley, Liang He, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13745–13753.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and automated essay scoring. In *arXiv: Computation and Language*.

Lawrence M. Rudner and Tahung Liang. 2002. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning, and Assessment*, 1(2):3–21.

Wei Song, Kai Zhang, Ruiji Fu, Lizhen Liu, Ting Liu, and Miaomiao Cheng. 2020. Multi-stage pre-training for automated chinese essay scoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6723–6733.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. Skipflow:incorporating neural coherence features for end-to-end automatic text scoring. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5948–5955.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088.

Yucheng Wang, Zhongyu Wei, Yaqian Zhou, and Xuanjing Huang. 2018. Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 791–797.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 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: EMNLP 2020*, pages 1560–1569.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.

A Appendix

All the segment-scales we explore range from 10 to 190. The interval between two neighbor scales is 20. As the combination number of all segment-scales is exponential, we use a greedy search method to find the best combination.

1. Initialize the segment-scale value set R as the document-scale and token-scale.

2. Experiment the combination of each segment-scale with the token-scale and document-scale essay representation, and compute the average QWK on develop set for all segment-scales, which is denoted as QWK_{ave}. The scale with higher QWK compared to QWK_{ave} is added to the candidate scale list L and the scales in L are sorted according to their QWK values from large to small.

3. For each i from 1 to $|L|$, we perform experiments on the combination of the first i segment-scales in L with the token-scale and document-scale. The combination segment-scales with the best performance on develop set are added to the segment-scale value set R