Enhancing Automated Essay Scoring Performance via Fine-tuning Pre-trained Language Models with Combination of Regression and Ranking

Ruosong Yang[†‡], **Jiannong Cao**[†], **Zhiyuan Wen**[†], **Youzheng Wu**[‡], **Xiaodong He**[‡]
[†]Department of Computing, The Hong Kong Polytechnic University
[‡]JD AI Research
[†]{csryang, csjcao, cszwen}@comp.polyu.edu.hk
[‡]{wuyouzheng1, xiaodong.he}@jd.com

Abstract

Automated Essay Scoring (AES) is a critical text regression task that automatically assigns scores to essays based on their writing quality. Recently, the performance of sentence prediction tasks has been largely improved by using Pre-trained Language Models via fusing representations from different layers, constructing an auxiliary sentence, using multi-task learning, etc. However, to solve the AES task, previous works utilize shallow neural networks to learn essay representations and constrain calculated scores with regression loss or ranking loss, respectively. Since shallow neural networks trained on limited samples show poor performance to capture deep semantic of texts. And without an accurate scoring function, ranking loss and regression loss measures two different aspects of the calculated scores. To improve AES's performance, we find a new way to fine-tune pre-trained language models with multiple losses of the same task. In this paper, we propose to utilize a pre-trained language model to learn text representations first. With scores calculated from the representations, mean square error loss and the batch-wise ListNet loss with dynamic weights constrain the scores simultaneously. We utilize Quadratic Weighted Kappa to evaluate our model on the Automated Student Assessment Prize dataset. Our model outperforms not only state-of-the-art neural models near 3 percent but also the latest statistic model. Especially on the two narrative prompts, our model performs much better than all other state-of-the-art models.

1 Introduction

Automated Essay Scoring (AES) automatically evaluates the writing quality of essays. Essay assignments evaluation costs lots of time. Besides, the same instructor scoring the same essay at different times may assign different scores (intra-rater variation), different raters scoring the same essay may assign different scores (inter-rater variation) (Smolentzov, 2013). To alleviate teachers' burden and avoid intra-rater variation, as well as inter-rater variation, AES is necessary and essential. An early AES system, e-rater (Chodorow and Burstein, 2004), has been used to score TOEFL writings.

Recently, large pre-trained language models, such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), etc. have shown the extraordinary ability of representation and generalization. These models have gained better performance in lots of downstream tasks such as text classification and regression. There are many new approaches to fine-tune pre-trained language models. Sun et al. (2019a) proposed to construct an auxiliary sentence to solve aspect-based sentiment classification tasks. Cohan et al. (2019) added extra separate tokens to obtain representations of each sentence to solve sequential sentence classification tasks. Sun et al. (2019b) summarized several fine-tuning methods, including fusing text representations from different layers, utilizing multi-task learning, etc. To our knowledge, there are no existing works to improve AES tasks with pre-trained language models. Before introducing our new way to use pre-trained language models, we briefly review existing works in AES firstly.

Existing works utilize different methods to learn text representations and constrain scores, which are the two key steps in AES models. For text representation learning, various neural networks are used to learn essay representations, such as Recurrent Neural Network (RNN) (Taghipour and Ng, 2016; Tay et al., 2018), Convolutional Neural Network (CNN) (Taghipour and Ng, 2016), Recurrent Convolutional Neural Network (RCNN) (Dong et al., 2017), etc. However, simple neural networks like RNN and CNN focus on word-level information, which is difficult to capture word connections in

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1560–1569
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

long-distance dependency. Besides, shallow neural networks trained on a small volume of labeled data are hard to learn deep semantics. As for score constraints, prediction and ranking are two popular solutions. From the prediction perspective, the task is a regression or classification problem (Taghipour and Ng, 2016; Tay et al., 2018; Dong et al., 2017). Besides, from the recommendation perspective, learning-to-rank methods (Yannakoudakis et al., 2011; Chen and He, 2013) aim to rank all essays in the same order as that ranked by gold scores. However, without precise score mapping functions, only regression constraints could not ensure the right ranking order. And only ranking based models could not guarantee accurate scores. In general, there are two key challenges for the AES task. One is how to learn better essay representations to evaluate the writing quality, the other one is how to learn a more accurate score mapping function.

Motivated by the great success of pre-trained language models such as BERT in learning text representations with deep semantics, it is reasonable to utilize BERT to learn essay representations. Since self-attention is a key component of the BERT model, it can capture the interactions between any two words in the whole essays (long texts). Previous work (Sun et al., 2019b) shows that fusing text representations from different layers does not improve the performance effectively. For the AES task, the length of essays approximates the length limit of the BERT model, so it is hard to construct an auxiliary sentence. Meanwhile, only score labels are available; it is also difficult to utilize multi-task learning. Summarized existing works in AES, they utilize regression loss or ranking loss, respectively. Regression loss requires to obtain accurate score value, and ranking loss aims to get precise score order. Unlike multi-task learning requires different fully-connected networks for different tasks, we propose to constrain the same task with multiple losses to fine-tune the BERT model. In addition, it is impossible to rank all essays in one batch so that the model is required to learn more accurate scores. During training, the weight of the regression loss is increasing while that of ranking loss is decreasing.

In this paper, we propose R^{2}BERT (BERT Model with Regression and Ranking). In our model, BERT is used to learn text representations to capture deep semantics. Then a fully connected neural network is used to map the representations to scores. Finally, regression loss and batch-wise ranking loss constrain the scores together, which are jointly optimized with dynamic combination weights. To evaluate our model, an open dataset, Automated Student Assessment Prize (ASAP), is used. With the measurement of Quadratic Weighted Kappa (QWK), our model outperforms state-of-the-art neural models on average QWK score of all eight prompts near 3 percent and also performs better than the latest statistical model. Especially on the two narrative Prompts (7 and 8), only the regression based model performs comparably even better compared with other models. And our model with combined loss gains much better performance. To explain the model's effectiveness, we also illustrate the attention weights on two example essays (an argumentative essay and a narrative essay). The self-attention can capture most conjunction words that reveal the logical structure, and most key concepts that show the topic shifting of the narratives.

In summary, our contributions are:

- We propose a new method called multi-loss to fine-tune BERT models in AES tasks. We are also the first one to combine regression and ranking in these tasks. The experiment results show that the combined loss could improve the performance significantly.
- Experiment results also show that our model achieves the best average QWK score and outperforms other state-of-the-art neural models almost on each prompt.
- To show the effectiveness of self-attention in the BERT model, we illustrate the weights of different words on two examples, including one argumentative essay and one narrative essay.

2 Related Works

Ke and Ng (2019) summarized recent works on automated essay scoring. In general, there are three parts to solve the AES task, namely text representation learning, score mapping function, and score constraints. Almost all works utilize a linear combination function to map each text representation to a score. In the rest, we introduce various score constraints with used approaches for text representation learning.

According to different score constraints, existing works fall into three categories, namely prediction, recommendation, and reinforcement learning based models.

Prediction is the most general approach, including classification and regression. For classification, the models directly predict labels that point to different scores. In comparison, regression models constrain calculated scores to be the same as gold ones. Generally, hand-crafted features and neural network based features are two popular methods to learn text representations. Early works mainly focus on the construction of hand-crafted features such as statistical features and linguistic features. There are several early AES systems including e-rater (Chodorow and Burstein, 2004), PEG (Project Essay Grade) (Shermis and Burstein, 2003), and IntelliMetric (Elliot, 2003). e-rater utilized ten linguistic features, including eight representing aspects of writing quality and two representing content. PEG used a larger feature set with more than 30 elements of writing quality. IntelliMetric aggregated all the features into five types, namely Focus/Coherence, Organization, Elaboration/Development, Sentence Structure, and Mechanics/Conventions. Cozma et al. (2018) combined string kernel and word embeddings to extract features. With the success of deep learning, researchers start to utilize various neural networks to learn text representations. Taghipour and Ng (2016) explored several neural networks, such as Long Short-Term Memory (LSTM) and CNN. Finally, they found that the ensemble model combining LSTM and CNN performs best. Dong et al. (2017) proposed a hierarchical text model that utilized CNN to learn sentence representations, and LSTM was used to learn text representations. Tay et al. (2018) introduced a model called SKIPFLOW, which aimed to capture neural coherence features of the text via considering the adjacent hidden states in the LSTM model.

In the recommendation view, learning to rank approaches is another popular method to solve this task. Yannakoudakis et al. (2011) firstly addressed this problem as a rank preference problem. Based on statistical features, RankSVM, a pairwise learning to rank model, was used as score constraint. Chen and He (2013) utilized listwise learning to rank model to learn a ranking model based on several linguistic features.

Reinforcement learning based models are also possible solutions. Wang et al. (2018b) utilized dilated LSTM to learn text representations. Then scores calculation was guided by quadratic weighted kappa based reward function.

For text representation, previous works only consider the relations among sentences. In this paper, we focus on all the interactions between any two words. Besides, existing works only utilize regression or ranking loss, respectively. We combine two losses dynamically in our model.

3 R^{2}BERT

In this section, we first introduce the framework of our model, briefly review the BERT model, as well as self-attention. In addition, we will illustrate the regression model as well as some useful tricks. Finally, we will show batch-wise learning to rank model and the combination metric.

Our model, as shown in Figure 1, takes a batch of essays as input. With preprocessing (adding a special token, [CLS], at the beginning of each essay), each token is transformed into its embedding and sent into the BERT model. The representations of all essays are the output vectors mapping to [CLS]. Essay scores could be obtained by passing the representations into the Score Mapping Function. They are constrained by regression loss and ranking loss, which are optimized jointly with the dynamic combination. As shown in the color bar, the weight of regression loss is gradually increasing, while that of ranking loss is decreasing.

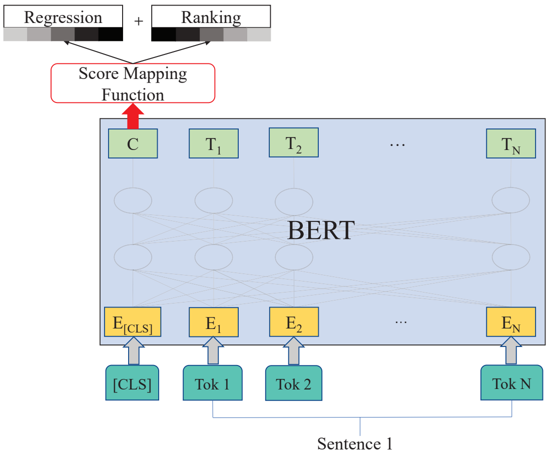

Figure 1: R^{2}BERT Framework

3.1 BERT

BERT (Devlin et al., 2019) refers to Bidirectional Encoder Representations from Transformers, which is one of the most popular models in recent years. More specifically, BERT is an extremely large pre-trained language model, which was trained on enormous corpora, totally more than 3000M words. Meanwhile, two target tasks, namely masked language model and next sentence prediction, are used to train the model. Many downstream tasks of natural language processing have gained benefits by utilizing pre-trained BERT to get text representation such as sentence classification, question answer, common sense inference, etc. To benefit regression problems, the target task is replaced by a fully connected neural network. Then the whole BERT model is fine-tuned on the new dataset.

Generally BERT has two parameter intensive settings:

BERT$_{\texttt{BASE}}$: 12 layers, 768 hidden dimensions and 12 attention heads (in transformer) with the total number of parameters, 110M;

BERT$_{\texttt{LARGE}}$: 24 layers, 1024 hidden dimensions and 16 attention heads (in transformer) with the total number of parameters, 340M.

3.2 Self-attention

Self-attention (Vaswani et al., 2017) is the key to the success of BERT, which is a mechanism that a sequence calculates the word weights with itself. Given a text, we construct a matrix W with three copies Q, K, V, referring to query, key, and value, in which each column is the word embedding. The new words' representations are calculated via the attention as shown in Formula 1, and Formula 2, where d is the size of word embedding, n_Q, n_K and n_V denote the number of words in each text, $Q[i]$ is the i_{th} word representation in the query text Q.

$$\texttt{Att}(Q,K) = [\texttt{softmax}(\frac{Q[i]\cdot K^T}{\sqrt{d}})]_{i=0}^{n_Q-1} \quad (1)$$

$$V_{att}(Q,K,V) = \texttt{Att}(Q,K)\cdot V \in R^{n_Q \times d} \quad (2)$$

3.3 Feature Extraction

Given a sample essay $t = \{w_1, w_2, .., w_N\}$ as input, where N is the number of the words, we preprocess it to a new sequence $t' = \{[\texttt{CLS}], w_1, w_2, .., w_N\}$, where $[\texttt{CLS}]$ is a special token. Assuming $\texttt{BERT}(\cdot)$ is the pre-trained BERT model, we can obtain the hidden representations of all the input words, $h = \texttt{BERT}(t') \in R^{r_h * |t'|}$, where $|t'|$ is the length of the input sequence and r_h is the dimension of the hidden state. Finally, the hidden representation mapping to $[\texttt{CLS}]$, $r = h_{[\texttt{CLS}]}$, is used as the text representation.

3.4 Regression

With obtained text representation r, a fully connected neural network $\texttt{FCNN}(\cdot)$ is used as the score mapping function. More specifically, $\texttt{FCNN}$ is a linear combination function, where $\boldsymbol{W}$ is the weight matrix and $\boldsymbol{b}$ is the bias as shown in Formula 3. To learn better parameters, the mean score of all training essays is used to initialize the bias $\boldsymbol{b}$. In addition, $\sigma = \texttt{Sigmoid}(\cdot)$, a non-linear activation function is used to normalize the calculated score into $[0, 1]$ as shown in Formula 4.

$$\texttt{FCNN}(r) = \boldsymbol{W}r + \boldsymbol{b} \quad (3)$$

$$s' = \sigma(\texttt{FCNN}(r)) \quad (4)$$

Mean square error is a widely used loss function for regression tasks. Given a dataset $D = \{(t_i, s_i)|i \in [1:m]\}$, m is the number of samples, and t_i refers to the i_{th} essay. Besides, s_i is the gold score of the i_{th} essay. The regression objective L_m is shown in Formula 5.

$$L_m = \texttt{MSE}(s, s') = \frac{1}{m}\sum_{i=1}^{m}(s_i - s'_i)^2 \quad (5)$$

3.5 Batchwise Learning to Rank Model

ListNet (Cao et al., 2007) ranks a list of objectives each time and measures the accordance between the predicted ranking list and the ground truth label. In our problem, all the essays are a large list. However, it is impossible to rank all the essays in one batch. We sacrifice the accuracy and only rank essays in each batch, which we called batch-wise ListNet.

Before introducing the objective of ListNet, we will give several basic definitions. Suppose that given a set of essays which are identified with the numbers $\{1, 2, ..., m\}$. A permutation π on the essays is defined as a bijection from $\{1, 2, ..., m\}$ to itself. The permutation is written as $\pi =< \pi(1), \pi(2), ..., \pi(m) >$, where $\pi(i)$ refers to the essay at position i in the permutation. And we also assume any permutation is possible. The set of all possible permutations is denoted as Ω_m. As aforementioned, we assume the batch size is m, and the calculated score of the essay pointed by $\pi(i)$ is $s'_{\pi(i)}$. As given by the original paper (Cao et al., 2007), the permutation probability is defined as Formula 6. And $\Phi(\cdot)$ is an increasing and strictly positive function.

$$P_{s'}(\pi) = \prod_{j=1}^{n} \frac{\Phi(s'_{\pi(j)})}{\sum_{k=j}^{n}\Phi(s'_{\pi(k)})} \quad (6)$$

The top one probability $P_{s'}(j)$ is defined as Formula 7, where j refers to each essay in the batch.

$$P_{s'}(j) = \frac{\Phi(s'_j)}{\sum_{k=1}^{n} \Phi(s'_k)} \tag{7}$$

With the use of top one probability, given two lists of scores s and s' as aforementioned, Cross Entropy could be used to represent the distance (batchwise loss function L_r) between the two score lists as shown in Formula 8.

$$L_r = \mathtt{CE}(s, s') = -\sum_{j=1}^{n} P_s(j) \log(P_{s'}(j)) \tag{8}$$

3.6 Combination of Regression and Ranking

The key problem of loss combination is to determine the weight of each loss. In the scoring scenario, teachers always prefer to score each essay rather than ranking all the essays. Besides, using batch-wise learning to rank approach could not guarantee precise global order. Referring to the combination method proposed in (Wu et al., 2009), the weight of ranking loss is decreasing, and that of regression loss is increasing during training. The weight calculation is followed by Formula 9, where τ_e is a σ function about e calculated as Formula 10.

$$L = \tau_e \times L_m + (1 - \tau_e) \times L_r \tag{9}$$

$$\tau_e = \frac{1}{1 + \exp(\gamma(E/2 - e))} \tag{10}$$

In Formula 10, E is the total number of the epochs, and e is the value of current epoch, γ is a hyper-parameter which is chosen such that $\tau_1 = 0.000001$.

4 Experiment

In this section, the ASAP dataset is introduced firstly. Then we illustrate experiment settings and evaluation metrics. In addition, baseline models, experiment results, and analyses are shown. Furthermore, we also visualize the attention weights of different words in two examples.

4.1 Dataset

The automated Student Assessment Prize dataset comes from a Kaggle competition[1], which contained eight essay prompts with different genres, including argumentative essays, response essays, and narrative essays. Each essay was given a score by the instructors. Some statistical information is shown in Table 1.

Set	#Essays	Genre	Avg Len.	Range
1	1783	ARG	350	2-12
2	1800	ARG	350	1-6
3	1726	RES	150	0-3
4	1772	RES	150	0-3
5	1805	RES	150	0-4
6	1800	RES	150	0-4
7	1569	NAR	250	0-30
8	723	NAR	650	0-60

Table 1: Statistics of the ASAP dataset; Range means the score range, For genre, ARG, RES, and NAR map to argumentative essays, response essays and narrative essays respectively.

4.2 Experiment Settings

Following previous work, we also utilize 5-fold cross-validation to evaluate all models with 60/20/20 split for train, validation, and test sets, which are provided by (Taghipour and Ng, 2016). We implement our model based on the Pytorch implementation of BERT [2] and use the BERT$_{\mathtt{BASE}}$ model due to the limit of GPU memory. Besides, we truncate all the essays with the max length of 512 words, following the setting of BERT. Also, for the limit of our GPU memory, the batch size is set to 16. Since essays in the ASAP dataset is much longer than that in GLUE (Wang et al., 2018a), we fine-tune our model for 30 epochs and select the best model based on the performance on the validation set. We adjust the fine-tuning learning rate from 1e-5 to 9e-5 with the step 1e-5, and 4e-5 performs best. And γ in Formula 10 is set to 0.99999. For tokenization and vocabulary, we all use the preprocessing tools provided by the BERT model. We also normalize all score ranges to within [0,1]. All the scores are rescaled back to the original prompt-specific scale for calculating Quadratic Weighted Kappa scores. Following previous works, we conduct the evaluation in prompt-specific fashion.

4.3 Evaluation Metric

Following previous works, Quadratic Weighted Kappa (QWK) is used as the evaluation metric, which measures the agreement between calculated scores and gold ones.

To calculate QWK, a weight matrix W is constructed firstly, as shown in Formula 11, where i and j are gold scores and calculated scores respec-

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

[2] https://github.com/huggingface/pytorch-transformers

		Dataset/Prompts								
ID	Model	1	2	3	4	5	6	7	8	Average
1	LSTM(last)	0.165	0.215	0.231	0.436	0.381	0.299	0.323	0.149	0.275
2	BiLSTM(last)	0.226	0.276	0.239	0.502	0.375	0.412	0.361	0.188	0.322
3	LSTM(mean)	0.582	0.517	0.516	0.702	0.604	0.670	0.661	0.566	0.602
4	BiLSTM(mean)	0.591	0.491	0.498	0.702	0.643	0.692	0.683	0.563	0.608
5	*EASE(SVR)	0.781	0.630	0.621	0.749	0.782	0.771	0.727	0.534	0.699
6	*EASE(BLRR)	0.761	0.621	0.606	0.742	0.784	0.775	0.730	0.617	0.705
7	CNN+LSTM	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
8	LSTM-CNN-att	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
9	RL1	0.766	0.659	0.688	0.778	0.805	0.791	0.760	0.545	0.724
10	SKIPFlOW	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.764
11	*HISK+BOSWE	**0.845**	**0.729**	0.684	0.829	0.833	0.830	0.804	0.729	0.785
12	RankingOnly	0.791	0.687	0.665	0.811	0.797	0.821	0.821	0.651	0.756
13	RegressionOnly	0.800	0.679	0.679	0.822	0.803	0.797	0.837	0.725	0.768
14	R^{2}BERT	0.817	0.719	**0.698**	**0.845**	**0.841**	**0.847**	**0.839**	**0.744**	**0.794**

Table 2: QWK evaluation scores on ASAP dataset (* means statistical model)

tively, and N is the number of possible ratings.

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

In addition, a matrix O is calculated, such that $O_{i,j}$ denotes the number of essays obtained a rating i by the human annotator and a rating j by the AES system. Another matrix E with the expected count is calculated as the outer product of histogram vectors of the two ratings. The matrix E is then normalized such that the sum of elements in E is the same as that of elements in O. Finally, with given matrices O and E, the QWK score is calculated according to Formula 12.

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

4.4 Baselines and Implementation Details

In this section, we list several baseline models as well as state-of-the-art models.

- ***EASE** A statistical model called Enhanced AI Scoring Engine (EASE) is an AES system that is publicly available, open-source[3], and also achieved excellent results in the ASAP competition. EASE utilizes hand-crafted features such as length-based features, POS tags, and word overlap, as well as different regression techniques. Following previous works, we report the results of EASE with the settings of Support Vector Regression (SVR) and Bayesian Linear Ridge Regression (BLRR).

- **LSTM** We use two layers LSTM and biLSTM, as well as mean pooling and last output to obtain the essay representations. Mean pooling means the average vector of all the hidden states, while the last output refers to the last hidden state. Then, a fully connected linear layer, as well as σ activation function, is used to gain scores. In these four models, GloVe (Pennington et al., 2014) is used to initialize the word embedding matrix, and the dimension is 300.

- **CNN+LSTM** This model is proposed in Taghipour and Ng (2016), which assembled CNN and LSTM to gain scores. We use the performance reported in the paper.

- **LSTM-CNN-att** Dong et al. (2017) proposed to use attention mechanisms and hierarchical neural networks to learn the representation of the essays. We also use the experiment results reported in their paper.

- **RL1** Wang et al. (2018b) proposed a reinforcement learning based model. In that paper, QWK is used as the reward function, and classification is used to gain the scores. The performance reported in the paper is used.

- **SKIPFlOW** Tay et al. (2018) proposed the

[3] https://github.com/edx/ease

ID	Model	First 512	Last 512
1	RankingOnly	0.657	0.644
2	RegressionOnly	0.724	0.725
3	R^{2}BERT	0.743	0.745

Table 3: QWK evaluation scores on Prompt 8 of ASAP Dataset with different parts of the whole essays

model, which considered the coherence when learning text representations. Experiment results in the paper are used.

- ***HISK+BOSWE** Cozma et al. (2018) proposed another statistical model. It utilized string kernel and word embedding to extract text features and gained higher performance.

Our models We not only show the performance of R^{2}BERT but also the results of the regression only version (RegressionOnly) and the ranking only version (RankingOnly). All experiments are conducted on a Linux machine running a single Tesla P40 GPU.

4.5 Experiment Results and Analysis

Table 2 shows the empirical results of all deep learning models as well as the statistical models. First, the comparison between LSTM based models is discussed. The mean pooling performs better than the last output in all LSTM based models. Since essays in the dataset contain hundreds of words, it is difficult for LSTM to capture longer dependency. Compared with the last output, average pooling could alleviate the aforementioned problem. Meanwhile, bidirectional LSTM based models perform comparably even better than the unidirectional ones. Because the bidirectional models could capture complete context information. However, these models show lower performance than that of EASE. It means well-designed hand-crafted features are more effective than simple neural networks. These models still perform worse than state-of-the-art models.

Additionally, we firstly compare published state-of-the-art results. RL1 (Wang et al., 2018b), the reinforcement learning based model, shows pretty lower performance in recent works. Since it utilizes dilated LSTM to learn essay representations, which ignores sentence-level structure information. It still outperforms basic LSTM based models, which shows the effectiveness of the QWK reward function. CNN+LSTM (Taghipour and Ng, 2016) is an ensemble model that shows comparable performance compared with LSTM-CNN-att (Dong et al., 2017), the hierarchical model, on Prompt 1,2,4,5,6,7, and even gains much higher performance on Prompt 3. Both models outperform LSTM based models. It means that the ensemble model could make up shortages of single neural networks and performs comparably with hierarchical models. Besides, LSTM-CNN-att and SKIPFLOW (Tay et al., 2018) both are hierarchical models. They capture the explicit structure through modeling the relationship of adjacent sentences (semantics) in each essay. So they perform better in Prompt 1 and 2, which contain argumentative essays. Especially the SKIPFLOW model even gains much better performance on Prompt 1. LSTM-CNN-att also performs better on Prompt 8. However, a well-designed statistical model, HISK+BOSWE Cozma et al. (2018), outperforms all previous neural models, which also performs best on the two argumentative prompts.

Compared with previous state-of-the-art neural models, the RegressionOnly model outperforms all other neural models on the average QWK score, which shows the great power of the pre-trained language model (BERT) in capturing deeply semantic information. Especially on the two narrative prompts (Prompt 7 and 8), the RegressionOnly model outperforms other models by a large margin, which shows that self-attention is more suitable for narrative essays since it can capture key concepts in narrative essays as shown in Figure 2. RankingOnly model shows much lower performance on Prompt 8 as well as average QWK score, maybe because it is difficult to utilize batch-wise order to reconstruct the global order perfectly.

R^{2}BERT outperforms RegressionOnly and RankingOnly models on each prompt by a large margin except Prompt 7. The result means that ranking and regression are surely two complementary objectives, and a combination via dynamic weights could improve the performance effectively. In general, R^{2}BERT gains a much higher average QWK score compared with the aforementioned neural models and almost performs best on each prompt except Prompt 1. It illustrates a successful way to enhance BERT on downstream tasks. Only utilizing BERT to learn text representations is not enough. Suitable auxiliary objectives are also necessary. More importantly, our model also outperforms HISK+BOSWE, the latest statistical

Prompt ID	Prompt 1	Prompt 7
Prompt	Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.	Write a story about a time when you were patient or write a story about a time when someone you know was patient or write a story in your own way about patience.
Attention Example	dear newspaper, computers have a positive effect on people because they teach hand-eye coordination, give people the ability to learn about faraway places and people and allow people to talk online with other people. the invention of computers is the single most important event of the @date1. @person1, a professor at @organization3 says that "the invention of computers has led to hundreds even thousands of new discoveries. this week alone, @caps3 have discovered @num1 new drugs that could put an end to cancer."	have you ever been in a situation when you know something good is coming or is going to happen and you just @caps1t control yourself? you ask your parents, when and they say, soon just have some patience! well this has happened to me multiple times, such as when we were going to @location1 or @location2, but on this special occasion, getting our new dog. i decided to be a mature teenager and be patient. it was @date1 @time1, the day my family was getting a dog and i was so excited. my stomach was filled with butterflies...

Figure 2: Self-attention visualization on examples of Prompt 1 and 7

model, which proves the great power of neural networks.

BERT limits the length of each input text with a maximum of 512 words. In Prompt 8, the average length of all essays is about 650 words, which is larger than the limit. We use the first 512 words or the last 512 words instead of the whole essay. Table 3 shows the experimental results. Our three models achieve similar performance. How to fully use the whole essays with BERT is a direction in future works. In Table 2, we use the average performance as the result of Prompt 8 in each model.

In Figure 2, we visualize the word weights of self-attention of two essays, including an argumentative essay from Prompt 1 and a narrative essay from Prompt 7. For the limit of the page, we only demonstrate part of each essay. In the figure, the word in darker red gains lower attention weight. The argumentative example needs to convince people that computers can benefit our life. Self-attention has identified several connectors such as ”because”, ”and”, ”even”, and some words indicating arguments including ”about”, ”that” etc. These words show the explicit logical structure of argumentative essays. The narrative example uses the example of getting a dog to show his/her patience. Self-attention capture the story details such as ”dog”, ”parent”, ”family”, ”stomach”, ”butterflies”, as well as the topic words ”patient” and ”patience”. All these words show the topics shifting of narratives.

4.6 Runtime and Memory

In this section, we analyze the runtime and memory, which means the total number of parameters. Since little previous work provided the source code so that it is difficult to estimate the total number of parameters accurately. Our three models only utilize different losses, so they have the same number of parameters. In summary, we only compare LSTM, BiLSTM, and R^{2}BERT model. Firstly, we estimate the total number of parameters for the three models. Then we record the total training time on all training samples in Prompt 6. Since simple neural networks need more training epochs to converge, yet BERT model only needs less training epochs to fine-tune. To compare the inference time, we record the time for inference per sample. All results are shown in table 4. It is obvious that BERT has more parameters and spends much more training and inference time. However, the inference time of each sample is near 1 second, which is practical in the real educational scenarios.

Model	TR	IPS	#Param
LSTM	2m53s	0.0013s	1.4M
BiLSTM	3m15s	0.0014s	1.4M
R^{2}BERT	22m20s	0.9103s	110M

Table 4: Comparison of Runtime and Memory. TR means the total training runtime on the train set and IPS means inference runtime per each test sample. #Param refers to the number of parameters.

5 Conclusion and Future works

From experimental results, we can obtain several conclusions: 1) BERT is a significantly effective model to improve the performance of downstream natural language processing tasks. 2) Regression loss and ranking loss are two complementary losses. 3) Simply fine-tuning on BERT is not enough. Multi-loss objective is an effective approach to fine-tune the BERT model. 4) Self-attention is useful to capture conjunction words and key concepts in essays. In the future, we will investigate how to utilize the whole long text with the pre-trained BERT model.

Acknowledgments

The work was supported by Hong Kong RGC Collaborative Research Fund with project code C6030-18GF and Hong Kong Red Swastika Society Tai Po Secondary School with project code P20-0021.

References

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Martin Chodorow and Jill Burstein. 2004. Beyond essay length: evaluating e-rater®'s performance on toefl® essays. *ETS Research Report Series*, 2004(1):i–38.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Daniel S Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3684–3690.

Mădălina Cozma, Andrei 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 (Volume 2: Short Papers)*, pages 503–509, Melbourne, Australia. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Scott Elliot. 2003. Intellimetric: From here to validity. *Automated essay scoring: A cross-disciplinary perspective*, pages 71–86.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: a survey of the state of the art. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6300–6308. AAAI Press.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.*

Mark D Shermis and Jill C Burstein. 2003. *Automated essay scoring: A cross-disciplinary perspective*. Routledge.

André Smolentzov. 2013. Automated essay scoring: Scoring essays in swedish.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. 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)*, pages 380–385.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

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, Austin, Texas. Association for Computational Linguistics.

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, (AAAI-18), the 30th innovative Applications of Artificial Intelligence*, pages 5948–5955.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yucheng Wang, Zhongyu Wei, Yaqian Zhou, and Xuanjing Huang. 2018b. 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, Brussels, Belgium. Association for Computational Linguistics.

Mingrui Wu, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. 2009. Smoothing dcg for learning to rank: A novel approach using smoothed hinge functions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1923–1926. ACM.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

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