AESPrompt: Self-supervised Constraints for Automated Essay Scoring with Prompt Tuning

Qiuyu Tao
Department of Computer Science
Chongqing University
Chongqing, China
TaoQiuyu@cqu.edu.cn

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

Rongzhen Li
Department of Computer Science
Chongqing University
Chongqing, China
lirongzhen@cqu.edu.cn

***Abstract*—Automated essay scoring(AES) aims to automatically assign scores to essays based on the quality of writing. Previous approaches have made many attempts with pre-trained BERT for essay scoring and achieved the state-of-the-art. However, these approaches mainly rely on the high computation cost and ignore the high similarity between text representations. In this paper, we propose a lightweight prompt-tuning framework, AESPrompt, to capture the significant semantic features of the text efficiently. We construct one continuous prompt for each layer of the frozen language model to help the language model understand the essay scoring task. Specially, we design task-related self-supervised constraints to capture discourse structure in terms of coherence and cohesion further to enhance the generalization and discourse awareness of the prompt. Experimental results on the public dataset ASAP illustrate that our approach performs competitively in the full data settings and outperforms in one-shot data settings significantly compared with fine-tuning BERT.**

***Index Terms*—Automated Essay Scoring, BERT, Prompt Tuning**

I. Introduction

Automated essay scoring(AES) aims to assign a score based on the essay quality, for essays written on a specific topic. AES is a necessary task in educational applications which can provide an efficient approach to score large-scale text and reduce human efforts remarkably. Early works in AES mainly leveraged the handcraft features such as such as grammaticality, spelling errors, and the length of essays [1]. Although AES systems based on feature engineering are explainable, it is expensive to design scoring rubrics for the new writing topics.

Existing works are mainly based on Convolution Neural Network(CNN) and Recurrent Neural Network(RNN) to learn text representations. The key challenge of neural-network-based AES systems is to learn a better text representation that can capture deep semantic features as much as possible. However, those neural networks requires more annotated essays for training. Shallow neural networks trained on limited samples show poor performance to capture deep semantics of texts which may obstruct an AES system to further ensure correct scoring.

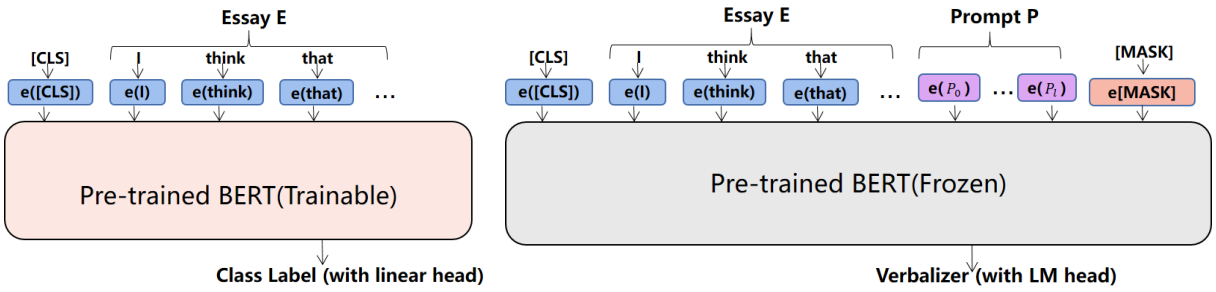
DOI reference number: 10.18293/SEKE2022-095

In recent years, pre-trained language models(PLMs) such as the BERT [2], have improved performance in many natural language downstream tasks such as text classification and sentiment analysis, which shows its extraordinary representation ability. The key component of the BERT model is the self-attention mechanism [3], which can capture the relationship between any words in the essay even the long text. Although some prior approaches utilize methods to fine-tune BERT [4], [5], these approaches are dependent on high computation costs which tune all model parameters and need to store a full copy of the model for each writing topic. Besides, previous works ignore the significant gap between pre-training and downstream tasks, which restricts BERT from reaching its full potential.

The prompt-based tuning method is proposed to narrow the gap between downstream and pre-train tasks [6]. Unlike the traditional fine-tuning method, prompt-based tuning reformulates natural language understanding (NLU) as a masked language modeling task, as the fig 1(b) shows. Formally, we make a template function $x_{prompt} = \tau(x)$ to concatenate the input with prompts and one answer slot [Z]. For a masked language model, the slot [Z] is fill with the [MASK] token. For instance, when it applies to AES, we can simply define a template $\tau(x) =$ "$[x]$ $Assign\ the\ essay\ on\ a\ scale\ of$ 0 to $4.[MASK]$". By feeding a supervised example $\{x_{prompt}, y\}$ into the masked language model M, we can determine the essay score with PLMs predicting '1' or '2' at the mask position. Prompt-based tuning can help PLMs better understand the task, meanwhile introducing no new parameters within PLMs and making it easier to fine tune.

PLMs with the fine-tuning need to store all model parameters for each downstream task. However, discrete prompts can be sub-optimal for the continuous PLMs. A recent line of work proposes the prompt-tuning paradigm [6]–[8] to adapt large PLMs to downstream tasks cheaply. Prompt tuning freezes all the parameters in PLMs but only tunes the prompts, making the method more efficient. Also, the prompts are initialized randomly and learned end-to-end, reducing the cost of manually designing the template.

Motivated by the above observations, in this paper, we propose AESPrompt, a novel prompt-tuning framework for



(a) Fine Tuning (b) Prompting

Fig. 1: Paradigms of fine-tuning(figure a) and Prompting(figure b) for automated essay scoring.

AES. We first construct a multi-layer prompt that prepends a continuous embedding into the input sequence for each layer of PLMs. Specifically, the prompts in different layers are independent, bringing more tunable parameters than other prompt-tuning methods. At the same time, it is still much smaller than the full PLMs. To further inject essay scoring self-supervised constraints into the prompt, we propose AES-related self-supervised learning to constrain the prompt including the "Discourse Indicator shuffle" and the "Paragraph Reordering Detection". Our main contributions can be summarized as follows:

- We propose AESPrompt, a prompt-tuning framework for the essay scoring task. To the best of our knowledge, this is the first approach to incorporate a prompt-based method for essay scoring.
- To better optimize the continuous prompts, we propose AES-related self-supervised constraints, including discourse indicator and paragraph order.
- We conduct experiments on the ASAP dataset with the $BERT_{base}$ model. Experimental results not only illustrate the effectiveness of AESPrompt in full data settings but also reinforce the stability in low-resource settings.

II. Related Work

A. Automated Essay Scoring

Early studies about the AES task starts with feature engineering. The systems use textual features designed by human experts [1]. The latter type of researches use deep neural networks to extract features automatically. Taghipour and Ng [9] first propose a neural method based on CNN and LSTM to learn essay representation for essay scoring. Many works improve AES based on that [10]–[14] BERT has achieved state-of-art results on many downstream NLP tasks. some prior works find BERT sentence embedding is useful for the ASAP data [5], [13], [15], [16]. TSLF [15] calculates the semantic score, coherence score, and prompt-relevant score during the first stage, and then concatenates handcraft features for further training. While Nadeem et al. [13] finds that token and sentence embedding from BERT makes no significant improvement. Their work explores discourse-based pre-training tasks and contextualized embedding and proposes a discourse-aware neural framework. R^2BERT [5] model is proposed to solve the essay scoring task and essay ranking task jointly. The model is fine-tuned by a multi-loss approach which combines the scoring MSE loss and a ranking error loss based on ListNet.

B. Prompt-based learning

Prompt-based methods are inspired by the birth of GPT-3 [17], which reformulate downstream tasks to language modeling tasks with textual templates and a verbalizer. The prompting method is first applied as a knowledge probe [18]. However, handcraft prompts heavily depend on the experience of designers, so some works explore automatically generating discrete prompts via gradient-based search [7], [19]. Shin et al. [8]propose an approach to generate prompts in vocabulary automatically. Compared with the fine-tuning method, the prompting method freezes all model parameters, which may lead to the volatile performance of the model in many cases. [20], [21]. Prompt tuning is proposed to only tune the continuous prompts and outperforms prompting in many tasks. Han et al. [22] propose prompt tuning with rules for text classification, Chen et al. [23] applied prompt-tuning with synergistic optimization on relation extraction. Recently, some works focus on optimizing continuous prompts for every layer of pre-trained model [24], [25]. In this paper, we propose a novel Prompt Tuning framework for the AES task. Besides, we inject AES-related self-supervised to constraint the prompt. To our knowledge, we are the first to apply prompt-based method to Automated essay scoring.

III. AESPrompt

In this section, we introduce our AESPrompt framework as shown in Fig 2.

A. Prompt Encoder

The overall framework involves a language model to learn text representation, which is then used for essay scoring. Following the deep prompt tuning approach as in P-Tuning-v2 (PT2) [26] which is an NLU version of prefix-tuning [24]. PT2 keeps all pre-train language model parameters frozen and only tunes the prompt parameters. We regard the AES task as a regression task and predict the score via [CLS] token. First,

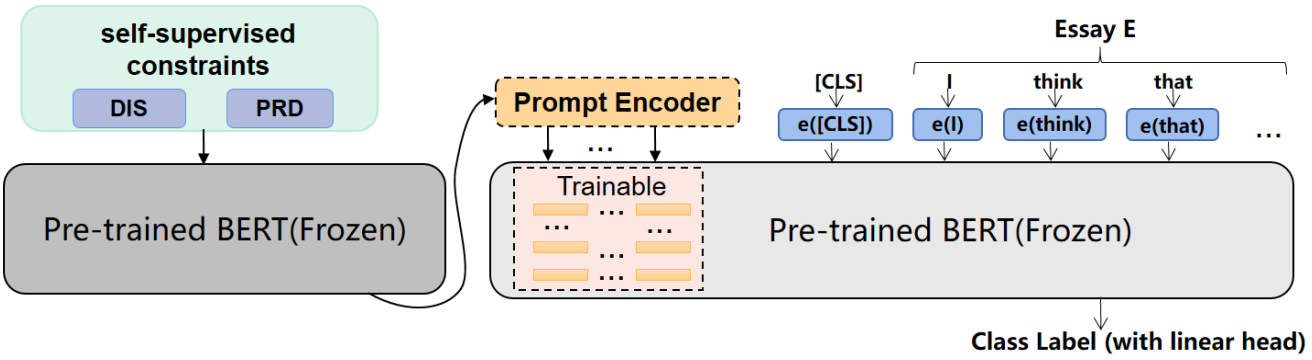


Fig. 2: Model architecture of AESPrompt. We design two self-supervised constraints for Prompt tuning including **D**iscourse **I**ndicator **S**huffle(DIS) and **P**aragraph **R**eordering **D**etection(PRD).

for a given sample essay $x = \{w_1, w_2, \ldots, w_n\}$, it is should be tokenized into a new sequence $\tilde{x} = \{[CLS], e_1, e_2, \ldots, e_n\}$, where n is the number of words and [CLS] is a special classification token. We conduct M to obtain the hidden representations of the inputs $h = M(\tilde{x}) \in \mathbb{R}^{|\tilde{x}| \times d}$, where $|\tilde{x}|$ is the sequence length. To reduce the objective gap between pre-training and AES, PT2 prepends prompt for each layer of M as additional keys $K^P \in \mathbb{R}^{L \times d}$ and values $V^P \in \mathbb{R}^{L \times d}$ to the multi-head self-attention mechanism, where L is the prompt length and d is the dimension of word embedding. The new text representation is obtained through attention as the show below:

$$\mathrm{Att}(Q, K) = \mathrm{softmax}(\frac{Q[K^P, K]^T}{\sqrt{d}}) \quad (1)$$

$$V_{att}(Q, K, V) = \mathrm{Att}(Q, K) \cdot [V^P, V] \quad (2)$$

where $[,]$ refers to the concatenation operation. Then, the hidden representation is mapped to CLS token $h_{[CLS]}$. Since score ranges are different from each other, during training the gold scores are normalized into the range of [0,1] first. Then during the test process, map the predicted scores to the original score ranges. We can thus conduct a linear layer with activation function to project the $h_{[CLS]}$ to a scalar value as formula (3), where W is weight matrix and b is a bias initialized by the mean gold score of training data [9].

$$\acute{y} = \mathrm{Sigmoid}\left(\mathrm{W}h_{[\mathrm{cls}]} + b\right) \quad (3)$$

B. Prompt Tuning with self-supervised constraints

Simply random initializing prompts with continuous embeddings brings difficulties to optimization. Fortunately, neural networks can utilize related tasks to improve performance through pre-training. To inject essay scoring self-supervised constraints into prompt, we design self-supervised learning with discourse indicator shuffle and paragraph reordering to pre-train our prompts. The self-supervised constraints further enhance the prompt's generalization and document structure awareness. Instead of using additional data, we generate pre-train data from the original data as shown in Fig 3. We introduce the details in the following sections.

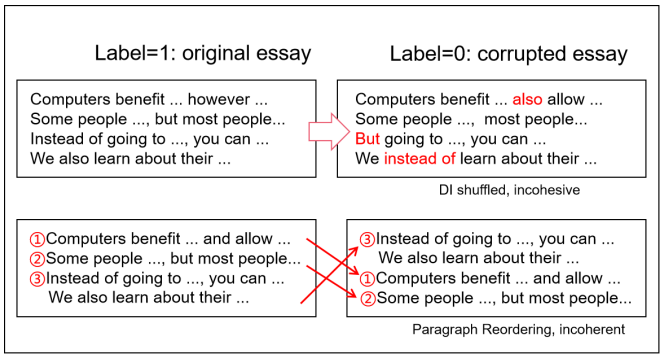


Fig. 3: Proposed self-supervised constraints which utilizing coherent/cohesive and incoherent/incohesive texts for Prompt Tuning

1) Discourse Indicator Shuffle: To strengthen the bidirectional representation on the AES task, we construct the discourse indicator shuffle task. The discourse indicator refers to the conjunction indicating the relationship between sentences (e.g. "however", "else" and "while"). Though DIs has a somewhat empty meaning, without sufficient DIs in a piece of writing, a text would lack logic and the connection between different sentences and paragraphs will be unfluent. We design a binary classification to detect whether discourse indicators are shuffled or not. To simplify, for one DI token is chosen, 1) we replace the token with other DI randomly 60% with the time. 2) delete the DI 20% with the time, 3) unchanged the token 20% of the time. For example, "they use an online catalog because it's cheaper" is cohesive. "they use an online catalog but it's cheaper" and "they use an online catalog, it's cheaper" is incohesive.

2) Paragraph Reordering Detection: The AES task is based on understanding not only the relationship between two sentences but also paragraphs while the relationship between paragraphs is not modeled directly when pre-training. With the hypothesis that many student essays follow a logical structure like, "introduction-body-conclusion". We propose to reorder paragraphs that divide the document into three parts and each part consists of one or more complete sentences.

And then, we permute them into a certain permutation. Since the permutations show great influence in representation learning, we choose the permutations with the maximal average Hamming distance [27]. We use three possible permutations $P = \{(1,2,3),(2,3,1),(3,1,2)\}$ in our experiments. We label the essay with permutation $P_i = \{(1,2,3)\}$ as coherent and label another two permutations as incoherent.

C. Training

a) The scoring Task: is treated as a regression task. We use the Adam optimization algorithm to minimize the mean squad error (MSE) function. Given a training essays of size N, y_i and $\acute{y_i}$ are the corresponding gold and predicted score for i-th eassy, separately. The loss function is shown in Formula(4) :

$$\mathrm{L_s}(\mathrm{x_i}, \mathrm{y_i}) = \mathrm{MSE}(\mathrm{y_i}, \acute{y_i}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \acute{y_i})^2 \tag{4}$$

b) Self-supervised Constraints: We create the training instances for self-supervised constraints as section 3.2 mentioned. Since the tasks are treated as binary classification tasks, we use the cross-entropy loss function for self-supervised constraints, respectively. Where $\mathrm{c_i}$ and $\acute{c_i}$ are the corresponding gold and predicted label of the input essay $\tilde{x_i}$. Note that $\mathrm{c_i}$ is automatically assigned in the corruption process where an original essay has a label of 1 and an artificially corrupted essay has a label of 0.

$$\mathrm{L_d}(\tilde{x_i}) = -\sum_{j=1}^{M} \mathrm{c_i}\log(\acute{c_i}) - (1-\mathrm{c_i})\log(1-\acute{c_i}) \tag{5}$$

IV. Experiments

TABLE I: Details of ASAP Dataset.

Set	Score Range	Type of essay	Mean length
1	2-12	persuasive	350
2	1-6	persuasive	350
3	0-3	source dependent response	150
4	0-3	source dependent responses	150
5	0-4	source dependent responses	150
6	0-4	source dependent responses	150
7	0-30	narrative	250
8	0-60	narrative	650

In this section, we first introduce the ASAP dataset and evaluate metrics. And then we describe the experimental setup and present the results.

A. Dataset and Metrics

The Automated Student Assessment Prize(ASAP) dataset is provided by a Kaggle competition which contains eight different essay sets written by students from grade 7 to grade 10. This dataset has become the most widely used in the field of AES which is composed of 12976 labeled essays. More details about the ASAP are summarized in Table I.

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

We employ the quadratic weighted kappa(QWK) as the evaluation metric, which is the official evaluation metric adopted by ASAP competition. Quadratic weight kappa measures the agreement between gold scores and automated scores. The QWK is calculated as follows, an N by N weight matrix is calculated first according to formula (6):

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{6}$$

where i refers to the gold score, j refers to the predicted score (assigned by the AES model) and N is the total number of essays. Second, we construct the confusion matrix O, that $O_{i,j}$ corresponds to the number of essays rated i by human rater and rated j by AES model. Then, an expected matrix E is calculated as the outer product between gold scores and predict scores. The matrix E is normalized such that E and O have the same sum. Finally, from the three matrices, the QWK is calculated as formula (7):

$$k = 1 - \frac{\sum_{i,j} W_{i,j}O_{i,j}}{\sum_{i,j} W_{i,j}E_{i,j}} \tag{7}$$

B. Experimental settings

We explore the following setups to train AESPrompt models for ASAP essays :

1) Training using only ASAP essay data;
2) Pretraining with either DIS or PRD data, followed by training with the essay data.
3) Pretraining with DIS and PRD data, followed by training with essay data.

For all our experiments, we use the "BERT-base-uncased" model as the base model, and the training is implemented on PyTorch with an Nvidia A6000 GPU. In our work, the linear learning rate policy is used to tune the parameters, and the max learning rate is set to 1e-3. In full data experiments, closely following the settings as [9], we conduct five-fold cross-validation with a 3:1:1 split for training, validation, and test to evaluate our method. We report the average QWK across the five folds. For one-shot data experiments, we follow Gao et. al [7], which assumes development data has the same size as train data to select model and hyper-parameters. And we repeat the sampling of one-shot labled data 5 times and the average results are reported. Consistently, we set prompt length to 40, as a result, the tunable parameters are only 80k, compared with 110M parameters of BERT fine-tuning, our method only needs to store additional 0.7% for each essay set.

C. Main Results

We evaluate the performance on the ASAP dataset, the main results are shown in Table II. To put our results in perspective, we compare our method with several baseline models. Enhanced AI Scoring Engine (EASE) is a statistical model based

https://github.com/huggingface/transformers
https://github.com/edx/ease

TABLE II: The performance (QWK) of all comparison methods on ASAP dataset. The best measures are in bold. * denotes statistical model.

settings	Models	1	2	3	4	5	6	7	8	avg
full-data	EASE(SVR)*	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
	EASE(BLRR)*	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705
	CNN+LSTM	0.821	0.688	0.694	0.805	0.807	**0.819**	0.808	0.644	0.761
	LSTM-CNN-att	0.822	0.682	0.672	**0.814**	0.803	0.811	0.801	0.705	0.764
	SKIPFLOW	**0.832**	0.684	**0.695**	0.788	**0.815**	0.810	0.800	0.697	0.765
	BERT	0.809	0.661	0.692	0.808	0.800	0.801	**0.834**	0.720	0.765
	P-Tuning-V2	0.781	0.640	0.677	0.758	0.794	0.798	0.825	0.717	0.749
	AESPrompt(DIS)	0.802	0.680	0.680	0.765	0.807	0.801	0.825	0.722	0.760
	AESPrompt(PRD)	0.788	0.650	0.670	0.784	0.793	0.803	0.826	**0.727**	0.755
	AESPrompt(ALL)	0.808	**0.689**	0.685	0.790	0.803	0.806	0.833	0.724	**0.767**
one-shot	BERT	0.625	0.545	0.431	0.515	0.647	0.485	0.664	0.646	0.572
	P-Tuning-V2	0.568	0.522	0.554	0.649	0.681	0.610	0.664	0.613	0.607
	AESPrompt(DIS)	0.680	0.532	0.585	0.660	0.698	0.617	0.669	0.598	0.630
	AESPrompt(PRD)	0.658	0.542	0.566	0.667	0.685	0.613	0.678	0.603	0.627
	AESPrompt(ALL)	**0.682**	**0.544**	**0.590**	**0.672**	**0.701**	**0.622**	**0.683**	**0.620**	**0.639**

on hand-crafted features followed by support vector regression (SVR) and bayesian linear ridge regression (BLRR) [28]. CNN+LSTM [9] is proposed to assemble CNN and LSTM to predict the essay rating. CNN-LSTM-Att [12] introduces hierarchical neural networks with attention mechanism to learn the representation of essays. SKIPFLOW [14] considers the coherence when learning text representations. BERT [2] is employed as an encoder for the AES task. P-Tuning-v2 [26] performs deep prompt tuning, which prepends prefix prompts in the input of model's hidden layer. AESPrompt(DIS) employs the discourse indicators shuffle constraint. AESPrompt(PRD) only includes the paragraph reordering detection constraint. AESPrompt(ALL) employs both two constraints. The BERT fine-tuning and SKIPFLOW give a strong baseline, the average QWK across eight sets is 0.765 in the full-data setting. LSTM-CNN-att and SKIPFLOW both are hierarchical models which explicitly capture the adjacent semantics in each essay. So they perform better in set 1, 3, 4, 5 and 6. We can see that the AESPrompt method slightly outperforms in a resource-rich setting. AESPrompt shows obvious advantages on two narrative essay sets(set 7 and 8). By incorporating the self-supervised constraints, the proposed framework dramatically improves the accuracy of PT2 at an average of 0.018 QWK.

To further evaluate the potential of our method, we conduct one-shot setting experiments on the ASAP dataset. We compare our approach with BERT fine-tuning and P-Tuning-v2. AESPrompt significantly outperforms the BERT fine-tuning and P-tuning-v2 in one-shot settings, which shows AESPrompt appears to be more beneficial in low-resource settings. Specifically, AESPrompt can obtain grains of up to 11.7% improvement on average compared with BERT fine-tuning. We can find out that AESPrompt outperforms in all sets. What is more, We also observe that our results suffer from high variance. The performance fluctuates up to 15% QWK under different randomly sampled D_{train} and D_{dev}. In one-shot settings, truncating text that exceeds the length may have a great impact on AESPrompt. We will explore these problems in the future.

TABLE III: Comparison of Runtime and Memory. TR means the total training time on the train set and IPS means inference runtime per each test sample. Parameters refer to the number of tuned parameters.

Model	TR	IPS	Parameters
BERT fine-tuning	256	0.067	110M
P-Tuning-v2	179	0.062	80k
AESPrompt	185	0.062	80k

D. Ablation Study

We explore the effects of the self-supervised constraints for the AESPrompt, by removing each of them individually. These self-supervised constraints include: discourse indicators shuffle, and Paragraph reordering detection. As shown in Table II, after removing one of them from AESPrompt, the performance decrease a lot. These indicate that the self-supervised constraints we proposed can enhance the prompts discourse awareness from paragraph level and discourse indicator level. In addition, the performance of AESPrompt(PRD) is worse than AESPrompt(DIS) which indicates that using the RPD constraint alone may fail to benefit the general regression model.

E. Runtime and Memory

Our secondary evaluation is based on the runtime and resource usage which means the total number of parameters. In summary, we main compare BERT fine-tuning, P-Tuning-v2 and AESPrompt model as Table III shows. Firstly, we estimate the total tuned parameters for the three models. Then, We take essay set 1 as an example to compare the model runtime. Since the prompt tuning needs more training epochs to converge than BERT fine-tuning that we record the total training time for each method. And we record the inference time on one sample to compare the efficiency of inference. In our approach, we freeze all parameters in the language model that reduce the storage and computation consumption. It's practical in real educational scenarios that AESPrompt can reach a reasonable

performance on scoring task only needs to store additional 80k parameters for a new essay scoring set.

V. Conclusion

In this work, we propose a lightweight prompt tuning framework with self-supervised constraints, AESPrompt, for automated essay scoring. Specifically, we propose two AES-related self-supervised constraints to pre-train the prompt which further reduces the intrinsic gap between the language model distribution and the target data distribution. In this way, both full-data and the one-shot performance can be boosted. Compared with standard BERT fine-tuning, our method is lightweight, which only tunes 80k parameters compared with 110M. Experimental results show that the proposed method achieves significant improvement on one-shot AES and competitive results on full-data AES. In this case, our approach is meaningful for the practical of PLMs in automated essay scoring. In the future, we plan to explore how to design unified task formats and the corresponding auxiliary task on eight sets.

Acknowledgment

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

References

[1] H. Yannakoudakis, T. Briscoe, and B. Medlock, "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*, 2011, pp. 180–189.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv preprint arXiv:1706.03762*, 2017.

[4] E. Mayfield and A. W. Black, "Should You Fine-Tune BERT for Automated Essay Scoring?" in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Seattle, WA, USA → Online: Association for Computational Linguistics, 2020, pp. 151–162.

[5] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "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*. Online: Association for Computational Linguistics, 2020, pp. 1560–1569.

[6] T. Schick and H. Schütze, "Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference," *arXiv preprint arXiv:2001.07676*, 2021.

[7] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *ACL/IJCNLP (1)*, 2021.

[8] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[9] K. Taghipour and H. T. Ng, "A Neural Approach to Automated Essay Scoring," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016, pp. 1882–1891.

[10] D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic Text Scoring Using Neural Networks," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 715–725, 2016.

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

[12] F. Dong, Y. Zhang, and J. Yang, "Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 153–162.

[13] F. Nadeem, H. Nguyen, Y. Liu, and M. Ostendorf, "Automated Essay Scoring with Discourse-Aware Neural Models," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 484–493.

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

[15] J. Liu, Y. Xu, and Y. Zhu, "Automated Essay Scoring based on Two-Stage Learning," *arXiv preprint arXiv:1901.07744*, 2019.

[16] A. Sharma, A. Kabra, and R. Kapoor, "Feature enhanced capsule networks for robust automatic essay scoring," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 365–380.

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[18] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, "Language Models as Knowledge Bases?" *arXiv preprint arXiv:1909.01066*, 2019.

[19] K. Hambardzumyan, H. Khachatrian, and J. May, "Warp: Word-level adversarial reprogramming," *arXiv preprint arXiv:2101.00121*, 2021.

[20] B. Lester, R. Al-Rfou, and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 3045–3059.

[21] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "Gpt understands, too," *arXiv preprint arXiv:2103.10385*, 2021.

[22] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "PTR: Prompt Tuning with Rules for Text Classification," *arXiv:2105.11259*, 2021.

[23] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, and H. Chen, "Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction," *arXiv preprint arXiv:2104.07650*, 2021.

[24] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597.

[25] G. Qin and J. Eisner, "Learning How to Ask: Querying LMs with Mixtures of Soft Prompts," *arXiv preprint arXiv:2104.06599*, 2021.

[26] X. Liu, K. Ji, Y. Fu, Z. Du, Z. Yang, and J. Tang, "P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks," *arXiv preprint arXiv:2110.07602*, 2021.

[27] Y. Cao, H. Jin, X. Wan, and Z. Yu, "Domain-adaptive neural automated essay scoring," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1011–1020.

[28] P. Phandi, K. M. A. Chai, and H. T. Ng, "Flexible domain adaptation for automated essay scoring using correlated linear regression," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 431–439.