PMAES: Prompt-mapping Contrastive Learning for Cross-prompt Automated Essay Scoring

Yuan Chen and Xia Li[*]
School of Information Science and Technology,
Guangdong University of Foreign Studies, Guangzhou, China
{yuanchen, xiali}@gdufs.edu.cn

Abstract

Current cross-prompt automated essay scoring (AES) is a challenging task due to the large discrepancies between different prompts, such as different genres and expressions. The main goal of current cross-prompt AES systems is to learn enough shared features between the source and target prompts to grade well on the target prompt. However, because the features are captured based on the original prompt representation, they may be limited by being extracted directly between essays. In fact, when the representations of two prompts are more similar, we can gain more shared features between them. Based on this motivation, in this paper, we propose a learning strategy called "prompt-mapping" to learn about more consistent representations of source and target prompts. In this way, we can obtain more shared features between the two prompts and use them to better represent the essays for the target prompt. Experimental results on the ASAP++ dataset demonstrate the effectiveness of our method. We also design experiments in different settings to show that our method can be applied in different scenarios. Our code is available at https://github.com/gdufsnlp/PMAES.

1 Introduction

Automated Essay Scoring (AES) aims to evaluate the quality of essays automatically. Compared with human grading process, a robust AES system can not only reduce the work of teachers, but also improve the consistency of grading (Hearst, 2000;Weigle, 2002) and make it broadly available to language learners.

AES has been studied for many years. Early studies focus more on handcrafted features, such as lexical features (Rudner and Liang, 2002;Attali and Burstein, 2006;Yannakoudakis et al., 2011). With the rise of deep learning, many studies based on neural networks for prompt-specific settings have been proposed and achieved better results (Dong et al., 2017; Tay et al., 2018; Liao et al., 2021;Xie et al., 2022). These studies follow the same setting, that is, both rated training essays and unrated test essays belong to the same prompt.

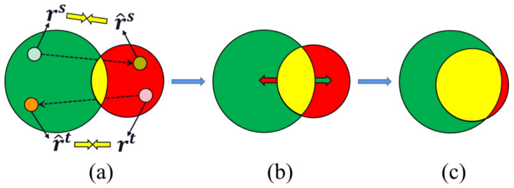

Figure 1: A summary of our motivations. The green circle represents the source prompt's representations, while the red circle represents the target prompt's representations. The yellow area reflects the features shared by both prompts.

Another type of work is cross-prompt AES. In this setting, labeled training essays are from source prompts and unlabeled test essays are from a different target prompt. Existing studies mainly focus on obtaining sufficient shared features between source and target prompts to grade the target prompt essays effectively. Some of them obtain shared features by extracting handcrafted features (Phandi et al., 2015; Ridley et al., 2020; Ridley et al., 2021) while others learn shared features by optimizing additional training objectives, such as the multi-task learning (Cummins et al., 2016), two-stage strategy (Jin et al., 2018; Li et al., 2020) and self-supervised learning task (Cao et al., 2020). Although these methods can effectively capture shared features between different prompts, we argue that these features are captured based on the original representations of the essays from source and target prompts. It may be limited by directly extracting the shared features among them.

Intuitively, when the representations of the essays from the source and target prompts are more consistent, they can share more knowledge between

[*] Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1489–1503
July 9-14, 2023 ©2023 Association for Computational Linguistics

them. To this end, we propose a prompt representation learning framework for cross-prompt AES (PMAES) in which we design a prompt-mapping contrastive learning strategy to effectively learn about more consistent representations of source and target prompts. To do this, we design a mapping operation to project each essay from the source prompt to the target prompt and get its mapping representation specific to the target prompt. For each essay on the source prompt (let's say r^s), we first determine how similar it is to all the essays in the target prompt by their original representations (e.g., by taking the dot product with the inverse matrix of the representations of all the essays in the target prompt) as the weights of the r^s to each target essay. Then, we employ a learnable parameter matrix (specifically, a prompt-mapping matrix) to acquire the weighted representation of the source prompt essay projected on the target prompt to express the mapping representation of the source essay r^s (let's say $\hat{r}^s$). These source essay representations and source mapping representations are treated as the source-to-target mapping pairs (r^s, $\hat{r}^s$). By decreasing the distance between the essays in these mapping pairs, we may gradually reduce the discrepancy between the source and target prompts and finally make the representations of the two prompt essays more consistent. It is worth noting that the above description is about mapping from source to target. Naturally, we also perform target-to-source prompt mapping operations to further learn a more consistent representations of the two prompts, which will be described in Section 3.4.

As demonstrated in Figure 1, given the original essay representations of a source and a target prompt (which we marked in green and red, respectively), there are very few shared features between them under the original representations (which we marked in yellow). When we train the model using our proposed prompt-mapping approach, the representations of the two prompts may become more similar, which enables more shared features across the two prompts. We show them in Figure 1(b) and Figure 1(c). As the shared features increase, we can get more accurate representations of target prompt essays and grade them more accurately. To summarize, the main contributions of our work are as follows:

1) To the best of our knowledge, this is the first attempt to explore the learning of consistent representations of different prompts by introducing a prompt-mapping learning strategy in order to obtain more shared features between the source and target prompts.

2) We conduct comprehensive experiments on the ASAP++ dataset, and the results show that our approach outperforms the state-of-the-art model on both single-overall and multi-attribute scoring tasks. Also, the prompt consistency experiments show that our method can make source and target prompts much more similar to each other.

3) We further design three types of source-target settings. The results show that our approach can be adapted to multiple scenarios.

2 Related Work

2.1 Prompt-specific AES

Prompt-specific AES aims to train and test essays on the same prompt. Early studies (Rudner and Liang, 2002;Attali and Burstein, 2006; Mohler and Mihalcea, 2009; Persing and Ng, 2013; Sakaguchi et al., 2015; Sultan et al., 2016) rate essays by extracting handcrafted features to train a machine learning model. Recently, with the rise of deep learning, a growing number of studies (Taghipour and Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017; Dasgupta et al., 2018; Li et al., 2018; Tay et al., 2018; Uto et al., 2020; Hussein et al., 2020; Ma et al., 2021; Liao et al., 2021; Wang et al., 2022; Xie et al., 2022) propose scoring models based on neural networks and achieve promising results.

2.2 Cross-prompt AES

Cross-prompt AES aims to train models from labeled source prompt essays and rate target prompt essays. Phandi et al. (2015) train the Bayesian linear ridge regression algorithm from the source prompt using manual features, then test it directly on the target prompt. Cummins et al. (2016) adopt multi-task learning to address the problem of prompt adaptation. Jin et al. (2018) propose a two-stage approach for the problem of cross-prompt AES. In the first stage, they train a RankSVM on prompt-independent features to obtain pseudo-labels for target prompt essays. In the second stage, a neural network model learns more prompt-dependent features in the pseudo-labeled essays. Li et al. (2020) also adopts a two-stage approach to train a model to learn common knowledge and provide pseudo labels for target prompt essays in the first stage, then use a Siamese framework to learn more prompt-dependent features in the second stage. Cao et al. (2020) train sentence reordering and noise identification tasks with adversarial training to improve the domain adaptability of the model. Ridley et al. (2020) utilize the handcrafted features to provide prompt agnostic information and achieve good results. Ridley et al. (2021) expand this prompt-agnostic information for multi-attribute scoring tasks.

2.3 Contrastive Learning

Contrastive learning is an unsupervised learning method originally used in computer vision (Hadsell et al., 2006). The main idea is to gradually bring the anchor and its positive samples closer together in a shared semantic space while distinguishing the anchor from other samples, such as the work of Chen et al. (2020). Recently, contrastive learning has shown satisfactory results in textual representation learning. Data augmentation is a general strategy for obtaining positive samples, such as translation (Han et al., 2022), synonym replacement (Wang et al., 2021), word repetition (Wu et al., 2022) or textual representation perturbation (Gao et al., 2021; Yan et al., 2021).

3 Our Approach

The whole architecture of our approach is shown in Figure 2. It contains three components: shared encoder, scorer and prompt-mapping contrastive learning. The shared encoder provides a shared representation for the other two components, the scorer is used to predict the score, and the prompt-mapping contrastive learning is used to maximize the consistency of source and target prompts.

3.1 Task Definition

Given source prompt data $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{P}$ and target prompt data $D_t = \{x_i^t\}_{i=1}^{Q}$, where $x_i^{s/t}$ is the i-th essay in source/target prompt, P and Q are the number of essays in the source and target prompts. For single-overall scoring task, y_i^s is the overall score of source prompt essay x_i^s, and for multi-attribute scoring task, $y_i^s = \{y_i^{s1}, y_i^{s2}, ..., y_i^{sK}\}$ is the set of attribute scores, and y_i^{s1} is the overall score. The task of our approach is to train a model with D_s and D_t as inputs and output the score of all target prompt essays. The complete algorithm is shown in Algorithm 1.

Algorithm 1: Procedure of our approach

```
Input: $\{(x_i^s, y_i^s)\}_{i=1}^{P}$, $\{x_i^t\}_{i=1}^{Q}$
Output: shared encoder $\mathcal{F}$, scorer $\mathcal{G}$
Calculate $I_s$ and $I_t$ using Eq. 14;
for sampling mini-batch do
    $r_i^s = \mathcal{F}(x_i^s)$, $r_i^t = \mathcal{F}(x_i^t)$;
    Calculate $\hat{r}_i^s$ and $\hat{r}_i^t$ using Eq. 15;
    Calculate $\mathcal{L}_{s\to t}$ and $\mathcal{L}_{t\to s}$ using Eq. 16
      and Eq. 17 ;
    $\mathcal{L}_{pm} = \mathcal{L}_{s\to t} + \mathcal{L}_{t\to s}$;
    if single-overall scoring task then
        Calculate $z_i^s$ using Eq. 5;
        Calculate $\hat{y}_i^s$ using Eq. 6;
        Calculate $L_{aes\_so}$ using Eq. 7;
        if epoch=1 then
            Update $\mathcal{F}$ and $\mathcal{G}$ minimizing
              $\mathcal{L}_{aes\_so}$;
        else
            Update $\mathcal{F}$ and $\mathcal{G}$ minimizing
              $\mathcal{L}_{pm}$ and $\mathcal{L}_{aes\_so}$;
    if multi-attribute scoring task then
        Calculate $\{z_i^{sk}\}_{k=1}^{K}$ using Eq. 8;
        Calculate $\{\hat{y}_i^{sk}\}_{k=1}^{K}$ using Eq. 9;
        Calculate $\mathcal{L}_{aes\_ma}$ using Eq. 10;
        Calculate $\mathcal{L}_{cor}$ using Eq. 13;
        Update $\mathcal{F}$ and $\mathcal{G}$ minimizing
          $\mathcal{L}_{aes\_ma}$, $\mathcal{L}_{pm}$ and $\mathcal{L}_{cor}$;
```

3.2 Shared Encoder

To better encode essays, we use the hierarchical structure proposed by Dong et al. (2017) as a shared encoder, in which the sentence-level representation is extracted by CNN and attention pooling from words, and LSTM and another attention pooling are used to capture essay-level representation from all sentences. In this paper, as with Ridley et al. (2021), we use POS embedding[1] to represent the essay text due to their ability to obtain better generalized representations. Suppose each essay is composed of n sentences, and each sentence contains m words. We use w_i to denote the POS embedding of each word for convenience. Then, the sentence-level representation is captured by CNN with attention pooling:

$$c_i = \mathrm{CNN}([w_i : w_{i+l-1}]),\ i = 1, 2, ..., m \quad (1)$$

$$s_t = \mathrm{attention}([c_1 : c_m]) \quad (2)$$

[1] We use the NLTK (http://www.nltk.org)

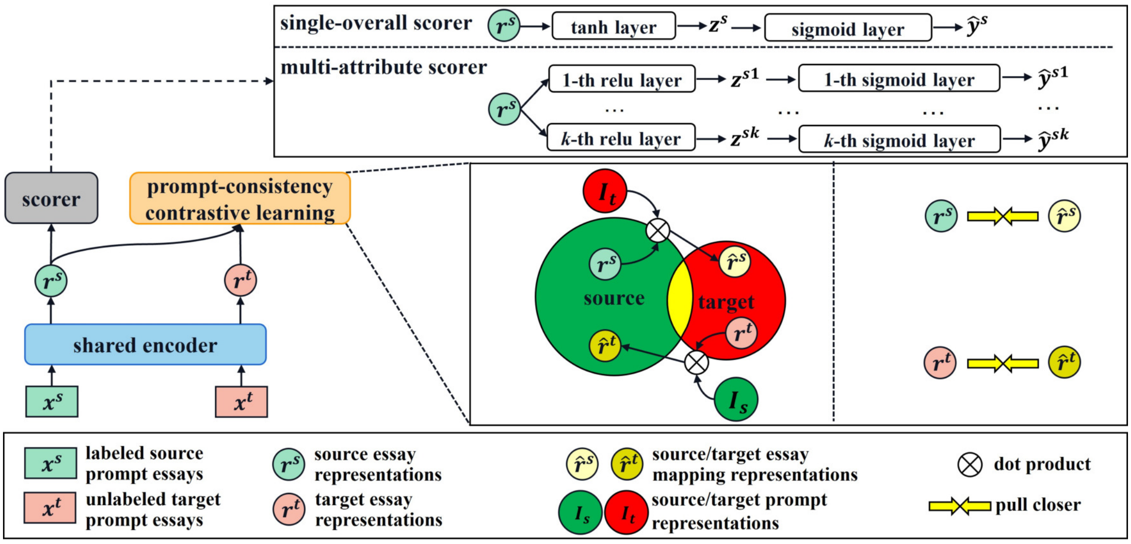

Figure 2: Overview of our model. Image on the left shows the whole architecture of our model and image on the right shows the operation of prompt-mapping contrastive learning.

where l is the kernel size of CNN, c_i is the output of the convolution operation applied to i-th POS embedding, and s_t is the representation of t-th sentence.

The essay-level representation is captured by LSTM with another attention pooling:

$$h_t = \text{LSTM}(s_{t-1}, s_t),\ t = 1, 2, ..., n \tag{3}$$

$$r = \text{attention}([h_1 : h_n]) \tag{4}$$

where h_t is the output of LSTM at the t-th time step, and r is the final essay representation.

3.3 Scorer

In this paper, we evaluate our approach both on single-overall scoring task and multi-attribute scoring task. Therefore, we have two types of scorers, corresponding to two forms of loss function. We also use the same handcrafted features as Ridley et al. (2021), denoted as **f**.

3.3.1 Single-overall Scorer

For single-overall scoring task, firstly, we concatenate the essay representation r and handcrafted features **f**, denoted as $[r; \mathbf{f}]$. Then, feeding it into a $\tanh$ dense layer to get z. Finally, another dense layer with sigmoid activation is applied to predict the overall score $\hat{y}$. The corresponding equations are as follows (Eq. 5 and Eq. 6):

$$z = \tanh(W_z[r; \mathbf{f}] + b_z) \tag{5}$$

$$\hat{y} = \sigma(W_y z + b_y) \tag{6}$$

where W_z and W_y are the trainable weight matrices, b_z and b_y are the bias vectors, σ is the sigmoid function. We use mean squared error (MSE) as the loss function, defined as follows:

$$\mathcal{L}_{aes\_so} = \frac{1}{N}\sum_i^N (\hat{y}_i - y_i)^2 \tag{7}$$

where N is the number of essays in a batch.

3.3.2 Multi-attribute Scorer

For multi-attribute scoring task, we first input the essay representation r into a specific relu dense layer to get the representation z^k of the k-th attribute. Then, concatenating z^k with **f** and feeding into a specific sigmoid dense layer to predict the k-th attribute score $\hat{y}^k$. The corresponding equations are as follows (Eq. 8 and Eq. 9):

$$z^k = \text{relu}(W_z^k r + b_z^k) \tag{8}$$

$$\hat{y}^j = \sigma(W_y^k[z^k; \mathbf{f}] + b_y^k) \tag{9}$$

where W_z^k and W_y^k are the trainable weight matrices , b_z^k and b_y^k are the bias vectors. Suppose the total number of attributes is K, the multi-attribute scoring loss is defined as follows:

$$\mathcal{L}_{aes\_ma} = \frac{1}{NK}\sum_i^N \sum_k^K (\hat{y}_i^k - y_i^k)^2 \tag{10}$$

It should be noted that not all essays have all attributes (as shown in Table 5). So we use the mask mechanism proposed by Ridley et al. (2021)

to account for the attributes without gold scores when calculating the loss.

$$mask_i^k = \begin{cases} 1, \; if \; y_i^k \in y_i \\ 0, \; otherwise \end{cases} \tag{11}$$

$$y_i = y_i \; \otimes \; mask_i \; , \hat{y}_i = \hat{y}_i \; \otimes \; mask_i \tag{12}$$

In addition, we believe that when predicting one attribute score, the other attributes can provide useful information for it. Therefore, we propose an inter-attribute correlation loss L_{cor}.

$$\mathcal{L}_{cor} = \frac{1}{K} \sum_{i}^{N} \sum_{k}^{K} -\log\left(\sum_{j, j \neq k}^{K} g(z_i^k, z_i^j) \right) \tag{13}$$

where $g(z_i^k, z_i^j) = \exp(\cos(z_i^k, z_i^j)/\rho)$, $\cos(\cdot)$ is the cosine similarity function, and ρ is a hyper-parameter. The goal of L_{cor} is to maximize the mutual information among all attributes.

3.4 Prompt-mapping Contrastive Learning

In order to capture more shared features between the source and target prompts, we propose a prompt-mapping contrastive learning strategy to learn about more consistent representations of source and target prompts. For convenience, let's take the source-to-target prompt mapping as an example to describe our method in detail. The target-to-source prompt mapping is the same operation.

Firstly, we use shared encoder $\mathcal{F}$ to encode all source and target prompt essays in training data to obtain the source prompt representation $I_s \in \mathbb{R}^{P*u}$ and the target prompt representation $I_t \in \mathbb{R}^{Q*u}$ (as shown in Eq. 14), where u is the number of LSTM hidden units, P and Q are the number of source and target prompt essays.

$$I_s = \mathcal{F}(\{x_i^s\}_{i=1}^P), \; I_t = \mathcal{F}(\{x_i^t\}_{i=1}^Q) \tag{14}$$

Next, we will obtain source-to-target mapping pairs. First, we take each source essay representation, let's say r_i^s, to dot product with $I_t^\top$, where $I_t^\top \in \mathbb{R}^{u*Q}$ is the transpose of I_t, which is used to obtain how similar it is to all the essays in the target prompt as the weights of the r_i^s to each target prompt essay. After that, we use a learnable parameter matrix $W_s \in \mathbb{R}^{Q*u}$ to acquire the weighted representations of the source prompt essays projected on the target prompt to express the source mapping representation $\hat{r}_i^s$, as shown in Eq. 15. In this way, r_i^s and $\hat{r}_i^s$ can form the source-to-target mapping pair (r_i^s, $\hat{r}_i^s$).

Similarly, for the target-to-source mapping pairs, $\hat{r}_i^t$ can be obtained by using r_i^t, $I_s^\top \in \mathbb{R}^{u*P}$ and $W_t \in \mathbb{R}^{P*u}$, and finally get the target-to-source mapping pair (r_i^t, $\hat{r}_i^t$).

$$\hat{r}_i^s = W_s \cdot (r_i^s \otimes I_t^\top), \; \hat{r}_i^t = W_t \cdot (r_i^t \otimes I_s^\top) \tag{15}$$

where $\otimes$ is the dot product operation.

Finally, we take the mapping pairs (r_i^s, $\hat{r}_i^s$) and (r_i^t, $\hat{r}_i^t$) as the positive pairs. For the selection of negative samples, we follow the work of SimCLR (Chen et al., 2020) which takes the other samples in the same batch as the negative samples. The contrastive learning loss functions of mapping from source to target and from target to source are defined as follows:

$$\mathcal{L}_{s \to t} = \sum_{i}^{N_s} -\log \frac{f(r_i^s, \hat{r}_i^s)}{\sum_{j}^{N_s} f(r_i^s, r_j^s) + f(r_i^s, \hat{r}_j^s)} \tag{16}$$

$$\mathcal{L}_{t \to s} = \sum_{i}^{N_t} -\log \frac{f(r_i^t, \hat{r}_i^t)}{\sum_{j}^{N_t} f(r_i^t, r_j^t) + f(r_i^t, \hat{r}_j^t)} \tag{17}$$

where $f(a, b) = \exp(\cos(a, b)/\tau)$, $\cos(\cdot)$ is cosine similarity function, τ is temperature hyper-parameter, N_s and N_t are the batch size of source prompt essays and target prompt essays. The prompt-mapping contrastive learning loss is defined as:

$$\mathcal{L}_{pm} = \mathcal{L}_{s \to t} + \mathcal{L}_{t \to s} \tag{18}$$

The total loss of single-overall scoring task is:

$$\mathcal{L}_{so} = \mathcal{L}_{aes\_so} + \lambda_1 \mathcal{L}_{pm} \tag{19}$$

The total loss of multi-attribute scoring task is:

$$\mathcal{L}_{ma} = \mathcal{L}_{aes\_ma} + \lambda_1 \mathcal{L}_{pm} + \lambda_2 \mathcal{L}_{cor} \tag{20}$$

where λ_1 and λ_2 are weighted hyper-parameters.

4 Experiments

4.1 Datasets and Evaluation Metrics

We conduct the experiments on the ASAP++ (Mathias and Bhattacharyya, 2018) dataset, which is an extension of the ASAP[2] dataset. Each essay has an overall score and multiple attribute scores. The statistics are provided in Appendix A.

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

Model	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
Single-overall scoring task									
Hi att †	0.372	0.465	0.432	0.523	0.586	0.574	0.514	0.323	0.474
PAES †	0.746	0.591	0.608	**0.641**	0.727	**0.609**	0.707	0.635	0.658
PMAES (ours)	**0.758**	**0.674**	**0.658**	0.625	**0.735**	0.578	**0.749**	**0.718**	**0.687**
Multi-attribute scoring task									
Hi att ‡	0.315	0.478	0.317	0.478	0.375	0.357	0.205	0.265	0.349
AES aug ‡	0.330	0.518	0.299	0.477	0.341	0.399	0.162	0.200	0.341
PAES ‡	0.605	0.522	0.575	0.606	**0.634**	0.545	0.356	0.447	0.536
CTS no att ‡	0.619	0.539	0.585	0.616	0.616	0.544	0.363	0.461	0.543
CTS ‡	0.623	0.540	0.592	**0.623**	0.613	0.548	0.384	0.504	0.553
PMAES (ours)	**0.656**	**0.553**	**0.598**	0.606	0.626	**0.572**	**0.386**	**0.530**	**0.566**

Table 1: Main results of single-overall scoring task and multi-attribute scoring task for each prompt. The results of multi-attribute scoring task is the average QWK score across all attribute for each prompt. † refers to the results of rerunning the code. ‡ refers to the results from Ridley et al. (2021).

Model	Overall	Cont	Org	WC	SF	Conv	PA	Lan	Nar	Avg.
Hi att ‡	0.453	0.348	0.243	0.416	0.428	0.244	0.309	0.293	0.379	0.346
AES aug ‡	0.402	0.342	0.256	0.402	0.432	0.239	0.331	0.313	0.377	0.344
PAES ‡	0.657	0.539	0.414	0.531	0.536	0.357	0.570	0.531	0.605	0.527
CTS no att ‡	0.659	0.541	0.424	0.558	0.544	0.387	0.561	0.539	0.605	0.535
CTS ‡	0.670	0.555	0.458	0.557	0.545	0.412	0.565	0.536	0.608	0.545
PMAES (ours)	**0.671**	**0.567**	**0.481**	**0.584**	**0.582**	**0.421**	**0.584**	**0.545**	**0.614**	**0.561**

Table 2: Main results of multi-attribute scoring task. This table shows the average QWK score across all prompts for each attribute. ‡ refers to the results from Ridley et al. (2021).

We use Quadratic Weighted Kappa (QWK) as the evaluation metric to measure the consistency between the real scores and the predicted scores, which is the general evaluation metric in AES tasks (Jin et al., 2018;Li et al., 2020;Ridley et al., 2021).

4.2 Implementation Details

We use the same data partition as the current state-of-the-art model (Ridley et al., 2021), that is for each prompt as target prompt, then the rest of prompts are set to be source prompt. For example, assume the target prompt is P8, then the source prompt consists of P1$\sim$P7. We use labeled source prompt essays and unlabeled target prompt essays as training data, and the same unlabeled target prompt essays as test data. The validation data is from labeled source prompt essays.

We use the same handcrafted features proposed by (Ridley et al., 2020) in single-overall and multi-attribute scoring task, including features of Length-based, Readability, Text Complexity, Text Variation and Sentiment. We use the length of the longest essay in the dataset as the padding length to ensure that the essay information can be retained as much as possible. We use 50-dimension POS embedding as input and train all models for 50 epochs. We report the average results across five random seeds. More details are provided in Appendix B.

4.3 Baseline Models

We compare with the existing models on single-overall scoring task and multi-attribute scoring task. For single-overall scoring task, we use **Hi att** (Dong et al., 2017) and **PAES** (Ridley et al., 2020) as baseline models, which are both the single-overall scoring models. For multi-attribute scoring task, we use **Hi att** (Dong et al., 2017), **AES aug** (Hussein et al., 2020), **PAES** (Ridley et al., 2020), **CTS no att** (Ridley et al., 2021) and the current state-of-the-art model **CTS** (Ridley et al., 2021) as the comparison models. The details of baseline models are described as follow:

(1) **Hi att**: Dong et al. (2017) propose a hierarchical structure with attention pooling for single-overall scoring task, which scores essays by extracting the sentence- and essay-level features.

Model	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
Single-overall scoring task									
PMAES	**0.758**	**0.674**	**0.658**	0.625	**0.735**	0.578	**0.749**	**0.718**	**0.687**
w/o $\mathcal{L}_{pm}$	0.602	0.551	0.621	**0.646**	0.727	**0.602**	0.745	0.665	0.645
Multi-attribute scoring task									
PMAES	**0.656**	**0.553**	**0.598**	0.606	0.626	0.572	**0.386**	**0.530**	**0.566**
w/o $\mathcal{L}_{cor}$	0.646	0.539	0.592	**0.611**	0.630	**0.580**	0.373	0.509	0.560
w/o $\mathcal{L}_{pm}$	0.650	0.545	0.589	0.606	0.620	0.578	0.383	0.453	0.553
w/o $\mathcal{L}_{pm}$ & $\mathcal{L}_{cor}$	0.625	0.525	0.594	0.607	**0.637**	0.557	0.377	0.469	0.549

Table 3: Ablation results of single-overall scoring task and multi-attribute scoring task for each prompt. The results of multi-attribute scoring task is the average QWK score across all attributes for each prompt.

Model	Overall	Cont	Org	WC	SF	Conv	PA	Lan	Nar	Avg.
PMAES	**0.671**	**0.567**	**0.481**	**0.584**	**0.582**	**0.421**	**0.584**	0.545	0.614	**0.561**
w/o $\mathcal{L}_{cor}$	0.669	0.562	0.461	0.573	0.569	0.405	0.583	**0.546**	**0.619**	0.554
w/o $\mathcal{L}_{pm}$	0.666	0.546	0.450	0.573	0.573	0.385	0.578	0.538	0.614	0.547
w/o $\mathcal{L}_{pm}$ & $\mathcal{L}_{cor}$	0.664	0.553	0.432	0.548	0.554	0.398	0.583	0.539	0.614	0.543

Table 4: Ablation results for multi-attribute scoring task, this table shows the average QWK score across all prompts for each attribute.

(2) **AES aug**: Hussein et al. (2020) convert the model proposed by Taghipour and Ng (2016) into a multi-task architecture, which can be used to rate the multi-attribute scores at the same time.

(3) **PAES**: Ridley et al. (2020) apply a neural model with handcrafted features for single-overall scoring.

(4) **CTS**: Ridley et al. (2021) propose the first model for the cross-prompt multi-attribute scoring task, in which they develop a trait-attention mechanism to establish interactions between different attributes.

(5) **CTS no att**: This model (Ridley et al., 2021) has the same shared- and private-layers as CTS, and removes the trait-attention mechanism.

5 Results and Analysis

5.1 Main Results

We report the main results on single-overall scoring task and multi-attribute scoring task.

For single-overall scoring task, we use Hi att and PAES as baseline models, which are both single-overall scoring models. As shown in Table 1, compared with Hi att and PAES, PMAES achieves the best results, improving the average QWK score by 21.3% and 2.9%, respectively, which proves the effectiveness of our approach on this task.

For multi-attribute scoring task, following Ridley et al. (2021), we report the results from two dimensions. For the average QWK score across all attributes for each prompt (Table 1), we can see that our approach achieves 0.566 average QWK score, which outperforms all baseline models. For the average QWK score across all prompts for each attribute (Table 2), PMAES not only achieves the state-of-the-art average performance but also gets best performance on all prompts, which shows the significant improvement of PMAES for this task. Based on the above results, we can see that PMAES is suitable for both grading a single overall score and multiple attribute scores.

Meanwhile, we discover that PMAES fails to perform well in P4 and P6 as target prompts. Through analysis, we find that essays in P4 and P6 are source-dependent types and were written by 10th graders. Their writing requirements are relatively difficult. P4 requires students to write a response to figure out the source author's thoughts, while P6 requires students to summarize academic excerpts. We believe that P4 and P6 share a few features with other prompts. In this case, the way our method maps P4/P6 and the source prompt to each other may lead to a low-scoring performance.

5.2 Ablation Studies

We conduct the ablation experiments both on single-overall scoring task and multi-attribute scoring task, which are shown in Table 3 and Table 4.

For single-overall scoring task, as shown in Table 3, we can see that if training model without $\mathcal{L}_{pm}$, the average QWK score drops by 4.2%, and the QWK scores of the majority of prompts also drop significantly. Especially in P1 and P2, the QWK scores drop by 15.6% and 12.3%. It proves that our proposed prompt-mapping contrastive learning is effective in this task.

For multi-attribute scoring task, we also show the results from two dimensions. Firstly, as shown in Table 3, it can be seen that the average QWK score drops by 0.6% after removing $\mathcal{L}_{cor}$ and by 1.3% after removing $\mathcal{L}_{pm}$, which demonstrates that both $\mathcal{L}_{pm}$ and $\mathcal{L}_{cor}$ contribute to improve the scoring performance, and $\mathcal{L}_{pm}$ contributes more. When we remove these two components (w/o $\mathcal{L}_{pm}$ & $\mathcal{L}_{cor}$), the average QWK score drops by 1.7%. This shows that $\mathcal{L}_{pm}$ and $\mathcal{L}_{cor}$ can promote each other and further improve the scoring performance. Secondly, for the dimension of the average QWK score across all prompts for each attribute, we show the results in Table 4. The average QWK score drops by 0.7% after removing $\mathcal{L}_{cor}$, by 1.4% after removing $\mathcal{L}_{pm}$ and by 1.8% after removing both components. It further demonstrates the effectiveness of our model. We also can see that when we remove both of them, the QWK scores drop on almost all attributes. Especially on *Organization*, after removing $\mathcal{L}_{pm}$ and $\mathcal{L}_{cor}$, the QWK score drops significantly (by 4.9%).

Based on the above results, it can be found that our proposed approach can effectively improve the model scoring performance in the single-overall scoring task and the multi-attribute scoring task.

5.3 Analysis of Prompt Consistency

To further investigate the effectiveness of prompt-mapping contrastive learning on prompt consistency, we present our analysis using two methods: 1) Measuring the distance between source and target prompts using the Maximum Mean Discrepancy (MMD, Gretton et al., 2012). 2) Visualizing the essay representations of source and target prompts by using t-SNE (Van der Maaten and Hinton, 2008) to observe the degree of the consistency of prompts.

5.3.1 MMD for Prompt Consistency

Maximum Mean Discrepancy (MMD) is a kernel-based method that measures the distance between two matrices based on their respective mean embeddings. Inspired by previous work (Thota and Leontidis, 2021; Yue et al., 2022), we quantify the degree of consistency by calculating the MMD distance between the source and target prompt essay representation matrices. A smaller distance indicates a greater degree of consistency between the source and target prompts, whereas a larger distance indicates a lesser degree of congruence. More details are provided in Appendix C

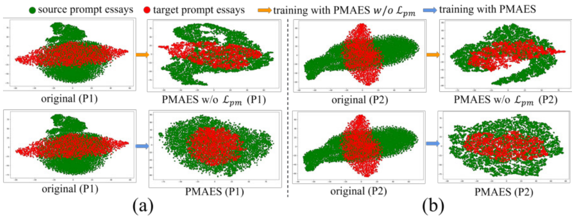

Figure 3: Visualization of prompt representations after training with PMAES w/o $\mathcal{L}_{pm}$ and PMAES. (a) and (b) represent the change of source and target prompt representations with P1 and P2 as target prompts.

5.3.2 Visualization for Prompt Consistency

We use the t-SNE (Van der Maaten and Hinton, 2008) toolkit to visualize the representations of all essays on source and target prompts in training data to demonstrate prompt representations, which are generated by shared encoder under random initialization (original), training with PMAES w/o $\mathcal{L}_{pm}$ and PMAES, respectively.

Firstly, as shown in Figure 3(a) and Figure 3(b), we show the visualization results of source and target prompt essay representations with P1 and P2 as target prompts. Taking Figure 3(a) for example, we can see that a clear discrepancy exists in the original representations of source prompt (green) and target prompt (red). After training with PMAES w/o $\mathcal{L}_{pm}$, the prompt representations become more discrete, while prompt representations generated by PMAES are undoubtedly more consistent and close to each other. The same phenomenon occurs in Figure 3(b).

Secondly, to further show how the prompt representations change as the number of training epochs increases, we visualize the essay representations generated by the epochs 0 (original), 4, 14, 34 and 50 during training w/o $\mathcal{L}_{pm}$ and PMAES with P1 as the target prompt. As shown in Table 4, the top row shows the results of training with PMAES w/o $\mathcal{L}_{pm}$, and the bottom row shows the results of training with PMAES. The results show that the representations generated by these two models are relatively divergent at the beginning of training. As

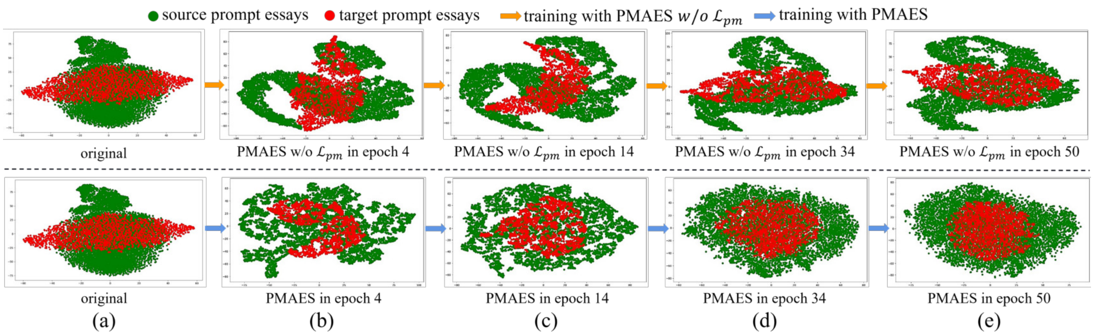

Figure 4: Visualization of changes in prompt representations during training with PMAES w/o $\mathcal{L}_{pm}$ (top) and PMAES (bottom), when P1 is the target prompt. (a), (b), (c), (d) and (e) represent the visualization of essay representations at epoch 0 (original), 4, 14, 34 and 50, respectively.

the training epochs increase, PMAES makes the prompt representations gradually consistent, while PMAES w/o $\mathcal{L}_{pm}$ makes them gradually discrete.

Based on the results of MMD and visualization analysis, it can be seen that w/o $\mathcal{L}_{pm}$ not only fails to maintain the consistency of source and target prompts, but also damages it. In contrast, our approach can significantly make these two prompts more consistent to improve scoring performance.

5.4 Results of Different Source-target Settings

Most of the current cross-prompt AES studies train on multiple prompts (source prompt) and test on a single prompt (target prompt), namely the many-to-one setting, which is the general setting in cross-prompt AES and is shown in Section 5.1. To verify the performance of our approach in many practical settings, we conduct comprehensive experiments for different source-target settings. More details are provided in Appendix D.

6 Conclusions

In this paper, we propose a new method for cross-prompt AES that aims to capture more shared features between the source and target prompts. Specifically, we design prompt-mapping contrastive learning to decrease the distance between the mapping pairs from source-to-target and target-to-source simultaneously and finally make the representations of the two prompts more consistent. Experimental results demonstrate that our approach achieves the state-of-the-art on both single-overall scoring task and multi-attribute scoring task. We further design experiments for three source-target settings, which proves that our approach can be adapted to multiple scenarios.

Limitations

Our approach achieves promising results in cross-prompt AES by enhancing the consistency between source and target prompts. We believe that this idea can also be used to other cross-domain or domain adaptation tasks. In addition, as can be seen from Table 1, our approach fails to perform well in some cases. We think that forcing the representations of two prompts to be closer during model training may result in more errors when the prompts' grading rubrics, writing genres, and writing requirements are quite different. Therefore, there are two possible directions can be explored for future research: 1) More fine-grained shared features can be extracted to improve scoring performance. 2) Score-aware information can be integrated into model to improve source and target prompts consistency.

Acknowledgements

This work is supported by the National Natural Science Foundation of China [grant number: 61976062].

References

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

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained multi-task learning for automated essay scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–799.

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.

Yann Dauphin, Harm De Vries, and Yoshua Bengio. 2015. Equilibrated adaptive learning rates for non-convex optimization. *Advances in neural information processing systems*, 28.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1072–1077. The 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), Vancouver, Canada, August 3-4, 2017*, pages 153–162. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Xu Han, Yuqi Luo, Weize Chen, Zhiyuan Liu, Maosong Sun, Zhou Botong, Hao Fei, and Suncong Zheng. 2022. Cross-lingual contrastive learning for fine-grained entity typing for low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2241–2250.

Marti A Hearst. 2000. The debate on automated essay grading. *IEEE Intelligent Systems and their Applications*, 15(5):22–37.

Mohamed A Hussein, Hesham A Hassan, and Mohammad Nassef. 2020. A trait-based deep learning automated essay scoring system with adaptive feedback. *International Journal of Advanced Computer Science and Applications*, 11(5).

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 (Volume 1: Long Papers)*, pages 1088–1097.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xia Li, Minping Chen, and Jian-Yun Nie. 2020. Sednn: shared and enhanced deep neural network model for cross-prompt automated essay scoring. *Knowledge-Based Systems*, 210:106491.

Xia Li, Minping Chen, Jianyun Nie, Zhenxing Liu, Ziheng Feng, and Yingdan Cai. 2018. Coherence-based automated essay scoring using self-attention. In *Chinese computational linguistics and natural language processing based on naturally annotated big data*, pages 386–397. Springer.

Dongliang Liao, Jin Xu, Gongfu Li, and Yiru Wang. 2021. Hierarchical coherence modeling for document quality assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13353–13361.

Junteng Ma, Xia Li, Minping Chen, and Weigeng Yang. 2021. Enhanced hierarchical structure features for automated essay scoring. In *China Conference on Information Retrieval*, pages 168–179. Springer.

Sandeep Mathias and Pushpak Bhattacharyya. 2018. Asap++: Enriching the asap automated essay grading dataset with essay attribute scores. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 567–575.

Isaac Persing and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269.

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, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13745–13753.

Robert Ridley, Liang He, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Prompt agnostic essay scorer: A domain generalization approach to cross-prompt automated essay scoring. *arXiv preprint arXiv:2008.01441*.

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

Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 1049–1054.

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075.

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 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 AAAI conference on artificial intelligence*, volume 32.

Mamatha Thota and Georgios Leontidis. 2021. Contrastive domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 2209–2218. Computer Vision Foundation / IEEE.

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.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Dong Wang, Ning Ding, Piji Li, and Haitao Zheng. 2021. Cline: Contrastive learning with semantic negative examples for natural language understanding. 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)*, pages 2332–2342.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3416–3425.

Sara Cushing Weigle. 2002. *Assessing writing*. Cambridge University Press.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907.

Jiayi Xie, Kaiwei Cai, Li Kong, Junsheng Zhou, and Weiguang Qu. 2022. Automated essay scoring via pairwise contrastive regression. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2724–2733.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. 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)*, pages 5065–5075.

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.

Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. 2022. Domain adaptation for question answering via question classification. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1776–1790. International Committee on Computational Linguistics.

A Statistics of Datasets

The ASAP++ dataset includes 12,978 English writings in response to eight prompts. Table 5 displays the statistics for both ASAP and ASAP++.

B Implementation Details

The implementation details of our model are presented as follows:

For single-overall scoring task, we optimize only the $\mathcal{L}_{aes\_so}$ in the first epoch, which is used to initialize the model weights, and optimize the $\mathcal{L}_{aes\_so}$ and $\mathcal{L}_{pm}$ in the rest epochs. We set the kernel size as 3, the number of filters as 100 for CNN and the number of hidden units as 50 for LSTM. We use

Prompt ID	No. of Essays	Avg. Len.	Attributes	Score Range	
				Overall	Attribute
1	1,783	350	Cont, Org, WC, SF, Conv	2 - 12	1 - 6
2	1,800	350	Cont, Org, WC, SF, Conv	0 - 6	1 - 6
3	1,726	150	Cont, PA, Lan, Nar	0 - 3	0 - 3
4	1,772	150	Cont, PA, Lan, Nar	0 - 3	0 - 3
5	1,805	150	Cont, PA, Lan, Nar	0 - 4	0 - 4
6	1,800	150	Cont, PA, Lan, Nar	0 - 4	0 - 4
7	1,569	300	Cont, Org, Conv	0 - 30	0 - 6
8	723	650	Cont, Org, WC, SF, Conv	0 - 60	2 - 12

Table 5: Statistics of ASAP and ASAP++ Datasets. Cont: Content, Org: Organization, WC: Word Choice, SF: Sentence Fluency, Conv: Conventions, PA: Prompt Adherence, Lan: Language and Nar: Narrativity.

Model	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
original	0.902	0.968	0.378	0.475	0.331	0.277	0.187	2.016	0.692
w/o $\mathcal{L}_{pm}$	2.366	1.778	0.868	1.249	0.570	0.759	0.343	2.542	1.309
PMAES	**0.180**	**0.167**	**0.093**	**0.077**	**0.054**	**0.043**	**0.046**	**1.168**	**0.228**

Table 6: MMD distance of single-overall scoring task under different settings. "original", "PMAES w/o $\mathcal{L}_{pm}$" and "PMAES" indicate the essay representations of source and target prompt essays generated by randomly initialized, trained with PMAES w/o $\mathcal{L}_{pm}$ and trained with PMAES, respectively.

Adam (Kingma and Ba, 2015) as the optimizer with the learning rate = 0.0001, $\tau = 0.1$ and $\lambda_1 = 0.5$. We use the model with the highest QWK score in the development set to evaluate the test set.

For multi-attribute scoring task, the detailed parameters are as follows: the kernel size is 5, the number of filters is 100 for CNN and the number of hidden units is 100 for LSTM. The optimizer is RMSprop (Dauphin et al., 2015) with the learning rate = 0.001, $\tau = 0.001$, $\rho = 0.1$, $\lambda_1 = 0.5$ and the $\lambda_2 = 0.1$. We take the model with the highest average QWK score of all attributes in the development set to evaluate the test set.

C MMD for Prompt Consistency

The MMD distance can be calculated by the following equation:

$$\mathrm{MMD} = \left\| \frac{1}{P} \sum_{i=1}^{P} \phi(r_i^s) - \frac{1}{Q} \sum_{j=1}^{Q} \phi(r_j^t) \right\|_H^2 \quad (21)$$

where $\phi(\cdot)$ denotes the function that is used to map the original variable to the Reproducing Kernel Hilbert Space (RKHS), P and Q are the number of source and target prompt essays in the training data, r_i^s and r_j^t are the representation of source and target prompt essays.

Source→Target	PMAES	w/o $\mathcal{L}_{pm}$
P1,P2→P3,P4	0.537	0.426
P3,P4→P1,P2	0.673	0.407
P5,P6→P7,P8	0.447	0.381
P7,P8→P5,P6	0.528	0.439
P1∼P4→P5∼P8	0.682	0.672
P5∼P8→P1∼P4	0.675	0.559

Table 7: Experiment results of the many-to-many setting on the single-overall scoring task. This table shows the average QWK scores of all prompts in target prompt. P1,P2→P3,P4 refers to source prompt consists of P1 and P2, and target prompt consists of P3 and P4. The same meaning to other source-target pairs.

We take the essay representation matrices of source and target prompts generated by shared encoder to calculate the MMD distance. In order to better show the effectiveness of our proposed prompt-mapping contrastive learning in improving the consistency of source and target prompts, we use the shared encoding layer representations obtained at three settings: random initialization (original), training PMAES without $\mathcal{L}_{pm}$ (w/o $\mathcal{L}_{pm}$), and training with PMAES. We show the results in Table 6. As can be seen, compared with PMAES, w/o $\mathcal{L}_{pm}$ leads to an increase in MMD distance, which indicates that the prompt consistency is bro-

T S	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
One-to-many setting									
P1	-	0.526 I 0.598	0.457 I 0.552	0.533 I 0.560	0.557 I 0.673	0.423 I 0.517	0.701 I 0.733	0.344 I 0.405	0.506 I 0.577
P2	0.354 I 0.450	-	0.192 I 0.426	0.325 I 0.485	0.210 I 0.434	0.144 I 0.269	0.222 I 0.451	0.488 I 0.552	0.276 I 0.438
P3	0.428 I 0.780	0.222 I 0.620	-	0.652 I 0.658	0.772 I $\underline{0.747}$	0.613 I 0.626	0.576 I 0.709	0.087 I 0.297	0.479 I 0.634
P4	0.436 I 0.742	0.220 I 0.542	0.639 I 0.656	-	0.745 I $\underline{0.735}$	0.635 I $\underline{0.629}$	0.601 I $\underline{0.532}$	0.153 I 0.348	0.490 I 0.598
P5	0.540 I 0.742	0.323 I 0.570	0.563 I 0.621	0.614 I 0.628	-	0.598 I 0.608	0.634 I 0.641	0.141 I 0.271	0.488 I 0.583
P6	0.655 I $\underline{0.592}$	0.438 I 0.558	0.396 I 0.505	0.406 I 0.575	0.448 I 0.535	-	0.477 I $\underline{0.407}$	0.320 I 0.565	0.449 I 0.534
P7	0.666 I 0.667	0.500 I 0.612	0.490 I 0.507	0.457 I 0.534	0.535 I $\underline{0.509}$	0.396 I $\underline{0.346}$	-	0.427 I 0.562	0.496 I 0.534
P8	0.408 I 0.416	0.313 I 0.466	0.404 I 0.441	0.459 I 0.502	0.062 I 0.155	0.029 I 0.099	0.390 I 0.497	-	0.295 I 0.368
One-to-one setting									
P1	-	0.371 I 0.483	0.477 I 0.553	0.529 I 0.531	0.608 I 0.659	0.470 I 0.513	0.736 I $\underline{0.731}$	0.362 I 0.421	0.507 I 0.556
P2	0.516 I 0.598	-	0.200 I 0.420	0.316 I 0.497	0.239 I 0.400	0.121 I 0.273	0.217 I 0.460	0.516 I 0.549	0.304 I 0.457
P3	0.458 I 0.782	0.382 I 0.519	-	0.656 I 0.657	0.758 I 0.759	0.597 I 0.633	0.599 I 0.716	0.088 I 0.265	0.506 I 0.619
P4	0.513 I 0.717	0.309 I 0.482	0.591 I 0.638	-	0.749 I $\underline{0.742}$	0.604 I 0.616	0.598 I $\underline{0.531}$	0.164 I 0.346	0.504 I 0.582
P5	0.424 I 0.750	0.275 I 0.606	0.583 I 0.627	0.608 I 0.637	-	0.599 I 0.612	0.601 I $\underline{0.555}$	0.113 I 0.325	0.458 I 0.588
P6	0.665 I 0.719	0.454 I 0.534	0.386 I 0.579	0.466 I 0.621	0.459 I 0.609	-	0.466 I 0.503	0.334 I 0.374	0.461 I 0.563
P7	0.633 I 0.660	0.461 I 0.607	0.485 I $\underline{0.452}$	0.460 I 0.505	0.510 I 0.512	0.463 I $\underline{0.343}$	-	0.428 I 0.574	0.491 I 0.522
P8	0.405 I 0.452	0.447 I $\underline{0.217}$	0.308 I 0.385	0.246 I 0.486	0.198 I $\underline{0.172}$	0.077 I 0.192	0.423 I 0.451	-	0.301 I 0.336

Table 8: Experiment results of one-to-many and one-to-one setting on single-overall scoring task. "$a \mid b$" refers to the result of PMAES w/o $\mathcal{L}_{pm}$ and PMAES, respectively. "$a \mid \underline{b}$" indicates that PMAES w/o $\mathcal{L}_{pm}$ outperforms PMAES, and the rest is the opposite.

ken. In contrast, PMAES can significantly reduce the MMD distance, which indicates that our approach is effective in improving prompt consistency. These results prove that our approach can effectively improve the consistency of source and target prompts.

D Results of Different Source-target Settings

We argue that there are different situations may exist in practical settings. For example, source prompt and target prompt are all containing multiple prompts (namely many-to-many), source prompt contains only one prompt and target prompt contains multiple prompts (namely one-to-many), or source prompt and target prompt both contain only one prompt (namely one-to-one). To this end, we conduct comprehensive experiments for these settings to verify the performance of our approach in multiple scenarios.

D.1 Results of Many-to-many Setting

The experimental results of the many-to-many setting are shown in Table 7. For convenience, we design 6 source-target pairs for this setting. Since each prompt has its own score range, we calculate the QWK score for each prompt separately, and report the average QWK score of all prompts in target prompt. As shown in Table 7, PMAES outperform w/o $\mathcal{L}_{pm}$ in all source-target pairs with the QWK scores increase by 11.1%, 26.6%, 6.6%, 8.9%, 1.0% and 11.6%. The results demonstrate that our approach is suitable for many-to-many setting.

D.2 Results of One-to-many Setting

Table 8 (top subtable) shows the experimental results of the One-to-many setting. Same as many-to-many setting, we also calculate the QWK score for each target prompt individually. In this setting, source prompt contains only one prompt, and target prompt consists of the remaining 7 prompts. Compared with w/o $\mathcal{L}_{pm}$, the average QWK scores of PMAES increase by 7.1%, 16.2%, 15.5%, 10.8%, 9.5%, 8.5%, 3.8%, 7.3%, respectively. This proves that our approach is also remarkable in one-to-many setting.

D.3 Results of One-to-one Setting

The experimental results of the one-to-one setting are shown in Table 8 (bottom subtable). For each prompt, we take each of the remaining 7 prompt as its corresponding target prompt to construct one-to-one source-target pairs. We use "alb" form to represent the performance of without using and using prompt-mapping contrastive learning, where "a" denotes QWK score of w/o $\mathcal{L}_{pm}$ and "b" denotes QWK score of PMAES. It can be observed that PMAES outperforms PMAES w/o $\mathcal{L}_{pm}$ in most source-target pairs. The average QWK score for each prompt as the source prompt are all improved, it can be demonstrated that our approach is stable and effective in one-to-one setting.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section after Conclusion

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section1

☒ A4. Have you used AI writing assistants when working on this paper?
No using AI writing assistants.

B ☑ Did you use or create scientific artifacts?

Section 4.1

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix A

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

C ☒ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.2

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3.2 and Section 4.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.