Improving Domain Generalization for Prompt-Aware Essay Scoring via Disentangled Representation Learning

Zhiwei Jiang*†, Tianyi Gao*, Yafeng Yin, Meng Liu, Hua Yu,
Zifeng Cheng, Qing Gu
State Key Laboratory for Novel Software Technology, Nanjing University, China
jzw@nju.edu.cn, mf21330021@smail.nju.edu.cn, yafeng@nju.edu.cn
{mf1933061,huayu.yh,chengzf}@smail.nju.edu.cn, guq@nju.edu.cn

Abstract

Automated Essay Scoring (AES) aims to score essays written in response to specific prompts. Many AES models have been proposed, but most of them are either prompt-specific or prompt-adaptive and cannot generalize well on "unseen" prompts. This work focuses on improving the generalization ability of AES models from the perspective of domain generalization, where the data of target prompts cannot be accessed during training. Specifically, we propose a prompt-aware neural AES model to extract comprehensive representation for essay scoring, including both prompt-invariant and prompt-specific features. To improve the generalization of representation, we further propose a novel disentangled representation learning framework. In this framework, a contrastive norm-angular alignment strategy and a counterfactual self-training strategy are designed to disentangle the prompt-invariant information and prompt-specific information in representation. Extensive experimental results on datasets of both ASAP and TOEFL11 demonstrate the effectiveness of our method under the domain generalization setting.

1 Introduction

Automated Essay Scoring (AES), which aims to score essays written for specific prompts, is helpful in reducing the burden of scoring staff in various writing tests (Ke and Ng, 2019). Over the past few years, supervised deep learning has achieved remarkable success on the prompt-specific AES task (Taghipour and Ng, 2016; Farag et al., 2018; Tay et al., 2018), which assumes that the training and test data are from the same prompt. However, in many real-world scenarios, the training and test data often come from different prompts, which leads to a performance degradation of prompt-specific AES model on the out-of-distribution target prompt (Dong and Zhang, 2016; Cozma et al., 2018).

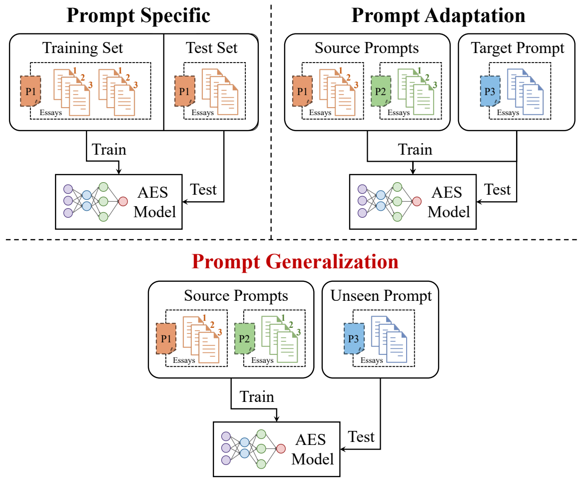


Figure 1: Comparison among prompt specific, prompt adaptation, and prompt generalization settings.

Many researchers have tried to adapt the AES model from source prompts to the target prompt, with limited labeled data (Cozma et al., 2018; Cao et al., 2020) or only unlabeled data (Jin et al., 2018) in target prompt. Despite their success, they need to access the data of target prompts during training and may fail to work when the target prompt is unavailable during training.

To this end, in this paper, we focus on the prompt generalization setting. As shown in Figure 1, we aim to train the AES model only based on source prompts and enable it to generalize well on "unseen" prompt(s). Existing prompt-generalized AES methods are relatively few, mainly including the generic method based on non-content handcrafted features (Yigal et al., 2010) and the prompt-agnostic method based on non prompt-specific hybrid features (Ridley et al., 2020). These methods discard the prompt-specific content features to alleviate the negative impact brought by domain shift, whereas they cannot score essays comprehensively.

To achieve more comprehensive essay scoring,

[*] Both authors contributed equally to this research.
[†] Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12456–12470
July 9-14, 2023 ©2023 Association for Computational Linguistics

we consider extracting features from perspectives of both prompt-invariant essay quality and prompt-specific prompt adherence. Therefore, we propose a prompt-aware neural AES model, which can extract the essay quality features based on an essay encoder such as the pre-trained BERT (Devlin et al., 2019) and extract the prompt adherence features based on a text matching module.

Although this AES model can be directly trained with data of source prompts, there are still two problems hindering its generalization on unseen prompts. (1) The essay quality features extracted by encoder such as BERT may encode both quality and content information and they are entangled in the features. How to disentangle independent quality information from features is the first problem. (2) Both prompt adherence features and essay quality features are extracted based on essay. Thus, from the view of causality (Pearl, 2009), the essay is a confounder of both features, leading to a spurious correlation between prompt adherence and essay quality. For example, the model may learn a correlation that high-quality essays often have good prompt adherence, whereas this correlation is spurious since an essay may have different adherence but unchanged quality under different prompts. Then, how to disentangle the spurious correlation to make these two kinds of features independently contribute to the final score is the second problem.

To address the above problems, we propose a disentangled representation learning framework. For the first problem, we design a contrastive norm-angular alignment strategy, which addresses the quality-content disentanglement by reflecting quality with norm and reflecting content with angular direction. For the second problem, we design a counterfactual self-training strategy, which addresses the quality-adherence disentanglement by self-training with quality-invariant and adherence-variant counterfactual data.

The contributions of this paper are as follows:

- We propose a prompt-aware neural network model for comprehensive essay scoring under the prompt generalization setting.
- We propose a novel disentangled representation learning framework to further improve the generalization ability of the AES model.
- Extensive experiments are conducted on two public datasets, and the results demonstrate the effectiveness of our method.

2 Related Work

Automated Essay Scoring Research on automated essay scoring has spanned the last 50 years (Ke and Ng, 2019; Klebanov and Madnani, 2020). From the perspective of essay representation, existing AES methods can be categorized into the early handcrafted features based methods (Page, 1994; Foltz et al., 1999; Persing et al., 2010; Somasundaran et al., 2014; Persing and Ng, 2014), recent neural network based methods (Dong and Zhang, 2016; Tay et al., 2018; Jiang et al., 2021), and hybrid features based methods (Uto et al., 2020a; Shibata and Uto, 2022). These methods can be further grouped into three scoring paradigms: prompt specific (Taghipour and Ng, 2016; Farag et al., 2018; Tay et al., 2018), prompt adaptation (Cozma et al., 2018; Cao et al., 2020; Jin et al., 2018; Ridley et al., 2021), and prompt generalization (Yigal et al., 2010; Ridley et al., 2020). While prompt-specific methods can achieve good performance, prompt-adaptive and prompt-generalized methods can reduce the annotation labor in target prompts.

Domain Generalization Domain generalization (DG) has been intensively studied in recent years (Wang et al., 2022). Existing DG methods can be categorized into three groups: (1) data augmentation (Zhao et al., 2020; Reich et al., 2022) which generates diverse samples to help generalization, (2) representation learning (Shen et al., 2021; Bui et al., 2021) which tries to learn domain-invariant representation or disentangle the features into domain-shared and domain-specific parts for better generalization, and (3) learning strategy (Segù et al., 2023; Lake, 2019) which tries to learn general knowledge by ensemble learning or meta-learning. This work considers improving generalization in terms of both data augmentation and representation learning.

Disentangled Representation Learning Disentangled representation learning has recently been used in many NLP tasks, such as style transfer (John et al., 2019; Nangi et al., 2021), machine reading comprehension (Wu et al., 2022), and negation and uncertainty modeling (Vasilakes et al., 2022). Most of these methods disentangle the underlying explanatory factors by separating features into several independent low-dimensional spaces, where commonly-used techniques include adversarial loss (John et al., 2019), information measure (Cheng et al., 2020), and counterfactual reasoning (Nangi et al., 2021). This work tries two types of representation disentanglements: one disentangles two factors respectively with norm and angular direction, while the other disentangles the spurious correlation based on counterfactual reasoning.

3 Proposed Method

3.1 Task Definition

The prompt-generalized AES task can be defined as follows: given K source prompts (i.e., domains) $\mathcal{P}_S = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_K\}$ as the training set, where the i-th prompt $\mathcal{P}_i$ has N_i labeled instances $\{x^i_j, y^i_j\}^{N_i}_{j=1}$. Each instance x^i_j is a text pair (e^i_j, p^i_j) and y^i_j is the holistic score of essay e^i_j under the prompt p^i_j, where p^i_j is the prompt text of the i-th source prompt $\mathcal{P}_i$. The objective is to learn a model from multiple source prompts that can be generalized to the target unseen prompt $\mathcal{P}_T$ with unknown distribution.

3.2 Overview

We propose a Prompt-Aware Neural Network (PANN) model for essay scoring, and a Disentangled Representation Learning (DRL) framework to improve its generalization on unseen prompts. Specifically, PANN takes both essays and prompts as inputs and extracts both prompt-invariant essay quality features and prompt-specific prompt adherence features for comprehensive essay scoring. DRL is designed in a pre-training and fine-tuning paradigm. In the pre-training stage, a contrastive norm-angular alignment strategy is designed to pre-train the essay quality features, aiming at disentangling the quality information and content information in features. In the fine-tuning stage, a counterfactual self-training strategy is employed to fine-tune the whole PANN, aiming at disentangling the spurious correlation between essay quality features and prompt adherence features. Finally, the fully-trained PANN is used for essay scoring on target unseen prompts.

3.3 Model Architecture of PANN

Our PANN contains three main components: the Essay Quality network (**EQ-net**) which only takes essay as input and is expected to extract prompt-invariant essay quality features, the Prompt Adherence network (**PA-net**) which takes both essay and prompt as inputs and is expected to extract prompt-specific prompt adherence features, and the Essay Scoring Predictor (**ESP**) which combines

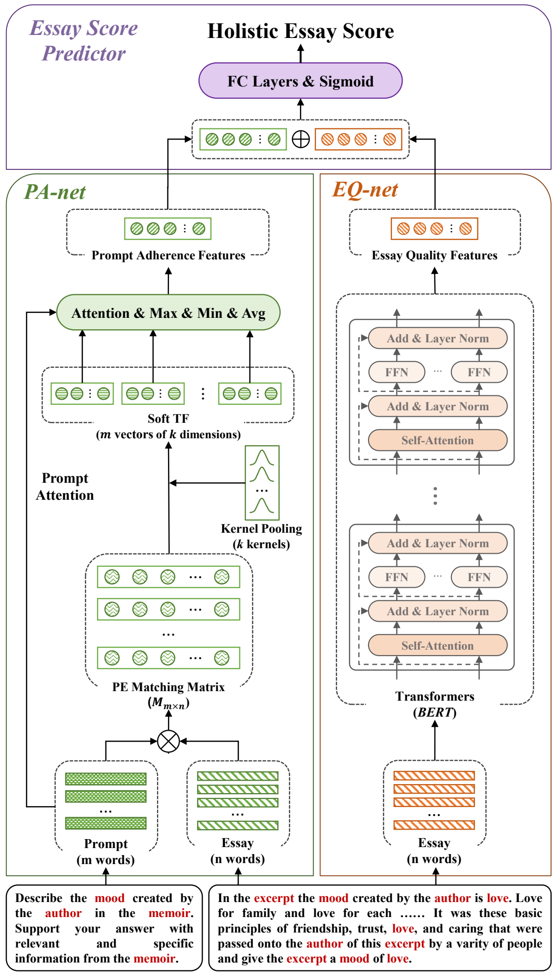
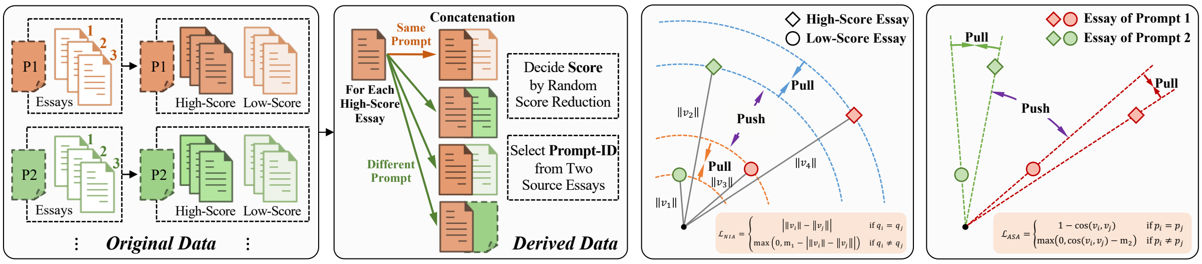


Figure 2: Model architecture of PANN

both kinds of features to predict a holistic score. The architecture of PANN is illustrated in Figure 2.

For EQ-net, we employ a Transformer-based neural network $f_\varphi(\cdot)$ to extract features v_i of an input essay e_i, where $v_i = f_\varphi(e_i; \varphi)$ refers to the essay quality features and φ indicates the network parameters. This module is not limited to a specific architecture and can be various existing AES encoders. Here, we initialize EQ-net with the pre-trained BERT (Devlin et al., 2019), which has been proven to be effective and to have good generalization in various NLP tasks, including essay scoring (Mayfield and Black, 2020; Uto et al., 2020a).

For PA-net, we design an interaction-based text matching model $f_\vartheta(\cdot)$ to extract features u_i of an input prompt-essay pair (p_i, e_i), where $u_i = f_\vartheta(p_i, e_i; \vartheta)$ refers to the prompt adherence features and ϑ indicates the network parameters. Since such interaction-based text matching model can focus only on the word-level similarities between essays and prompts, it can avoid encoding informa-



(a) Data Augmentation (b) Norm-Invariant Alignment (c) Angular-Shift Alignment

Figure 3: Illustration of the Contrastive Norm-Angular Alignment strategy for quality-content disentanglement.

tion related to the essay quality, such as syntax and coherence, thus making the features more specific to prompt adherence. More details of PA-net are given in Appendix A.

For ESP, we feed the combined features to several fully-connected (FC) layers followed by a linear layer with sigmoid activation for essay score prediction:

$$\hat{y}_i = sigmoid(W_s \times \sigma([v_i \oplus u_i]) + b_s) \tag{1}$$

where $\oplus$ represents the concatenation of vectors and $\sigma(\cdot)$ refers to the FC transformations.

3.4 Disentangled Representation Learning

In PANN, we design two sub-networks (i.e., PA-net and EQ-net), and expect them to capture the information of prompt adherence and essay quality respectively. However, the EQ-net may encode both prompt-invariant quality information and prompt-related content information, and the content information often shifts across prompts, which may hinder the generalization of EQ-net. Besides, both PA-net and EQ-net take essay as input, which makes the essay become a confounder of prompt adherence features and essay quality features, leading to a spurious correlation between them. In DRL, we correspondingly design two strategies to address these representation entanglements.

3.4.1 Quality-Content Disentanglement

We propose a Contrastive Norm-Angular Alignment (CNAA) strategy to disentangle the quality and content information in essay quality features. This strategy is designed based on the **norm invariant** and **angular shift** assumption, which assumes that the quality and content information can be disentangled by aligning features in terms of norm and angle respectively. **For norm invariant**, we expect that essays of similar quality can be distributed with similar norms and that these norms may be invariant across prompts. **For angular shift**, we expect that essays of similar content (i.e., prompt) can be distributed with similar angles but these angles should shift across prompts.

Data Augmentation. To prepare data for contrastive norm-angular alignment, as shown in Figure 3(a), we first extract all high-score and low-score essays from the training set to form the original data $\mathcal{D}_o$. Two thresholds δ_h and δ_l are used for essay filtering. For each essay $e_i \in \mathcal{D}_o$, apart from its score y_i, we assign extra quality label q_i and content label c_i to it, where $q_i \in \{0, 1\}$ denotes quality type (i.e., $q_i = 0$ when $y_i \geq \delta_h$ and $q_i = 1$ when $y_i \leq \delta_l$) and $c_i \in \{1, ..., K\}$ denotes content type (i.e., the prompt-ID). Therefore, the original data can be denoted as $\mathcal{D}_o = \{(e_i, y_i, q_i, c_i)\}_{i=1}^{N_o}$.

We further construct derived data $\mathcal{D}_d$ by synthesizing four kinds of essays based on text concatenation, as shown in Figure 3(a). For each synthesized essay $e'_k = e_i \oplus e_j$ (or $e_i \oplus p_j$ where p_j can be viewed as a special essay), we decide its score y'_k by randomly reducing the score $\max(y_i, y_j)$ by $a \sim \mathcal{N}(\mu, \sigma)$ and randomly select a prompt-ID c_i or c_j as its content label c'_k. Two reasons motivate us to randomly select a score lower than $\max(y_i, y_j)$ for a synthesized essay. First, concatenating two essays may reduce the quality (e.g., coherence and organization) of the higher-score one. Second, concatenating two essays from different prompts may reduce essay's prompt adherence to both prompts. The essays with high score or low score are selected to form the derived data $\mathcal{D}_d = \{(e'_i, y'_i, q'_i, c'_i)\}_{i=1}^{N_d}$.

Norm-Invariant & Angular-Shift Alignment. We implement the norm-angular alignment based on pairwise contrastive learning, which includes norm-invariant quality alignment and angular-shift content alignment.

Specifically, we sample essay pairs (e_i, e_j) from augmented data, where e_i is sampled from $\mathcal{D}_o$ and e_j is sampled from $\mathcal{D}_o \cup \mathcal{D}_d$. Given a pair of essays

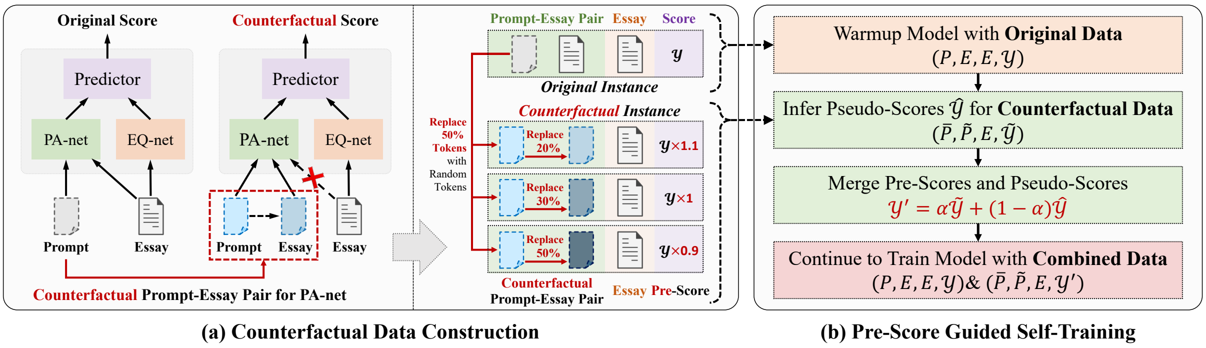


Figure 4: Illustration of the Counterfactual Self-Training (CST) strategy for quality-adherence disentanglement.

(e_i, e_j), we can first get their essay quality features (v_i, v_j) based on EQ-net.

Then, as shown in Figure 3(b), we can align features in perspective of quality information based on the Norm-Invariant Alignment (**NIA**) loss:

$$\mathcal{L}_{NIA} = \begin{cases} |\|v_i\| - \|v_j\||, & \text{if } q_i = q_j; \\ \max(0, m_1 - |\|v_i\| - \|v_j\||), & \text{if } q_i \neq q_j, \end{cases} \quad (2)$$

where m_1 denotes the margin between two quality types. Simultaneously, as shown in Figure 3(c), we can align features in perspective of content information based on the Angular-Shift Alignment (**ASA**) loss:

$$\mathcal{L}_{ASA} = \begin{cases} 1 - cos(v_i, v_j), & \text{if } c_i = c_j; \\ \max(0, cos(v_i, v_j) - m_2), & \text{if } c_i \neq c_j, \end{cases} \quad (3)$$

where m_2 denotes the margin between any two content types (i.e., prompts).

Finally, the overall loss of this strategy is:

$$\mathcal{L}_{CNAA} = \mathcal{L}_{NIA} + \mathcal{L}_{ASA} \quad (4)$$

3.4.2 Quality-Adherence Disentanglement

We propose a Counterfactual Self-Training (**CST**) strategy to disentangle the spurious correlation between essay quality features and prompt adherence features. While we do not call upon the mathematical machinery of causality (Pearl, 2009), we draw inspiration from the underlying philosophy to construct counterfactual data, where we try to ask and answer: "*What would the final score have been if the essay had a different prompt adherence, while its essay quality remained the same?*" As shown in Figure 4, with the counterfactual data, PANN can be fine-tuned based on our desinged pre-score guided self-training.

Counterfactual Data Construction. Due to the disentangled structure of PA-net and EQ-net, we can easily change the prompt adherence features by controlling the input of PA-net while maintaining the essay quality features unchanged. As shown in Figure 4(a), for each instance (p_i, e_i, e_i, y_i) with the input form of PANN (i.e., first two inputs p_i and e_i for PA-net while the third input e_i for EQ-net), we can generate three counterfactual instances $(\overline{p}_i, \widetilde{p}_i^{20}, e_i, \widetilde{y}_i^{20})$, $(\overline{p}_i, \widetilde{p}_i^{30}, e_i, \widetilde{y}_i^{30})$, and $(\overline{p}_i, \widetilde{p}_i^{50}, e_i, \widetilde{y}_i^{50})$, where $\overline{p}_i$ is constructed by randomly replacing 50% tokens of p_i with random tokens, $\widetilde{p}_i^z$ is constructed by randomly replacing z% tokens of $\overline{p}_i$ with random tokens, and $\widetilde{y}_i^z$ is the pre-score of the text pair $(\overline{p}_i, \widetilde{p}_i^z)$. Here we make an empirical guess for these pre-scores to highlight their differences in the degree of matching, where $\widetilde{y}_i^{20} = y_i \times 1.1$, $\widetilde{y}_i^{30} = y_i \times 1$, and $\widetilde{y}_i^{50} = y_i \times 0.9$.

Pre-Score Guided Self-Training. Unlike conventional self-training strategies that directly predict the pseudo-labels for unlabeled data, we combine both the pre-score and the predicted pseudo-score of each counterfactual instance as its final score. In this way, the prior knowledge we provide in the pre-scores and the model's knowledge encoded in the pseudo-scores can be well merged.

Specifically, we first warm up PANN on the original training set for several epochs based on the MSE (Mean Squared Error) loss function:

$$\mathcal{L}_{AES} = -\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2, \quad (5)$$

where y_i and $\hat{y}_i$ denote the ground-truth and the predicted score of essay e_i respectively. Then, we employ the trained PANN to infer a pseudo-score $\hat{y}_i$ for each counterfactual instance $(\overline{p}_i, \widetilde{p}_i, e_i, \widetilde{y}_i)$, and calculate its score y'_i:

$$y'_i = \alpha\widetilde{y}_i + (1 - \alpha)\hat{y}_i, \quad (6)$$

where α is a tradeoff parameter. Finally, we continue to train PANN on the combination of the original training set and these counterfactual instances.

4 Experiments

4.1 Datasets and Experiment Settings

We use two public datasets for the experiments of prompt-generalized essay scoring. The first is the ASAP (Automated Student Assessment Prize) dataset[1], which contains 12,978 essays from eight prompts of different genres (i.e., ARG, RES, and NAR) scored in various ranges. The second is the TOEFL11 (Blanchard et al., 2013), which contains 12,100 essays sampled from eight prompts and scored by three levels (low/medium/high). These two datasets are widely used by current studies on AES (Dong and Zhang, 2016; Jin et al., 2018; Nguyen and Litman, 2018). The detailed statistics of these two datasets are listed in Table 1.

For prompt-generalized essay scoring, we design experiments on two datasets using prompt-wise leave-one-out validation. One prompt is used as test set, while the remaining seven prompt are randomly divided into training set and validation set by a ratio of 4 to 1. The model achieving the best performance on validation set is used for testing. To measure the performance of essay scoring, we adopt the widely-used Quadratic Weighted Kappa (QWK) (Dong and Zhang, 2016; Jin et al., 2018). To reduce randomness, under each case, 5 runs are performed, and the average results are reported.

4.2 Implementation Details

In our PANN model, for PA-net, the number of kernels is set to 8. The μ_k of eight kernels is uniformly selected from $[-1, 1]$ with equal interval, while the kernel width σ_k is set to 0.1. For EQ-net, the essay encoder is initialized with the weights of the 'uncased BERT-based model'[2]. For the essay scoring predictor, the number of FC layers is set to 2. For the data augmentation in CNAA strategy, the μ and σ of random score reduction is set to 0.4 and 1 respectively. For the ASAP dataset, we select thresholds δ_l and δ_h with grid search ($\delta_l \in [0.2, 0.5]$ and $\delta_h \in [0.6, 0.9]$) and finally set $\delta_l = 0.3$ and $\delta_h = 0.8$. For the TOEFL11 dataset, we directly use the three-level interval division defined by the dataset, without the need to set specific δ_l and δ_h values. For score merging in CST strategy, the tradeoff parameter α is set to 0.8. For model training, the Adam optimizer is adopted, and the learning rate is set to 5×10^{-5}. For the training of AES models, the ground-truth scores of essays are rescaled into $[0, 1]$. For the results evaluation, the predicted scores are rescaled to the original score range of the corresponding prompts. Our model is implemented in PyTorch1.4 and trained on 1 NVIDIA Tesla V100 GPU. The number of parameters in our model is 112.52M. The computational budget for running PANN and PANN+DRL with one epoch is 0.036 and 0.059 GPU hours, respectively.

[1]https://www.kaggle.com/c/asap-aes/data
[2]https://huggingface.co/BERT-base-uncased

Dataset	Prompt	#Essay	Genre	Avg Len	Range
ASAP	1	1,783	ARG	350	2-12
	2	1,800	ARG	350	1-6
	3	1,726	RES	150	0-3
	4	1,772	RES	150	0-3
	5	1,805	RES	150	0-4
	6	1,800	RES	150	0-4
	7	1,569	NAR	250	0-30
	8	723	NAR	650	0-60
TOEFL11	1	1656	ARG	332	l/m/h
	2	1562	ARG	331	l/m/h
	3	1396	ARG	283	l/m/h
	4	1509	ARG	302	l/m/h
	5	1648	ARG	349	l/m/h
	6	960	ARG	203	l/m/h
	7	1686	ARG	335	l/m/h
	8	1683	ARG	340	l/m/h

Table 1: Statistics of the ASAP and TOEFL11 datasets. For column Genre, ARG denotes argumentative essays, RES denotes response essays, and NAR denotes narrative essays. The last column lists the score ranges.

4.3 Comparison with Other Methods

We compare our method with the following methods under prompt-generalized setting, including three types of methods: handcrafted features based, neural network based, and hybrid.

• **BLRR** (Phandi et al., 2015) and **RankSVM** (Jin et al., 2018) are based on handcrafted features, where correlated Bayesian linear regression and rankSVM are used for prediction respectively.

• Neural AES models: **2L-LSTM** (Alikaniotis et al., 2016), **HCNN** (Dong and Zhang, 2016), **CNN-LSTM-MoT** (Taghipour and Ng, 2016), and **CNN-LSTM-Att** (Dong et al., 2017).

• **BERT** has recently been used for AES (Mayfield and Black, 2020; Cao et al., 2020; Uto et al., 2020b), which is also used to initialize our EQ-net. **BERT-Dual** indicates the **BERT** with essay-prompt text pair as dual input.

• **PAES** (Ridley et al., 2020) is a prompt-generalized hybrid model, but it needs to use the available target-prompt essays to normalize feature values of the entire test set. We denote the ratio of

Dataset	Method	Target Unseen Prompt								
		P1	P2	P3	P4	P5	P6	P7	P8	Avg.
ASAP	**BLRR**	0.472	0.45	0.325	0.507	0.663	0.563	0.492	0.257	0.466
	RankSVM†	0.737	0.467	0.464	0.511	0.669	0.529	0.586	0.408	0.546
	PAES-Target$_{40\%}$†	**0.798**	0.628	0.659	0.653	0.756	0.626	0.724	0.64	0.686
	PAES-Target$_{20\%}$†	–	–	–	–	–	–	–	–	0.650
	2L-LSTM	0.432	0.390	0.473	0.647	0.622	0.494	0.495	0.337	0.486
	HCNN	0.479	0.403	0.532	0.576	0.604	0.543	0.349	0.433	0.490
	CNN-LSTM	0.473	0.367	0.506	0.620	0.609	0.485	0.454	0.313	0.478
	CNN-LSTM-ATT	0.418	0.314	0.473	0.589	0.556	0.566	0.517	0.330	0.470
	BERT	0.609	0.499	**0.666**	**0.681**	0.724	0.637	0.699	0.537	0.632
	BERT-Dual	0.270	0.484	0.578	0.529	0.542	**0.671**	0.232	0.586	0.487
	PANN (Ours)	0.762	**0.686**	0.637	0.673	**0.778**	0.664	**0.742**	**0.677**	**0.702**
TOEFL11	**BLRR**	0.273	0.388	0.462	0.441	0.413	0.398	0.388	0.406	0.396
	RankSVM	0.575	0.524	0.645	0.607	0.548	0.558	0.56	0.549	0.571
	2L-LSTM	0.483	0.348	0.500	0.483	0.508	0.565	0.451	0.469	0.476
	HCNN	0.457	0.509	0.619	0.463	0.569	0.587	0.480	0.558	0.530
	CNN-LSTM	0.510	0.530	0.606	0.557	0.586	0.582	0.458	0.549	0.547
	CNN-LSTM-ATT	0.525	0.503	0.612	0.555	0.634	0.612	0.501	0.511	0.557
	BERT	0.592	0.645	0.656	0.593	0.662	0.685	0.633	0.613	0.635
	BERT-Dual	0.683	0.658	0.706	0.685	0.672	0.680	0.661	0.673	0.677
	PANN (Ours)	**0.701**	**0.662**	**0.722**	**0.686**	**0.697**	**0.705**	**0.700**	**0.685**	**0.695**

Table 2: QWK measures achieved in target unseen prompts on both ASAP and TOEFL11 datasets. The best measures are in bold. † denotes that the data is referenced from its original paper.

target data it uses for feature normalization.

The results are listed in Table 2. As shown, our *PANN* model can outperform most baseline methods by a large margin and achieve the best overall performance on both datasets (i.e., 0.702 on ASAP and 0.695 on TOEFL11). This indicates that our method is effective for prompt-generalized essay scoring. Besides, *BERT* performs good and stably on both datasets, but *BERT-Dual* performs significantly different on two datasets (i.e., 0.487 on ASAP and 0.677 on TOEFL11). This may be because, compared with *BERT*, which only takes essays as input, *BERT-Dual* takes both prompt and essay as its inputs, making its performance easily affected by the prompt-specific information. While all eight prompts of TOEFL11 are of the same genre (i.e., argumentative essay) and their prompt are of the same template, ASAP contains three genres and the templates of different prompts vary a lot. This may make *BERT-Dual* easier to generalize well on TOEFL11, but harder to generalize on ASAP. This also indicates that prompt-specific information is useful for essay scoring, but is easily entangled with the prompt-invariant information and thus affects the generalizability.

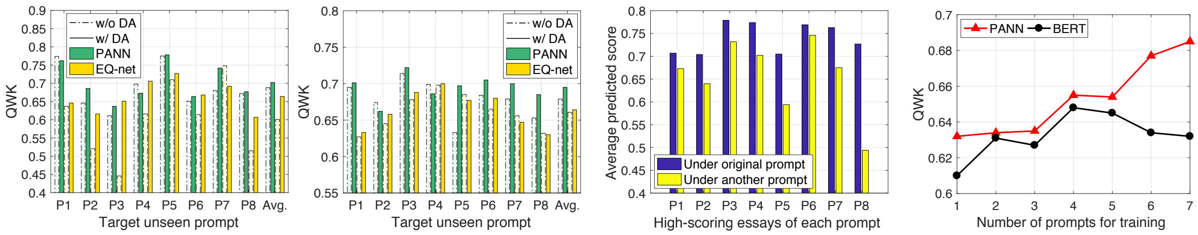
By observing other baseline methods, we can find that the neural models without pre-training perform significantly worse than *BERT*. The hand-crafted features based methods (e.g. *RankSVM*) perform stably on both datasets and can outperform many neural AES models. *PAES-Target*$_{40\%}$ achieves good performance on ASAP, but it needs 40% of essays from the target prompt for feature normalization and cannot work well when only a handful of target prompt essays are given.

4.4 Ablation Study

We then explore the effect of the components (i.e., PA-net and EQ-net) and the disentangled representation learning framework (i.e., NIA, ASA, and CST) on the performance of PANN, by adding each of them one by one. As shown in Table 3, the performance of combining the two components (i.e., PA-net+EQ-net) is better than the individual performance of either PA-net or EQ-net. This indicates that both PA-net and EQ-net can provide useful information for essay scoring. By observing the disentangled representation learning framework, we can find that the performance of EQ-net is improved when EQ-net is pre-trained with NIA and ASA together (i.e., 0.632 to 0.664 on ASAP and 0.635 to 0.666 on TOEFL11). But when EQ-net is pre-trained only with one of them, the performance is degraded on TOEFL11. Similar phenomenon can be observed for PA-net+EQ-net. This may be because these two losses need to be used simultaneously to disentangle quality and content information. Besides, CST strategy also needs to be used together with CNAA strategy to achieve better performance. In summary, all components and disentanglement strategies contribute to the final performance of PANN.

Dataset	Model Setting	Target Unseen Prompt								
		P1	P2	P3	P4	P5	P6	P7	P8	Avg.
ASAP	**PA-net**	0.719	0.370	0.484	0.408	0.709	0.650	0.635	0.523	0.562
	EQ-net	0.609	0.499	**0.666**	0.681	0.724	0.637	0.699	0.537	0.632
	+ **NIA**	0.618	0.599	0.596	0.677	0.751	0.653	0.645	0.586	0.641
	+ **ASA**	0.565	0.587	0.658	0.682	0.763	0.659	0.608	0.555	0.635
	+ **NIA&ASA**	0.646	0.616	0.651	**0.706**	0.727	**0.668**	0.692	0.607	0.664
	PA-net + EQ-net	0.698	0.592	0.616	0.645	0.731	0.610	0.576	0.579	0.631
	+ **NIA**	0.705	0.623	0.623	0.652	0.734	0.625	0.588	0.588	0.642
	+ **ASA**	0.694	0.597	0.598	0.622	0.725	0.609	0.552	0.607	0.626
	+ **NIA&ASA**	**0.772**	0.657	0.630	0.697	0.776	0.651	0.707	**0.691**	0.698
	+ **CST**	0.727	0.580	0.630	0.658	0.758	0.606	0.624	0.610	0.649
	+ **NIA&ASA&CST**	0.762	**0.686**	0.637	0.673	**0.778**	0.664	**0.742**	0.677	**0.702**
TOEFL11	**PA-net**	0.500	0.294	0.543	0.488	0.474	0.429	0.475	0.463	0.458
	EQ-net	0.592	0.645	0.656	0.593	0.662	0.685	0.633	0.613	0.635
	+ **NIA**	0.684	0.377	0.655	0.676	0.574	0.580	0.526	0.563	0.579
	+ **ASA**	0.661	0.289	0.657	0.680	0.605	0.659	0.580	0.447	0.572
	+ **NIA&ASA**	0.633	0.658	0.688	0.700	0.677	0.680	0.647	0.643	0.666
	PA-net + EQ-net	0.650	0.636	0.678	0.635	0.654	0.628	0.682	0.631	0.649
	+ **NIA**	0.642	0.649	0.676	0.658	0.675	0.576	0.647	0.614	0.642
	+ **ASA**	0.547	0.645	0.668	0.666	0.678	0.484	0.612	0.624	0.616
	+ **NIA&ASA**	0.685	0.661	0.682	**0.705**	**0.717**	0.666	0.671	0.654	0.680
	+ **CST**	0.558	0.596	0.688	0.652	0.580	**0.715**	0.606	0.640	0.629
	+ **NIA&ASA&CST**	**0.701**	**0.662**	**0.722**	0.686	0.697	0.705	**0.700**	**0.685**	**0.695**

Table 3: Ablation study of our method on both datasets. 'NIA' and 'ASA' indicate two losses in CNNA strategy for the pre-training of EQ-net. 'CST' indicates the counterfactual self-training strategy for the fine-tuning of PANN.



(a) Effect of Data Augmentation in CNAA startegy (b) Effect of PA-net (c) Effect of data size

Figure 5: Effect of different components and factors on the essay scoring performance of our method.

4.5 Further Analysis

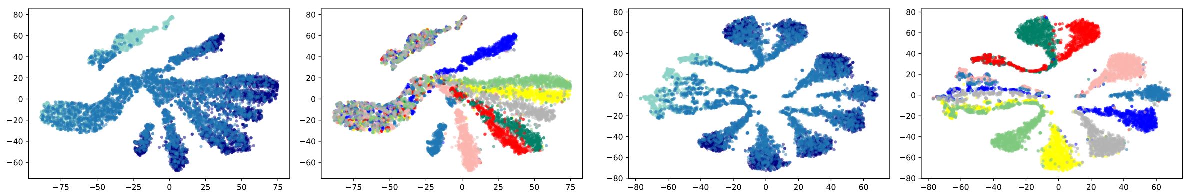
We further analyze the effects of more designs and factors on the performance of our method.

Effect of Data Augmentation We first analyze whether the data augmentation in CNAA strategy can boost the generalization ability of our method by plotting performance with and without using data augmentation. As shown in Figure 5(a), we can find that both PANN and EQ-net can benefit from data augmentation on most prompts of both datasets, especially on P3 of the ASAP dataset (left figure) and P5 of the TOEFL11 dataset (right figure).

Effect of PA-net We are also interested in whether PA-net can independently influence the final score prediction. For each target unseen prompt on ASAP, we select all high-scoring essays and predict their scores under their original prompt and another prompt. As shown in Figure 5(b), PANN predicts a lower average score for high-scoring essays under an unmatched prompt. While EQ-net output unchanged features under both settings, PA-net can be aware of the change in prompt.

Effect of Data Size We then analyze the effect of data size on performance by selecting one prompt as test set and adding remaining prompts for training one by one. Experiments are conducted on TOEFL11, since it contains essays of the same genre (i.e., ARG). As shown in Figure 5(c), the prediction performance of our PANN is on the rise with the growth of the data size, while BERT shows a trend of first rising and then falling. This indicates that our representation disentanglement strategies can deal well with the entangled information brought by the growth of prompts, so that the model can benefit from the data growth.

Feature Visualization To further analyze the learned latent space of CNAA strategy, we visualize the distributions of essay quality features with



(a) Entangled EQ-features: score (left), prompt (right) (b) Disentangled EQ-features: score (left), prompt (right)

Figure 6: Feature visualization for EQ-net with (a) direct training and (b) our CNAA strategy on TOEFL11 dataset. Three colors for score indicate low/medium/high and eight colors for prompt (pink indicates the unseen prompt).

t-SNE in Figure 6. For better comparison, we show feature distributions of EQ-net with and without using CNAA strategy. From Figure 6(a), we can find that scores of three levels are relatively well separated (left), but essays of different prompts are not completely separated, especially the essays with medium and low score (right). In contrast, as shown in Figure 6(b), when using our CNAA strategy, scores can be separated well according to different norms, and prompts can be separated well according to different angular directions.

5 Conclusion

In this paper, we focus on the prompt-generalized AES task. We propose the prompt-aware neural network model PANN to comprehensively evaluate the essays in terms of both prompt adherence and writing quality. To improve its generalization, we further propose a disentangled representation learning framework, including two representation disentanglement strategies. Experimental results demonstrate the effectiveness of the proposed method for prompt-generalized essay scoring.

Limitations

A major limitation of our work may be that our disentangled representation learning framework adopts some heuristic assumptions and designs in data augmentation and counterfactual data construction, and it remains to be seen whether they are applicable to other datasets and other languages. In particular, for the data augmentation of CNAA strategy, we assume that more data can be synthesized by text concatenation and we heuristically decide the quality and content label of synthesized data by some random strategies. Besides, for the counterfactual data generation, we mainly generate counterfactual samples and scores heuristically through our intuition and experience, rather than building a generation model based on counterfactual reasoning. Considering that some researchers have already developed some counterfactual data generation models for NLP tasks such as neural dialogue generation (Zhu et al., 2020), we are interested in whether it is possible and better to build a counterfactual data generation model for our method.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant Nos. 61972192, 62172208, 61906085, 41972111. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

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

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. Toefl11: A corpus of non-native english. *Ets Research Report*, 2013(2):i–15.

Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. 2021. Exploiting domain-specific features to enhance domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, pages 21189–21201. Curran Associates, Inc.

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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1011–1020. ACM.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541, Online. Association for Computational Linguistics.

Madalina Cozma, Andrei M. Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 503–509.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

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

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

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural automated essay scoring and coherence modeling for adversarially crafted input. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, pages 263–271.

Peter W. Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *Proceedings of EdMedia + Innovate Learning 1999*, pages 939–944, Seattle, WA USA. Association for the Advancement of Computing in Education (AACE).

Zhiwei Jiang, Meng Liu, Yafeng Yin, Hua Yu, Zifeng Cheng, and Qing Gu. 2021. Learning from graph propagation via ordinal distillation for one-shot automated essay scoring. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2347–2356. ACM / IW3C2.

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

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6300–6308.

Beata Beigman Klebanov and Nitin Madnani. 2020. Automated evaluation of writing - 50 years and counting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7796–7810. Association for Computational Linguistics.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Elijah Mayfield and Alan W. Black. 2020. Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2020, Online, July 10, 2020*, pages 151–162.

Sharmila Reddy Nangi, Niyati Chhaya, Sopan Khosla, Nikhil Kaushik, and Harshit Nyati. 2021. Counterfactuals to control latent disentangled text representations for style 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 2: Short Papers)*, pages 40–48, Online. Association for Computational Linguistics.

Huy V Nguyen and Diane J Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ellis Batten Page. 1994. Computer grading of student prose, using modern concepts and software. *Journal of Experimental Education*, 62(2):127–142.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1534–1543.

Peter Phandi, Kian Ming Adam 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.

Aaron Reich, Jiaao Chen, Aastha Agrawal, Yanzhe Zhang, and Diyi Yang. 2022. Leveraging expert guided adversarial augmentation for improving generalization in named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1947–1955, Dublin, Ireland. Association for Computational Linguistics.

Robert Ridley, Liang He, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13745–13753. AAAI Press.

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. *CoRR*, abs/2008.01441.

Mattia Segù, Alessio Tonioni, and Federico Tombari. 2023. Batch normalization embeddings for deep domain generalization. *Pattern Recognit.*, 135:109115.

Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for intent classification and out-of-domain detection in SLU. 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 2443–2453, Online. Association for Computational Linguistics.

Takumi Shibata and Masaki Uto. 2022. Analytic automated essay scoring based on deep neural networks integrating multidimensional item response theory. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2917–2926, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of the 25th International conference on computational linguistics*, pages 950–961.

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

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. In *Proceedings of the 32nd Conference on Artificial Intelligence(AAAI-18)*, pages 5948–5955.

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

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020b. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6077–6088.

Jake Vasilakes, Chrysoula Zerva, Makoto Miwa, and Sophia Ananiadou. 2022. Learning disentangled representations of negation and uncertainty. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8380–8397, Dublin, Ireland. Association for Computational Linguistics.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Linjuan Wu, Shaojuan Wu, Xiaowang Zhang, Deyi Xiong, Shizhan Chen, Zhiqiang Zhuang, and Zhiyong Feng. 2022. Learning disentangled semantic representations for zero-shot cross-lingual transfer in multilingual machine reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 991–1000, Dublin, Ireland. Association for Computational Linguistics.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 287–296.

Attali Yigal, Bridgeman Brent, and Trapani Catherine. 2010. Performance of a generic approach in automated essay scoring. *Journal of Technology Learning & Assessment*, 10(3):17.

Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. 2020. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Advances in Neural Information Processing Systems*, volume 33, pages 14435–14447. Curran Associates, Inc.

Qingfu Zhu, Wei-Nan Zhang, Ting Liu, and William Yang Wang. 2020. Counterfactual off-policy training for neural dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3438–3448, Online. Association for Computational Linguistics.

A Details of PA-net

PA-net aims to generate a prompt adherence feature vector u for an input prompt $p = \{w_p^1, w_p^2, \cdot, w_p^m\}$ and essay $e = \{w_e^1, w_e^2, \cdot, w_e^n\}$ pair. As shown in Figure 2, PA-net achieves this goal via three main operations: PE matching matrix construction, kernel pooling, and prompt attention.

PE matching matrix refers to a matrix which represents the semantic matching information of word pairs from a prompt and essay pair. To construct the PE matching matrix, PA-net first uses an embedding layer to map each word w^i into an L-dimension word embedding t^i: $w^i \Rightarrow t^i$. Then, a matching layer is used to construct a PE matching matrix $M \in R^{m \times n}$ based on the mapped prompt $p = \{t_p^1, t_p^2, \cdots, t_p^m\}$ and essay $e = \{t_e^1, t_e^2, \cdots, t_e^n\}$. Each element $M_{i,j}$ is the semantic similarity between a prompt word t_p^i and an essay word t_e^j, which is measured by cosine similarity (Yang et al., 2016):

$$M_{i,j} = \cos(t_p^i, t_e^j).$$

Kernel pooling (Xiong et al., 2017) is an operation used to convert a vector u to a value $\phi(u)$ by applying a kernel function on vector u. For the row M_i of a PE matching matrix corresponding to the i-th prompt word, PA-net applies K kernels on M_i to pooling and maps it into a K-dimensional feature vectors $\phi(M_i)$:

$$\phi(M_i) = \{\phi_1(M_i), \phi_2(M_i), \cdots, \phi_K(M_i)\}.$$

The effect of kernel function ϕ depends on the kernel used. To measure the matching degree of prompt word w_p^i with all the essay words, we use the RBF kernel:

$$\phi_k(M_i) = \sum_{j=1}^{n} \exp\left(-\frac{(M_{ij} - \mu_k)^2}{2\sigma_k^2}\right)$$

where μ_k and σ_k represent the mean and width of the kernel. We can infer from the equation that the more word pairs with similarities $M_{ij} \in M_i$ close to the mean μ_k, the higher the value of $\phi_k(M_i)$ can reach. Compared to exact matching which is equivalent to term frequency, the RBF kernel function defines a soft term frequency (soft-TF), which allows words that related but not exactly matched contribute to the final matching result.

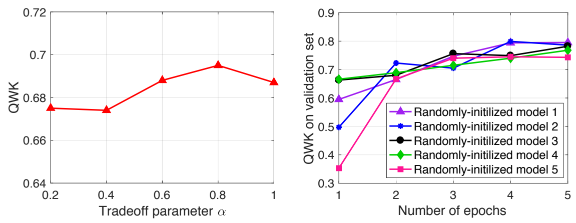
Prompt attention is an attention mechanism which converts m K-dimensional soft-TF vectors $\phi(M_i)$ into a K-dimensional prompt adherence feature vector v_p. Other pooling functions (e.g., average, min, and max pooling) that treat all words in the prompt with equal importance, are used as simultaneously. In practice, we find that only part of the key words in the prompt should be paid attention when measuring the prompt adherence of essays. Therefore, it is necessary to quantify the contributions of each word in the prompt. Unlike the general attention mechanism (Dong et al., 2017), prompt attention generates the attention weights based on the word embedding of prompt words, and apply the attention weights to the combination of soft-TF vectors. Given a prompt $p = \{t_p^1, t_p^2, \cdots, t_p^m\}$, the attention weight α_i for soft-TF can be defined as:

$$\alpha_i = \frac{\exp(u_i^\top u_p)}{\sum_{j=1}^{m} \exp(u_j^\top u_p)},$$

$$u_i = tanh(W_p \cdot t_p^i + b_p)$$

where u_p is a context vector, u_i is the hidden state of the i-th word in the prompt, W_p and b_p are the weight matrix and the bias vector respectively. Formally, the prompt adherence feature vector v_p is a weighted sum of soft-TF vectors $\phi(M_i)$ as:

$$v_p = \sum_{i=1}^{m} \alpha_i \phi(M_i).$$



(a) Effect of parameter α (b) Effect of training epoch

Figure 7: Effect of hyper-parameters.

B Effect of Hyper-parameters

For the hyper-parameter search, we use grid search to search for the best values and select the value that performs the best on the validation set. For example, we study the effect of the tradeoff parameter α by varying it from 0.2 to 1 with a step of 0.2. We take the experiments on the TOEFL11 dataset as an example and report the average performance of all eight prompts. As shown in Figure 7(a), the overall fluctuation of the line is not dramatic, and

Setting	$\delta_l = \mathbf{0.2}$ $\delta_h = \mathbf{0.9}$	$\delta_l = \mathbf{0.3}$ $\delta_h = \mathbf{0.8}$	$\delta_l = \mathbf{0.4}$ $\delta_h = \mathbf{0.7}$	$\delta_l = \mathbf{0.5}$ $\delta_h = \mathbf{0.6}$
QWK	0.695	**0.762**	0.723	0.687

Table 4: Effect of the thresholds δ_l and δ_h.

the maximum difference is within 0.02. The best performance is achieved at $\alpha = 0.8$. This indicates that our method is robust to this parameter, and our guessed pre-score needs a larger weight than the predicted score, which implies that our guessed pre-score can provide more counterfactual information for the improvement of prompt generalization.

We then explore the effect of training epochs. As shown in Figure 7(b), we select P6 of the TOEFL11 dataset as the test prompt and list the performance of five randomly-initilized models. We can see that all models can coverage in about 5 epochs on the validation set. Therefore, in our experiments, we only run each model for 5 epochs and select the epoch with best performance on the validation set for testing. For each case, we run the experiments five times and report the average results.

Finally, we explore the effect of the thresholds δ_l and δ_h. We define $\delta_l \in [0, 1]$, $\delta_h \in [0, 1]$, and $\delta_h > \delta_l$. Thus, the score range of essays can be divided into three intervals: $[0, \delta_l]$, (δ_l, δ_h), and $[\delta_h, 1]$. Since the score range of the TOEFL11 dataset is naturally divided into three intervals, we only set thresholds for the ASAP dataset. To observe the effect of interval changes on performance more clearly, we consider choosing the values of thresholds δ_l and δ_h symmetrically. As shown in Table 4, we select P1 of the ASAP dataset as the test prompt and list four different interval divisions. We can see that the combination of $\delta_l = 0.3$ and $\delta_h = 0.8$ achieves the best performance, while other more extreme divisions resulted in poorer performance. This may be because extreme divisions lead to an insufficient or excessive number of essays with low or high scores, resulting in insufficient training or inadequate discrimination between high-score and low-score essays, respectively.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The Limitations section

☑ A2. Did you discuss any potential risks of your work?
The Limitations section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4

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

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
4.1

☑ 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)?
4.1

☒ 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?
These two datasets are widely used for essay scoring and does not have these problems.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4.1

☑ 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.
4.1

C ☑ Did you run computational experiments?

4

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

☑ 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?
4.3, 4.4, Appendix B

☑ 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.)?
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.?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.