MLPAS: Encoder-only Essay Scoring with Multi-level Disentanglement

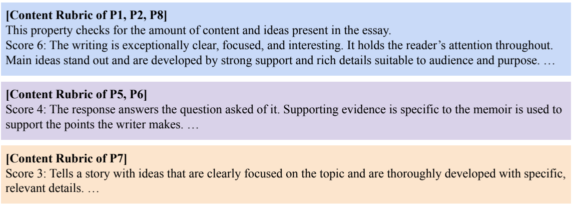
Anonymous COLING 2025 submission

Abstract

The application of language models in essay scoring has gained significant attention in recent years, typically evaluating a single model across multiple prompts. However, in a multi-prompt setup, it is crucial to understand the varying aspects of different prompts. In such settings, there are notable variations even in a trait with the same name across prompts, often overlooked in existing research. We propose introducing multi-level disentanglement into a Transformer encoder-only framework for essay scoring, preserving fine-grained semantic differences across such traits. Our method not only improves the quality of essay scoring, but also reduces memory usage and latency. Experimental results demonstrate that our framework achieves the highest agreement with human essay ratings over four SOTA approaches. The codes will become available upon acceptance.

1 Introduction

Recently, automated essay scoring (AES) has garnered significant attention due to advancements in various language models. In particular, methods leveraging pre-trained BERT-based models have been proposed, demonstrating superior performance compared to traditional approaches (Yang et al., 2020; Wang et al., 2022; Jiang et al., 2023). However, the majority of these studies focus on scoring for a single prompt or a single overall score. To address these limitations, several multi-trait scoring methods have been developed (Mathias and Bhattacharyya, 2020a; Ridley et al., 2021; Kumar et al., 2022; Do et al., 2023). The latest approaches employ pre-trained Transformer models with an encoder-decoder structure to learn the relationships between traits in a multi-prompt, multi-trait essay scoring (Do et al., 2024). This model generates text including scores in a sequence-to-sequence (Seq2Seq) manner, achieving better performance in multi-prompt, multi-trait essay scoring.



[Content Rubric of P1, P2, P8]
This property checks for the amount of content and ideas present in the essay.
Score 6: The writing is exceptionally clear, focused, and interesting. It holds the reader’s attention throughout. Main ideas stand out and are developed by strong support and rich details suitable to audience and purpose. …

[Content Rubric of P5, P6]
Score 4: The response answers the question asked of it. Supporting evidence is specific to the memoir is used to support the points the writer makes. …

[Content Rubric of P7]
Score 3: Tells a story with ideas that are clearly focused on the topic and are thoroughly developed with specific, relevant details. …

Figure 1: The evaluation criteria for *content* trait vary depending on the prompts.

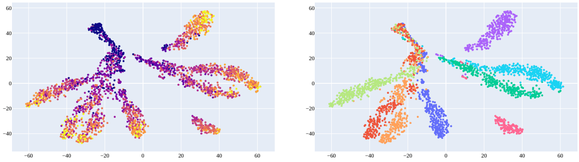


Figure 2: T-SNE visualisation of the embeddings extracted from the language model trained to evaluate *content* scores. (Colors: Left-Score, Right-Prompt)

In these studies, traits have been simply treated as equivalent if the names are identical across prompts. Yet, this approach overlooks the fact that different prompts may have different evaluation criteria for traits, or different semantic factors embedded in the essay that affect the score. Figure 1 shows that even the same trait can contains different evaluation criteria depending on the prompt. Figure 2 illustrates the gap in essay embedding distributions of the same trait (i.e., *content*) between essay prompts. Even essays with high scores on the same trait, the embedding distributions vary significantly depending on the prompt. Hence, overlooking such differences and treating representations as if they share the same distribution over different prompts collapses prompt-specific information.

Despite that a recent work proposed prompt-wise disentanglement in multi-prompt essay scoring (Jiang et al., 2023), it still lacks granularity because it only evaluates the *overall* trait and overlooks the variations in how the same trait manifests across different prompts in its disentanglement phase. Therefore, this semantic variation over the same traits underscores the necessity of

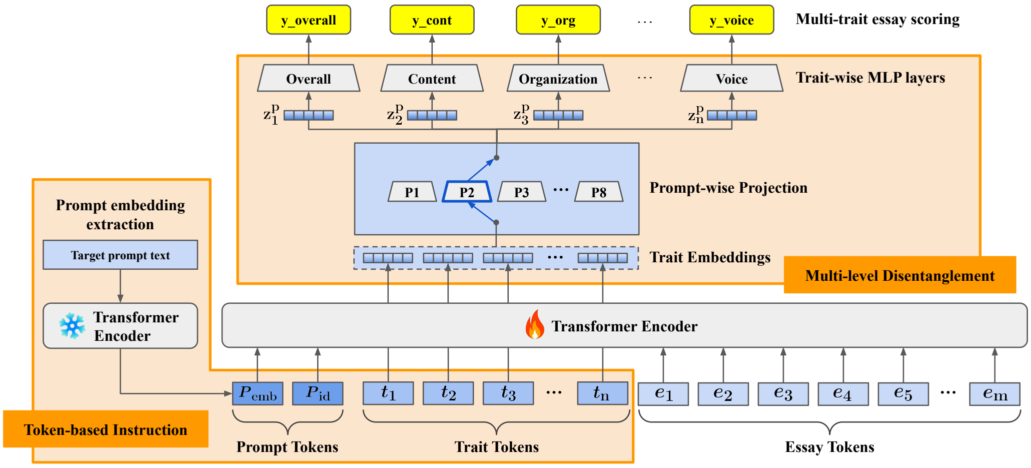


Figure 3: **Overview of MLPAS**: Token-based instruction provides prompt-specific information about the target essay to guide the model in outputting the essay score for each trait token, while multi-level disentanglement dynamically adjusts the projection layers for fine-grained, multi-level disentanglement over prompts and traits.

treating them differently at a fine-grained level according to each prompt. In addition, the state-of-the-art Seq2Seq essay scoring framework is not appropriate for achieving trait-level disentanglement. It turns out that a strong dependency between traits does exist in score generation. The order of traits generated by auto-regressive decoding significantly affects the scoring performance of the framework (Do et al., 2024). Besides, the Seq2Seq fashion results in high latency due to the need to maintain a Transformer decoder in addition to the encoder, and its expensive inference cost of auto-regressive decoding.

In this paper, we propose a novel multi-level disentanglement framework for multi-prompt essay scoring, **MLPAS** (**M**ulti-**L**evel **P**rompt-**A**daptive Multi-trait Essay **S**coring), designed to achieve fine-grained disentanglement at both prompt- and trait-levels. In this framework, as shown in Figure 3, we introduce the notion of *token-based instruction*, a set of instruction tokens as input to the Transformer encoder. The semantic embedding of the prompt text is extracted by a frozen text embedding model, followed by the concatenation with a learnable prompt ID token. They are then concatenated with multiple learnable trait tokens as the final instruction input. It provides prompt-specific information about the target essay to guide the model in outputting the essay score for each trait token. Furthermore, we leverage *multi-level disentanglement*, which introduces prompt-wise projection and trait-wise predictions to select the most suitable parameters for the target prompt and traits. Here, the proposed method enables fine-grained disentanglement at both prompt- and trait-levels by adjusting the projection layers on demand.

While each component is simple, the merger of them facilitates fine-grained, multi-level disentanglement with an efficient encoder-only Transformer architecture. This advantage of our framework enables a more accurate essay scoring with low latency. Our main contributions are:

- We propose a novel multi-level disentanglement framework MLPAS for multi-prompt, multi-trait essay scoring, enabling fine-grained disentanglement at both prompt- and trait-levels.

- MLPAS demonstrates the highest agreement with human essay ratings compared to existing SOTA essay scoring methods.

- We experimentally demonstrate that our method can achieve better performance even without a decoder, while also being more efficient in terms of parameter count and latency.

2 Related Work

2.1 Single-prompt Essay Scoring

Early research on AES primarily focused on prompt-specific essay scoring, training and testing each model for each essay prompt (Tay et al., 2018; Dong et al., 2017; Uto et al., 2020). Recently, pre-trained Transformer models, such as BERT (Devlin et al., 2019), have been successfully applied to prompt-specific essay scoring, yielding significant improvements in overall scoring accuracy (Yang et al., 2020; Wang et al., 2022). For a more precise scoring, several studies have proposed to use multiple traits in addition to the single overall score (Mathias and Bhattacharyya, 2018, 2020b; Hussein et al., 2020). Specifically, Kumar et al. (2022) proposed multi-task learning with BERT-based architectures. However, it requires training separate models for each trait, leading to substantial resource inefficiency.

2.2 Multi-prompt Essay Scoring

In real-world essay scoring, there are many types of essay prompts, and it is inefficient to create a separate model for each prompt. To address this issue, multi-prompt essay scoring models have been proposed that can evaluate essays from multiple prompts simultaneously. Multi-prompt essay scoring can be categorized into cross-prompt setting and prompt-adaptive setting.

The cross-prompt setup entails applying the evaluation model trained on seen prompts to unseen ones, thereby generalizing its effectiveness across various prompts. There have been numerous prior studies in this setup, but most of them have focused solely on evaluating a single overall score (Ridley et al., 2020; Cao et al., 2020; Jiang et al., 2023). A few studies have expanded it to mulit-trait scoring, yet they primarily focused on utilizing Part-Of-Speech(POS) tagging, while neglecting semantic distinctions at both the prompt- and trait-levels (Ridley et al., 2021; Do et al., 2023; Chen and Li, 2023, 2024).

The prompt-adaptive setup involves adjusting scoring criteria based on the specific prompt to enhance the accuracy of essay evaluation (Mathias and Bhattacharyya, 2020a; Kumar et al., 2022). The most recent work (Do et al., 2024) uses a pre-trained Seq2Seq Transformer, thereby achieving higher performance in multi-prompt, multi-trait essay scoring. Nevertheless, they are inadequate for capturing the fine-grained level of semantic difference across prompts and traits, due to their text generation instability, with high latency, posing an extra challenge for an efficient essay scoring.

In this study, our scope is to develop an efficient encoder-only essay scoring framework, which considers prompt- and trait-level semantic differences with multi-level disentanglement, under the prompt-adaptive setup.

3 Preliminaries

A prior study utilized the pre-trained encoder-decoder Transformer to generate trait scores in an auto-regressive manner (Do et al., 2024). In the study, only essay prompt ID and essay tokens were used as inputs to the Transformer encoder, and then the decoder generated a single text sequence consisting of trait and score pairs as:

$$\hat{S} = Dec(Enc([P_{\text{id}}|E])) \in \mathbb{R}^v. \qquad (1)$$

Here, $\hat{S}$ denotes the generated text sequence, Dec denotes the Transformer decoder, and P_{id} denotes the essay prompt ID. This text sequence is parsed to predict the final trait scores as:

$$\hat{Y} = g(\hat{S}) \in \mathbb{R}^n. \qquad (2)$$

Here, g is a parser that extracts a scalar value corresponding to each trait score from the text sequence.

This method outperformed existing other essay scoring methods, but overlooked the representation disentanglement, and also suffered from high latency and additional parameters due to its auto-regressive nature. To address these limitations, we propose an encoder-only model that facilitates fine-grained disentanglement at both prompt- and trait-levels and replaces the complicated auto-regressive task with a simple regression task for high effectiveness and efficiency.

4 Method: MLPAS Framework

4.1 Overview

We adopt an encoder-only architecture combined with multi-level disentanglement to achieve better performance with reduced latency and fewer parameters. We use a pre-trained Transformer encoder to extract trait-wise features from prompts and essays. The three token sets are concatenated into an integrated sequence S^*.

$$S^* = [P^*|T|E] \in \mathbb{R}^{(2+n+m)\times d}. \qquad (3)$$

Here, P^* represents the concatenated prompt embedding and prompt ID token embedding. The Transformer encoder processes S^* to generate the representation Z^*:

$$Z^* = \text{Enc}(S^*) \in \mathbb{R}^{(2+n+m)\times d}. \qquad (4)$$

The representation Z^*, which integrates the prompt, trait tokens, and essay, enables efficient multi-trait score prediction by passing the target trait representation set Z^*_T through a non-auto-regressive prediction function f, denoted as:

$$\hat{Y} = f(Z_T) \in \mathbb{R}^{n\times d}. \qquad (5)$$

This process enables the model to generate predictions for the target traits based on the extracted information from the prompt, essay, and target traits. Unlike the previous method described in Eq. (1) and Eq. (2), we do not follow the auto-regressive approach to predict final scores, which then allows us to predict each trait in parallel via regression.

4.2 Main Components

Token-based Instruction Prior studies often combine essay text and prompt IDs into a single sequence (Do et al., 2024), but this approach struggles to utilize semantic information in the prompts and learning distinct trait-specific representations. To address these limitations, we introduce three types of instruction tokens. The first type reflects the semantic information contained in the prompt through prompt embedding, the second type refers to special tokens assigned according to the prompt ID, and the third type consists of trait tokens, crucial for capturing trait-specific information. We assign a special token for each trait, and if the trait is not relevant to the prompt, it is masked with an *NA* token.

The prompt embedding and the prompt ID token together form P^* as:

$$P^* = [p_{\text{emb}} \| p_{\text{id}}] \in \mathbb{R}^{2\times d}. \tag{6}$$

The fixed pre-traiend Transformer encoder and average pooling are used to obtain prompt embedding $p_{\text{emb}} \in \mathbb{R}^d$. This approach can be applied regardless of the length of the prompt, because it needs a single extracted token, and has the advantage of being able to extract more fine-grained information by calculating the attentions with the prompt embedding at the essay text token level.

The final input sequence is formalized as:

$$S^* = [\underbrace{p_{\text{emb}}, p_{\text{id}}}_{\text{Prompt Tokens}}, \underbrace{t_1, t_2, \ldots, t_n}_{\text{Trait Tokens}}, \underbrace{e_1, e_2, \ldots, e_m}_{\text{Essay Tokens}}]. \tag{7}$$

This input sequence is fed into the Transformer encoder to get the sequence representation.

Multi-level Disentanglement We design a multi-level disentanglement approach to predict trait scores from trait token embeddings[1] to consider prompt-wise, trait-wise differentiation. We dynamically select the projection layer for multi-level disentanglement. It involves a two-step process: (1) *Prompt-wise projection* and (2) *Trait-wise multi-layer perceptrons (MLPs)*.

Firstly, we extract the trait token embeddings from the sequence embedding Z^*. Let $Z^*_T \in \mathbb{R}^{n\times d}$ represent the extracted trait token embeddings and $z_i \in \mathbb{R}^d$ represent the embedding of trait t_i:

$$Z^*_T = [z_1, z_2, \ldots, z_n]. \tag{8}$$

Prompt-wise projection captures the distinct characteristics of each prompt within the trait embeddings. Then, for each prompt, unique weights and biases are utilized:

$$Z^p_T = W^p \cdot Z^*_T + b^p. \tag{9}$$

Here, $W^p \in \mathbb{R}^{d\times d}$ and $b^p \in \mathbb{R}^d$ represent the weights and biases for projecting a specific prompt p, a set of specified projection parameters are dynamically selected based on the given prompts[2].

Trait-wise MLPs predicts the scores for each trait from the adapted trait embeddings:

$$\hat{y}_t = \text{MLP}_t(z^p_t). \tag{10}$$

In the equation, MLP_t represents the multi-layer perceptron used for predicting the score of trait t, z^p_t is the adapted embedding of trait t, and $\hat{y}_t$ is the predicted score of trait t.

These components capture trait variability across prompts by addressing both prompt-specific and trait-specific differences, leading to superior performance in multi-prompt, multi-trait essay scoring.

4.3 Model Training

We train our model in a multi-prompt, multi-trait setting, consistent with prior studies (Mathias and Bhattacharyya, 2020a; Kumar et al., 2022; Do et al., 2024). The training set comprises a mixture of essays from various prompts, each evaluated on multiple traits.

We update the model by calculating the Mean Squared Error (MSE) loss for each trait, masking the non-existent traits during the process:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sum_{t=1}^{T} m_{i,t}}\sum_{t=1}^{T} m_{i,t}(\hat{y}_{i,t} - y_{i,t})^2. \tag{11}$$

N is the total number of samples and T is the total number of traits. For each sample i and trait t,

[1] The subset of the trait tokens in Eq. (8) from the Transformer encoder's output in Eq. (4).

[2] Similar to the recent work, we assumes that prompt ID is given, while it can be extended to the cross-prompt setting by averaging the projections of trained prompts. We demonstrate the generalization ability in Section 5.2.3.

$\hat{y}_{i,t}$ denotes the predicted score; $y_{i,t}$ the true score. The mask $m_{i,t}$ is 1 if trait t exists for sample i and 0 otherwise. This loss function ensures that only the traits present for each sample contribute to the loss, allowing the model to learn effectively from the data with differences in trait sets between prompts. We normalize scores to a range of 0 to 10 for consistency across prompts and traits. For evaluation using the Quadratic Weighted Kappa (QWK) metric, we then map these scores back to their original trait-specific ranges.

5 Evaluation

Dataset We use the well-known ASAP and ASAP++ datasets (Mathias and Bhattacharyya, 2018), each of which consists of a set of English essays on eight prompts written by U.S. high school students in grades 7 through 10. We combine the ASAP dataset with the ASAP++ dataset to utilize the evaluation trait scores for all prompts. Consistent with prior studies (Kumar et al., 2022; Do et al., 2024), we use the same source data and ran our experiments using the same 5-fold split. Table 1 shows the dataset compositions, revealing significant overlaps in traits across distinct domains.

Compared Baselines We evaluate our model against several existing methods in multi-prompt, multi-trait settings. **STL-LSTM** (Dong et al., 2017) is an LSTM-CNN-based model for essay scoring in single-task learning scenarios. **HISK** (Cozma et al., 2018) utilizes a histogram intersection string kernel with a support vector regressor. **MTL-BiLSTM** (Kumar et al., 2022) employs a BiLSTM architecture for essay scoring in multi-task learning setups. **ArTS** (Do et al., 2024) is our main baseline, a recently proposed T5-based encoder-decoder model specifically designed for automated essay scoring and is considered a top-performing baseline. We compare the performance of our model with that of these baselines to demonstrate the effectiveness of our model in multi-prompt, multi-trait essay scoring.

Training Configuration We use only the encoder of the pre-trained T5 model (Raffel et al., 2020) to initialize the encoder of MLPAS and extract prompt embeddings. Regarding training, we set the early stop tolerance to 5, the batch size to {8, 16}, the learning rate to {1e-4, 2e-4}, and the total epochs to 20. We run our experiments on an NVIDIA A100 GPU with 40GB VRAM.

Prompt	Essay Domain	# Essays	Available Traits
1	Computer usage	1,785	Over, Cont, WC, Org, SF, Conv
2	Library censorship	1,800	Over, Cont, WC, Org, SF, Conv
3	Cyclist setting	1,726	Over, Cont, PA, Nar, Lang
4	Story analysis	1,772	Over, Cont, PA, Nar, Lang
5	Memoir mood	1,805	Over, Cont, PA, Nar, Lang
6	Empire State Building	1,800	Over, Cont, PA, Nar, Lang
7	Patience story	1,569	Over, Cont, Org, Conv, Style
8	Laughter importance	723	Over, Cont, WC, Org, SF, Conv, Voice

Table 1: Overview of multi-prompt, multi-trait ASAP/ASAP++ datasets used in experiments. Over: Overall, Cont: Content, WC: Word Choice, Org: Organization, SF: Sentence Fluency, Conv: Conventions, PA: Prompt Adherence, Nar: Narrativity, Lang: Language.

Evaluation Metric We leverage the quadratic weighted kappa (QWK), a widely adopted metric in existing AES studies (Ke and Ng, 2019; Ramesh and Sanampudi, 2022). QWK is renowned for its effectiveness in capturing agreement between human-rated and model-predicted scores. We report the average QWK of the trained model, which performs best on the validation dataset. To ensure a comprehensive evaluation, we report the QWK scores aggregated for each trait, *i.e.*, trait-wise comparison, and for each prompt, *i.e.*, prompt-wise comparison.

5.1 Main Results

5.1.1 Agreement with Human Rating

Tables 2 and 3 present the results of the agreement between automated essay evaluators and human ratings at two different levels: one for trait-wise and the other for prompt-wise. Overall, the results show that MLPAS surpasses other models in most cases (9 out of 11 for trait-wise and 6 out of 8 for prompt-wise). While it does not achieve the highest score in some cases (2 for trait-wise and 2 for prompt-wise), the performance gap is marginal.

We particularly observe significant improvement in P8 (the minority w.r.t the number of training examples)[3]. This improvement is mainly attributed to the robustness of MLPAS against bias from majority prompts (P1-P7), caused by their traits overlapping with those in P8. In the case of "Voice", it is not directly affected by the major prompts, but may be indirectly influenced through the overlapping traits in P8 (e.g., "Content" in P1 influencing "Content" in P8, which in turn impacts "Voice" in P8). Our proposed method effectively reduces inter-prompt and inter-trait interference, which suggests that we can achieve performance improvements in these more subtle prompts and traits.

[3] On P8, the performance increase by 0.08 both for the non-overlapping trait (e.g., “Voice”) and for traits that overlap with those in the major prompts.

Model Type	Model	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑ (SD↓)
Non-Transformer	HISK	0.718	0.679	0.697	0.605	0.659	0.610	0.527	0.579	0.553	0.609	0.489	0.611 (-)
	STL-LSTM	0.750	0.707	0.731	0.640	0.699	0.649	0.605	0.621	0.612	0.659	0.544	0.656 (-)
	MTL-BiLSTM	0.764	0.685	0.701	0.604	0.668	0.615	0.560	0.615	0.598	0.632	0.582	0.638 (-)
Encoder-Decoder	ArTS (main baseline)	0.754	0.730	**0.751**	**0.698**	0.725	0.672	0.668	0.679	0.678	**0.721**	0.570	0.695 (±0.018)
Encoder-only	**MLPAS (ours)**	**0.771**	**0.746**	**0.751**	0.691	**0.732**	**0.710**	**0.701**	**0.701**	**0.699**	0.702	**0.650**	**0.714** (±0.007)

Table 2: **Trait-wise** agreement with human ratings (PA: Prompt Adherence, Lang: Language, Nar: Narrativity, Org: Organization, Conv: Conventions, WC: Word Choice, SF: Sentence Fluency).

Model Type	Model	P1	P2	P3	P4	P5	P6	P7	P8	AVG↑ (SD↓)
Non-Transformer	HISK	0.674	0.586	0.651	0.681	0.693	0.709	0.641	0.516	0.644 (-)
	STL-LSTM	0.690	0.622	0.663	0.729	0.719	0.753	0.704	0.592	0.684 (-)
	MTL-BiLSTM	0.670	0.611	0.647	0.708	0.704	0.712	0.684	0.581	0.665 (-)
Encoder-Decoder	ArTS (main baseline)	0.708	0.706	**0.704**	0.767	0.723	**0.776**	0.749	0.603	0.717 (±0.025)
Encoder-only	**MLPAS (ours)**	**0.718**	**0.726**	0.703	**0.771**	**0.727**	0.765	**0.753**	**0.683**	**0.731** (±0.007)

Table 3: **Prompt-wise** agreement with human ratings (P1-P8 denote the prompts).

Model	# Parameters	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑ (SD↓)
ArTS (t5-small)	60M	0.712	0.695	0.720	0.667	0.711	0.630	0.606	0.631	0.625	0.694	0.474	0.651 (±0.026)
ArTS (t5-base)	220M	0.754	0.730	0.751	0.698	0.725	0.672	0.668	0.679	0.678	0.721	0.570	0.695 (±0.018)
ArTS (t5-large)	770M	0.751	0.730	0.750	0.701	0.728	0.675	0.682	0.680	0.680	0.715	0.603	0.700 (±0.024)
MLPAS(T5-small)	37M	0.760	0.733	0.738	0.681	0.719	0.690	0.686	0.696	0.689	0.711	0.640	0.704 (±0.007)
MLPAS(T5-base)	113M	0.771	0.746	0.751	0.691	0.732	0.710	0.701	0.701	0.699	0.702	0.650	0.714 (±0.007)
MLPAS(T5-large)	342M	0.768	0.752	0.760	0.700	0.731	0.719	0.709	0.702	0.713	0.693	0.641	0.717 (±0.008)

Table 4: **Trait-wise** agreement with human scores when using different sizes of T5 backbones.

Model	# Parameters	P1	P2	P3	P4	P5	P6	P7	P8	AVG↑ (SD↓)
ArTS (T5-small)	60M	0.696	0.669	0.682	0.732	0.712	0.743	0.712	0.492	0.680 (±0.029)
ArTS (T5-base)	220M	0.708	0.706	0.704	0.767	0.723	0.776	0.749	0.603	0.717 (±0.025)
ArTS (T5-large)	770M	0.701	0.698	0.705	0.766	0.725	0.773	0.743	0.635	0.718 (±0.030)
MLPAS(T5-small)	37M	0.710	0.702	0.694	0.758	0.725	0.756	0.728	0.669	0.718 (±0.009)
MLPAS(T5-base)	113M	0.718	0.726	0.703	0.771	0.727	0.765	0.753	0.683	0.731 (±0.007)
MLPAS(T5-large)	342M	0.723	0.731	0.707	0.774	0.726	0.771	0.750	0.693	0.734 (±0.010)

Table 5: **Prompt-wise** agreement with human scores when using different sizes of T5 backbones.

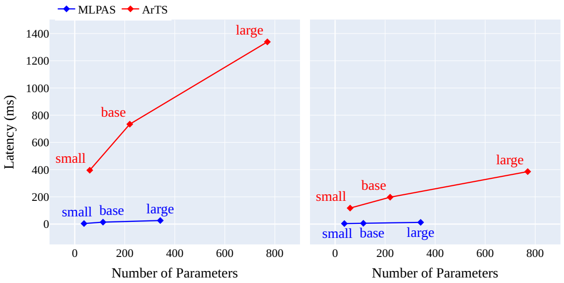
Therefore, our multi-level disentanglement successfully captures the distinct characteristics of each trait at a fine-granular level, minimizing the collapse of each trait by other prompts.

In addition, MLPAS outperforms ArTS, a Transformer model using the encoder-decoder structure, even though we eliminate the decoder part of the Transformer. While ArTS does not perform well on the "Overall" trait, which is the most important among all traits, MLPAS exhibits the best agreement on that trait, highlighting the potential of using the encoder-only structure compared to the encoder-decoder structure.

5.1.2 Model Size and Latency

A crucial aspect when using Transformers is scalability, achieved by replacing the backbone with a larger one. However, the increasing latency by the larger model is the main bottleneck to hinders its practical use for online essay evaluation. Hence, we have conducted an in-depth study on the trade-off between model size and latency.

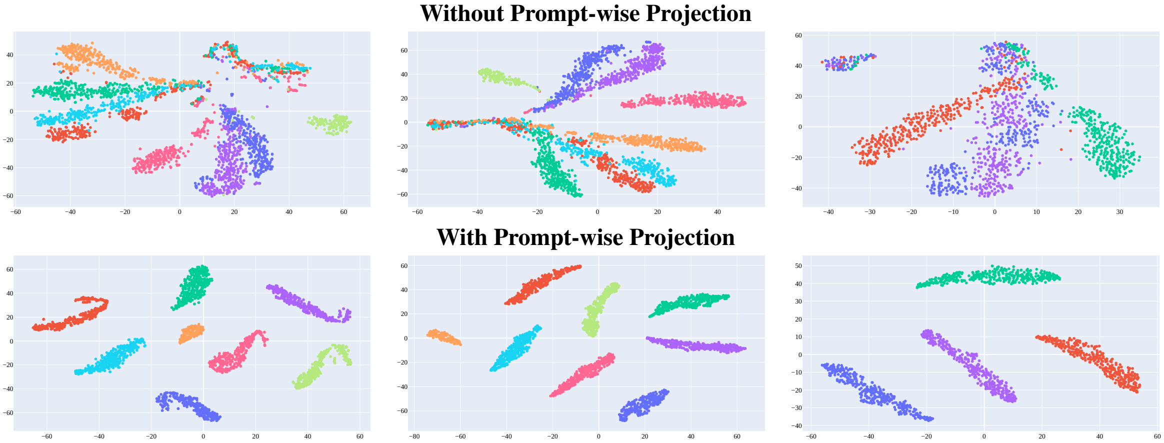
Figure 4 illustrates the trade-off between model size and latency (in milliseconds) over three different sizes of T5 backbones, namely small, base, and large from left to right. Notably, MLPAS exhibits a much better trade-off between them, achieving faster inference speed (lower latency) even with larger size T5 backbones. This improvement is attributed to (1) the removal of the Transformer decoder and (2) the replacement of a computationally heavy auto-regressive decoding task with a simple and efficient regression task.



(a) batch size = 1. (b) batch size = 4.

Figure 4: Trade-off between model size (# parameters) and latency (ms) when using two different batch sizes.

Tables 4 and 5 present the QWK scores of ArTS and MLPAS, from three different sizes of backbones. In general, both methods show good scalability w.r.t the model size, considering that they achieve higher performance as the size of models



(a) Overall.

(b) Content.

(c) Narrativity.

Figure 5: T-SNE visualization of the traits with the same name across different prompts, with and without the prompt-wise projection of MLPAS, Trait embeddings just before regression are used, with each prompt color coded.

Model	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑ (SD↓)
MLPAS	**0.771**	**0.746**	0.751	0.691	**0.732**	**0.710**	**0.701**	**0.701**	**0.699**	**0.702**	**0.650**	**0.714** (±0.007)
(1) w/o essay prompt	0.759	0.735	**0.755**	0.690	0.728	0.690	0.698	0.695	0.689	0.693	0.622	0.705 (±0.004)
(2) w/o prompt ID token	0.768	0.740	**0.755**	0.692	0.727	0.700	0.694	0.698	0.692	0.698	0.607	0.706 (±0.009)
(3) w/o trait tokens	0.770	0.741	0.752	**0.695**	**0.732**	0.692	0.693	**0.701**	0.690	0.683	0.633	0.707 (±0.005)
(4) w/o prompt-wise projection	0.760	0.734	0.754	0.682	0.727	0.696	0.696	0.703	0.697	0.694	0.635	0.707 (±0.009)

Table 6: Ablation study results based on traits.

gets larger (see the last "AVG" column for the tables). Nevertheless, our framework, MLPAS has better scalability compared with ArTS: (1) MLPAS (w. T5-small) achieves the average QWK score even better than ArTS (w. T5-large); (2) MLPAS (w. T5-large) is even faster than ArTS (w. T5-small) due to our efficient framework design, *i.e.*, the encoder-only Transformer; and (3) MLPAS needs much fewer trainable parameters than ArTS, thus leading to less computational cost. These advantages over ArTS suggest that our model offers substantial advantages in real-world use cases.

5.1.3 Impact of Multi-level Disentanglement

We investigate the impact of multi-level disentanglement by MLPAS on traits with the same name across different prompts. Figure 5 visualizes the trait embedding just before the regression layer with and without our prompt-wise projection. We observe a distinct difference with and without multi-level disentanglement by MLPAS. Without the prompt-wise projection, the unique characteristics of the trait in each prompt are likely to collapse (*i.e.*, the overlap of trait embeddings across prompts). In contrast, the trait embeddings produced by MLPAS exhibit a clear separation across prompts. Therefore, our multi-level disentanglement indeed helps each trait embedding to keep their individual semantic over prompts.

5.2 In-depth Analysis

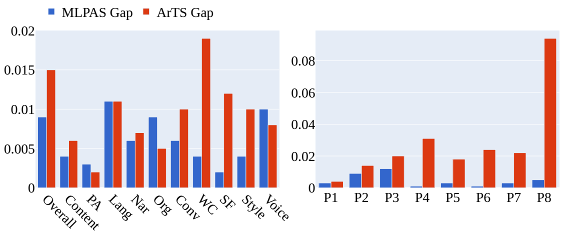
5.2.1 Ablation of Each Component

We have conducted an ablation study to assess the impact of each component in MLPAS. For the construction of token-based instruction, three components can be adjusted: (1) essay prompt embedding; (2) prompt ID, and (3) trait tokens. In addition, regarding the multi-level disentanglement, we can control (4) the use of the prompt-wise projection. Table 6 summarizes the results when each one of the four components is eliminated from the final model. The ablation results indicate that every component is essential for the best performance on essay evaluation.

Specifically, the most significant performance drop, on average, occurs when essay prompts are excluded. These results demonstrate the importance of utilizing the semantic information contained in prompts. Additionally, our observations indicate that our methods are particularly effective at leveraging the diverse semantic information contained in multiple prompts more efficiently.

5.2.2 Impact of Trait Ordering

The sensitivity to the order of traits in the input is a concern in the encoder-decoder ArTS (Do et al., 2024), due to its auto-regressive decoding. However, with the task replaced by a simple regression,



(a) Trait-wise Gap.

(b) Prompt-wise Gap.

Figure 6: Absolute QWK score gaps before and after reversing the order of traits in the input.

MLPAS alleviates the impact of the order on the predicted essay scores. Figure 6 well supports this by comparing the absolute gap of QWK scores before and after reversing the order of traits in the input. The smaller gaps over the ArTS at both levels demonstrate that MLPAS is more robust to the trait order than ArTS.

5.2.3 Generalization for Unseen Prompts

We conduct cross-prompt experiments to evaluate the generalization ablility of our method to unseen prompts. We compare the performance of MLPAS with SOTA cross-prompt AES models[4]. As in previous studies, we train the model on all prompts except the target prompt. To generalize to unobserved prompts in training, we apply prompt-wise z-score normalization to the embeddings from prompt-wise projection. We train the trait-wise MLP layers with fix the normalized embeddings, to avoid overfitting to the training dataset. In the inference step, we assume that the target prompt ID is unknown, and therefore use average pooled embeddings utilizing all the trained prompt-wise projections.

As illustrated in Figure 7, our method outperformed recent cross-prompt AES models in both average score and across the majority of prompts and traits (6 out of 9 for traits and 6 out of 8 for prompts). Our approach leverages pre-trained language models, offering a clear advantage over existing POS tagging-based methods by utilizing the latest advancements in language model technology.

5.2.4 Comparison with GPT-4

One interesting aspect is to compare MLPAS with recent LLMs. The foundational model, like GPT-4, is capable of performing human-like evaluation with prompt tuning (Kojima et al., 2022; Liusie et al., 2024; Zhang et al., 2024). Our finding is that the naive use of LLMs is inappropriate for accurate essay scoring due to the unclear scoring standards.

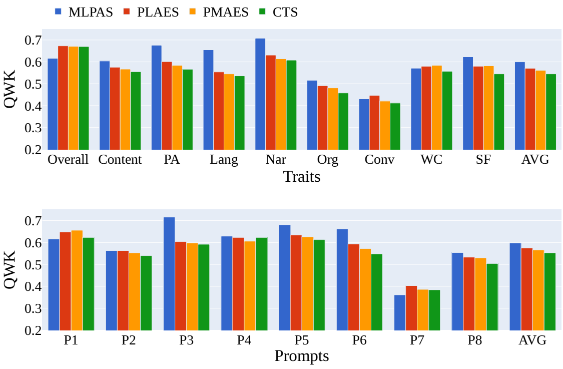


Figure 7: Cross-prompt performance comparison.

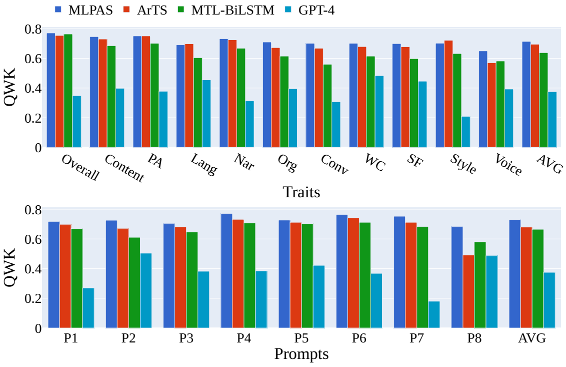


Figure 8: Agreement with human ratings using fine-tuned small models and LLMs as essay evaluators.

More specifically, we provide relevant traits and ranges of scores as prompts to GPT-4 for automatic essay scoring (see Appendix for the prompt details). Figure 8 illustrates the comparison of GPT-4's zero-shot evaluation performance with other models across various traits and prompts. The results indicate that GPT-4 performs poorly compared to other fine-tuned small language models, such as MLPAS and ArTS, highlighting the necessity of fine-tuning with clearly defined criteria for scoring.

6 Conclusion

In this work, we propose a multi-level disentanglement framework named MLPAS for essay scoring. This framework benefits from token-based instruction and prompt- and trait-levels disentanglement. They help the model keep the distinct semantic knowledge of each trait across prompts, even if the trait is shared across multiple prompts. Notably, MLPAS outperforms the SOTA models while achieving improved efficiency in terms of latency and parameter numbers. Through visualizations, we demonstrate the efficacy of the multi-level disentanglement. Furthermore, our findings underscore the limitations of zero-shot essay scoring with LLMs, highlighting the effectiveness of lightweigth fine-tuned essay scoring models.

[4] CTS (Ridley et al., 2021), PMAES (Ridley et al., 2020), and PLAES (Chen and Li, 2024).

7 Limitations

While our proposed model demonstrates significant improvements in multi-prompt, multi-trait essay scoring, several limitations must be acknowledged. Our experiments were conducted exclusively on benchmark datasets, and the performance and adaptability of our model in real-world applications with diverse essay topics and prompts remain to be tested. The current evaluation does not account for the introduction of new prompts over time, making it essential to investigate how well the model can adapt to these new prompts sequentially without significant retraining. Additionally, our study did not extensively explore the model's performance for the condition where the number of available essay samples for training is limited. In many practical situations, there may be insufficient data for certain prompts or traits, which could affect the model's robustness and generalization capabilities. Addressing these limitations in future research will be important for enhancing the applicability and effectiveness of the model in real-world essay scoring scenarios.

8 Ethics Statement

Potential Risks Our study was conducted using a constrained dataset, and our proposed method does not guarantee impartial essay scoring outcomes. Models for essay scoring may exhibit biases in their predictions based on the training data employed. The ASAP and ASAP++ datasets utilized in our research could potentially introduce biases towards specific demographic groups (Mathias and Bhattacharyya, 2018). However, it should be noted that demographic information was not provided in these datasets. To mitigate privacy concerns, any personally identifiable information within the essays has been anonymized.

Use of Scientific Artifacts Our research leveraged open-source tools including PyTorch (Paszke et al., 2019) and scikit-learn (Pedregosa et al., 2011), alongside pre-trained language models such as T5 obtained via the Huggingface (Wolf et al., 2019) library. The experiments were conducted using the ASAP and ASAP++ datasets, accessible for non-commercial research purposes. For experiments involving LLMs, we utilized OpenAI's API under their sharing and publication policy (OpenAI, 2022).

Use of Ai Assistants We only used ChatGPT to provide a better expression and to refine the wording. Some of the code used in the experiment was written with the assistance of Copilot.

References

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 '20, page 1011–1020, New York, NY, USA. Association for Computing Machinery.

Yuan Chen and Xia Li. 2023. PMAES: Prompt-mapping contrastive learning for cross-prompt automated essay scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1503, Toronto, Canada. Association for Computational Linguistics.

Yuan Chen and Xia Li. 2024. PLAES: Prompt-generalized and level-aware learning framework for cross-prompt automated essay scoring. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12775–12786, Torino, Italia. ELRA and ICCL.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Mădălina Cozma, Andrei Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 503–509, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Heejin Do, Yunsu Kim, and Gary Lee. 2024. Autoregressive score generation for multi-trait essay scoring.

In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1659–1666, St. Julian's, Malta. Association for Computational Linguistics.

Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. Prompt- and trait relation-aware cross-prompt essay trait scoring. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1538–1551, Toronto, Canada. Association for Computational Linguistics.

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

Mohamed Abdellatif Hussein, Abd El-Latif Hesham, 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.

Zhiwei Jiang, Tianyi Gao, Yafeng Yin, Meng Liu, Hua Yu, Zifeng Cheng, and Qing Gu. 2023. Improving domain generalization for prompt-aware essay scoring via disentangled representation learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12456–12470, Toronto, Canada. 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, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Rahul Kumar, Sandeep Mathias, Sriparna Saha, and Pushpak Bhattacharyya. 2022. Many hands make light work: Using essay traits to automatically score essays. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1485–1495, Seattle, United States. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM comparative assessment: Zero-shot NLG evaluation through pairwise comparisons using large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian's, Malta. Association for Computational Linguistics.

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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sandeep Mathias and Pushpak Bhattacharyya. 2020a. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91, Seattle, WA, USA → Online. Association for Computational Linguistics.

Sandeep Mathias and Pushpak Bhattacharyya. 2020b. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91, Seattle, WA, USA → Online. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`. Last accessed on 2024-01-15.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. *Proceedings of the AAAI*

Conference on Artificial Intelligence, 35(15):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. *Preprint*, arXiv:2008.01441.

Yi Tay, Minh Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Ruohong Zhang, Yau-Shian Wang, and Yiming Yang. 2024. Generation-driven contrastive self-training for zero-shot text classification with instruction-following LLM. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 659–673, St. Julian's, Malta. Association for Computational Linguistics.

A Additional Experiments

In this section, we present additional experiments that were not included in the main text due to space constraints. These include ablation study results based on traits, exploration of diverse backbone models, and the use of high-scoring essays as an alternative to prompts. These experiments offer valuable insights into the characteristics of our model and suggest directions for future research.

A.1 Ablation Study

Tables 7 and 8 presents the results of our ablation study based on prompts. Excluding components of the model generally led to lower performance compared to using all components. Notably, omitting the essay prompts resulted in the largest average performance drop.

Interestingly, the *Prompt Adherence (PA)* trait increased when the essay prompts were omitted. This can be attributed to the fact that P3-P6, which involve the *Prompt Adherence* trait, are source-dependent essay types; however, the dataset's essay prompts lack the source text. For instance, P5 requires describing the mood of a given memoir, but the prompt does not provide the content of the memoir. This lack of source text is a common issue in P3-P6. To address these issues, we have conducted additional experiments, which are detailed in the next subsection.

A.2 Utilizing High-Score Essays as Prompts

To address the absence of source text in the prompts for source-dependent essays, we have experimented with the idea of replacing the prompts with high-scoring essays. For each prompt, we randomly sample one essay with the highest overall score and an average across all traits of at least 80% of a max score. We use these sampled essays as proxies for the prompts to observe changes in model performance. Additionally, we instruct GPT-4 to rewrite these sampled essays and prompts as high-quality essays.

Tables 9 and 10 summarize the results of this experiment. "Gold Essay" refers to the sampled high-scoring essays, while "Gold Essay GPT-4" refers to the essays transformed by GPT-4. In both cases, the average performance is lower than the traditional method, but the Prompt Adherence score is higher. These results suggest that the absence of source text in source-dependent essay types can be mitigated by using other high-quality essays. We also observed that the decrease in average scores was smaller when using essays rewritten by GPT-4 compared to using the sampled essays as is. This finding indicates that data augmentation with LLMs can effectively fill gaps in the existing dataset.

A.3 Evaluating Performance with Various Backbone

We have conducted multiple experiments to verify the effectiveness of our proposed method with various Transformer backbone models. Specifically, we applied BART (Lewis et al., 2020) and FLAN-T5 (Chung et al., 2022) models at different scales, in addition to the T5-based models previously used. Tables 11 and 12 present the results of these experiments.

Our findings indicate that the proposed method is effective across different Transformer backbone models. Except for the BART-base model, our approach consistently achieves higher average QWK performance compared to the ArTS using the T5-base model. Given that the BART-base encoder contains approximately 70 million parameters, these results demonstrate the robustness and effectiveness of our method across a broad range of pre-trained transformer encoders.

A.4 Impact of Trait Order

One of the advantages of our proposed encoder-only model is its stability, with minimal performance variation based on the order of traits. Previous work generates trait scores sequentially through the decoder, which inherently leads to order sensitivity. Experiments have shown that certain orderings can achieve higher performance. In contrast, our method learns representations for each trait simultaneously through the encoder, eliminating dependence on the order of generation. Tables 13 and 14 provide specific data supporting this claim, as illustrated in Figure 6. Our findings show that the encoder-only approach exhibits greater stability across different trait orders. This stability is particularly notable in the case of the minor prompt P8. Such robustness enables us to streamline the process of identifying the optimal trait sequence, thereby reducing the time and resources involved in experimentation.

A.5 LLM-based Prompt Augmentation

We conduct experiments using LLM-based prompt augmentation to evaluate our model's ability to make accurate inferences across a variety of essay prompts. Each prompt is augmented with 30 semantically similar but textually different prompts generated using GPT-3.5-turbo. This augmentation is applied during both the training and testing phases to observe performance changes.

Tables 15 and 16 present the results of these experiments. In the "Prompt Augmentation" column, $A \rightarrow B$ indicates that prompt A was used during the training phase, while prompt B was used during the testing phase. Our findings show performance drop was not significant when varying prompts during training. Moreover, using different prompts in the training phase sometimes led to improved performance. Performance decreased when testing with various prompts, but our model still achieved higher performance than ArTS, which does not utilize prompts. These results highlight not only the robustness of our model but also suggest that leveraging the power of LLMs can enhance the performance of relatively smaller models.

A.6 Cross-prompt Setup

We provide additional explanations on how to extend MLPAS for cross-prompt setups. By leveraging the prompt-wise projection and normalization techniques, MLPAS can be adapted to handle unseen prompts during inference. We apply prompt-wise z-score normalization to the embeddings produced by the prompt-wise projection. Specifically, given the embedding Z_T^p for a specific prompt p, we normalize it using:

$$\tilde{Z}_T^p = \frac{Z_T^p - \mu_p}{\sigma_p}$$

where μ_p and σ_p represent the mean and standard deviation of Z_T^p for prompt p, respectively. $\tilde{Z}_T^p$ is the normalized embedding for the given prompt p after normalization. This normalization ensures that the embeddings are standardized, mitigating the distribution gap between prompts and risk of overfitting to the training dataset.

The trait-wise MLP layers are then trained using these normalized embeddings to predict the trait scores. At this stage, the input embeddings are fixed and only the trait-wise MLP layers are updated.

$$\hat{y}_t = \text{MLP}_t(\tilde{Z}_T^p)$$

where $\hat{y}_t$ denotes the predicted score for trait t, MLP_t represents the multi-layer perceptron used for predicting the score of trait t.

During inference, we assume that the target prompt ID is unknown, we use average pooling across all the trained prompt-wise projections to generate the embeddings. The average pooled embedding is computed as:

$$\overline{Z}_T = \frac{1}{K} \sum_{p=1}^{K} Z_T^p$$

where K is the number of trained prompts and $\overline{Z}_T$ is the generalized embedding for unseen prompt sample. This pooling approach allows the model to make generalized predictions even when encountering unseen prompts.

Model	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑ (SD↓)
MLPAS	**0.771**	**0.746**	0.751	0.691	**0.732**	**0.710**	**0.701**	**0.701**	**0.699**	**0.702**	**0.650**	**0.714** (±0.007)
(1) w/o essay prompt	0.759	0.735	**0.755**	0.690	0.728	0.690	0.698	0.695	0.689	0.693	0.622	0.705 (±0.004)
(2) w/o prompt ID token	0.768	0.740	**0.755**	0.692	0.727	0.700	0.694	0.698	0.692	0.698	0.607	0.706 (±0.009)
(3) w/o trait tokens	0.770	0.741	0.752	**0.695**	**0.732**	0.692	0.693	**0.701**	0.690	0.683	0.633	0.707 (±0.005)
(4) w/o prompt-wise projection	0.760	0.734	0.754	0.682	0.727	0.696	0.696	0.703	0.697	0.694	0.635	0.707 (±0.009)

Table 7: Ablation study results based on traits.

Model	P1	P2	P3	P4	P5	P6	P7	P8	AVG↑ (SD↓)
MLPAS	**0.718**	**0.726**	0.703	0.771	**0.727**	0.765	**0.753**	**0.683**	**0.731** (±0.007)
(1) w/o essay prompt	0.714	0.724	0.703	0.765	0.717	0.768	0.746	0.650	0.723 (±0.009)
(2) w/o prompt ID token	0.712	0.716	**0.705**	0.768	0.725	0.768	**0.753**	0.664	0.726 (±0.008)
(3) w/o trait tokens	0.714	0.715	0.703	**0.776**	0.719	**0.777**	0.743	0.666	0.727 (±0.006)
(4) w/o prompt-wise projection	0.715	0.716	0.703	0.767	0.716	0.767	0.743	0.662	0.724 (±0.008)

Table 8: Ablation study results based on prompts.

Model	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG (SD)
Original Prompt	0.771	0.746	0.751	0.691	0.732	0.710	0.701	0.701	0.699	0.702	0.650	0.714 (±0.007)
Gold Essay	0.766	0.737	0.757	0.694	0.728	0.692	0.694	0.701	0.686	0.676	0.610	0.704 (±0.015)
Gold Essay GPT-4	0.764	0.741	0.757	0.693	0.727	0.698	0.699	0.701	0.697	0.711	0.644	0.712 (±0.006)

Table 9: Trait-wise prompt replacement experiment results.

Model	P1	P2	P3	P4	P5	P6	P7	P8	AVG (SD)
Original Prompt	0.718	0.726	0.703	0.771	0.727	0.765	0.753	0.683	0.731 (±0.007)
Gold Essay	0.711	0.712	0.707	0.769	0.727	0.765	0.740	0.666	0.725 (±0.012)
Gold Essay GPT-4	0.712	0.722	0.699	0.766	0.723	0.773	0.750	0.675	0.728 (±0.009)

Table 10: Prompt-wise prompt replacement experiment results.

Backbone	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑ (SD↓)
T5-small	0.760	0.733	0.738	0.681	0.719	0.690	0.686	0.696	0.689	0.711	0.640	0.704 (±0.007)
T5-base	0.771	0.746	0.751	0.691	0.732	0.710	0.701	0.701	0.699	0.702	0.650	0.714 (±0.007)
T5-large	0.768	0.752	0.760	0.700	0.731	0.719	0.709	0.702	0.713	0.693	0.641	0.717 (±0.008)
BART-large	0.742	0.733	0.744	0.674	0.713	0.679	0.678	0.687	0.692	0.709	0.634	0.699 (±0.009)
BART-base	0.754	0.711	0.733	0.660	0.697	0.654	0.655	0.666	0.657	0.681	0.611	0.680 (±0.010)
FLAN-T5-small	0.760	0.728	0.742	0.674	0.720	0.679	0.666	0.681	0.670	0.719	0.635	0.698 (±0.009)
FLAN-T5-base	0.761	0.730	0.748	0.689	0.726	0.672	0.678	0.685	0.673	0.695	0.607	0.697 (±0.009)
FLAN-T5-large	0.776	0.745	0.760	0.686	0.721	0.700	0.702	0.689	0.698	0.707	0.645	0.712 (±0.005)

Table 11: Trait-wise QWK performance of MLPAS with various backbone models.

Backbone	P1	P2	P3	P4	P5	P6	P7	P8	AVG↑ (SD↓)
T5-small	0.710	0.702	0.694	0.758	0.725	0.756	0.728	0.669	0.718 (±0.009)
T5-base	0.718	0.726	0.703	0.771	0.727	0.765	0.753	0.683	0.731 (±0.007)
T5-large	0.723	0.731	0.707	0.774	0.726	0.771	0.750	0.693	0.734 (±0.010)
BART-large	0.687	0.694	0.690	0.762	0.703	0.745	0.750	0.671	0.713 (±0.004)
BART-base	0.681	0.665	0.678	0.746	0.715	0.743	0.728	0.635	0.699 (±0.006)
FLAN-T5-small	0.711	0.690	0.689	0.752	0.725	0.757	0.747	0.639	0.714 (±0.008)
FLAN-T5-base	0.704	0.708	0.708	0.768	0.720	0.759	0.747	0.623	0.717 (±0.009)
FLAN-T5-large	0.719	0.711	0.706	0.775	0.722	0.766	0.762	0.675	0.729 (±0.008)

Table 12: Prompt-wise QWK performance of MLPAS with various backbone models

Model	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑(SD↓)
ArTS	0.754	0.730	0.751	0.698	0.725	0.672	0.668	0.679	0.678	0.721	0.570	0.695 (±0.018)
ArTS-rev	0.739	0.724	0.749	0.687	0.718	0.667	0.658	0.660	0.666	0.711	0.562	0.686 (±0.021)
MLPAS	0.771	0.746	0.751	0.691	0.732	0.710	0.701	0.701	0.699	0.702	0.650	0.714 (±0.007)
MLPAS-rev	0.763	0.742	0.754	0.680	0.726	0.701	0.695	0.705	0.698	0.706	0.660	0.712 (±0.012)
ArTS Gap	0.015	0.006	0.002	0.011	0.007	0.005	0.010	0.019	0.012	0.010	0.008	0.009 (-)
MLPAS Gap	0.009	0.004	0.003	0.011	0.006	0.009	0.006	0.004	0.002	0.004	0.010	0.002 (-)

Table 13: Traits-wise results for the trait ordering. "-rev" indicates traits order reversing.

Model	P1	P2	P3	P4	P5	P6	P7	P8	AVG↑ (SD↓)
ArTS	0.696	0.669	0.682	0.732	0.712	0.743	0.712	0.492	0.680 (±0.029)
ArTS-rev	0.700	0.683	0.702	0.763	0.730	0.767	0.734	0.586	0.708 (±0.027)
MLPAS	0.718	0.726	0.703	0.771	0.727	0.765	0.753	0.683	0.731 (±0.007)
MLPAS-rev	0.715	0.717	0.691	0.770	0.724	0.766	0.750	0.679	0.726 (±0.008)
ArTS Gap	0.004	0.014	0.020	0.031	0.018	0.024	0.022	0.094	0.028 (-)
MLPAS Gap	0.003	0.009	0.012	0.001	0.003	0.001	0.003	0.005	0.004 (-)

Table 14: Prompt-based results for the trait ordering. "-rev" indicates traits order reversing.

Model	Prompt Augmentation	Overall	Content	PA	Lang	Nar	Org	Conv	WC	SF	Style	Voice	AVG↑ (SD↓)
MLPAS (T5-small)	origin → origin	0.760	0.733	0.738	0.681	0.719	0.690	0.686	0.696	0.689	0.711	0.640	0.704 (±0.007)
	GPT-3.5-turbo → origin	0.762	0.734	0.740	0.686	0.717	0.690	0.693	0.697	0.693	0.688	0.644	0.704 (±0.009)
	origin → GPT-3.5-turbo	0.751	0.730	0.738	0.674	0.718	0.681	0.675	0.688	0.682	0.657	0.585	0.689 (±0.007)
	GPT-3.5-turbo → GPT-3.5-turbo	0.756	0.731	0.739	0.686	0.713	0.689	0.694	0.697	0.701	0.679	0.649	0.703 (±0.013)

Table 15: Performance comparison on traits with LLM-based prompt augmentation.

Model	Prompt Augmentation	P1	P2	P3	P4	P5	P6	P7	P8	AVG↑ (SD↓)
MLPAS (T5-small)	origin → origin	0.710	0.702	0.694	0.758	0.725	0.756	0.728	0.669	0.718 (±0.009)
	GPT-3.5-turbo → origin	0.715	0.709	0.689	0.754	0.726	0.753	0.737	0.675	0.720 (±0.008)
	origin → GPT-3.5-turbo	0.708	0.711	0.685	0.745	0.720	0.750	0.723	0.646	0.711 (±0.005)
	GPT-3.5-turbo → GPT-3.5-turbo	0.706	0.716	0.686	0.752	0.720	0.753	0.726	0.684	0.718 (±0.013)

Table 16: Performance comparison on prompts with LLM-based prompt augmentation.

B Experiment Prompts for GPT-4

Here are the prompts used in our experiment to evaluate GPT-4's zero-shot essay scoring capability. The prompts included the essay scoring instruction, the essay text, and the specific traits, along with their respective maximum and minimum scores. Below are some examples of the prompts:

> Evaluate the following essay based on the given traits and their score ranges. Provide only numeric scores without any explanation. Format the scores as 'Trait: Score' for each trait.
>
> "Patience, whats the first word that comes to your mind when you hear that word? Waiting? I know that's the main word in my mind. Here is a story when I was very patient. Every kid dreads meap testing it was @DATE1 and we have to be completely silent. It's hard enough to be quite. But it's harder when your next to your friends, and your a girl. We had to do meap writing, writing is the most wrost for me. But when I was done I felt really good and quiet for everyone else. That's the story when I had to be patient."
>
> Overall: 0 to 30
> Content: 0 to 6
> Organization: 0 to 6
> Conventions: 0 to 6
> Style: 0 to 6

> Evaluate the following essay based on the given traits and their score ranges. Provide only numeric scores without any explanation. Format the scores as 'Trait: Score' for each trait.
>
> The author concluded the story in this manner so that the audience would feel sympathy and understanding for the things that Saeng and her family were going through in this hard time. It makes the reader feel sympathy for Saeng because all she can think of to make herself feel better is her home town and how she got taken away.
>
> Overall: 0 to 3
> Content: 0 to 3
> Prompt_adherence: 0 to 3
> Language: 0 to 3
> Narrativity: 0 to 3