TRATES: Trait-Specific Rubric-Assisted Cross-Prompt Essay Scoring

Sohaila Eltanbouly, Salam Albatarni, Tamer Elsayed
Computer Science and Engineering Department, Qatar University, Doha, Qatar
{se1403101, sa1800633, telsayed}@qu.edu.qa

Abstract

Research on holistic Automated Essay Scoring (AES) is long-dated; yet, there is a notable lack of attention for assessing essays according to individual traits. In this work, we propose *TRATES*, a novel trait-specific and rubric-based cross-prompt AES framework that is generic yet specific to the underlying trait. The framework leverages a Large Language Model (LLM) that utilizes the trait grading rubrics to generate trait-specific features (represented by assessment questions), then assesses those features given an essay. The trait-specific features are eventually combined with generic writing-quality and prompt-specific features to train a simple classical regression model that predicts trait scores of essays from an unseen prompt. Experiments show that *TRATES* achieves a new state-of-the-art performance across all traits on a widely-used dataset, with the generated LLM-based features being the most significant.

1 Introduction

Automated Essay Scoring (AES), stemming from Page's early study (Page, 1966), has seen notable progress in addressing writing evaluation. It covers holistic scoring (giving a single score for overall proficiency) and trait scoring (assessing specific writing aspects like organization). While holistic scoring offers a broad assessment of writing ability, trait scoring provides detailed feedback to aid students in targeted skill improvement. Due to the complexity of evaluating different traits, holistic scoring has been the predominant focus in AES research. Various methodologies have been introduced, ranging from approaches relying on handcrafted features (Phandi et al., 2015) to language model-based ones (Xie et al., 2022).

Cross-prompt AES is gaining momentum in recent AES research, with the goal of training a model that can effectively score essays from *unseen* prompts. This approach is not only more practical, reflecting real-world scenarios where models must generalize across diverse prompts, but also more challenging, as it demands the ability to adapt to variations in writing styles, topics, and prompt structures with high accuracy. Different approaches have been proposed for cross-prompt AES, including feature-based approaches (Li and Ng, 2024b), multi-task learning (Ridley et al., 2021), and contrastive learning (Chen and Li, 2024, 2023).

As Large Language Models (LLMs) represent the recent advancements in NLP, there is a growing trend towards the integration of LLMs in AES (Naismith et al., 2023; Do et al., 2024). The common approach involves providing the LLM with task descriptions, rubrics, and essays for scoring. However, this method has not achieved the desired results, falling short of basic baseline performance (Yancey et al., 2023; Mansour et al., 2024). Other studies employed LLMs as conversational models, where scoring is performed on multiple steps (Stahl et al., 2024; Lee et al., 2024). While these approaches improve upon basic LLM prompting, they still lag behind baseline performance. This highlights the need for a hybrid approach integrating the advanced text analysis capabilities of LLMs with the previously well-established AES methods.

In this work, we address the challenge of *trait-based cross-prompt AES* with a novel approach that redefines the role of LLMs. Rather than following the conventional paradigm of "*Given this essay, provide a score.*", we propose a hybrid framework, ***TRATES***,[1] a Trait-specific and Rubric-Assisted Cross-Prompt AES framework, that leverages generic writing quality features in addition to *trait-specific* features, automatically generated via an LLM, that are easy to interpret, thus providing direct feedback to students on sub-trait aspects.

At its core, *TRATES* leverages an LLM to generate and extract trait-specific features given a trait rubric in two stages. Initially, a set of questions is

[1] Spelled differently from 'Traits' with same pronunciation.

Findings of the Association for Computational Linguistics: ACL 2025, pages 20528–20543
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

generated from the rubric to assess a specific trait. The LLM then answers each question individually, given the essay. The intuition is to streamline the rubric to facilitate the assessment of various aspects within the trait, rather than assessing the trait as a whole. Finally, trait-specific features are combined with writing-quality and prompt-specific features to train a cross-prompt classical *regression* model that predicts the trait scores of essays from unseen prompts. This approach allows us to handle the scoring of various traits within a unified framework while automatically tailoring rubric-based features for each specific trait, a task that would be difficult to achieve without the capabilities of LLMs. Our contribution is five-fold:

1. We present *TRATES*, a *novel* rubric-assisted framework for trait-based cross-prompt AES that provides a simple and generic pipeline that can be used to score any trait while generating features specific to the trait. It combines the strengths of powerful LLMs (for feature generation and extraction) with the simplicity of a basic regression model (for scoring).
2. *TRATES* establishes a new state-of-the-art (SOTA) performance on *all* traits on a widely-used dataset.
3. We conduct an ablation study to show the significance of each feature category.
4. We assess the generalizability of *TRATES* over two different datasets.
5. We publicly release all trait-specific features to enable future research.[2]

The remainder of this paper is organized as follows. Section 2 outlines the related work. Section 3 defines the cross-prompt AES problem. Section 4 details our proposed *TRATES* framework. Section 5 discusses our experimental setup. Section 6 presents the results and offers a comprehensive analysis. Finally, Section 7 concludes with few suggested future work directions.

2 Related Work

In this section, we review AES studies, focusing on cross-prompt trait scoring and LLM integration.

Cross-Prompt AES Early cross-prompt studies concentrated on holistic scoring (Jin et al., 2018; Li et al., 2020; Ridley et al., 2020), employing methods ranged from simple neural networks to hybrid models combining neural networks with engineered features. However, holistic scoring falls short in providing detailed feedback. To address this, Ridley et al. (2021) introduced a new task: cross-prompt trait scoring, and developed a POS-embedding-based neural model. Building on this, Do et al. (2023) enhanced the architecture by incorporating prompt-text features, achieving SOTA performance on the ASAP dataset. Several learning methods have been applied to cross-prompt trait scoring, including multi-task learning (Li and Ng, 2024b), contrastive learning (Chen and Li, 2023), and meta-learning (Chen and Li, 2024).

LLM for AES Research on LLMs for AES is expanding but remains limited. Promising results were demonstrated by Do et al. (2024) with fine-tuning T5 for prompt-specific scoring. Aside from fine-tuning, several LLM prompting techniques were applied to AES (Mansour et al., 2024). GPT-4 showed improvement with few-shot examples; however, it failed to outperform a simple XGBoost baseline (Yancey et al., 2023). Similarly, GPT-3.5, used for scoring and feedback generation, was unsuccessful (Han et al., 2024). Moreover, Stahl et al. (2024) explored various LLM prompting strategies for scoring and feedback generation, including impersonation and chain of thought. Recently, Lee et al. (2024) proposed a Multi Trait Specialization framework that engages the LLM in a conversation to score the essay holistically.

Most of the proposed LLM-based approaches above focused on zero-shot holistic essay scoring, raising multiple concerns regarding inconsistent evaluation and hallucinations. In contrast, *TRATES* introduces a novel use of LLMs, serving as feature generators and extractors, rather than direct graders. It then trains a regression model using LLM outputs (besides other features), resulting in more effective scoring while eliminating the need for direct scoring with the LLM, which has been shown to be ineffective. Additionally, while most of the work targets holistic scoring, our framework focuses on trait scoring and ensures a more comprehensive evaluation. Moreover, unlike *TRATES*, none of the related work that utilized LLMs has outperformed or even reached SOTA performance. Finally, this work addresses key gaps in cross-prompt trait scoring; unlike prior approaches, our framework integrates the scoring rubric to extract trait-specific features that are easily interpretable.

[2] https://github.com/Sohaila-se/TRATES

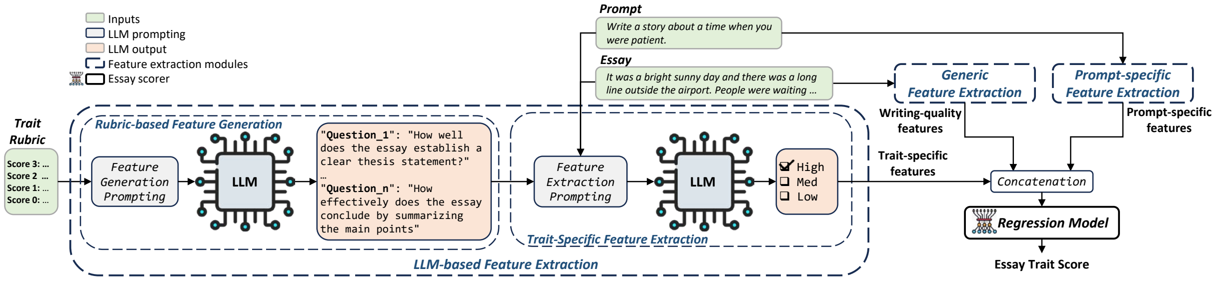
Inputs
LLM prompting
LLM output
Feature extraction modules
Essay scorer
Prompt
Write a story about a time when you were patient.
Essay
It was a bright sunny day and there was a long line outside the airport. People were waiting ...
Trait Rubric
Score 3: ...
Score 2 ...
Score 1: ...
Score 0: ...
Rubric-based Feature Generation
Feature Generation Prompting
LLM
"Question_1": "How well does the essay establish a clear thesis statement?"
…
"Question_n": "How effectively does the essay conclude by summarizing the main points"
Feature Extraction Prompting
LLM
☑ High
❑ Med
❑ Low
Trait-Specific Feature Extraction
LLM-based Feature Extraction
Generic Feature Extraction
Prompt-specific Feature Extraction
Writing-quality features
Prompt-specific features
Trait-specific features
Concatenation
Regression Model
Essay Trait Score

Figure 1: Overview of our *TRATES* framework.

3 Cross-prompt Trait-based AES

In trait-based AES, a prompt p is defined as a tuple (a_p, T_p, E_p), where a_p is the prompt task description (a paragraph or so describing the writing task), T_p is a set $\{(t, r_t)\}$ of traits; each trait t (an aspect of student writing, such as organization or sentence fluency) is associated with a rubric r_t (a set of criteria used to evaluate the specific trait t), and E_p is a set $\{(e, \{s_{e,t}\})\}$ of essays written for the prompt p; each essay e is associated with a score $s_{e,t}$ for each trait $t \in T_p$. While, generally, each prompt has its own trait rubrics, those rubrics are usually common across different prompts for specific traits.

The cross-prompt problem is set up as follows. We aim to build a model that is trained on a set of source prompts P_{src} to score the (traits of) essays written for a target prompt $p_{trg} \notin P_{src}$. We note that only the task description of the target prompt is available to the model at inference time.

4 *TRATES* Framework

In essay scoring, the rubric generally serves as a scoring guide that sets out the criteria for different performance levels. For example, a brief 3-level rubric for the *organization* trait could be:

> ***Score 2:*** *Organization and connections between ideas are logically sequenced.*
> ***Score 1:*** *Organization and connections between ideas are weak.*
> ***Score 0:*** *No organization evident.*

Human evaluators, typically teachers, use the rubric to assess the student essays, which enhances grading consistency and transparency. Moreover, it provides students with detailed feedback pinpointing areas for improvement. Given the rubric's importance in *manual* essay scoring, our primary objective is to develop an *automated* rubric-based scoring framework that offers the same benefits as manual scoring, focusing on assessing essay traits.

The main challenge in automatically scoring essay traits lies in identifying the essay representation that best reflects the characteristics of each trait. To address that, we propose *TRATES*, a Trait-specific and Rubric-Assisted AES framework, illustrated in Figure 1. *TRATES* leverages the rubric to identify *trait-specific* features (represented by automatically-generated questions), then extracts the values of those features (by answering the generated questions) from the essays, using an LLM. Those features combined with prompt-specific and generic features are used to train a (relatively-simple) regression model for scoring essay traits.

The major advantage of *TRATES* is that it is generic enough to be applied to *any* trait, given the rubric used to assess that trait, while being able to generate and extract different features that are more specific to that trait. In this section, we discuss the main components of the framework in detail.

4.1 Rubric-based Feature Generation

The first component aims to automatically generate (or identify) essay features that are trait-specific and rubric-based. To achieve that, we use an LLM to convert a given trait rubric into a set of assessment questions (acting as *sub-traits*) that constitute trait-specific features. This helps assess the various individual aspects associated with each trait, rather than considering all the aspects in one assessment step, allowing for a more fine-grained assessment of the essay traits.

To generate the trait-specific features Q_t for a trait t, an LLM $\mathcal{M}$ is prompted, with LLM prompt instructions I_{gen}, to formulate a set of assessment questions Q_t based on the given trait rubric r_t:

$$Q_t = \mathcal{M}(r_t | I_{gen}) \qquad (1)$$

where $Q_t = \{q_1, q_2, ..., q_n\}$ is the set of n sub-traits generated from r_t. The LLM is prompted to formulate the questions to rate the essay's aspects as *high*, *medium*, or *low*. The rationale is (1) we opt for simple-to-assess questions, (2) we avoid numeric ratings because the generated questions may not always be formulated so that higher ratings imply better quality, and (3) we prioritize clear and easily understandable responses. The LLM-prompt is shown in Figure 2. We note that the same LLM-prompt template is used with *any* LLM and trait, with adjustments made to include the specific essay type (if any), trait name, and rubric (as is).

4.2 Trait-specific Feature Extraction

We next use the same LLM $\mathcal{M}$ to answer each sub-trait question $q_i \in Q_t$, given the LLM prompt instructions I_{ans} (see Appendix A) and essay e.

$$v(e, q_i) = \mathcal{M}(e, q_i | I_{ans}) \tag{2}$$

where $v(q_i) \in \{\text{high}, \text{medium}, \text{low}\}$ is the answer (i.e., feature value) to the question q_i for essay e.

4.3 Prompt-Specific Feature Extraction

Training data for cross-prompt AES typically involves essays of various types, such as persuasive and narrative. The writing requirements for each type can differ significantly; for instance, persuasive essays must be supported by arguments, whereas narrative essays often rely on creative writing. Additionally, essays from different prompts may vary in length depending on the specific writing requirements and the grade level of the students, which can greatly impact the assigned score.

Since different prompts exhibit distinct characteristics, we define a set O of *prompt-specific* features that represents properties of the writing prompts, including *essay type* (e.g., persuasive or narrative), *expected essay length* (aligned with task requirements), *source length* (for tasks that utilize external sources), and the *grade level* of the students. The rationale behind these features is to enhance the model's ability to distinguish between different prompts and to establish a connection between prompts with similar characteristics, indicating to the regression model that these prompts share certain writing characteristics.

4.4 Generic Feature Extraction

While the LLM-based features are trait-specific, there are other features that are *generic* that can help capture different aspects of writing proficiency. We considered five categories of features that were widely utilized in the literature (Ridley et al., 2020, 2021): (i) *length-based* features (G_L), such as the number of words and sentences in the essay, (ii) *readability* (G_D) features, which measure how difficult the essay is to read, (iii) *text variations* (G_T) features covering the usage of different part-of-speech tags (POS) and punctuation, (iv) *text complexity* (G_C) features, which evaluate the structural complexity of essays, and (v) *sentiment* (G_S) features assessing the tone of the essays. This feature set is denoted by $G = \{G_L, G_D, G_T, G_C, G_S\}$. The full feature list G is presented in Appendix B.

4.5 Trait Scoring

At the final stage of the framework, the concatenated list of extracted values of the trait-specific Q_t, prompt-specific O, and writing quality G features from each essay of the set of source prompts P_{src} are used to train a cross-prompt and trait-specific regression model $R(t)$, which is then used to predict the scores of essays from an unseen prompt p_{trg}. Note that the feature sets O and G are common for all traits, while Q_t differs across traits, resulting in a trained model for each trait.

We use regression models instead of classifiers to provide our model with the ability to predict the essay score with granularity similar to real-world scenarios. We opted for a *shallow neural network* to maintain simplicity in the regression model and to account for the expected relatively small amount of training data.

4.6 Addressing Cross-prompt Challenges

The main challenge in cross-prompt AES lies in integrating various writing tasks with different characteristics and requirements into a single scoring model; differences between those tasks can manifest at various levels, such as variations in essay types that have distinct writing requirements, different scoring rubrics with unique criteria and score ranges, and the varying quality of writing expected across grade levels. To ensure the model's generalizability to unseen prompts, it must account for potential differences that may arise during inference. In this section, we discuss how *TRATES* attempts to address these challenges.

Different scoring rubrics Having different essay types implies having different rubrics. This imposes a challenge because what qualifies as a strong essay under one rubric may differ significantly from another. Given that our methodology relies on the scoring rubric, we utilized the rubrics from all source prompts for the LLM-based feature extraction, assuming that their diversity would cover the scoring criteria of different essay types.

Different score ranges One challenge with using prompts with different rubrics is that each may have a different scoring range. This poses a challenge when training a regression model, as it requires scores to be standardized within a unified range. A direct scaling method is to apply min-max scaling within each prompt, ensuring that all score ranges are mapped to a common scale. However, this ignores grade level differences, where the quality required for a maximum score varies. For example, an 8^{th}-grade essay might earn a maximum score for clear ideas and basic structure, while a 12^{th}-grade essay requires sophisticated analysis and advanced writing for the same score. A min-max scaling would give the same unified score for both. To address this issue, we propose a score-scaling method based on incremental adjustments relative to the highest grade level; scores for a given grade level are scaled within a fixed range, where the minimum score remains constant while the maximum score decreases by one level for each grade below the highest. This ensures the scaled scores are accurately comparable across prompts with different grade levels and rubrics.

5 Experimental Setup

In this section, we outline the setup used to conduct our experiments, including the datasets, selected LLMs, baselines, and implementation details.

Datasets In our main experiments, we used the Automated Student's Assessment Prize (ASAP)[3] and ASAP++ (Mathias and Bhattacharyya, 2018) datasets combined, which are widely used for AES evaluation. ASAP has 8 prompts (P1-P8) with trait annotations for P7 and P8 only; ASAP++ extends it by scoring traits for P1-P6. Table 6 describes the dataset. The traits are: Content (CNT), Organization (ORG), Word Choice (WC), Sentence Fluency (SF), Conventions (CNV), Prompt Adherence (PA), Language (LNG), and Narrativity (NAR).

To test the generalizability of *TRATES*, we conducted additional experiments over the ELLIPSE dataset (Crossley et al., 2023) (Table 7), which comprises about 6.5k essays written by English Language Learners for 44 prompts. Each essay is assessed over 6 traits (cohesion (COH), syntax (SYN), vocabulary (VOC), phraseology (PHR), grammar (GRM), and conventions (CNV)) using a standardized rubric with a scoring range [1-5] and increments of 0.5 points.

LLMs Selection The LLMs we experimented with are chosen based on 4 criteria: (1) we opt for open-source models for accessibility, reproducibility, and cost-effectiveness, (2) they are based on different foundation models, (3) considering efficiency and resource constraints, we limit our selection to smaller-scale LLMs of size ranging from 7B to 9B parameters, and finally, (4) we consider the highest-ranked models on the Arena Elo benchmark (at the time of experiments) that match the criteria above.[4] For each LLM, we used its corresponding checkpoint available on Hugging Face.

Accordingly, we selected three LLMs: (i) **Starling**-LM-7B-beta (Zhu et al., 2023); (ii) **Llama**-3.1-8B-Instruct (Touvron et al., 2023); and (iii) **Gemma**-2-9b-it-SimPO (Meng et al., 2024). More details are provided in Appendix E.

Baselines To evaluate *TRATES*, we compare it with 3 baselines. The first two are considered the *SOTA* for cross-prompt AES on the ASAP dataset.

- **ProTACT** (Do et al., 2023) a cross-prompt model that leverages prompt attention, topic-coherence features, and trait-similarity loss.
- **Li & Ng** (Li and Ng, 2024b) adopted a simple neural architecture with different sets of features, developing feature-based models.
- Zero-shot LLM (**LLM-D**) utilizes an LLM to *directly* scores essays with zero-shot prompt. The details are presented in Appendix F.

For evaluation, we use Quadratic Weighted Kappa (QWK) (Cohen, 1968), a common measure for AES that assesses the agreement between the scores of two raters.

Training and Hyperparameter tuning For ASAP, we used leave-one-prompt-out cross-validation. The hyperparameters of the models are tuned using sequential hyperparameter tuning, where one hyperparameter is optimized at a time while keeping others fixed. The hyperparameters and their search space are listed in Table 8. We used early stopping with a patience of 10 epochs, and ReduceLROnPlateau learning-rate scheduler with a patience of 5 epochs and a factor of 0.1. The batch size was fixed to 128 for all the experiments. LLM's answers (high/medium/low) were mapped

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

[4] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

to numerical values (3/2/1). We report the performance of our models on the *unseen target prompts* for each trait and on average. The performance of ProTACT and Li & Ng baselines is reported as in their corresponding studies for ASAP dataset. For ELLIPSE dataset, we trained the ProTACT model using their implementation.[5]

Feature Generation Trait-specific features are generated from each unique rubric. Tables 9 and 10 show the feature count and examples for ASAP, while Table 11 presents the same for ELLIPSE.

Prompt-specific and Generic Features The prompt-specific features are extracted from the metadata provided with the dataset. The generic features, described in Table 5, are extracted using (Ridley et al., 2021) code.[6]

Feature Normalization Previous studies typically normalize features within individual prompts, assuming that the feature distribution of the unseen prompt is known (Do et al., 2023; Li and Ng, 2024b). However, this assumption does not align with real-world scenarios. Instead, we aim to develop a generalizable model that can handle unseen prompts of any size. To achieve this, we use the minimum and maximum values from the *training* dataset for normalization and consistently apply those values to the test data during inference.

Score-scaling Due to inconsistency in score ranges of ASAP prompts, we scaled all scores to [0-6]. We also implemented score scaling based on grade levels (Section 4.6), ensuring the maximum score for lower grade levels is appropriately reduced. More details are provided in Appendix I. The predicted scores are scaled back to their original range for evaluation to ensure a fair comparison with previous work.

6 Results and Discussion

In this section, we discuss the results of our experiments addressing 4 research questions: **RQ1**: Are features generated and extracted via LLMs effective? (6.1), **RQ2**: Can *TRATES* with trait-specific and generic features improve the performance? (6.2), **RQ3**: Which feature category holds greater significance? (6.3), and **RQ4**: How well does *TRATES* generalize? (6.4). Furthermore, we discuss the inference cost in Section 6.5.

6.1 Effectiveness of LLM Features (RQ1)

We first examine the effectiveness of models trained only on the LLM-based features (LLM-F) without the other feature categories. The LLM-F models have the same architecture and training setup as *TRATES*. Table 1 (rows f-h) presents their QWK performance. The results indicate that the extracted features demonstrate strong predictive capabilities across different LLMs. Notably, the LLM-F models clearly outperform the LLM direct scoring (LLM-D) (Table 1, rows c-e) by an average of 9 points. This highlights that LLMs struggle with zero-shot essay scoring, which is consistent with previous findings (Mansour et al., 2024).

Among the three LLMs, Llama exhibits the lowest performance, followed by Starling, while Gemma excels. Gemma generates the fewest questions, averaging 8.6 features per trait. This suggests that its generated features are concise and better aligned with the actual criteria in the rubrics compared to the other LLMs. In contrast, Llama generates an average of 20 features per trait; yet, its performance lags behind the others, indicating that some of the generated features introduce noise and do not align well with the scoring criteria.

Remarkably, the ORG and CNV traits were the most difficult to score. In ASAP, these traits include prompts with different types and grade levels, making it more challenging to capture the characteristics of different prompts. Conversely, the CNT and PA traits achieved the highest QWK scores. Content-based traits are known to be difficult to predict, as they cannot be assessed heuristically, and the model must understand the prompt and the content of the essay in order to score it effectively (Li and Ng, 2024a). Nevertheless, the features derived from LLMs alone were able to achieve a reasonable QWK, underscoring the predictive capabilities inherent in these features.

Although LLM-F models exhibit significantly lower performance than the baselines, the extracted features show good predictive capability for all traits (reflected in *positive* QWK values). However, the results indicate that the trait-specific features are *not sufficient* for effective essay trait scoring, suggesting the necessity of integrating other features for a comprehensive scoring framework.

6.2 Effectiveness of *TRATES* (RQ2)

Table 1 (rows i-k) shows the performance of *TRATES* (combining all types of features) with the

[5] https://github.com/doheejin/ProTACT

[6] https://github.com/robert1ridley/cross-prompt-trait-scoring

	Model	LLM	ORG	WC	SF	PA	NAR	LNG	CNV	CNT	Avg
a	ProTACT	-	0.518	0.599	0.585	0.619	0.639	0.596	0.450	0.596	0.575
b	Li & Ng	-	0.478	0.459	0.452	0.617	0.637	0.556	0.439	0.592	0.529
c	LLM-D	Starling	0.281	0.318	0.290	0.322	0.289	0.221	0.157	0.282	0.270
d		Llama	0.323	0.183	0.209	0.487	0.467	0.484	0.168	0.332	0.332
e		Gemma	0.345	0.375	0.390	0.337	0.382	0.337	0.263	0.326	0.344
f	LLM-F	Starling	0.345	0.355	0.387	0.520	0.471	0.396	0.306	0.457	0.405
g		Llama	0.345	0.227	0.322	0.426	0.428	0.316	0.301	0.438	0.350
h		Gemma	0.329	0.546	0.456	0.533	0.525	0.412	0.429	0.546	0.472
i	*TRATES*	Starling	0.518	0.593	**0.612**•	**0.624**•	**0.668**•	**0.608**•	0.501•	**0.636**•	**0.595**•
j		Llama	0.522•	0.579	0.587•	0.572	0.600	0.541	0.497•	0.631•	0.566
k		Gemma	**0.547**•	**0.622**•	**0.612**•	0.599	0.600	0.521	**0.556**•	0.632•	0.586•

Table 1: QWK Performance of *TRATES* with the different LLMs compared to the baselines. **Bold** values indicate the best performance per trait, while values with • outperform the SOTA.

three LLMs. Notably, *TRATES* outperforms the SOTA models on average over all traits using Starling and Gemma by 2 and 1 points, respectively, while it trails by 1 point with Llama. In terms of individual traits, Starling outperforms SOTA in 6 out of 8 traits (with on-par performance for the remaining two) , Gemma in 5, and Llama in 4. More importantly, *TRATES* establishes new SOTA performance for all traits on the ASAP dataset.

Interestingly, while Gemma demonstrated the best performance with LLM-F, it did not achieve the same with *TRATES*. This suggests its extracted features overlap with other features, leading to less improvement when combined. In contrast, Starling features perform better when integrated with other features, suggesting their extracted features are more unique and complementary.

We note that ORG, CNV, and CNT traits showed improvements with the three LLMs, with an average of 1.1, 6.8, and 3.7 points, respectively. Besides the positive impact of trait-specific features, we believe that the added prompt-specific features play a significant role in this improvement. These features are particularly applicable to those traits because the prompts within those traits exhibit the greatest diversity in terms of grade levels and essay types.

In contrast, PA, NAR, and LNG traits showed less improvements, with only Starling outperforming the SOTA. It is important to note that these traits already had good QWK scores with the feature-based baseline (Li & Ng), indicating the effectiveness of the generic features. Although *TRATES* did not outperform on those traits with Llama and Gemma, *TRATES* with Starling showed improvements of 0.5, 3, and 1.20 points, respectively. This emphasizes the significant influence of the choice of LLM on the trait scoring performance.

6.3 Feature Category Ablation Study (RQ3)

In this section, we address an important question regarding the significance of feature categories within our framework. This analysis helps in understanding whether each category is contributing positively to the prediction of trait scores. We conducted this experiment for *TRATES* with Starling. Table 2 presents the *drop* in QWK when the regression model is trained with all feature categories but one.

The trait-specific features stand out as the most significant category across all traits except organization. Notably, these are the only features that vary between traits, underscoring their critical role. This highlights the importance of incorporating *trait-specific* features to effectively capture the unique requirements of each trait. It is also important to note that these are the only features that are generated automatically, whereas all other features require manual feature engineering.

The prompt-specific category comes next by positively contributing to 6 traits. For WC and SF, the negative impact can be attributed to the limited diversity of prompts within those traits. Within the generic features, the length-based category positively contributed to 6 traits; it contains a diverse range of features that pertain to various essay aspects, such as words and sentences, which may explain the negative values observed in some traits, as not all features in the category are equally relevant to every trait. Additionally, readability features had the greatest impact on LNG trait, which is expected as it emphasizes grammar and spelling accuracy.

Category	Size	ORG	WC	SF	PA	NAR	LNG	CNV	CNT	Avg
Trait-specific	18.2	2.23	**3.73**	**2.00**	**10.22**	**12.27**	**13.28**	**8.73**	**8.35**	**7.60**
Prompt-specific	4	4.57	-3.35	-1.69	4.76	8.94	3.28	3.34	5.28	3.14
Length-based	16	3.39	1.53	-0.33	-0.70	4.57	2.94	3.49	3.42	2.29
Readability	12	0.97	-1.43	-2.63	2.26	1.49	7.39	2.83	2.58	1.68
Text complexity	5	1.17	-1.09	-0.20	2.19	5.59	5.82	-1.62	2.47	1.79
Text variations	43	**7.27**	-0.70	-0.95	1.06	3.59	-0.44	3.46	0.10	1.67
Sentiment	5	2.01	-1.75	-0.58	0.64	3.45	4.71	1.01	0.23	1.22

Table 2: **Drop** in QWK performance of *TRATES* with Starling on ASAP when excluding one feature category. **Bold** and underlined values indicate the most and second-most important categories for each trait, respectively.

The results indicate that all feature categories contribute positively on average. This also supports our proposed approach, where instead of relying exclusively on an LLM-based system, we leverage the classic feature categories. Moreover, the study clearly underscores the necessity for developing trait-specific AES systems, as the significance of features varies across different traits, demonstrating that not all features are relevant for every trait except for our trait-specific features.

6.4 Evaluation on ELLIPSE Dataset (RQ4)

Establishing a new SOTA performance on ASAP has pushed us to check whether *TRATES* can exhibit similar performance on another dataset. We chose ELLIPSE because it contains several (44) prompts, making it well-suited for cross-prompt setup. Additionally, being a dataset for English learners allows us to test the applicability of *TRATES* across different types of learners. It is worth mentioning that this is the *first-ever* cross-prompt AES study on that dataset.

To conduct this study, all the steps of *TRATES* are repeated with Starling (the best-performing LLM on ASAP). We adopted an 11-fold cross-validation approach, where each fold has 4 prompts, setting 1 unseen fold for testing.[7] The results are presented in Table 3. We compare *TRATES* with 4 baselines: ProTACT′ (a variant of ProTACT without the topic coherence (TC) features, as their extraction was not included in the provided implementation), LLM-D, LLM-F, and a feature-based model that is trained on the generic and prompt-specific features (GP-F).

We first note that *TRATES* clearly outperforms all the baselines in all traits, with a margin of at least 6.5 points on average. However, the performance of all models on ELLIPSE is lower than that on ASAP. This disparity can be attributed to several factors. First, the features utilized in ProTACT′ and GP-F were primarily developed on and tested for ASAP, raising concerns about their generalizability to other datasets, particularly for scoring essays written by English learners, whose writing styles and errors often differ from those of native speakers. Nonetheless, GP-F outperforms LLM-F by 7 points, likely due to the number of features; GP-F is trained on 89 features, whereas the average number of features for LLM-F from ELLIPSE is only 7. Additionally, the rubrics in ELLIPSE are more concise than those for ASAP, resulting in fewer generated features. The features generated from ASAP rubrics range from 5 to 13, while those from ELLIPSE rubrics range from 4 to 10.

Finally, we recall that the results reflect the average score across 44 prompts (compared to 8 in ASAP), while QWK for individual prompts ranges from 0.25 to 0.87. It is also worth noting that assessing essays within this dataset is inherently difficult, as indicated by an inter-rater agreement between the human annotators, yielding a kappa value of less than 0.6 (Crossley et al., 2023), which is not far from the performance of *TRATES*.

6.5 Inference Cost

In addition to *TRATES* predictive performance, the practical deployment of *TRATES* depends on its inference efficiency. We evaluate the inference cost of *TRATES* in terms of processing time, focusing on the feature extraction stages and essay scoring.

Table 4 presents the average inference time per essay for each trait with the Starling LLM. For the ORG, WC, SF, CNV, and CNT traits, we report the average inference time over the 1,783 essays of prompt P1. For PA, NAR, and LNG traits, the reported times are averaged over the 1,726 essays of prompt P3. All timings were obtained on an Azure VM equipped with an NVIDIA A10 GPU and an

[7] `https://github.com/Sohaila-se/TRATES`

Model	COH	SYN	VOC	GRM	CNV	PHR	Avg
ProTACT′	0.33	0.35	0.42	0.29	0.36	0.36	0.35
GP-F	0.45	0.49	0.48	0.40	0.50	0.46	0.46
LLM-D	0.24	0.29	0.17	0.28	0.23	0.20	0.24
LLM-F	0.35	0.38	0.38	0.45	0.44	0.34	0.39
TRATES	**0.52**	**0.54**	**0.52**	**0.51**	**0.56**	**0.53**	**0.53**

Table 3: QWK performance of *TRATES* with Starling on ELLIPSE. **Bold** values are the best per trait.

	ORG	WC	SF	PA	NAR	LNG	CNV	CNT
Num. of Trait-specific Features	25	9	13	18	20	17	15	31
Essay Average Length	350	350	350	100	100	100	350	350
Trait-specific FE Time (msec)	5,655	2,017	2,933	2,252	2,477	2,084	3,379	7,004
Generic + Prompt FE Time (msec)	139	139	139	43	43	43	139	139
Regression Model Time (msec)	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
Total Time (msec)	5794	2156	3072	2295	2520	2127	3518	7143

Table 4: Average inference time of *TRATES*, using the Starling LLM, per essay of prompt P1 for the ORG, WC, SF, CNV, and CNT traits, and P3 for the other traits. 'FE' denotes Feature Extraction stages. Time is in milliseconds.

AMD EPYC 74F3 24-Core Processor. The model was loaded using the Hugging Face Transformers library with FP16 precision.

As expected, the majority of the inference cost is attributed to LLM-based feature extraction, with times ranging from 2.02 to 7 seconds, depending on the number of features per trait. Traits from P3 show lower inference times compared to those from P1, primarily due to the shorter essay length, which reduces the LLM prompt size and speeds up processing. In contrast, the time for extracting the generic and prompt-specific features is 0.04 seconds for P3 and 0.14 seconds for P1, whereas the final scoring step using the neural network regression model is extremely fast and consistent across all traits, taking only 0.12 milliseconds. These results highlight that the LLM component highly dominates the overall inference cost.

These findings demonstrate the feasibility of deploying *TRATES* in applications with moderate-latency requirements. Although latency remains a limitation and warrants further research, AES is not considered a real-time task and can tolerate the current processing delays.

7 Conclusion and Future Work

In this paper, we introduced *TRATES*, a cross-prompt rubric-based framework utilizing LLMs, traditional features, and classical regression models for scoring essay traits. The framework is designed to reconceptualize the use of LLMs as feature generators and extractors. The results demonstrate the effectiveness of *TRATES* yielding new SOTA performance on ASAP dataset and setting the first baseline for ELLIPSE dataset. Our findings underscore the positive impact of the added LLM-based trait-specific features on enhancing the accuracy of trait scoring, emphasizing the necessity for novel models tailored to individual traits.

In future work, we plan to extend the framework to holistic scoring and explore different prompting techniques for LLM-based trait-specific feature generation. Moreover, studying the usefulness of the generated sub-traits as feedback to the students remains an important direction for future work.

Acknowledgment

This work was supported by NPRP grant# NPRP14S-0402-210127 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

8 Limitations

While *TRATES* is advancing the use of LLMs for AES and extending their application beyond conversational tasks, some limitations are present in this study. Firstly, the study focused solely on relatively small LLMs, raising questions about whether larger ones would perform better in generating trait-specific questions and features.

Furthermore, the performance of *TRATES* remains unexplored in the context of holistic scoring, as holistic rubrics are often highly prompt-specific.

Moreover, our score-scaling approach was designed to address the discrepancies between the rubrics present in the ASAP dataset. The assigned scaling ranges were intuitively determined after a thorough examination of the various rubrics. It is debatable whether essays graded using different rubrics can be effectively combined to develop an AES system; the criteria for assessing essay quality may not be applicable across various rubrics, leading to inconsistencies in scores assigned to essays written for different prompts. Therefore, the research community should collaboratively establish a standardized mapping approach to align the rubrics of existing datasets within a unified framework. This would facilitate the creation of more realistic AES systems that can leverage diverse datasets for training and testing, addressing a wide range of writing proficiencies and real scenarios.

Another limitation of this study is that the effectiveness of the generated questions as feedback for students was not assessed. However, the conceptual framework was discussed with experts in psychometrics and education fields, and the potential value of such feedback was acknowledged. We are dedicated to making these questions available to the community for future research. Moreover, the efficiency remains a limitation of our proposed framework and is an important area for future improvement.

Finally, the effectiveness of this work heavily relies on the quality of the grading rubric.

References

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.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Scott Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Youngmeen Kim, Wesley Morris, Meg Benner, Aigner Picou, and Ulrich Boser. 2023. The english language learner insight, proficiency and skills evaluation (ellipse) corpus. *International Journal of Learner Corpus Research*, 9(2):248–269.

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.

Jieun Han, Haneul Yoo, Junho Myung, Minsun Kim, Hyunseung Lim, Yoonsu Kim, Tak Yeon Lee, Hwajung Hong, Juho Kim, So-Yeon Ahn, and Alice Oh. 2024. LLM-as-a-tutor in EFL writing education: Focusing on evaluation of student-LLM interaction. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 284–293, Miami, Florida, USA. Association for Computational Linguistics.

Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097, Melbourne, Australia. Association for Computational Linguistics.

Zixuan Ke and Vincent Ng. 2019. Automated Essay Scoring: A Survey of the State of the Art. In *IJCAI*, volume 19, pages 6300–6308.

Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing large language models' proficiency in zero-shot essay scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198, Miami, Florida, USA. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024a. Automated essay scoring: A reflection on the state of the art. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024b. Conundrums in cross-prompt automated essay scoring: Making sense of the state of the art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7661–7681, Bangkok, Thailand. Association for Computational Linguistics.

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

Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. Can large language models automatically score proficiency of written essays? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.

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)*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.

Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Peter Phandi, Kian Ming A Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 431–439.

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

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

Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. Exploring LLM prompting strategies for joint essay scoring and feedback generation. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

A *TRATES* LLM Prompts

Our framework uses two LLM prompts: one for generating assessment questions based on the rubric (Figure 2) and another for extracting feature values by answering questions about the input essay (Figure 3).

Your task is to formulate a set of assessment questions from the given rubric to be used to evaluate the <*trait*> of essays written by <*grade-level range*> grade students.
Here are some instructions to follow:
- Formulate the questions to rate the essay's aspects as High/Medium/Low
- The questions should start with "How would you rate ...".
- Keep the questions short and concise.
- Each question should address only one scoring criterion from the rubric.
- Structure your response in a numbered list from 1 to n, as follows:
1- <question 1?>
n- <question n?>
—
Rubric: <*rubric*>
Questions:

Figure 2: LLM prompt for questions (i.e., features) generation.

Category	Feature	Description
Length-based	mean_word	Average word length.
	word_var	Variance of word length.
	mean_sent	Average sentence length.
	sent_var	Variance of sentence length.
	ess_char_len	Total number of characters in the essay.
	word_count	Total word count in the essay.
	prep_comma	The number of prepositions and commas in the essay.
	characters_per_word	Average number of characters per word.
	syll_per_word	Average syllables per word.
	type_token_ratio	The ratio of unique words to the total number of words.
	syllables	Total syllable count.
	wordtypes	Total number of unique word types.
	sentences	Total number of sentences.
	long_words	The total number of words with characters ≥ 7.
	complex_words	The total number of complex words with syllable count ≥ 3.
	complex_words_dc	Complex words based on Dale-Chall readability formula.
Readability	spelling_err	Count of spelling errors.
	automated_readability	Calculated based on the average characters per word and average words per sentence.
	linsear_write	Linsear Write readability score.
	Kincaid	Measures the grade level based on average words per sentence and average syllables per word.
	Coleman-Liau	Measures readability using average characters per 100 words and average sentences per 100 words.
	FleschReadingEase	Estimates reading ease based on average syllables per word and average words per sentence.
	GunninGP-FgIndex	Measures readability by analyzing average sentence length and the percentage of complex words.
	LIX	A readability measure based on sentence length and the number of long words.
	SMOGIndex	Index estimates the years of education needed to understand a text.
	RIX	Calculates text difficulty based on the number of long words and the number of sentences.
	DaleChallIndex	Uses a list of familiar words to calculate readability.
Text Complexity	clause_per_s	Average number of clauses per sentence.
	mean_clause_l	Average clause length.
	max_clause_in_s	Maximum number of clauses in a sentence.
	sent_ave_depth	Average depth of sentence syntactic trees.
	ave_leaf_depth	Average depth of leaf nodes in syntactic trees.
Text Variations	unique_word	Count of unique words.
	stop_prop	Proportion of stop words.
	",", "."	Proportion of commas, and periods.
	VBG, VBZ, VBP, VB, VBD, VBN	Proportion of the different forms of verbs.
	NN, NNP, NNS	Proportion of the different forms of nouns.
	JJ, JJS, RBR, JJR	Proportion of the different forms of adjectives.
	RB, WRB	Proportion of the Adverbs and Wh-adverbs.

Category	**Feature**	**Description**
	PRP, WP, PRP$	Proportion of the different forms of pronouns.
	IN, MD, RP, CC, TO, WDT	Proportion of Other grammatical categories.
	DT, CD	Proportion of the numerals and determiners.
	POS	Proportion of genitive marker.
	tobeverb	Count of "to be" verbs.
	auxverb	Count of auxiliary verbs.
	conjunction	Count of conjunctions.
	pronoun	Count of pronouns.
	preposition	Count of prepositions.
	nominalization	Count of nominalized forms (e.g., "decision" from "decide").
	begin_w_pronoun	The number of sentences that begin with a pronoun.
	begin_w_interrogative	The number of sentences that begin with an interrogative word.
	begin_w_article	The number of sentences that begin with an article.
	begin_w_subordination	The number of sentences that begin with a subordinating conjunction.
	begin_w_conjunction	The number of sentences that begin with a coordinating conjunction.
	begin_w_preposition	The number of sentences that begin with a preposition.
Sentiment	positive_sentence_prop	Proportion of positive sentences.
	negative_sentence_prop	Proportion of negative sentences.
	neutral_sentence_prop	Proportion of neutral sentences.
	overall_positivity_score	Overall positivity score based on sentiment analysis.
	overall_negativity_score	Overall negativity score based on sentiment analysis.

Table 5: The list of the generic writing-quality features for the five categories.

Prompt	Scores	Ave Length	Essays	CNT	ORG	WC	SF	CNV	PA	LNG	NAR
P1	1 - 6	350	1783	✓	✓	✓	✓	✓			
P2	1 - 6	350	1800	✓	✓	✓	✓	✓			
P3	0 - 3	100	1726	✓					✓	✓	✓
P4	0 - 3	100	1772	✓					✓	✓	✓
P5	0 - 4	125	1805	✓					✓	✓	✓
P6	0 - 4	150	1800	✓					✓	✓	✓
P7	0 - 3	300	1569	✓	✓			✓			
P8	1 - 6	600	723	✓	✓	✓	✓	✓			

Table 6: A description of the ASAP and ASAP++ Datasets: Scores, Average essay length in terms of words, and Traits.

You will be given a <*essay_type*> essay written in response to the given prompt by a student in <*grade_level*>th grade. Your task is to answer an assessment question with high/medium/low to evaluate the <*trait*> of the essay.

—

Follow the following format.
Prompt: the topic to which the essay responds.
Essay: the essay you need to evaluate.
Assessment Question: the question you need to answer about the essay.
Answer (High, Medium, or Low): your answer to the question.

—

Prompt: <*task_prompt*>
Essay: <*essay_text*>
Evaluation Question: <*question*>
Answer (High, Medium, or Low):

Figure 3: LLM prompt for feature extraction.

Parameter	Value
#Essays	6,482
Grade levels	8–12
Number of prompts	44
Average prompt size	147 essays
Average essay length	427 words
Score range	1–5 (0.5 increments)

Table 7: ELLIPSE dataset statistics

B Generic Writing-Quality Features

We used five categories of generic writing-quality features to cover different quality dimensions of the essay when scoring the different traits. The features categories are:

1. *Length-based* features, such as the number of words and sentences in the essay; these features are considered the most intuitive indicators for writing quality, which have been used extensively over the years (Mathias and Bhattacharyya, 2018; Chen and Li, 2023),
2. *Readability* features, which measure how difficult the essay is to read. This category includes readability scores, such as the automated readability index and the Flesch–Kincaid test (Ke and Ng, 2019),
3. *Text variations* features covering the usage of part-of-speech tags (POS) and punctuation.
4. *Text complexity* features, which evaluate the structural complexity of essays by analyzing the number of clauses per sentence and the sentence depths.
5. *Sentiment* features assessing the tone of the essays, capturing the proportion of positive, negative, and neutral sentences.

Table 5 presents all the considered features in the five categories with their description.

C ASAP Dataset Statistics

The ASAP++ comprises 12,978 essays written in English in response to 8 prompts. Each of these prompts has a different number of responses and a different score range. There are a total of 10 different traits; each is covered in a subset of the prompts, except the content trait which is covered in all prompts. It's important to note that the traits of voice and style are only addressed in one prompt, making them unsuitable for cross-prompt experiments. Table 6 presents the statistics and the details of the dataset.

D ELLIPSE Dataset Statistics

The ELLIPSE dataset comprises 6,482 essays written in response to 44 different prompts. These essays are written by English Language Learners across grade levels 8 through 12. All essays are evaluated using a standardized rubric that assesses 6 key traits: cohesion, syntax, vocabulary, phraseology, grammar, and conventions, in addition to a holistic score. Table 7 presents the statistics and details of this dataset.

E LLMs Selection

We selected three LLMs for our experiments:

1. **Starling**-LM-7B-beta (Zhu et al., 2023): a fine-tuned model based on Mistral-7B and Openchat-3.5 via Reinforcement Learning from AI Feedback. It uses the Starling-RM-34B reward model with PPO for policy optimization. It leverages the Nectar ranking dataset and an improved training pipeline.[8]
2. **Llama**-3.1-8B-Instruct (Touvron et al., 2023): a fine-tuned model based on Llama-3 via supervised fine-tuning and Reinforcement Learning with Human Feedback. It is optimized for multilingual dialogue tasks and outperforms many open-source and closed models on industry benchmarks.[9]
3. **Gemma**-2-9b-it-SimPO (Meng et al., 2024) a fine-tuned model based on gemma-2-9b-it

[8] https://huggingface.co/Nexusflow/Starling-LM-7B-beta

[9] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

#	Hyperparameter	Default value	Possible values
1	Loss function	MSE	MSE, Weighted-MSE
2	Learning rate	0.001	0.01,0.001,0.0001
3	Hidden layers	1	1, 2, 3
3	Neurons per layer	32	16, 32,
4	Activation	ReLU	ReLU, SELU, LeakyReLU, Tanh, ELU
5	L2 regularization	0	0, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 0.1
6	Dropout	0	0.0, 0.1, 0.2, 0.3, 0.4, 0.5

Table 8: The hyperparameters search space. Parameters are listed in the order of tuning, along with their default values. The number of hidden layers and their sizes are tuned together.

	ORG	WC	SF	PA	NAR	LNG	CNV	CNT	Avg
Starling	25	9	13	18	20	17	15	31	18.5
Llama	30	20	10	16	16	18	20	32	20.3
Gemma	14	8	5	6	6	6	11	13	8.63

Table 9: The number of LLM-generated questions from all the unique rubrics of each trait for the ASAP dataset.

Trait	Prompts	Example
ORG	P1, P2	Does the essay have a clear introduction, body, and conclusion?
	P7	How well does the essay establish a clear thesis statement or central idea?
	P8	How would you rate the essay's transitions among sentences, paragraphs, and ideas in terms of their smoothness and effectiveness?
WC	P1, P2, P8	How would you rate the attempts at colorful language in the essay? Are there any instances where the language seems overdone or forced?
SF	P1, P2, P8	How well does the essay demonstrate variety and control in its sentence structure, length, and beginnings?
PA	P3, P4	How would you rate the essay's adherence to the prompt's topic?
	P5, P6	How would you rate the clarity of the essay's main points and arguments?
NAR	P3, P4	How would you rate the essay's overall interest and engagement for the reader?
	P5, P6	How would you rate the essay's overall interest and engagement?
LNG	P3, P4	How would you rate the essay's vocabulary range and usage?
	P5, P6	How would you rate the essay's grammar and spelling?
CNV	P1, P2, P8	How would you rate the essay's capitalization? Are there any significant errors or inconsistencies?
	P7	How would you rate the essay's spelling accuracy, considering the grade level?
CNT	P1, P2, P8	How well do the main ideas stand out in the essay?
	P3, P4	How well does the essay maintain focus on the topic and avoid digressing?
	P5, P6	How would you rate the essay's language and style?
	P7	How would you rate the essay's language and tone?

Table 10: Examples of the assessment questions generated by Starling LLM for each unique rubric in the ASAP dataset. Each row represents a rubric in the dataset with the prompts associated with this rubric.

Trait	#Questions	Example
Cohesion	5	How would you rate the essay's overall cohesion and organization?
Syntax	4	How would you rate the essay's use of punctuation and capitalization?
Vocabulary	6	How would you rate the essay's control of word choice and word forms?
Grammar	10	How would you rate the essay's use of punctuation?
Conventions	10	How would you rate the essay's spelling accuracy?
Phraseology	5	How would you rate the essay's avoidance of noticeable repetitions and misuses of phrases?

Table 11: Examples of the assessment questions generated by Starling LLM for each trait in the ELLIPSE dataset.

via Simple Preference Optimization (SimPO). SimPO is an offline method for enhancing LLM training with preference optimization datasets.[10]

F Zero-shot LLM Scoring

You will be given a <*essay_type*> essay written in response to the given prompt. Your task is to score the <*trait*> of the essay as per the given rubric.

—

Follow the following format.
Prompt: the topic to which the essay responds.
Rubric: the grading rubric to score the essay.
Essay: the essay you need to evaluate.
Score: the score of the essay as per the given rubric (only one number).

—

Prompt: <*task_prompt*>
Rubric: <*trait_rubric*>
Essay: <*essay_text*>
Score:

Figure 4: LLM prompt for Zero-shot LLM direct scoring (LLM-D).

The primary purpose of including this baseline is to demonstrate the performance of LLMs in AES, highlighting that direct LLMs utilization lags significantly behind other established methods in the field. The results of this baseline underscore the necessity for alternative ways to integrate LLMs in AES, rather than relying on them solely for scoring purposes. Figure 4 illustrates the LLM prompt used for zero-shot essay scoring.

G Hyperparameters

The hyperparameters are tuned using sequential tuning with QWK as the scoring function. For each fold, the best hyperparameters on the validation set are used to train the regression model and evaluate its performance on the test set. The hyperparameters search space is presented in Table 8.

H Rubric-based Assessment Questions

Table 9 shows the total number of generated questions by each LLM from the rubrics of each trait. Additionally, Table 10 presents examples of the assessment questions generated by the Starling LLM for each rubric. These rubrics are derived from the ASAP and ASAP++ datasets (Mathias and Bhattacharyya, 2018), which were utilized in our evaluation. The data indicates that Llama tends to generate the most numerous questions, followed by Starling and then Gemma.

For ELLIPSE dataset, one rubric for each trait is used to score all the essays in the dataset. Table 11 illustrates the number of generated questions from each rubric, and an example of the generated questions for each trait.

I Score-scaling

The rationale for scaling essay scores based on grade level stems from the fact that writing expectations are generally lower in the earlier grades compared to higher ones. To develop an AES system capable of accommodating all educational levels, we propose scaling the scores for each grade level within a defined range during the training process. This range is then converted back to the original scoring range for evaluation. We assigned the following score ranges to the different grade levels in the ASAP dataset: [0,4] for grade 7, [0,5] for grade 8, and [0,6] for grade 10, showing that the maximum score is decreased by one score level (1 point) for each lower grade relative to the maximum grade level within the dataset. Note that this is applicable only when the scoring rubrics for the different prompts in the dataset are different. Hence, we applied score-scaling only to the ASAP dataset.

[10] https://huggingface.co/princeton-nlp/gemma-2-9b-it-SimPO