Conundrums in Cross-Prompt Automated Essay Scoring: Making Sense of the State of the Art

Shengjie Li and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75080-0688
{sxl180006,vince}@hlt.utdallas.edu

Abstract

Cross-prompt automated essay scoring (AES), an under-investigated but challenging task that has gained increasing popularity in the AES community, aims to train an AES system that can generalize well to prompts that are unseen during model training. While recently-developed cross-prompt AES models have combined essay representations that are learned via sophisticated neural architectures with so-called prompt-independent features, an intriguing question is: are complex neural models needed to achieve state-of-the-art results? We answer this question by abandoning sophisticated neural architectures and developing a purely feature-based approach to cross-prompt AES that adopts a simple neural architecture. Experiments on the ASAP dataset demonstrate that our simple approach to cross-prompt AES can achieve state-of-the-art results.

1 Introduction

Automated essay scoring (AES) is the task of assigning a single score (also known as the holistic score) to an essay that summarizes its overall quality. Traditional work on AES has focused on *within-prompt* scoring, where an AES model is trained on manually annotated essays written for a given prompt and subsequently applied to essays written for the same prompt. Although considerable success has been made on within-prompt scoring, there is a key weakness associated with these prompt-specific models. When they are applied to essays written for a different prompt, their performance often deteriorates considerably. Hence, in practice, before they are applied to score essays written for a new prompt, they need to be retrained on scored essays written for the new prompt. However, manually scoring essays is a time-consuming process and requires a lot of expertise.

To address the aforementioned weakness, researchers have begun working on *cross-prompt* essay scoring, where the goal is to train a model on annotated essays that are *not* written for the target prompt and apply the resulting model to essays written for the target prompt. In other words, the goal of cross-prompt scoring is to train a model on essays written for existing prompts so that it can accurately score essays written for a new prompt without the need to retrain it on essays from the new prompt. Cross-prompt scoring is a very challenging task. To understand why, consider the task of scoring essays written for the prompt "Write a persuasive essay on why one should (or should not) support Obamacare". Intuitively, a high-scoring essay should provide evidence(s) that can adequately support the claim of why one should (or should not) support Obamacare. However, determining whether an argument is persuasive could require domain knowledge (in this case knowledge about Obamacare), which the model may not possess in the absence of training data for the new prompt.[1]

Research on cross-prompt scoring is still in its infancy. The vast majority of the recently-developed cross-prompt scoring models (e.g., Ridley et al. (2021); Chen and Li (2023); Do et al. (2023)) are composed of two parts: (1) learning a representation of an essay that is specific for the task of AES by training Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks using AES data; and (2) employing a set of *prompt-independent* features derived from the input essay. Designing prompt-independent features is by no means the focus of cross-prompt scoring research: the features being used in existing cross-prompt AES models were designed by AES researchers in the pre-neural NLP era during which the focus of AES research was on feature engineering. The first part (learning a task-specific essay representation) is what cross-prompt scoring

[1] While one could rely on large language models (LLMs) for such background knowledge, this is not a general solution: LLMs do not possess knowledge of events that took place after the date on which they were trained.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7661–7681
August 11-16, 2024 ©2024 Association for Computational Linguistics

researchers have been trying to improve over the past few years, resulting in AES models that have become increasingly sophisticated.

Much of the recent work on AES, including cross-prompt AES, has focused on improving performance numbers. While improving performance numbers is an important goal of AES research, what is crucially missing is an understanding of what has been improved. For instance, as mentioned above, state-of-the-art cross-prompt AES models score essays by exploiting both a learned essay representation and a set of prompt-independent features. Before continuing to develop increasingly sophisticated AES models, we should perhaps step back and ask an intriguing question: can we achieve state-of-the-art results if we abandon complex neural architectures and, like the AES researchers in the pre-neural NLP era, adopt a purely feature-based approach with a simple architecture?

Given the above discussion, our goal in this paper is to gain a better understanding of the state of the art in cross-prompt AES by answering the aforementioned question. Unlike recently-developed cross-prompt AES models, which focus on designing increasingly sophisticated AES models to learn essay representations, we abandon entirely the idea of learning essay representations, focusing instead on a purely *feature-based* approach where we employ a *simple* neural architecture in conjunction with a feature set composed of features commonly used in existing cross-prompt AES models as well as our own features. Following previous work on cross-prompt AES (e.g., Ridley et al. (2021), Chen and Li (2023)), we evaluate our approach in two settings that differ in terms of whether essay *traits* (i.e., dimensions of essay quality such as ORGANIZATION and MECHANICS) are exploited when scoring an essay holistically. More specifically, in the first setting, essays are scored based on a set of input features, whereas in the second setting, essays are scored based on both the features and the automatically scored traits.

We evaluate our cross-prompt AES model on a standard evaluation dataset for AES research, ASAP, showing that when used in combination with a simple feature selection method, our model can achieve state-of-the-art results in the two evaluation settings mentioned above. We believe that the key ramifications of our results are two-fold. First, a purely feature-based approach for cross-prompt AES with a simple neural architecture can work at least as well as a sophisticated model that focuses on learning an essay's representation and combining it with a set of features. Second, feature selection plays a crucial role in our model despite the simplicity of our feature selection method. Furthermore, our results establish a new simple but strong baseline against which future cross-prompt AES models can be compared.

2 Related Work

In this section, we give a brief overview of related work on within- and cross-prompt scoring. For a detailed discussion of related work, we refer the reader to the books and surveys recently published in this area (e.g., Ke and Ng (2019), Beigman Klebanov and Madnani (2021), Li and Ng (2024a)).

2.1 Within-Prompt Scoring

Holistic scoring. Early approaches to heuristic scoring are heuristic-based, where the holistic score is typically computed as a weighted sum of the trait scores (Attali and Burstein, 2006). Given the lack of labeled data, the trait scores are also computed heuristically, with a focus on the easier-to-compute traits, such as those that are based on grammaticality and structure rather than content. After training data becomes available, researchers have focused on hand-engineering the features that are to be used to train classification or regression models for holistic scoring (Larkey, 1998; Burstein et al., 1998; Miltsakaki and Kukich, 2004; Yannakoudakis et al., 2011). With the advent of the neural NLP era, AES models have incorporated neural networks such as CNNs and LSTM networks to automatically extract features from the essays (Taghipour and Ng, 2016; Dong et al., 2017; Hussein et al., 2020; Kumar et al., 2022). More recently, many AES studies have utilized pre-trained language models to obtain essay embeddings, followed by fine-tuning the models to optimize performance (Uto et al., 2020; Cao et al., 2020; He et al., 2022; Wang et al., 2022).

Trait scoring. While early work on trait scoring has focused on computing the easier-to-compute traits in a heuristic manner (Attali and Burstein, 2006), later work has resorted to machine learning for scoring traits that are based on not only grammaticality and structure but also content, such as the clarity of the essay's thesis and the persuasiveness of the argument the essay makes (Higgins et al., 2004; Persing et al., 2010; Persing and Ng, 2013, 2014, 2015; Somasundaran et al., 2014; Mathias and Bhattacharyya, 2018). More recently, research has shifted from scoring the traits independently of each other to developing multi-task learning models where the trait scores are predicted jointly with the holistic score (He et al., 2022; Kumar et al., 2022).

2.2 Cross-Prompt Scoring

Holistic scoring. In the pre-neural NLP era, AES researchers have recast cross-prompt scoring as domain adaptation (Phandi et al., 2015; Cummins et al., 2016). In this setting, each essay prompt is viewed as a "domain", so that transfer learning techniques can be employed to adapt a model trained on the existing prompts (i.e., the "source domains") to a new prompt (i.e., the "target domain"). Note that Phandi et al.'s approach and Cummins et al.'s approach are both developed for a soft version of cross-prompt scoring where a small number of labeled essays from the target prompt are available for model training in addition to a large number of essays from the source prompts.

Recent cross-prompt scoring models are neural-based, aiming to learn a prompt-independent representation from the training essays and combine the resulting representation with a set of prompt-independent features to holistically score an essay written for a new prompt (Jin et al., 2018; Li et al., 2020; Ridley et al., 2020). These models, unlike the domain-adapted AES models, are *not* trained on any essays written for the target prompt.

Joint holistic and trait scoring. Like in within-prompt scoring, in cross-prompt scoring researchers have also developed models based on multi-task learning where the trait scores are jointly predicted with the holistic score. Ridley et al. (2021) propose an innovative approach that extends a cross-prompt holistic scoring model (Ridley et al., 2020) by (1) incorporating distinct decoding components tailored to individual traits and (2) using a cross-trait attention mechanism to capture potential interdependencies among traits. Chen and Li (2023) propose a multi-trait scoring system utilizing contrastive learning, which is designed to learn consistent essay representations across different prompts. This approach captures features shared by essays from different prompts, thereby helping the model generalize well across prompts. Do et al. (2023) introduce a prompt and trait relation-aware cross-prompt essay trait scorer, which leverages essay-prompt attention and topic-coherence features to capture prompt adherence in essays, even in the absence of labeled data.

3 Corpus

For model training and evaluation, we employ the widely-used ASAP[2] corpus and its and its extension, ASAP++ (Mathias and Bhattacharyya, 2018).

ASAP (Automated Student Assessment Prize), which is released as part of a Kaggle competition in 2012, is composed of essays manually annotated with their holistic scores. The essays are written for eight prompts, including two for persuasive essays, two for narrative essays, and four for source-dependent essays. Different rubrics are used for scoring prompts, and as a result, the score ranges for different prompts can be different. For instance, one prompt has a score range of 0 to 3, while another prompt has a score range of 1 to 6. The eight prompts and their statistics can be found in Appendix A.

ASAP++ is an extension of ASAP where each essay is additionally scored along different traits. Eight traits are scored, including CONTENT (how clear and focused the writing is and how well-developed the main ideas are), WORD CHOICE (how well the words convey the intended message), ORGANIZATION (how well-organized the essay is), PROMPT ADHERENCE (how adherent the essay is to the prompt), SENTENCE FLUENCY (whether the sentences in the essay are of high quality), CONVENTIONS (how well the essay demonstrates standard writing conventions), NARRATIVITY (how coherent and cohesive the response is), and LANGUAGE (how good grammar and spelling are). Note, however, that the traits that are used for essays written for different prompts can be different: some essays are scored along three traits while some are scored along six traits. Additional information on the traits can be found in Appendix B.

4 Approaches to Holistic Scoring

In this section, we describe our two feature-based approaches to holistic scoring. The first one does not exploit essay traits (Section 4.1), whereas the second one does (Section 4.2).

4.1 Holistic Scoring without Traits

Below we describe our features and model.

4.1.1 Features

At a high level, the features we employ can be divided into two broad categories: existing features and our proposed features.

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

4.1.1.1 Existing Features

The existing features we employ are taken from two AES systems, Ridley et al.'s (2020) system and Uto et al.'s (2020) system, as described below.

Ridley et al.'s (2020) features. These are a manually curated set of 86 prompt-independent features that include readability features (features computed using different readability indices, such as the Coleman–Liau index), text complexity features (features that capture syntactic complexity, such as the number of clauses per sentence), text variation features (features that capture variations in word and part-of-speech usage, such as the number of unique words), length-based features (any other count-based features such as the total number of words), and sentiment-based features (features that encode document- and sentence-level sentiment, such as the percentage of positive sentences).

Uto et al.'s (2020) features. Only three of the 25 features in Uto et al.'s feature set are not present in Ridley et al.'s (2020) feature set: the number of lemmas, the number of question marks, and the number of exclamation marks. As we will see in Section 4.3, Ridley et al. and Uto et al. employ different feature normalization methods, so we use all of Uto et al.'s features despite the overlap.

4.1.1.2 Our Proposed Features

Below are the additional features we propose.

Part-of-speech (POS) Bigram features. Hypothesizing that POS bigrams can help a model generalize to new prompts, we propose 902 features, each of which encodes the count of a POS bigram that appears in the training data. An example of a useful POS bigram feature is "CC NN", which can capture the presence of phrases that signal the elaboration of an idea, such as "for instance".

Prompt Adherence features. Crucially missing from our existing features are those that encode whether an essay is adherent to the prompt for which it is written. For this reason, we propose four features that aim to measure an essay's adherence to its prompt. The first feature computes the dot score between the embedding of an essay and that of its prompt[3]. To compute the remaining features, we (1) compute the dot score between the embedding of each sentence in the essay and that of the prompt, and (2) take the maximum, minimum, and average dot scores to be the feature values.

Top-N Words features. None of the aforementioned features are word-based features. We hypothesize that word-based features could be a useful addition to the feature set. Note, however, that word-based features may render the resulting model prompt-specific. As a result, we strike a balance by employing a group of 300 features that are computed based on the set of N words that appear most frequently in the training data. For each word w in this set, we collect three types of statistics as features: (1) the count of w in an essay, (2) the number of sentences in an essay that contains w, and (3) the percentage of sentences in an essay that contains w. We set N to 100 in our experiments.

Pronoun features. This is a group of 218 word-based features that are specialized for pronouns. Its design is motivated by our desire to investigate whether writing quality is correlated with the frequency with which certain types of pronouns are used. Specifically, this group is composed of six types of pronoun-related features: (1) the count of each pronoun (e.g., the count of "I"), (2) the count of pronouns belonging to each pre-defined pronoun group (e.g., the count of first person pronouns), (3) the number of sentences that contain each pronoun (e.g., the number of sentences that contain "I"), (4) the number of sentences that contain pronouns belonging to each pre-defined pronoun group (e.g., the number of sentences that contain first person pronouns), (5) the percentage of sentences that contain each pronoun (e.g., the percentage of sentences that contain "I"), and (6) the percentage of sentences that contain pronouns belonging to each pre-defined pronoun group (e.g., the percentage of sentences that contain first person pronouns).

A description of each of these features can be found in Appendix C.

4.1.1.3 Feature Analysis

To gain insights into which features are likely to be useful for holistic scoring, we rank the features based on the absolute average of the Pearson and Spearman Correlation Coefficients computed between the feature values and the holistic scores on the entire dataset. The top three features, which all encode word variation, include the number of word types and the number of word lemmas. The next feature, which counts the number of syllables in an essay, indirectly encodes essay length. This is followed by a feature that encodes text readability by counting the number of complex words. After that, we have two more length-based features, one

[3] We use the `all-mpnet-base-v2` model from Reimers and Gurevych's (2019) sentence-transformers package to obtain embeddings.

encoding the number of sentences and the other the number of words. While these top-ranked features appear to encode different aspects of an essay, intuitively they are all positively correlated with essay length. In other words, this analysis seems to suggest that the length of an essay is a strong indicator of its holistic score.

To gain further insights into which *categories* of features are likely to be useful for holistic scoring, we divide our features into nine categories. The first five categories are defined in Ridley et al. (2020) (see Section 4.1.1.1), whereas the last four categories come from our proposed features (see Section 4.1.1.2). These categories include: (1) readability-based (RB) features, (2) text complexity (TC) features, (3) text variation (TV) features, (4) length-based (LB) features, (5) sentiment-based (SB) features, (6) part-of-speech bigram (POSB) features, (7) prompt adherence (PA) features, (8) top-N words (TNW) features, and (9) pronoun (PRO) features. Overall, the most useful categories are TV and LB. They are followed by RB and PRO. TC, TNW, and PA are of middling importance. The least useful categories are SB and POSB.

The entire list of top-ranked features and their associated categories can be found in Appendix D.

4.1.2 Model

We employ as our model a multi-layer neural network, which takes as input a set of features and outputs a holistic score. We train four models, which differ in terms of the input features. Specifically, to determine whether the features we proposed provide any added value, we experiment with two feature sets, the EXISTING features, which are composed of only Ridley et al.'s and Uto et al.'s features, and ALL features, which are composed of the EXISTING features and all of our proposed features. To determine whether feature selection can improve model performance, we optionally apply feature selection to automatically select features for each of the two aforementioned feature sets. This results in four feature sets: the EXISTING feature set with and without feature selection, and the ALL feature set with and without feature selection.

Given the large number of features in our feature set, we need to employ an efficient feature selection method. For this reason, we choose to filter features based on their Pearson and Spearman Correlation Coefficients with the holistic scores as computed on the training set. Given our hypothesis that features with lower coefficients are less predictive of the holistic score, our feature selection method discards a feature if the minimum of its two coefficients is below a certain threshold.

4.2 Holistic Scoring with Traits

We experiment with two architectures when performing holistic scoring with traits.

4.2.1 Joint Architecture

In virtually all recent approaches to exploiting traits for holistic scoring, the trait scores and the holistic score are predicted *jointly* in a multi-task learning framework where the different tasks (i.e., holistic scoring and trait scoring) interact via a shared representation layer (see Section 2 for a discussion). To enable an apples-to-apples comparison between these approaches and ours, we experiment with an architecture for joint holistic and trait scoring.

Our joint model is the same as the one described in Section 4.1.2 except that the last layer of the network contains nine (rather than one) output nodes. Specifically, there is one output node for scoring each of the eight traits described in Section 3 and one output node for predicting the holistic score.

4.2.2 Pipeline Architecture

We also experiment with a *pipeline* architecture that is composed of *two* steps. In the first step, we score the traits. Then, in the second step, we use the trait scores predicted in the first step as input to predict the holistic score. Note that this pipeline architecture is reminiscent of early heuristic approaches to holistic scoring (e.g., *e-rater* (Attali and Burstein, 2006)), where a heuristic scorer scores an essay holistically by taking the weighted sum of the heuristically-computed trait scores. Recent approaches have avoided adopting a pipeline architecture because of the error propagation problem: since the results of trait scoring have generally been poor, a model that relies solely on the noisily predicted trait scores will unlikely yield accurate prediction of the holistic score.

Next, we describe how the models involved in our two-step pipeline approach are trained. Recall that in the first step, we train trait-specific models to predict trait scores. Specifically, we train one model to predict the score for each trait. The way each of these trait-specific models is trained is the same as the way the holistic scoring model was trained in Section 4.1.2. In particular, for each trait, we train four models, one for each of the four feature sets described in Section 4.1.2. In the second step, we train a model to predict holistic scores. Specifically, we train two versions of this model that also differ in terms of the input features. In the first version, we use all and only the gold trait scores as input. Note that while the gold trait scores are used for model training, the *predicted* trait scores will be used when the resulting model is applied to a test essay. Hence, the success of this model depends entirely on how accurately the traits are scored in the first step. More specifically, the more poorly the traits are scored in the first step, the less likely the model in the second step will perform well. To mitigate this error propagation problem, we consider a second version of the model that we train in the second step: we employ as features not only the gold traits but also the features we proposed in Section 4.1.1. This way the model could be trained to be less dependent on the traits, as it may choose to rely on not only the traits but also the other features.

4.3 Implementation Details

Input normalization. In accordance with Ridley et al. (2020), we apply min-max normalization to their features within each prompt, scaling them to the $[0, 1]$ range. Following Uto et al. (2020), we standardize their features within each prompt to achieve a mean of 0 and a standard deviation of 1. For all other features and the input traits for the holistic scoring model, we also standardize them to have a mean of 0 and a standard deviation of 1.

Training details. All models are trained with two hidden layers with sizes 128 and 64 respectively, using ReLU as the activation function and mean squared error (MSE) as the loss function. For the multi-task learning models, the loss is the sum of the MSE losses over all tasks. Since not all traits are applicable to all prompts, any score predicted for an inapplicable trait will not contribute to the loss. Each neuron in the final output layer uses the sigmoid activation function to predict a score between 0 and 1. The predicted scores are then scaled back to the valid score range accordingly. All models are trained for 15 epochs using AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as the optimizer, $0.1 \times \{\text{total number of update steps}\}$ as the number of warm-up steps, 0.5 as the dropout rate, and 11 as the random seed. We perform grid search for determining the feature selection threshold (by searching out of $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65\}$) and the learning rate (by searching out of $\{0.001, 0.003, 0.01, 0.03, 0.1\}$) that jointly maximize QWK[4], the standard metric for evaluating AES systems, on development data. It takes less than 48 hours to complete training for all experiments on two NVIDIA RTX A6000 48GB GPUs. The best hyperparameter values selected are reported in Appendix E.

5 Evaluation

5.1 Experimental Setup

In this section, we evaluate our feature-based approach to cross-prompt holistic scoring.

Datasets. We use ASAP/ASAP++ for model training and evaluation. Following previous work (Jin et al., 2018; Ridley et al., 2021), we conduct cross-prompt evaluation via leave-one-prompt-out cross-validation experiments. Since ASAP/ASAP++ contains eight prompts, we divide the essays into eight folds based on the prompt for which an essay is written. In each fold experiment, we use one fold for testing, one fold for development (parameter tuning), and the remaining six folds for model training.

Evaluation metric. We employ Quadratic Weighted Kappa (QWK) as our metric for holistic and trait-specific scoring. Since QWK is an agreement metric, higher values are better.

Evaluation settings. We employ two evaluation settings that differ in terms of whether traits are used for holistic scoring. In the first setting, only the input features are used to predict the holistic score of an essay. In the second setting, the trait-specific scores, which may be augmented with the input features, are used to predict holistic scores.

Baseline systems. For holistic scoring with traits as well as trait scoring, we employ seven baseline systems, namely, `Hi att` (Dong et al., 2017), `AES aug` (Hussein et al., 2020), `PAES` (Ridley et al., 2020), `CTS no att` (Ridley et al., 2021), `CTS` (Ridley et al., 2021), `PMAES` (Chen and Li, 2023), and `ProTACT` (Do et al., 2023). For holistic scoring without traits, we use as baseline systems all and only those systems mentioned above that have also reported holistic scoring results without traits, namely, `Hi att` (Dong et al., 2017), `PAES` (Ridley et al., 2020) and `PMAES` (Chen and Li, 2023). A description of each of these systems can be found in Appendix F.

[4] See https://www.kaggle.com/competitions/asap-aes/overview/evaluation for details.

	Model	1	2	3	4	5	6	7	8	Avg.
1	Hi att (Dong et al., 2017)	.372	.465	.432	.523	.586	.574	.514	.323	.474
2	PAES (Ridley et al., 2020)	.746	.591	.608	.641	.727	.609	.707	.635	.658
3	PMAES (Chen and Li, 2023)	.758	**.674**	.658	.625	.735	.578	**.749**	**.718**	.687
4	Existing Feats	.744	.601	.657	.653	.778	.620	.704	.430	.648
5	All Feats	.735	.538	.602	.587	.722	.584	.689	.523	.623
6	Existing Feats Filtered	**.829**	.612	.621	.621	.767	.655	.739	.570	.677
7	All Feats Filtered	.820	.601	**.685**	**.666**	**.786**	**.682**	.693	.654	**.698**

Table 1: Holistic scoring results without traits for each prompt. The results for Hi att and PAES are taken verbatim from Chen and Li (2023). The best result in each column is boldfaced.

5.2 Results and Discussion

5.2.1 Holistic Scoring without Traits

Results of holistic scoring without traits for each of the eight prompts, as well as the macro-averaged results over the prompts, are shown in Table 1. Rows 1–3 of the table show the results of the three baselines. As we can see, results on cross-prompt AES have improved over time, with the later systems performing better than the earlier ones.

Rows 4–7 of Table 1 show the results obtained via the four models of our feature-based approach, which differ in terms of which feature set is used. Rows 4 and 5 show the results of the two models without employing feature selection. On average, they underperform two of the baselines, PAES and PMAES, by 0.01–0.062 points in QWK. Comparing rows 4 and 5, we see that using only the EXISTING features yields better results than than using all of the features. In other words, not only are the additional features not useful, but their incorporation into the feature set hurts performance.

Results of our two models with feature selection are shown in rows 6 and 7 of Table 1. As can be seen, despite its simplicity, our feature selection method seems effective: QWK increases by 0.029 points and 0.075 points when the model employs only the EXISTING features and ALL features, respectively. The fact that after feature selection, using ALL features yields better results than using only the EXISTING features implies that there are useful features in our proposed feature set for holistic scoring. In addition, the fact that before feature selection, using ALL features fails to improve performance in comparison to using only the EXISTING features can be attributed to the presence of many noisy features in our proposed feature set.

5.2.2 Holistic Scoring with Traits

Results of holistic scoring with traits for each of the eight prompts, as well as the macro-averaged results over the prompts, are shown in Table 2. The first seven rows show the results of the seven baseline systems. Note that per-prompt results are not available for each of the baselines as they are not reported in the original papers. As can be seen, the majority of the baselines have QWK scores that hover around 0.67. The best-performing baseline is ProTACT, which achieves a QWK score of 0.674.

Next, consider the three baselines that appear in both Tables 1 and 2, namely, Hi att, PAES, and PMAES. When traits are used, we see a consistent precipitation in the average QWK score: QWK decreases by 0.001–0.021 points. At first glance, these results may appear surprising, as the the introduction of traits is meant to help predict the holistic scores. Nevertheless, jointly predicting the trait scores and the holistic score in these systems increases the complexity of the underlying learning task, adversely affecting holistic scoring. Note that the results for Hi att and PAES in Table 1 and Table 2 are obtained from different sources. Direct comparisons between them might not be accurate.

Rows 8–11 of Table 2 show the results of our joint approach where the holistic score is predicted jointly with the trait scores. These four rows differ in terms of the feature set used for model training. The trends we observed for the three baselines, Hi att, PAES, and PMAES, are also applicable to the four joint models. Specifically, comparing the results of the joint models with the corresponding holistic scoring results without traits in rows 4–7 of Table 1, we see that the average QWK scores drop in all four cases. This means that the addition of trait scoring hurts holistic scoring. It is perhaps not surprising: like in the three baselines, the increase in the complexity of the underlying learning task is likely responsible for the deterioration in performance in the joint models. The drops associated with the joint models (2.3–7.7%) are similar to those associated with the baselines (0.2–4.4%).

Rows 12–15 of Table 2 show the results of our two-step pipeline approach where the holistic scoring model in the second step is trained using only gold traits and tested using only predicted traits.

	Model	1	2	3	4	5	6	7	8	Avg.
1	Hi att (Dong et al., 2017)	–	–	–	–	–	–	–	–	.453
2	AES aug (Hussein et al., 2020)	–	–	–	–	–	–	–	–	.402
3	PAES (Ridley et al., 2020)	–	–	–	–	–	–	–	–	.657
4	CTS no att (Ridley et al., 2021)	–	–	–	–	–	–	–	–	.659
5	CTS (Ridley et al., 2021)	–	–	–	–	–	–	–	–	.670
6	PMAES (Chen and Li, 2023)	–	–	–	–	–	–	–	–	.671
7	ProTACT (Do et al., 2023)	–	–	–	–	–	–	–	–	.674
8	Joint: Existing Feats	.579	.630	.667	.638	.771	.574	.687	.518	.633
9	Joint: All Feats	.724	.451	.531	.546	.696	.557	.670	.544	.590
10	Joint: Existing Feats Filtered	**.829**	.551	.610	.635	.735	.595	.648	.393	.625
11	Joint: All Feats Filtered	.788	.550	.643	.623	.760	.612	.652	.670	.662
12	Step 1: Existing Feats; Step 2: GT	.654	**.619**	.493	.498	.709	.534	.251	**.679**	.554
13	Step 1: All Feats; Step 2: GT	.634	.476	.427	.439	.690	.592	.354	.534	.518
14	Step 1: Existing Feats Filtered; Step 2: GT	.656	.602	.505	.510	.715	.465	.290	.660	.550
15	Step 1: All Feats Filtered; Step 2: GT	.687	.499	.488	.457	.705	.601	.403	.547	.548
16	Step 1: Existing Feats: Step 2: GT+Feats	.738	.584	.697	.658	.776	.656	.688	.641	.680
17	Step 1: All Feats, Step 2: GT+Feats	.750	.557	.684	.646	.777	**.669**	.719	.623	.678
18	Step 1: Existing Feats Filtered; Step 2: GT+Feats	.753	.581	.692	**.664**	.780	.661	.701	.632	**.683**
19	Step 1: All Feats Filtered; Step 2: GT+Feats	.781	.556	**.692**	.639	**.783**	.668	**.724**	.617	.682

Table 2: Holistic scoring results with traits. The results for Hi att, AES aug and PAES are taken verbatim from Ridley et al. (2021). The best result in each column is boldfaced.

These four rows differ in terms of the feature set used to train the models for scoring traits in the first step. As we can see, the average QWK scores of these models hover around 0.55. In particular, these models all substantially underperform the corresponding models in Table 1, where traits are not used: QWK drops by 0.094–0.15 points. These drops can be attributed to error propagation: the trait scores predicted in the first step are not accurate enough to enable accurate prediction of the holistic score in the second step.

The results in rows 16–19 of Table 2 are produced using the same models underlying the results in rows 12–15, except that in the second step, both the traits and all of the features that survive feature selection are used to predict the holistic score. In particular, the same holistic scoring model is used to predict holistic scores in all four rows, so the performance differences observed in these four rows can be attributed solely to the differences in the trait scores predicted in the first step.

A few points deserve mention. First, comparing the results in rows 16–19 with the corresponding results in rows 12–15, we see that holistic scoring results substantially improve when the traits are augmented with our features. These results suggest that the features have indeed helped alleviate the error propagation problem caused by the poorly scored traits. Second, even when all the features are used, these results are still worse than the best holistic scoring results when traits are *not* used (row 7, Table 1), meaning that the use of traits has caused more harm than good to holistic scoring. Nevertheless, the models in rows 16–19 still outperform ProTACT, achieving state-of-the-art cross-prompt holistic scoring results when traits are used.

A natural question is: are traits simply not useful for holistic scoring? To answer this question, we compute the Pearson Correlation Coefficient for each trait with the holistic score and find that each trait is strongly correlated with the holistic score (see Appendix G). Furthermore, to determine whether the holistic scoring results with traits are bad because the trait scores are poorly predicted, we conducted an oracle experiment on ASAP++ in our previous work (Li and Ng, 2024b) in which we trained a linear regressor to predict the holistic score using only the gold traits as features and evaluated the resulting regressor using the same set of gold traits as features via leave-one-prompt-out cross validation. This experiment yields a QWK score of 0.88, suggesting that traits are useful for holistic scoring if they can be accurately scored.

5.2.3 Trait Scoring

In Table 2, we showed that using predicted trait scores does not improve holistic scoring results, so a question is: how accurately are the trait scores predicted? Table 3 shows the QWK results of scoring the eight traits when macro-averaged over the eight folds. The first seven rows show the baseline results. The next four rows show the results of trait scoring where the trait scores are predicted jointly with the holistic score. Note that these results are derived from the same four joint models that produce the holistic scoring results in rows 8–11 of

	Model	Content	Org	WC	SF	Conv	PA	Lang	Nar
1	Hi att (Dong et al., 2017)	.348	.243	.416	.428	.244	.309	.293	.379
2	AES aug (Hussein et al., 2020)	.342	.256	.402	.432	.239	.331	.313	.377
3	PAES (Ridley et al., 2020)	.539	.414	.531	.536	.357	.570	.531	.605
4	CTS no att (Ridley et al., 2021)	.541	.424	.558	.544	.387	.561	.539	.605
5	CTS (Ridley et al., 2021)	.555	.458	.557	.545	.412	.565	.536	.608
6	PMAES (Chen and Li, 2023)	.567	.481	.584	.582	.421	.584	.545	.614
7	ProTACT (Do et al., 2023)	**.596**	**.518**	**.599**	**.585**	**.450**	.619	**.596**	**.639**
8	Existing Feats - Joint	.548	.430	.516	.537	.296	.592	.543	.602
9	All Feats - Joint	.554	.427	.421	.467	.373	.603	.515	.595
10	Existing Feats Filtered - Joint	.544	.398	.422	.506	.241	.549	.537	.574
11	All Feats Filtered - Joint	.568	.458	.570	.434	.373	**.621**	.562	.614
12	Existing Feats - Independent	.569	.477	.507	.532	.362	.568	.558	.617
13	All Feats - Independent	.562	.393	.411	.454	.373	.559	.509	.605
14	Existing Feats Filtered - Independent	.562	.473	.508	.535	.386	.566	.554	.590
15	All Feats Filtered - Independent	.592	.478	.459	.452	.439	.617	.556	.637

Table 3: Trait scoring results (Org: Organization, WC: Word Choice; SF: Sentence Fluency; Conv: Conventions; PA: Prompt Adherence; Lang: Language; Nar: Narrativity). The results for Hi att, AES aug and PAES are taken verbatim from Ridley et al. (2021). The best result in each column is boldfaced.

Table 2. The last four rows show the results of trait scoring where the traits are scored independently of each other. These results are derived from the trait scoring models used in Step 1 of the pipeline architecture mentioned earlier.

A few points deserve mention. First, comparing the joint trait scoring results with the corresponding independent trait scoring results, we see that there is no clear winner. More specifically, while the independent versions consistently outperform their joint counterparts when scoring Content, Narrativity, and Conventions, the results are rather mixed w.r.t. the remaining traits. Second, the best-performing system for trait scoring is ProTACT, which achieves the highest QWK score on every trait except Prompt Adherence. As discussed earlier, virtually all systems achieve worse results on holistic scoring when traits are involved in the scoring process. Nevertheless, since Do et al. (2023) do not report holistic scoring results *without* traits, it is not clear whether the level of performance ProTACT has achieved on trait scoring can enable its predicted trait scores to benefit holistic scoring. Third, despite the fact that the best holistic scoring results are achieved *without* using traits, it by no means implies that we can safely ignore trait scoring for at least two reasons. First, trait scoring is important in its own right: if predicted accurately, trait scores can help inform an essay's writer which aspects of their essay need improvement. Second, as shown earlier in the oracle experiment, accurate trait scores can indeed improve holistic scoring.

Since ProTACT, which employs a sophisticated neural model for AES, has achieved better trait scoring results than ours, a relevant question is: does that mean a feature-based approach is insufficient for trait scoring? We believe the answer is no. Among the eight traits, Content and Narrativity are intuitively the most difficult to score accurately as they are dependent on an essay's content. Comparing one of our models, All Feats Filtered (row 15), with ProTACT (row 7), we see that the two perform comparably on Content and Narrativity. In contrast, All Feats Filtered underperforms ProTACT primarily on the remaining six traits, all of which are based on an essay's surface form or structure rather than its content and hence are intuitively easier to score. We therefore speculate that it is possible to improve the scoring of these traits by identifying additional features.

6 Conclusion

We examined the relatively under-studied task of cross-prompt essay scoring, seeking to understand (1) whether state-of-the-art performance on cross-prompt scoring could only be achieved using sophisticated models and (2) what role the features played in the scoring process. For this reason, we proposed a purely feature-based approach to cross-prompt scoring that combined existing features with those of our own, achieving state-of-the-art results on the ASAP dataset when our feature set was used to train a simple neural architecture. Not only does our work establish a strong baseline against which future work can be compared, but it serves to remind researchers that understanding which portions of a complex model are chiefly responsible for performance improvements is as important as demonstrating performance improvements itself.

Limitations

We believe that our work has several limitations. First, while we managed to augment a set of existing features with our own to create a feature set that, when combined with feature selection, achieves state-of-the-art results on cross-prompt scoring, we have not tested the full potential of a purely feature-based approach to cross-prompt scoring. More specifically, the EXISTING features that we employed came solely from Ridley et al. (2021) and Uto et al. (2020). In other words, there are many existing features that we have not incorporated into our feature set, particularly those that are proposed in the pre-neural NLP era. Repeating our experiments with a more comprehensive feature set composed of all of the features proposed in AES so far could help us discover the full potential of a feature-based approach and establish a stronger baseline against which future work can be compared. Second, while we have an augmented feature set that can achieve state-of-the-art results, we did not use it in combination with existing cross-prompt AES models, such as PMAES and ProTACT, to determine whether having better features can improve the performance of existing models.

References

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

Beata Beigman Klebanov and Nitin Madnani. 2021. *Automated Essay Scoring*. In Graeme Hirst, editor, *Synthesis Lectures in Human Language Technologies*. Morgan & Claypool Publishers.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. 1998. Automated scoring using a hybrid feature identification technique. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 206–210, Montreal, Quebec, Canada. Association for Computational Linguistics.

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.

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

Yaqiong He, Feng Jiang, Xiaomin Chu, and Peifeng Li. 2022. Automated Chinese essay scoring from multiple traits. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3007–3016, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192, Boston, Massachusetts, USA. Association for Computational Linguistics.

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

Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097, Melbourne, Australia. 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 28th International Joint Conference on Artificial Intelligence*, pages 6300–6308, Macao, China.

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.

Leah S. Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 90–95, New York, NY, USA. Association for Computing Machinery.

Shengjie Li and Vincent Ng. 2024a. Automated essay scoring: Recent successes and future directions. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, Jeju, Republic of Korea.

Shengjie Li and Vincent Ng. 2024b. ICLE++: Modeling fine-grained traits for holistic essay scoring. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguisitics*, Mexico City, Mexico. 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.

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).

Eleni Miltsakaki and Karen Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10:25–55.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

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

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, Baltimore, Maryland. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552, Beijing, China. Association for Computational Linguistics.

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, Lisbon, Portugal. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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. *ArXiv*, abs/2008.01441.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 950–961, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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

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.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

A Statistics on ASAP

In this section, we present additional statistics on the ASAP corpus. Table 4 displays the eight essay prompts featured in ASAP, along with the corresponding number of essays and the average word count for each prompt. The total number of essays in ASAP as well as the average number of words across all essays are also shown in this table.

B Traits in ASAP++

Recall that eight traits are used to annotate the essays in ASAP: CONTENT, WORD CHOICE, ORGANIZATION, SENTENCE FLUENCY, CONVENTIONS, NARRATIVITY, PROMPT ADHERENCE, and LANGUAGE. The rubrics used for scoring these traits can be found in Mathias and Bhattacharyya (2018). Not all the eight traits are applicable to every prompt. Table 5 shows the set of traits annotated for each essay prompt, whereas Table 6 shows the number of essays annotated for each trait.

C List of Features

Table 7 enumerates the features used in our models alongside their detailed descriptions and the categories to which they belong. Feature names are appended with superscripts for source identification. Specifically, features marked with superscript 1 are features derived using the textstat package[5]. Features marked with superscript 2 are computed using a readability package[6]. Those marked with superscript 3 are NLTK package-derived features[7]. Finally, those marked with superscript 4 are features obtained via the spaCy package[8].

D Feature Usefulness

To gain insights into which features are useful for holistic scoring, we display in Table 8 the 60 features in our feature set where the absolute average of the Pearson and Spearman Correlation Coefficients computed between the feature values and the holistic scores on the entire corpus surpasses 0.2. Notably, certain text variation, length-based, and readability features demonstrate high correlations with the holistic score. Conversely, the newly added features display relatively lower correlations, potentially attributable to their fine-grained characteristics.

E Hyperparameter Settings

In this section, we describe for each set of experiments the best hyperparameter values selected for each fold. Table 9 shows the best hyperparameters for holistic scoring without traits. Table 10 shows the best hyperparameters for trait scoring models. Table 11 shows the best hyperparameters for holistic scoring with traits.

F Baseline Systems

Next, we briefly describe our baseline systems.

`Hi att` (Dong et al., 2017) is a holistic scoring model that first employs a CNN on the input characters with both max pooling and average pooling to obtain the word embeddings. Then, another CNN layer with attention pooling is applied on the word embeddings for extracting sentence representations. After that, a LSTM network with attention pooling is applied to the resulting sentence representations to obtain the document representation. Finally, a linear layer with a sigmoid output neuron is used to predict the holistic score.

`PAES` (Ridley et al., 2020) is a holistic scoring model that is structurally similar to `Hi att`. A notable distinction of `PAES` is that its CNN layer is applied on top of POS tags instead of words or characters. Additionally, `PAES` incorporates handcrafted features as the input of the final linear layer.

`AES aug` (Hussein et al., 2020) builds on Taghipour and Ng's (2016) AES model by increasing the number of output neurons, with the goal of jointly predicting the trait scores and the holistic score. More specifically, each output neuron is used to predict either the holistic score or one of the trait scores. `AES aug` utilizes a CNN to extract n-gram level features, passes them through an LSTM

[5] https://github.com/textstat/textstat

[6] https://github.com/andreasvc/readability

[7] https://www.nltk.org/

[8] https://spacy.io/

	Prompt	Avg. # Words	Essays
1	Write a letter to the editor of a newspaper about how computers affect society today.	365.4	1783
2	Write a letter to the editor of a newspaper about censorship in libraries	380.7	1800
3	Write a review about an article called Rough Rough Road by Joe Kurmaskie. The article will be provided.	108.5	1726
4	Explain why the author concludes the story the way the author did. The short story will be provided.	94.3	1772
5	Describe the mood created by the author in the memoir. Support your answer with relevant and specific information from the memoir	122.1	1805
6	Describe the difficulties that builders of the Empire State Building faced because of allowing dirigibles to dock there.	153.2	1800
7	Write a story about a time when you were patient OR write a story about a time when someone you know was patient OR write a story in your own way about patience.	167.6	1569
8	We all understand the benefits of laughter. For example, someone once said, "Laughter is the shortest distance between two people." Many other people believe that laughter is an important part of any relationship. Tell a true story in which laughter was one element or part.	604.7	723
Overall		222.5	12978

Table 4: The eight writing prompts in ASAP.

ID	Traits
1	Content, Word Choice, Organization, Sentence Fluency, Conventions
2	Content, Word Choice, Organization, Sentence Fluency, Conventions
3	Content, Prompt Adherence, Narrativity, Language
4	Content, Prompt Adherence, Narrativity, Language
5	Content, Prompt Adherence, Narrativity, Language
6	Content, Prompt Adherence, Narrativity, Language
7	Content, Organization, Conventions
8	Content, Word Choice, Organization, Sentence Fluency, Conventions

Table 5: Traits applicable to each ASAP++ prompt.

Trait	# of Annotated Essays
Content	12978
Organization	5875
Word Choice	4306
Sentence Fluency	4306
Conventions	5875
Prompt Adherence	7103
Language	7103
Narrativity	7103

Table 6: Number of annotated essays in ASAP++ for each trait.

network, performs mean pooling, and then uses a linear layer for joint holistic and trait scoring.

CTS (Ridley et al., 2021) is the first model that explores cross-prompt multi-trait scoring. Similar to PAES, CTS utilizes a CNN with attention pooling on the POS tags of the input essays to obtain n-gram level features. For each trait, it applies a LSTM network with attention pooling to the n-gram representations to obtain trait-specific essay representations. These representations are then combined with hand-crafted features from Ridley et al. (2020), followed by a cross-trait attention mechanism so that information can be shared by different traits. Finally, the trait scores and the holistic score are predicted using a linear layer with sigmoid activation.

CTS no att (Ridley et al., 2021) is the same as CTS except that CTS no att does not utilize the cross-trait attention mechanism.

PMAES (Chen and Li, 2023) performs holistic scoring both with traits and without traits. It enhances the representations extracted by Hi att by applying a contrastive learning objective to learn consistent essay representations across different prompts. This approach captures features shared by essays from different prompts, thereby helping the model generalize well across prompts. Additionally, it incorporates the hand-crafted features from Ridley et al. (2020) before feeding the representations into the final linear layer.

ProTACT (Do et al., 2023) is the current state-of-the-art cross-prompt trait scoring model. The model extracts essay representations by employing CNNs and LSTM networks on the POS embeddings of the input essays. It also extracts prompt representations by applying the same network architecture to the sum of the POS embeddings and the GloVe embeddings (Pennington et al., 2014) for each word in the prompt. Prompt-aware essay representations are then derived using multi-head attention, with the prompt representations acting as the query and the essay representations as the key and value. These representations are subsequently concatenated with hand-crafted features and processed through a linear layer to predict both the holistic score and the trait-specific scores.

G Usefulness of Traits

To gain insights into whether the traits in ASAP++ are useful for holistic scoring, we present in Table 12 the Pearson Correlation Coefficient between the gold scores for each trait and the gold holistic scores. As we can see, the correlations are high across all traits, suggesting that these essay traits are potentially useful for improving cross-prompt holistic scoring if they can be scored accurately.

Feature Group	Feature Name	Description
	Ridley et al.'s (2020) Features (86 features)	
LB^{R}	word_count	The total number of words in the essay.
	mean_word	The average number of characters in each word.
	ess_char_len	The number of characters in the essay.
	mean_sent[3]	The average number of words in each sentence.
	characters_per_word[2]	The average number of characters in each word.
	avg_word_len	The average number of characters in each word.
	avg_words_per_sentence	The average number of words in each sentence.
	characters[2]	The number of characters in the essay.
	syllables[2]	The number of syllables in the essay.
	words[2]	The number of words in the essay.
	words_per_sentence[2]	The average number of words in each sentence.
	sentences_per_paragraph[2]	The average number of sentences in each paragraph.
	.[3]	The number of periods in the essay.
	,[3]	The number of commas in the essay.
	syll_per_word[2]	The average number of syllables in each word.
RB^{R}	automated_readability[1]	A readability metric that measures the readability of a text based on characters per word and words per sentence.
	linsear_write[1]	A readability metric developed for the U.S. Air Force to help them calculate the understandability of technical manuals, factoring in sentence length and words that are considered difficult.
	Kincaid[2]	A readability metric which estimate the readability of English texts based on sentence length and word length.
	ARI[2]	A readability metric that measures the readability of a text based on characters per word and words per sentence.
	Coleman-Liau[2]	A readability assessment that estimates the U.S. grade level required to understand a piece of text based on characters, words, and sentences.
	FleschReadingEase[2]	A readability metric that measures the readability of text based on syllables, words, and sentences. The scores are on a scale from 0 to 100, with higher scores indicating easier-to-read text.
	GunningFogIndex[2]	A readability metric that estimates the years of formal education a person needs to understand the text on the first reading.
	LIX[2]	A readability metric that considers sentence length and the percentage of long words (words with more than six characters) in a text.
	SMOGIndex[2]	A readability formula that estimates the education level needed to understand a piece of text by analyzing the number of polysyllabic words (words with three or more syllables) within the text.
	RIX[2]	A variant of the LIX readability index that only takes into account the average number of long words per sentence.
	DaleChallIndex[2]	A readability formula that uses word difficulty based on a list of familiar words, along with sentence length, to estimate the grade level required to understand a text.
	sentences[2]	The total number of sentences present in the essay.
	paragraphs[2]	The total number of paragraphs present in the essay.
	long_words[2]	The number of words that have 7 or more characters.
	complex_words[2]	The number of words that have 3 or more syllables.
	complex_words_dc[2]	The total number of words that are not in the Dale-Chall word list of 3000 words recognized by 80% of fifth graders.
TC^{R}	clause_per_s[4]	The average number of clauses per sentence.
	sent_ave_depth[4]	The average parse tree depth per sentence in each essay,
	ave_leaf_depth[4]	The average parse depth of each leaf node in the parse tree.
	max_clause_in_s[4]	The maximum number of clauses in the sentences of the essay.
	mean_clause_l[4]	The average number of words in each clause.
SB^{R}	overall_positivity_score[3]	Overall, how positive the essay is.
	overall_negativity_score[3]	Overall, how negative the essay is.

Continued on next page

Feature Group	Feature Name	Description
	positive_sentence_prop[3]	The percentage of positive sentences in the essay.
	neutral_sentence_prop[3]	The percentage of neutral sentences in the essay.
	negative_sentence_prop[3]	The percentage of negative sentences in the essay.
TV^{R}	sent_var[3]	The variance of the length of sentences in the essay.
	word_var[3]	The variance of the length of words in the essay.
	stop_prop	The percentage of stopwords in the essay.
	unique_word	The total number of unique words in the essay.
	type_token_ratio[2]	The number of unique words divided by the number of words.
	wordtypes[2]	The total number of unique words present in the essay.
	tobeverb[2]	The number of “to be” verbs in the essay.
	auxverb[2]	The number of auxilllary verbs in the essay.
	conjunction[2]	The number of conjunctions in the essay.
	pronoun[2]	The number of pronouns in the essay
	preposition[2]	The number of prepositions in the essay
	nominalization[2]	The number of nominalizations in the essay
	begin_w_pronoun[2]	The number of sentences in the essay that begin with a pronoun.
	begin_w_interrogative[2]	The number of sentences in the essay that begin with an interrogative.
	begin_w_article[2]	The number of sentences in the essay that begin with an article.
	begin_w_subordination[2]	The number of sentences in the essay that begin with a subordination.
	begin_w_conjunction[2]	The number of sentences in the essay that begin with a conjunction.
	begin_w_preposition[2]	The number of sentences in the essay that begin with a preposition.
	spelling_err[3]	The number of words that are not in The Brown corpus of the NLTK package.
	prep_comma[3]	The number of preprositions and commas in the essay.
	MD[3]	The number of tokens having a POS tag of MD in the text.
	DT[3]	The number of tokens having a POS tag of DT in the text.
	TO[3]	The number of tokens having a POS tag of TO in the text.
	PRP$[3]	The number of tokens having a POS tag of PRP$ in the text.
	JJR[3]	The number of tokens having a POS tag of JJR in the text.
	WDT[3]	The number of tokens having a POS tag of WDT in the text.
	VBD[3]	The number of tokens having a POS tag of VBD in the text.
	WP[3]	The number of tokens having a POS tag of WP in the text.
	VBG[3]	The number of tokens having a POS tag of VBG in the text.
	RBR[3]	The number of tokens having a POS tag of RBR in the text.
	CC[3]	The number of tokens having a POS tag of CC in the text.
	VBP[3]	The number of tokens having a POS tag of VBP in the text.
	JJS[3]	The number of tokens having a POS tag of JJS in the text.
	VBN[3]	The number of tokens having a POS tag of VBN in the text.
	POS[3]	The number of tokens having a POS tag of POS in the text.
	NNS[3]	The number of tokens having a POS tag of NNS in the text.
	WRB[3]	The number of tokens having a POS tag of WRB in the text.
	JJ[3]	The number of tokens having a POS tag of JJ in the text.

Continued on next page

Feature Group	Feature Name	Description
	CD^3	The number of tokens having a POS tag of CD in the text.
	NNP^3	The number of tokens having a POS tag of NNP in the text.
	RP^3	The number of tokens having a POS tag of RP in the text.
	RB^3	The number of tokens having a POS tag of RB in the text.
	IN^3	The number of tokens having a POS tag of IN in the text.
	VB^3	The number of tokens having a POS tag of VB in the text.
	VBZ^3	The number of tokens having a POS tag of VBZ in the text.
	NN^3	The number of tokens having a POS tag of NN in the text.
	PRP^3	The number of tokens having a POS tag of PRP in the text.
Uto et al.'s (2020) Features (25 features)		
LB^{U}	syllable_count	The number of syllables in the essay.
	num_words	The number of words in the essay.
	num_sentences	The number of sentences in the essay.
	lemma_count	The number of lemmas in the essay.
	,	The number of commas in the essay.
	!	The number of exclamation marks in the essay.
	?	The number of question marks in the essay.
TV^{U}	noun_count	The number of nouns in the essay.
	verb_count	The number of verbs in the essay.
	adverb_count	The number of adverbs in the essay.
	adjective_count	The number of adjectives in the essay.
	conjunction_count	The number of conjunctions in the essay.
	spelling_error_count	The number of spelling errors in the essay.
	stopwords_count	The number of stop words in the essay.
RB^{U}	ARI	A readability metric that measures the readability of a text based on characters per word and words per sentence.
	coleman_liau	A readability assessment that estimates the U.S. grade level required to understand a piece of text based on characters, words, and sentences.
	dale_chall	A readability formula that uses word difficulty based on a list of familiar words, along with sentence length, to estimate the grade level required to understand a text.
	difficult_words	The total number of words that are not in the Dale-Chall word list of 3000 words recognized by 80% of fifth graders.
	flesch_reading_ease	A readability metric that measures the readability of text based on syllables, words, and sentences. The scores are on a scale from 0 to 100, with higher scores indicating easier-to-read text.
	flesch_kincaid_grade	A readability metric which estimate the readability of English texts based on sentence length and word length.
	gunning_fog	A readability metric that estimates the years of formal education a person needs to understand the text on the first reading.
	linsear_write	A readability metric developed for the U.S. Air Force to help them calculate the understandability of technical manuals, factoring in sentence length and words that are considered difficult.
	smog_index	A readability formula that estimates the education level needed to understand a piece of text by analyzing the number of polysyllabic words (words with three or more syllables) within the text.
Part-of-speech Bigram Features (902 features)		
POSB	(DT, NN)	The number of appearance of the bigram (DT, NN)
	...	
Pronoun Features (218 features)		
PRO-Pronoun Count	pronoun_cnt_I	The number of pronoun "I" in the essay.
	...	

Continued on next page

Feature Group	Feature Name	Description
PRO-Pronoun Group Count	first_person_pronoun_cnt ...	The number of first person pronouns in the essay.
PRO-Sent Pronoun	sent_cnt_I ...	The number of sentences that contain "I"
PRO-Sent Pronoun Group	sent_first_person_pronoun ...	The number of sentences that contain first person pronouns.
PRO-Sent Pronoun Portion	percentage_sent_I ...	The percentage of sentences that contain pronoun "I".
PRO-Sent Pronoun Group Portion	percentage_sent_first_person ...	The percentage of sentences that contain first person pronouns.
	Prompt Adherence Features (4 features)	
PA	max_sentence_dot_score	Dot score between the embeddings of an essay and its prompt.
	mean_sentence_dot_score	The maximum dot score between the embeddings of sentences of an essay and its prompt.
	min_sentence_dot_score	The average dot score between the embeddings of sentences of an essay and its prompt.
	dot_score	The minimum dot score between the embeddings of sentences of an essay and its prompt.
	Top-N Words Features (300 features)	
TNW-Word Count	top_n_word_count_the ...	The count of "the" in the essay.
TNW-Sent Count	top_n_num_sent_have_the ...	The number of sentences in an essay that contains "the".
TNW-Sent Portion	top_n_percentage_sent_have_the ...	The percentage of sentences in an essay that contains "the".

Table 7: Description of the features along with their group information. Features marked with the superscript R are Ridley et al.'s (2020) features. Features marked with the superscript U are Uto et al.'s (2020) features. Group LB is composed of length-based features. Group RB is composed of readability-based features. Group TC is composed of text complexity features. Group TV is composed of text variation features. Group SB is composed of sentiment-based features. Group POSB is composed of the part-of-speech bigram features. Group PRO is composed of the pronoun-related features. Group PA is composed of the prompt adherence features. Group TNW is composed of the top-N words features.

Group	Feature	*PC*	*SC*	Group	Feature	*PC*	*SC*
TV^{R}	wordtypes	.694	.718	PRO	num_sent_have_third_person_pronoun	.454	.471
TV^{R}	unique_word	.689	.715	TV^{R}	conjunction	.449	.462
LB^{U}	lemma_count	.670	.690	PRO	num_sent_have_demonstrative_pronoun	.437	.448
LB^{U}	syllable_count	.666	.693	TV^{R}	type_token_ratio	-.460	-.408
RB^{R}	complex_words_dc	.653	.694	PRO	demonstrative_pronoun_count	.386	.405
RB^{R}	sentences	.648	.687	PRO	third_person_pronoun_count	.383	.405
LB^{U}	num_words	.655	.679	PRO	num_sent_have_indefinite_pronoun	.366	.386
LB^{R}	sentences_per_paragraph	.636	.671	TV^{R}	nominalization	.336	.357
RB^{R}	long_words	.623	.676	PRO	indefinite_pronoun_count	.329	.358
LB^{R}	ess_char_len	.606	.681	PRO	num_sent_have_that	.334	.339
TV^{U}	noun_count	.629	.657	TV^{R}	auxverb	.324	.344
LB^{R}	characters	.603	.679	TV^{R}	pronoun	.307	.331
TV^{U}	stopwords_count	.622	.644	TV^{R}	preposition	.309	.322
LB^{R}	syllables	.594	.671	TV^{R}	begin_w_article	.310	.291
RB^{U}	difficult_words	.599	.637	PRO	demonstrative_that	.292	.304
LB^{R}	words	.574	.661	PRO	num_sent_have_this	.298	.287
LB^{R}	word_count	.572	.663	TC^{R}	max_clause_in_s	.267	.298
TV^{R}	prep_comma	.590	.637	PRO	demonstrative_this	.282	.279
TV^{U}	verb_count	.597	.620	LB^{R}	,	.250	.289
RB^{R}	complex_words	.588	.629	PRO	num_sent_have_it	.257	.266
TV^{R}	preposition	.575	.625	PA	min_sentence_dot_score	-.267	-.227
LB^{U}	num_sentences	.565	.600	PRO	num_sent_have_first_person_pronoun	.213	.265
TV^{U}	adjective_count	.564	.590	TNW	top_n_percentage_sent_have_the	-.288	-.169
RB^{U}	coleman_liau	.524	.526	PRO	third_person_it	.219	.235
TV^{U}	adverb_count	.502	.527	PRO	first_person_pronoun_count	.192	.251
TV^{R}	spelling_err	.493	.531	PRO	num_sent_have_interrogative_pronoun	.222	.201
TV^{R}	pronoun	.493	.523	RB^{R}	SMOGIndex	.218	.202
TV^{R}	tobeverb	.487	.515	TV^{R}	NNP	.136	.277
LB^{U}	,	.475	.502	PRO	interrogative_pronoun_count	.211	.197
TV^{U}	conjunction_count	.476	.494	PRO	num_sent_have_her	.171	.234

Table 8: Features ranked by the absolute average of the Pearson and Spearman Correlation Coefficients computed between the feature values and the holistic scores. Only the features whose average correlation value exceeds 0.2 are shown. Features marked with the superscript R are Ridley et al.'s (2020) features. Features marked with the superscript U are Uto et al.'s (2020) features. Group LB is composed of length-based features. Group RB is composed of readability-based features. Group TC is composed of text complexity features. Group TV is composed of text variation features. Group PRO is composed of the pronoun-related features. Group PA is composed of the prompt adherence features. Group TNW is composed of the top-N words features. *PC* refers to Pearson's Correlation Coefficient, and *SC* refers to Spearman's Correlation Coefficient.

	1		**2**		**3**		**4**		**5**		**6**		**7**		**8**	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.001	-	.01	-	.01	-	.01	-	.003	-	.01	-	.01	-	.01	-
All Feats	.001	-	.001	-	.001	-	.001	-	.001	-	.001	-	.001	-	.003	-
Existing Feats Filtered	.01	.2	.03	.1	.01	.1	.01	.1	.01	.2	.01	.6	.003	.1	.03	.5
All Feats Filtered	.003	.6	.03	.6	.01	.5	.03	.5	.01	.6	.01	.5	.001	.1	.01	.4

Table 9: Best hyperparameters for holistic scoring without traits for every fold. "lr" refers to learning rate and "th" refers to the threshold selected by highest QWK score on the development set.

(a) Best hyperparameters for Content

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.001	-	.01	-	.01	-	.01	-	.01	-	.01	-	.01	-	.01	-
All Feats	.001	-	.001	-	.01	-	.001	-	.01	-	.001	-	.001	-	.001	-
Existing Feats Filtered	.001	.1	.01	.1	.01	.1	.01	.1	.01	.3	.01	.2	.01	.1	.01	.3
All Feats Filtered	.01	.5	.01	.1	.01	.4	.01	.1	.01	.3	.01	.5	.01	.6	.001	.3

(b) Best hyperparameters for Organization

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.01	-	.01	-	.001	-	.001	-	.001	-	.001	-	.01	-	.001	-
All Feats	.01	-	.01	-	.001	-	.001	-	.001	-	.001	-	.001	-	.001	-
Existing Feats Filtered	.01	.1	.01	.1	.01	.2	.01	.2	.01	.2	.01	.2	.01	.2	.01	.2
All Feats Filtered	.01	.4	.01	.4	.01	.3	.01	.3	.01	.3	.01	.3	.01	.2	.001	.4

(c) Best hyperparameters for Word Choice

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.001	-	.001	-	.01	-	.01	-	.01	-	.01	-	.01	-	.01	-
All Feats	.01	-	.01	-	.001	-	.001	-	.001	-	.001	-	.001	-	.01	-
Existing Feats Filtered	.01	.3	.01	.3	.01	.65	.01	.65	.01	.65	.01	.65	.01	.65	.01	.2
All Feats Filtered	.01	.3	.01	.65	.001	.65	.001	.65	.001	.65	.001	.65	.001	.65	.01	.6

(d) Best hyperparameters for Sentence Fluency

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.01	-	.01	-	.001	-	.001	-	.001	-	.001	-	.001	-	.01	-
All Feats	.001	-	.001	-	.01	-	.01	-	.01	-	.01	-	.01	-	.001	-
Existing Feats Filtered	.01	.2	.01	.2	.01	.6	.01	.6	.01	.6	.01	.6	.01	.6	.001	.3
All Feats Filtered	.01	.4	.01	.3	.01	.5	.01	.5	.01	.5	.01	.5	.01	.5	.01	.4

(e) Best hyperparameters for Conventions

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.01	-	.001	-	.001	-	.001	-	.001	-	.001	-	.01	-	.01	-
All Feats	.01	-	.001	-	.001	-	.001	-	.001	-	.001	-	.001	-	.01	-
Existing Feats Filtered	.001	.1	.001	.1	.01	.2	.01	.2	.01	.2	.01	.2	.01	.3	.01	.1
All Feats Filtered	.01	.2	.001	.1	.001	.3	.001	.3	.001	.3	.001	.3	.01	.3	.001	.3

(f) Best hyperparameters for Prompt Adherence

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.001	-	.001	-	.01	-	.01	-	.01	-	.01	-	.001	-	.001	-
All Feats	.001	-	.001	-	.01	-	.01	-	.01	-	.01	-	.001	-	.001	-
Existing Feats Filtered	.01	.2	.01	.2	.01	.6	.01	.1	.01	.3	.01	.6	.01	.2	.01	.2
All Feats Filtered	.01	.6	.01	.6	.01	.4	.01	.5	.001	.1	.001	.5	.01	.6	.01	.6

(g) Best hyperparameters for Language

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.001	-	.001	-	.01	-	.01	-	.01	-	.01	-	.001	-	.001	-
All Feats	.01	-	.01	-	.01	-	.01	-	.001	-	.001	-	.01	-	.01	-
Existing Feats Filtered	.001	.5	.001	.5	.01	.1	.01	.2	.01	.3	.01	.2	.001	.5	.001	.5
All Feats Filtered	.001	.1	.001	.1	.01	.1	.01	.1	.01	.1	.01	.5	.001	.1	.001	.1

(h) Best hyperparameters for Narrativity

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Existing Feats	.001	-	.001	-	.01	-	.01	-	.01	-	.001	-	.001	-	.001	-
All Feats	.01	-	.01	-	.01	-	.01	-	.001	-	.001	-	.01	-	.01	-
Existing Feats Filtered	.001	.3	.001	.3	.01	.1	.01	.4	.01	.5	.01	.1	.001	.3	.001	.3
All Feats Filtered	.001	.1	.001	.1	.01	.1	.01	.3	.01	.1	.001	.1	.001	.1	.001	.1

Table 10: Best hyperparameters for trait scoring for every fold.

	1		2		3		4		5		6		7		8	
	lr	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**	**lr**	**th**
Step 1: Existing Feats; Step 2: `GT`	.001	-	.01	-	.001	-	.01	-	.01	-	.001	-	.01	-	.001	-
Step 1: All Feats; Step 2: `GT`	.01	-	.01	-	.001	-	.001	-	.001	-	.001	-	.001	-	.001	-
Step 1: Existing Feats Filtered; Step 2: `GT`	.01	-	.01	-	.01	-	.01	-	.01	-	.01	-	.01	-	.01	-
Step 1: All Feats Filtered; Step 2: `GT`	.01	-	.01	-	.01	-	.001	-	.001	-	.001	-	.01	-	.001	-
Step 1: Existing Feats: Step 2: `GT+Feats`	.001	.5	.001	.4	.001	.6	.001	.2	.001	.65	.001	.6	.001	.1	.001	.6
Step 1: All Feats, Step 2: `GT+Feats`	.01	.5	.001	.1	.01	.65	.001	.2	.01	.6	.001	.6	.01	.1	.01	.65
Step 1: Existing Feats Filtered; Step 2: `GT+Feats`	.001	.5	.001	.4	.01	.6	.001	.5	.01	.65	.001	.6	.01	.1	.01	.6
Step 1: All Feats Filtered; Step 2: `GT+Feats`	.01	.5	.01	.1	.01	.6	.001	.2	.01	.4	.001	.6	.01	.1	.01	.65

Table 11: Best hyperparameters for holistic scoring with traits for every fold. Threshold does not apply to experiments whose step 2 is `GT`.

Trait	PC
Content	.898
Organization	.802
Word Choice	.990
Sentence Fluency	.990
Conventions	.802
Prompt Adherence	.831
Language	.793
Narrativity	.822

Table 12: Pearson's Correlation between each trait and the holistic score.