Rank-Then-Score: Enhancing Large Language Models for Automated Essay Scoring

Anonymous ACL submission

Abstract

In recent years, large language models (LLMs) achieve remarkable success across a variety of tasks. However, their potential in the domain of Automatic Essay Scoring (AES) remains largely underexplored. Moreover, compared to English data, the methods for Chinese AES is not well developed. In this paper, we propose Rank-Then-Score (RTS), a fine-tuning framework based on large language models to enhance their essay scoring capabilities. Specifically, we fine-tune the ranking model (Ranker) with feature-enriched data, and then feed the output of the ranking model, in the form of a candidate score set, with the essay content into the scoring model (Scorer) to produce the final score. Experimental results on two benchmark datasets, HSK and ASAP, demonstrate that RTS consistently outperforms the direct prompting (Vanilla) method in terms of average QWK across all LLMs and datasets, and achieves the best performance on Chinese essay scoring using the HSK dataset.

1 Introduction

Automatic Essay Scoring (AES) is a task that uses machine learning methods to score an essay, which shows great efficiency and objectivity compared to humans (Dikli, 2006). The traditional *prompt-specific* AES task (Prompt means Topic) focuses on essays within the same prompt, allowing the scoring model to more accurately capture the scoring criteria for that specific prompt. As a result, it aligns more closely with human scoring criteria and better meets the precision requirements for large-scale examinations (Attali and Burstein, 2006).

Previous research primarily focus on modeling the content of essays using neural network models (Taghipour and Ng, 2016; Dong et al., 2017). Subsequently, researchers try to explore the enhancement of performance by modeling various types of content-related information. Some works achieve satisfactory results in both cross-prompt and prompt-specific tasks by modeling statistical features (Ridley et al., 2020). Some studies achieve significant improvements, even reaching state-of-the-art (SOTA) levels, by incorporating ranking tasks into the scoring process (Yang et al., 2020; Xie et al., 2022). Compared to the absolute quality represented by scoring, ranking can reflect differences between essays through relative quality, thereby reducing biases caused by absolute scoring. This idea is also commonly used in the optimization process of Reward Models in Reinforcement Learning from Human Feedback (RLHF) (Li and Li, 2024).

In recent years, with the advancement of Large Language Models (LLMs), many text regression tasks and text evaluation tasks witness further progress (Vacareanu et al., 2024; Chen et al., 2023). Some studies explore AES methods based on fine-tuning, which achieve considerable improvements compared to zero-shot approaches (Stahl et al., 2024; Li and Ng, 2024).

However, existing methods that combine ranking with regression face a key challenge in the LLM era: the training logic of LLMs(focused on next-token prediction) is not directly compatible with regression or ranking loss formulations (Yang et al., 2020). Consequently, these methods struggle to transfer and leverage the full advantages of LLMs, such as their powerful semantic understanding and multi-task generalization capabilities. Meanwhile, some studies consider the integration of ranking task with the scoring task, but the results of such integration do not reach a level comparable to that of supervised small models, and further exploration is needed (Stahl et al., 2024).

Additionally, most current AES research focuses on the English language. However, when writing essays in different languages, such as Chinese, the evaluation criteria can vary significantly. Therefore, some researchers also explore AES methods in Chinese essays and achieve certain progress (Song et al., 2020a,b; He et al., 2022).

In this paper, we propose a novel Rank-Then-Score (RTS) pipeline, which leverages two key advantages: (1) Our multi-stage design (ranking → scoring) decouples complex tasks into manageable sub-problems, aligning with LLMs' strength in specialized fine-tuning; (2) Building on the success of LLMs as rankers in recommendation systems (Hou et al., 2024) and RLHF (Bai et al., 2022), our Ranker effectively narrows score intervals through pairwise comparisons, ensuring more accurate predictions.

Specificly, RTS comprises two models in two different tasks: **a Ranker and a Scorer**. We first randomly selected reference essays and incorporated essay-related features into the essays. Then, we employ a fine-tuned pairwise Ranker to compare the target essay with the reference essays in a manner analogous to a binary search tree. This process identifies a candidate set of scores for the target essay. Finally, the candidate score set, along with the target essay, is fed into the fine-tuned Scorer to obtain the predicted score.

Moreover, this paper also employs a new Chinese essay scoring dataset: the HSK dataset, which is derived from the essay data of the HSK examination (Cheng, 2022). We clean and filter the data, ultimately obtaining a dataset comprising 8,597 essays.

We conduct experiments on Chinese (HSK) and English (ASAP) (Hamner et al., 2012) datasets using various LLMs for ranking and scoring tasks. The fine-tuned RTS method is compared to a Vanilla baseline fine-tuned with standard instructions, and Quadratic Weighted Kappa (QWK) is used as the evaluation metric. As shown in the Table 2 and Table 3, the RTS method outperforms the Vanilla method across all settings. Specifically, on the HSK dataset, RTS achieves an improvement of 1.9% ($74.6\% \rightarrow 76.5\%$) over the Vanilla method, while on the ASAP dataset, it achieves improvements of 1.7% ($78.1\% \rightarrow 79.8\%$) and 1.1% ($78.3\% \rightarrow 79.4\%$) over the Vanilla method in different configurations. Additionally, on the HSK dataset, RTS surpasses other methods using smaller models. On the ASAP dataset, RTS is comparable to the R^{2}BERT (Yang et al., 2020) method and approaches the performance of the NPCR (Xie et al., 2022) method.

In summary, our contributions are as follows:

- We propose a method for integrating ranking and scoring mechanisms within LLMs, thereby enhancing the performance of LLMs for AES.
- We propose a **Binary-Search-Tree**-like approach to transform the results of pairwise ranking into inputs for the scoring model.
- We present a novel method that incorporates rich essay-related **features** into the scoring task.

2 Related Work

Automated Essay Scoring The development of AES is mainly driven by technological advancements and researchers' exploration of essay evaluation criteria. Early methods primarily relied on hand-crafted features (Yannakoudakis et al., 2011; Persing and Ng, 2013). Subsequently, many studies began to introduce neural network models and achieved excellent results (Taghipour and Ng, 2016; Dong et al., 2017; Farag et al., 2018). At the same time, methods that utilized features (Ridley et al., 2020; Chen and Li, 2023) and ranking (Yang et al., 2020; Xie et al., 2022) also emerged. In recent years, an increasing number of studies focus on Multi-Trait Scoring methods (Ridley et al., 2021; Li and Ng, 2024), which are widely applied in various essay scoring works.

After the emergence of LLMs, many researchers believed that the characteristic of LLMs performing well across various downstream tasks is worth leveraging for the AES task. Among them, the work of (Lee et al., 2024) explored the performance of the Multi-Trait method in a zero-shot setting on LLMs, while (Xiao et al., 2024) investigated the potential of fine-tuning LLMs to scoring. Recently, (Stahl et al., 2024) explores various instruction methods in the in-context learning of LLMs, achieving comprehensive results in this field.

Chinese AES In addition to these developments, there are also some advanced explorations in the field of Chinese AES. Firstly, in the context of pretrained methods, research on Chinese AES also directs its approach towards Multi-Trait Scoring (Song et al., 2020a,b). Moreover, (Gong et al., 2021) meticulously listed the majority of aspects that need to be considered in Chinese AES, providing significant guidance for future research. Following that, (He et al., 2022) proposed a new method based on multiple scorers, which achieved considerable improvement.

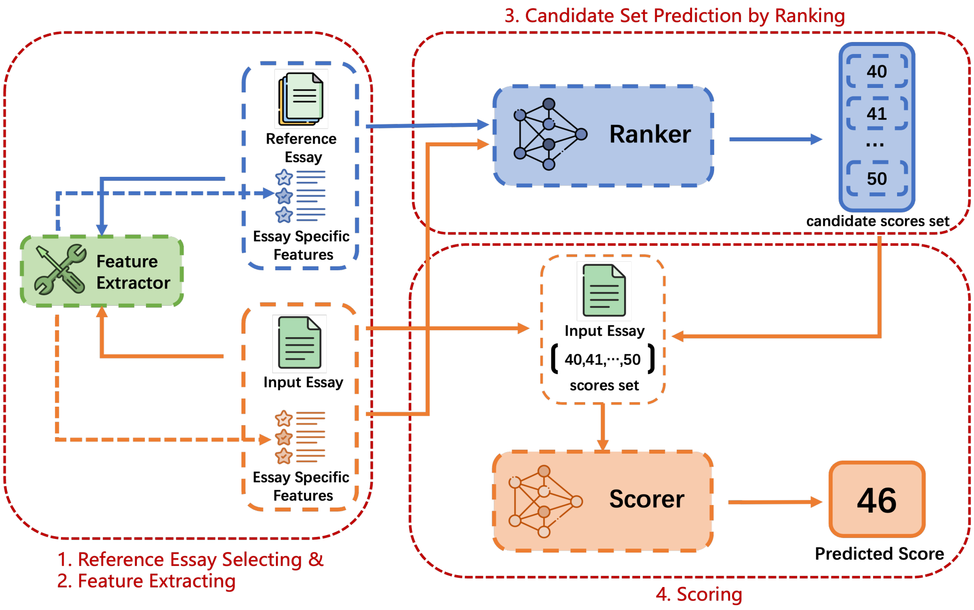

Figure 1: The overall architecture of RTS is illustrated in the figure. Excluding the training process, the method is divided into the following four steps: (1) Select reference essays. (2) use the feature extractor to identify the features of the essays, and incorporate these features into the essay content. (3) Utilize the Ranker to obtain the candidate score set of the current essay through pairwise ranking. (4) Feed the candidate score set, along with the essay, into the Scorer to generate final score.

However, it is unfortunate that there is a scarcity of research on Chinese AES based on LLMs, which is also the direction we are striving towards.

3 Method

The supervised fine-tuning-based AES method can be formalized as follows: given a set of essays $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ and a corresponding set of scores $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$, where each essay x_i is associated with a ground truth score y_i. Given a pre-trained model g_θ that is typically parameterized by θ. The goal is to train the base g_θ and obtain a new model $g_{\hat{\theta}}$ with $\hat{\theta}$ that predicts a score $\hat{y}_i = g_{\hat{\theta}}(x_i)$, making $\hat{y}_i$ close to y_i.

The RTS method divides the scoring process into two steps: (1) Training a pairwise ranking model (Ranker) to generate candidate score sets for target essays. (2) Training a scoring model (Scorer) to predict the real scores. The architecture of RTS is illustrated in Figure 1.

3.1 Pairwise Ranking

The task for the Ranker is as follows: given a target essay and a reference essay, the model outputs the index of the essay that has the higher score; We repeat the process above and transform the ranking results into a candidate score set.

We employ supervised fine-tuning method on an LLM, allowing it to accurately evaluate the quality of essays through ranking. We design a four-step approach to train the Ranker's pairwise ranking capability and generate the candidate score set:

1. **Reference Essay Selecting**: For each prompt, a subset of reference essays is selected to facilitate pairwise comparisons.

2. **Features Extracting**: This includes linguistic features, structural features, and semantic features to effectively represent the essays.

3. **Fine-tuning Pairwise Ranker**: We fine-tune the model using feature-augmented pairwise data.

4. **Candidate Set Prediction by Ranking**: By comparing the target essay with the reference essays, the model predicts the candidate score set for the target essay.

In Step 1, we select different reference scores for different prompts. Specifically, we adhere to the following two principles for selection:(1) the number of reference scores should not exceed 5, as exceeding this limit would increase inference costs. (2) when the number of scores is even, the two middle scores are selected; when the number is odd, the central score is selected. Afterwards, for each reference score, we randomly select **2 essays** as reference essays.

In Step 2, we utilize features related to the text to enhance the model's understanding of the essay. We first extract various types of feature for both Chinese and English data. For the ASAP dataset, we use the hand-crafted features proposed by (Ridley et al., 2020). For the HSK dataset, we adopt the feature categories used by (Li et al., 2022) in their readability assessment study and extract features by ourselves. The specific feature categories are detailed in the Appendix A.

Afterwards, we employ **LibSVM** to select a subset of beneficial features. Specifically, we use it to perform simple predictions on pairs (f, y), where f represents features and y represents scores, the F-score is defined by (Chen and Lin, 2006) as:

$$F_i \equiv \frac{\left(\bar{f}_i^{(+)} - \bar{f}_i\right)^2 + \left(\bar{f}_i^{(-)} - \bar{f}_i\right)^2}{\frac{1}{n_+ - 1}\sum_{j=1}^{n_+}\left(f_{j,i}^{(+)} - \bar{f}_i^{(+)}\right)^2 + \frac{1}{n_- - 1}\sum_{j=1}^{n_-}\left(f_{j,i}^{(-)} - \bar{f}_i^{(-)}\right)^2} \quad (1)$$

where $\bar{f}_i, \bar{f}_i^{(+)}, \bar{f}_i^{(-)}$ are the average of the ith feature of the whole, positive, and negative data; $f_{j,i}^{(+)}$ is the ith feature of the jth positive instance, and $f_{j,i}^{(-)}$ is the ith feature of the jth negative instance. Then, we select the top 10 features with the highest F-score as the final set of features. We concatenate the content of each essay with its features to obtain a feature-augmented essay representation, which serves as the input in the future steps.

In Step 3, in a training set of size M, for each feature-enhanced essay, we randomly select k essays with different scores to form pairwise data, which is used to fine-tune the Ranker. The instruction used for fine-tuning on ASAP is shown as Figure 2. The instruction used on HSK is shown in Appendix B

In Step 4, we adopt a **"Binary Search Tree"** approach, inspired by (Zhuang et al., 2024). By using the Ranker, we compare the target essay with the reference essays to determine the candidate score set. The detailed process is illustrated in Figure 3. We arrange the reference essays in a BST structure and begin pairwise ranking from the score closest to the median. After each round, we use the result to guide the selection of the next score, ultimately obtaining a candidate score set represented by the

```
System
Imagine you are a teacher's assistant in a middle
school tasked with reviewing a 7th to 10th grade
student's essay. You have been given two students'
essays and the prompt the student responded to.
Please choose the better of the two essays.
User
[Prompt]
{{prompt}}
(end of [Prompt])
[Student Essay1]
{{content}}
{{features}}
(end of [Student Essay1])
[Student Essay2]
{{content}}
{{features}}
(end of [Student Essay2])
Assistant
{{better essay id}}
```

Figure 2: Instruction for fine-tuning the Ranker. Contents to be filled are highlighted in red.

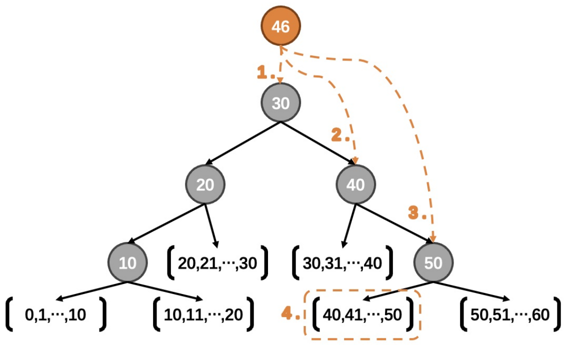

Figure 3: The BTS-like inference process.

leaf node. In special cases, when the Ranker determines that the target essay's score lies between two adjacent essays, we add an additional leaf node between the two reference essays to fix this issue as shown in Figure 4.

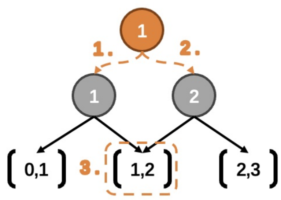

Figure 4: Another scenario of the BTS-like approach.

During each comparison with reference essays, we employ the following **Multi-Validation** method based on (Qin et al., 2023) to assess the difference. Given two essays e_1 and e_2, we define the comparison function $C(e_1, e_2)$ as follows:

$$C(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 \text{ is better than } e_2 \\ 0, & \text{if } e_2 \text{ is better than } e_1 \end{cases} \quad (2)$$

```
System
Imagine you are a teacher's assistant in a middle school tasked with reviewing a 7th to 10th grade student's essay. You have been given a student's essay and the prompt the student responded to.
User
[Prompt]
{{prompt}}
(end of [Prompt])
[Student Essay]
{{content}}
{{candidate score set}}
(end of [Student Essay])
Assistant
{{score}}
```

Figure 5: Instruction for fine-tuning the Scorer. Contents to be filled are highlighted in red.

In each round, we select a reference essay r_i and pair it with the target essay x to form the pair (x, r_i). By swapping the order of the two essays in the prompt, we obtain another pair (r_i, x). This process yields four comparison results:

$$\begin{cases} o_1 = C(x, r_1) \\ o_2 = C(r_1, x) \\ o_3 = C(x, r_2) \\ o_4 = C(r_2, x) \end{cases} \tag{3}$$

Define the statistics:

$$\begin{cases} S_{x>r_i} = o_1 + o_3 \\ S_{r_i>x} = o_2 + o_4 \end{cases} \tag{4}$$

where $S_{x>r_i}$ represents the number of times x is better than r_i, and $S_{r_i>x}$ represents the number of times r_i is better than x. The final result is defined as:

$$\text{result}(x, r_i) = \begin{cases} r_i{>}x, & S_{r_i>x} = 2 \land S_{x>r_i} < 2 \\ r_i{>}x, & S_{x>r_i} = 2 \land S_{r_i>x} < 2 \\ r_i{=}x, & \text{others} \end{cases} \tag{5}$$

3.2 Essay Scoring

We embed the candidate score set information into the data for scoring, fine-tuning the Scorer to endow the model with scoring capabilities. The instruction used for fine-tuning and evaluation in ASAP is as Figure 5. And instruction used in HSK is shown in Appendix B.

It is necessary to clarify that due to the overlap of the training data for the Scorer and the Ranker, theoretically, the accuracy of the candidate score set for fine-tuning of the training data is 100%. Consequently, we introduce some adjustment to the set to lower its accuracy to some extent, thereby achieving better effects.

4 Experimental Setup

4.1 LLMs

We conduct experiments using mainstream open-source LLMs for both Chinese and English tasks. For the Scorer, we select **Qwen2-7B-Instruct** (Yang et al., 2024) for the Chinese essay scoring task, and select two models of different sizes for English essay scoring: **LlaMA3.1-8B-Instruct** (Grattafiori et al., 2024) and **Mistral-NeMo-Instruct-2407** (MistralAI, 2024) to demonstrate the general applicability of our method. For the Ranker model, we select **Qwen2.5-1.5B-Instruct** (Yang et al., 2024) for Chinese pairwise ranking task, and select **LlaMA3.2-3B-Instruct** (Grattafiori et al., 2024) for English pairwise ranking task.

4.2 Datasets

We conduct experiments on both Chinese and English datasets.

For the Chinese data, we utilize the **HSK** (Hanyu Shuiping Kaoshi) dataset for the Chinese essay scoring task. The HSK dataset originates from the work of (Cheng, 2022), which comprises essay corpora collected from foreign candidates who took the advanced Chinese HSK examination between 1992 and 2005. After cleaning the flag for syntax errors in essays and removing essays with a score of 0 and those with insufficient word counts, we obtain a total of 10,329 essays. Finally, we select the 11 prompts with the largest number of essays for our experiments which contain 8,597 essays.

The **ASAP** (Automated Student Assessment Prize) dataset (Hamner et al., 2012) is famous in the field of English AES, which includes 12,978 essays written by students in grades 7 through 10. These essays are composed in response to 8 prompts covering a variety of genres and score ranges. More descriptions of the datasets are provided in Appendix C.

4.3 Evaluation Metric

We use **Quadratic Weighted Kappa (QWK)** to evaluate the discrepancy between predicted scores and gold scores. This metric is widely adopted in AES tasks for both Chinese and English essays (Taghipour and Ng, 2016; Ridley et al., 2020; He et al., 2022; Li et al., 2022).

When generating the candidate score set, we also use **accuracy** to evaluate the quality of the prediction of our "Binary Search Tree" approach.

4.4 Implementation Details

Reference Essay Selecting Following the previously specified rules, the selected reference scores are shown in Table 1. For each score, we randomly select **2 essays** as the reference essays for the current prompt.

Feature Extracting For both the HSK and ASAP datasets, we select the top 10 final features using LibSVM and F-score, which is shown in Appendix A.

Prompt	Range	Reference Score
HSK	40-100	50,60,70,80,90
ASAP1	2-12	5,9
ASAP2	1-6	3,4
ASAP3	0-3	1,2
ASAP4	0-3	1,2
ASAP5	0-4	2
ASAP6	0-4	2
ASAP7	0-30	5,10,15,20
ASAP8	0-60	10,20,30,40,50

Table 1: The score ranges and corresponding reference scores for both Chinese and English prompts are provided, where ASAP1-ASAP8 represent the prompt IDs in the ASAP dataset.

Pairwise Ranking Data We set $k = 5$ to generate pairwise data for fine-tuning the Ranker, resulting in a rank training dataset that is five times larger than the original scoring training dataset.

Scorer Performance through Candidate Score Calibration We reduce the accuracy of the Scorer training set according to the accuracy of Ranker's test set. We finally choose to reduce the accuracy of the Scorer training set by 15% compared to Ranker's test set.

4.5 Comparing Methods

We compare RTS with other excellent supervised method.

R^{2}BERT (Yang et al., 2020) Significant improvements are achieved by modifying the scoring loss to a combination of pairwise ranking and scoring losses.

NPCR (Xie et al., 2022) This is the state-of-the-art supervised prompt-specific AES method in the ASAP dataset. They utilized up to 50 reference essays to compare with the target essay, achieving excellent results. We also apply this method to the HSK dataset to compare its performance.

PAES (Ridley et al., 2020) A highly effective cross-prompt AES method that also incorporates features. The features we used for the ASAP dataset are adapted from this work.

Vanilla Fine-tuning the model directly without incorporating the candidate score set.

5 Results and Analysis

5.1 Main Results

The final experimental results are shown in Table 2 and Table 3. Overall, RTS is able to outperform the Vanilla method both in average QWK and QWK on almost all prompts, which shows that RTS has the enhancement effect not only on different datasets in different languages, but also on different LLMs.

Expanding on this, the improvement of HSK on average QWK is 1.9% ($74.6\% \rightarrow 76.5\%$), and the improvement of ASAP is 1.7% ($78.1\% \rightarrow 79.8\%$) and 1.1% ($78.3\% \rightarrow 79.4\%$) respectively, note that compared to the Vanilla method, RTS's improvement in average QWK is similar across datasets and models. In terms of each dataset, in HSK, RTS boosts ranged from 0.9% to 8.9%, with prompt 11 boosting the most by 8.9% ($64.3\% \rightarrow 73.2\%$). In the ASAP dataset, the boost ranges from 1.4% to 2.8%, with the largest boost being 2.8% ($72.4\% \rightarrow 75.2\%$) for LlaMA3.1-8B-Instruct on prompt3. All of these improvements indicate that the improvement effect of RTS is similar across different data and has good cross-language capabilities.

It is also worth noting that, on the HSK dataset, the RTS method also significantly outperforms the results of all the small models that perform extremely well on the ASAP dataset ($56.8\%, 65.1\% \rightarrow 74.6\%, 76.5\%$), which demonstrates that LLM has a very high potential for Chinese AES tasks.

However, if we look closely, we can see that except for prompt 6, 9 and 10 in HSK and prompt 7 in ASAP, where there is almost no improvement ($78.8\% \rightarrow 78.9\%$), prompt 2 decreases by 0.9% compared to Vanilla ($72.5\% \rightarrow 71.6\%$). We will analyze the reasons for this phenomenon in the next section with another set of experiments.

5.2 Upper Bound Analysis

Before verifying the RTS method, we first fine-tune Scorer with a candidate score set with 100% accuracy in order to see if our hypothesis is reasonable. We also validate the model by adding features to

Method	1	2	3	4	5	6	7	8	9	10	11	Avg
NPCR	0.435	0.650	0.541	0.574	0.657	0.586	0.501	0.529	0.609	0.630	0.540	0.568
PAES	0.450	**0.730**	0.690	0.679	0.658	0.663	0.751	0.662	0.702	0.720	0.458	0.651
Vanilla	0.625	0.725	0.774	0.696	0.812	0.788	0.854	0.758	0.754	0.776	0.643	0.746
RTS	**0.657**	0.716	**0.796**	**0.706**	**0.823**	**0.789**	**0.863**	**0.797**	**0.755**	**0.779**	**0.732**	**0.765**

Table 2: Performance of our method on the HSK dataset. The bolded data are the best performing results among all Models.The scorer model used by RTS is **Qwen2-7B-Instruct**.

Model	Method	1	2	3	4	5	6	7	8	Avg
R^2BERT	-	0.817	0.719	0.698	0.845	0.841	0.847	0.839	0.744	0.794
NPCR	-	0.856	0.750	0.756	0.851	0.847	0.858	0.838	0.779	0.817
LlaMA3.1-8B-Instruct	Vanilla	0.822	0.688	0.724	0.826	0.806	0.845	0.830	0.706	0.781
	RTS	**0.840**	**0.712**	**0.752**	**0.844**	**0.831**	**0.848**	0.830	**0.732**	**0.798**
Mistral-NeMo-Instruct-2407	Vanilla	0.823	0.688	0.705	0.836	0.801	0.838	**0.841**	0.728	0.783
	RTS	**0.835**	**0.710**	**0.730**	**0.840**	**0.821**	**0.839**	0.838	**0.740**	**0.794**

Table 3: Performance of our method on the ASAP dataset. The first row shows the NPCR method which is SOTA method on small models. the bolded data are the results where our method significantly outperforms Vanilla's method. The two LLMs used here are both their Instruct versions.

the Scorer in order to determine whether features are applicable in the RTS to where they are added to the Scorer. The result in the HSK dataset is shown in Figure 6.

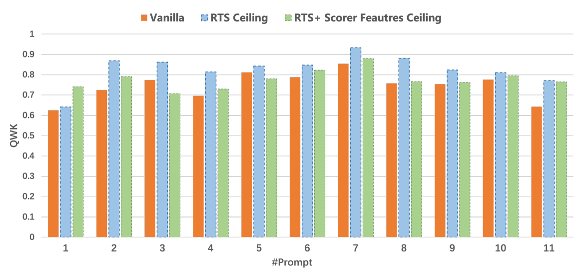

Figure 6: The results in the HSK dataset of two ceiling experiments are presented.

As shown in the figure, all prompts RTS have extremely high ceilings, so RTS methods are probably viable. However, because the accuracy of the candidate score set cannot reach the ideal state, it is difficult to reach the ceiling in practice. On the other hand, adding features to the Scorer downs the ceiling of the RTS method, which explains why we do not add features to the Scorer in the final RTS method.

5.3 Ablation Study of Features

Further, we explore the effects of incorporating features into different components of RTS, as shown in Table 4.

As can be seen in Table 4, RTS decreases by 1.4% (76.5% $\rightarrow$ 75.1%) after removing Ranker features. A feasible approach to addressing this issue is to determine the accuracy of pairwise rankings with and without features. The result of this experiment on HSK is shown in Figure 7.

Method	Avg
Vanilla	0.746
RTS	0.765
+ Scorer Features	0.719
- Ranker Features	0.751

Table 4: Results of adding features to scorer and removing features from ranker on HSK.

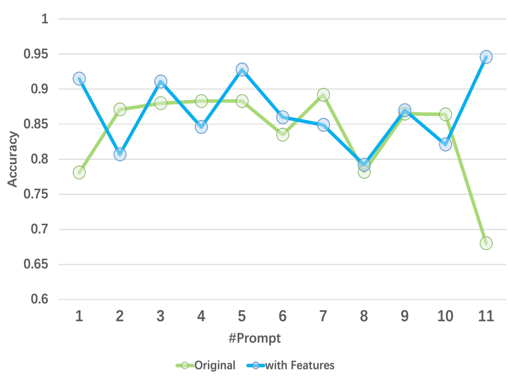

Figure 7: The impact of adding features on the Ranker's classification accuracy. The average accuracy rate after adding Features is 3.1% higher than that without adding features (83.7% $\rightarrow$ 86.8%).

From the perspective of average accuracy, Ranker's ranking ability is significantly improved after the addition of features, especially on some prompts. However, we can also clearly observe that the accuracy even decreases on four prompts, with prompt 2 decreasing the most significantly (87.1% $\rightarrow$ 80.7%). Not only does this shows that features is not facilitated in some prompt, but it also explains why RTS performance drops on prompt 2 in HSK dataset, which is observed in **5.1**.

5.4 Scorer Performance through Candidate Score Calibration

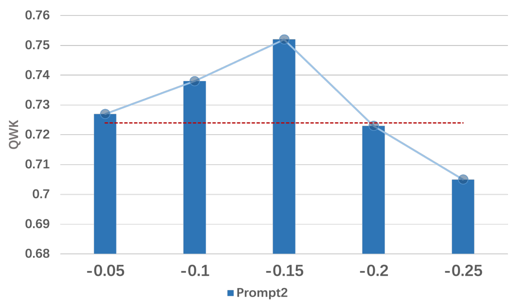

Figure 8: This is the result on Prompt 2 of the ASAP dataset. The x-axis represents the degree of adjustment. For example, "-0.1" indicates that the accuracy of the Ranker's test data is a, while the accuracy of the Scorer's candidate score set is "$a - 0.1$".

In the process of fine-tuning the Scorer, we observe that adjusting the accuracy of the candidate score set used in some of the fine-tuning data based on the accuracy of the Ranker, is able to improve the results of the Scorer. The results from the experiment, with different adjustment values, are shown in Figure 8. As illustrated, the Scorer's performance is optimal when the accuracy of the Ranker's test data differs by 0.15 from that of the Scorer's candidate score set.

5.5 Other Ranking and Scoring Combined Methods

We explore some other methods that can combine the ranking task and the scoring task on LLMs (Yang et al., 2020; Xie et al., 2022):

Scoring In Multiple Essays When we assume that the model can automatically learn the order among multiple essays from the history, we give the model 5 different essays at a time and let the model score them.

Simultaneous Generation of Scores And Rankings Based on the above assumption, we propose another method: generate both scores and rankings in all previous essays.

Both In One LLM Starting from the idea of Multi-Task, it is easy to think of a way to fine-tune the Scorer with pairwise ranking data. Therefore, we divide this method into two types of fine-tuning phases: first ranking and then scoring, and first scoring and then ranking.

The results of the above method compared to the RTS method on the HSK dataset are shown in Table 5. It can be clearly seen that the first two results prove that the assumption mentioned above is not true, and both methods have a lowering effect on the model. For the latter two methods, the S1R2 method has a better improvement, but it is still only 0.6% $(74.6\% \rightarrow 75.2\%)$ much less than the 1.9% $(74.6\% \rightarrow 76.5\%)$ boost of the RTS. The above results illustrate that of all the methods combining ranking and scoring, RTS is the one that performs best on LLMs.

Method	**Avg**
Vanilla	0.746
Scoring in 5 Essays	0.656
Simultaneous Generation	0.676
R1S2	0.509
S1R2	0.752
RTS	0.765

Table 5: Results of other methods on the HSK dataset. **R1S2** indicates ranking first, then scoring. **S1R2** indicates scoring first, then ranking

6 Conclusion

This study introduces RTS (Rank-Then-Score), a novel LLM-based fine-tuning method for Automated Essay Scoring (AES) across Chinese and English datasets. RTS combines two specialized LLMs: one fine-tuned for essay ranking and another fine-tuned for scoring, achieving superior improvements. Experiments show RTS significantly outperforms traditional Vanilla fine-tuning, particularly in Chinese dataset. Key findings include: (1) RTS has the best AES performance on LLMs; (2) Integrating features into the Ranker enhances quality discrimination more effectively than adding them to the Scorer; (3) RTS surpasses other ranking-scoring combinations on LLMs by enabling seamless integration with human grading standards. The method demonstrates exceptional cross-lingual adaptability and precision, offering a robust solution for scenarios requiring nuanced essay evaluation. This dual-model approach addresses subtle quality distinctions while maintaining alignment with manual assessment practices, marking a notable advancement in AES technology on LLMs.

Limitations

Firstly, our architecture comprises two LLMs. Although the Ranker employs a relatively smaller model, there is still room for optimization in the size of both models. Encouragingly, we experiment with using Qwen2.5-1.5B-Instruct as the Scorer, and on the HSK dataset, the Vanilla method still achieves an average QWK of 0.741. This demonstrates that our approach has the potential to perform well on even smaller LLMs. Such models can be more effectively utilized in practical applications.

Another issue that requires attention is the selection of reference essays. Although we achieve satisfactory results by randomly selecting reference essays, it is still necessary to explore whether different methods of selecting reference essays will significantly impact our approach.

Ethics Statement

Potential Risks Our method cannot guarantee fair evaluation, meaning that RTS may reinforce the LLMs' tendency to favor certain social groups in scoring. For example, the predicted results may assign higher scores to groups with specific L1 (first language) backgrounds compared to other groups. Additionally, the datasets we used (ASAP and HSK) may disproportionately represent certain demographic groups, potentially leading to biased conclusions.

Use of Scientific Artifact We utilize the open-source scikit-learn package (Pedregosa et al., 2011) to compute the Quadratic Weighted Kappa (QWK). For our experiments, we employ the ASAP dataset (Hamner et al., 2012) and the HSK dataset (Cheng, 2022), both of which are available for non-commercial research purposes. Both ASAP and HSK replace personally identifiable information in the essays with symbols. Features used in ASAP and the types of features referenced in HSK both originate from open-source code (Ridley et al., 2020; Li et al., 2022). The large language models used in this study, LlaMA 3 (Grattafiori et al., 2024), Mistral (MistralAI, 2024), and Qwen2 (Yang et al., 2024), are licensed under the LlaMA 3 Community license and Apache-2.0 license, respectively. Alllicenses permit their use for research purposes.

Computational Budget We utilize two NVIDIA A40 GPUs for model fine-tuning and a single NVIDIA A40 GPU for inference of each model, including Qwen2.5-1.5B-Instruct,Qwen2-7B-Instruct, LlaMA3.2-3B-Instruct, LlaMA3.1-8B-Instruct, and Mistral-NeMo-Instruct-2407. Each batch contains 8 samples. Fine-tuning the RTS method on both datasets take approximately 2 hours, while inference, including both the Ranker and Scorer, take a maximum of 12 seconds per sample. However, the inference time may vary depending on the model architecture and acceleration methods employed.

References

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv*.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the Use of Large Language Models for Reference-Free Text Quality Evaluation: An Empirical Study. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*. Association for Computational Linguistics.

Yi-Wei Chen and Chih-Jen Lin. 2006. Combining svms with various feature selection strategies. *Feature extraction: foundations and applications*, pages 315–324.

Yuan Chen and Xia Li. 2023. Pmaes: Prompt-mapping Contrastive Learning for Cross-prompt Automated Essay Scoring. In *Conference on Semantics in Text Processing (STEP)*, pages 1489–1503.

Yong Cheng. 2022. Analysis of csl writing quality based on grammatical richness. , (5):10–22.

Semire Dikli. 2006. An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1).

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 153–162. Association for Computational Linguistics.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, volume abs/1804.06898, pages 263–271. Association for Computational Linguistics.

Jiefu Gong, Xiao Hu, Wei Song, Ruiji Fu, Zhichao Sheng, Bo Zhu, Shijin Wang, and Ting Liu. 2021. Iflyea: A Chinese Essay Assessment System with Automated Rating, Review Generation, and Recommendation. In *Conference on Semantics in Text Processing (STEP)*, pages 240–248. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, and etc. Sravankumar. 2024. The Llama 3 Herd of Models. *arXiv*.

Ben Hamner, Jaison Morgan, lynnvandev, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring. https://kaggle.com/competitions/asap-aes. Kaggle.

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.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. *Large Language Models are Zero-Shot Rankers for Recommender Systems*. Springer Nature Switzerland.

Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing Large Language Models' Proficiency in Zero-shot Essay Scoring. In *Conference on Empirical Methods in Natural Language Processing*, volume abs/2404.04941, pages 181–198.

Shengjie Li and Vincent Ng. 2024. Conundrums in Cross-Prompt Automated Essay Scoring: Making Sense of the State of the Art. In *Annual Meeting of the Association for Computational Linguistics*, pages 7661–7681.

Wenbiao Li, Ziyang Wang, and Yunfang Wu. 2022. A Unified Neural Network Model for Readability Assessment with Feature Projection and Length-Balanced Loss. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7446–7457.

Wendi Li and Yixuan Li. 2024. Process reward model with q-value rankings. *Preprint*, arXiv:2410.11287.

MistralAI. 2024. Mistral-nemo. https://mistral.ai/en/news/mistral-nemo. Accessed: 2024.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Ron J. Weiss, J. Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *ArXiv*, abs/1201.0490.

Isaac Persing and Vincent Ng. 2013. Modeling Thesis Clarity in Student Essays. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 260–269.

Zhen Qin, R. Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *North American Chapter of the Association for Computational Linguistics*, volume abs/2306.17563.

Robert Ridley, Liang He, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated Cross-prompt Scoring of Essay Traits. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 13745–13753. Association for the Advancement of Artificial Intelligence (AAAI).

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.

Wei Song, Ziyao Song, Lizhen Liu, and Ruiji Fu. 2020a. Hierarchical Multi-task Learning for Organization Evaluation of Argumentative Student Essays. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3875–3881. International Joint Conferences on Artificial Intelligence Organization.

Wei Song, Kai Zhang, Ruiji Fu, Lizhen Liu, Ting Liu, and Miaomiao Cheng. 2020b. Multi-Stage Pretraining for Automated Chinese Essay Scoring. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6723–6733. Association for Computational Linguistics.

Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. Exploring LLM Prompting Strategies for Joint Essay Scoring and Feedback Generation. In *Workshop on Innovative Use of NLP for Building Educational Applications*, volume abs/2404.15845.

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. Association for Computational Linguistics.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024. From Words to Numbers: Your Large Language Model Is Secretly A Capable Regressor When Given In-Context Examples. *arXiv*, abs/2404.07544.

Changrong Xiao, Wenxing Ma, Qingping Song, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and

Qi Fu. 2024. Human-AI Collaborative Essay Scoring: A Dual-Process Framework with LLMs. *arXiv*.

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.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. *arXiv*, abs/2407.10671.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing Automated Essay Scoring Performance via Fine-tuning Pre-trained Language Models with Combination of Regression and Ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 180–189.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47. ACM.

A Types of Features

Idx	Dim	Feature Description
1	1	Total number of characters
2	1	Number of character types
3	1	Type Token Ratio (TTR)
4	1	Average number of strokes
5	1	Weighted average number of strokes
6	25	Number of characters with different strokes
7	25	Proportion of characters with different strokes
8	1	Average character frequency
9	1	Weighted average character frequency
10	1	Number of single characters
11	1	Proportion of single characters
12	1	Number of common characters
13	1	Proportion of common characters
14	1	Number of unregistered characters
15	1	Proportion of unregistered characters
16	1	Number of first-level characters
17	1	Proportion of first-level characters
18	1	Number of second-level characters
19	1	Proportion of second-level characters
20	1	Number of third-level characters
21	1	Proportion of third-level characters
22	1	Number of fourth-level characters
23	1	Proportion of fourth-level characters
24	1	Average character level

Table 6: Character features description.

Idx	Dim	Feature Description
1	1	Total number of sentences
2	1	Average characters in a sentence
3	1	Average words in a sentence
4	1	Maximum characters in a sentence
5	1	Maximum words in a sentence
6	1	Number of clauses
7	1	Average characters in a clause
8	1	Average words in a clause
9	1	Maximum characters in a clause
10	1	Maximum words in a clause
11	30	Sentence length distribution
12	1	Average syntax tree height
13	1	Maximum syntax tree height
14	1	Syntax tree height ≤ 5 ratio
15	1	Syntax tree height ≤ 10 ratio
16	1	Syntax tree height ≤ 15 ratio
17	1	Syntax tree height ≥ 16 ratio
18	14	Dependency distribution

Table 7: Sentence features description.

Idx	Dim	Feature Description
1	1	Total number of words
2	1	Number of word types
3	1	Type Token Ratio (TTR)
4	1	Average word length
5	1	Weighted average word length
6	1	Average word frequency
7	1	Weighted average word frequency
8	1	Number of single-character words
9	1	Proportion of single-character words
10	1	Number of two-character words
11	1	Proportion of two-character words
12	1	Number of three-character words
13	1	Proportion of three-character words
14	1	Number of four-character words
15	1	Proportion of four-character words
16	1	Number of multi-character words
17	1	Proportion of multi-character words
18	1	Number of idioms
19	1	Number of single words
20	1	Proportion of single words
21	1	Number of unregistered words
22	1	Proportion of unregistered words
23	1	Number of first-level words
24	1	Proportion of first-level words
25	1	Number of second-level words
26	1	Proportion of second-level words
27	1	Number of third-level words
28	1	Proportion of third-level words
29	1	Number of fourth-level words
30	1	Proportion of fourth-level words
31	1	Average word level
32	57	Number of words with different POS
33	57	Proportion of words with different POS

Table 8: Word features description.

Idx	Dim	Feature Description
1	1	Total number of paragraphs
2	1	Average characters in a paragraph
3	1	Average words in a paragraph
4	1	Maximum characters in a paragraph
5	1	Maximum words in a paragraph

Table 9: Paragraph features description.

Idx	Feature Name	Full Name
1	mean_word	Mean Word Length
2	word_var	Word Variance
3	mean_sent	Mean Sentence Length
4	sent_var	Sentence Variance
5	ess_char_len	Essential Character Length
6	word_count	Word Count
7	prep_comma	Preposition to Comma Ratio
8	unique_word	Unique Word Count
9	clause_per_s	Clauses per Sentence
10	mean_clause_l	Mean Clause Length
11	max_clause_in_s	Maximum Clauses in a Sentence
12	spelling_err	Spelling Error Count
13	sent_ave_depth	Sentence Average Depth
14	ave_leaf_depth	Average Leaf Depth
15	automated_readability	Automated Readability Index
16	linsear_write	Linsear Write Formula
17	stop_prop	Stopword Proportion
18	positive_sentence_prop	Positive Sentence Proportion
19	negative_sentence_prop	Negative Sentence Proportion
20	neutral_sentence_prop	Neutral Sentence Proportion
21	overall_positivity_score	Overall Positivity Score
22	overall_negativity_score	Overall Negativity Score

Table 10: Text Statistical Features and Their Full Names

Dataset	Features
HSK	Total Word Count, Character TTR (Type-Token Ratio), Word TTR, Proportion of Advanced Characters, Proportion of Advanced Words, Character-Level Weighted Score, Word-Level Weighted Score, Number of Sentences, Average Syntactic Tree Height, Maximum Syntactic Tree Height.
ASAP	Mean Word Length, Mean Sentence Length, Essay Character Length, Total Word Count, Number of Unique Words, Clauses per Sentence, Spelling Errors, Sentence Average Syntactic Depth, Automated Readability Index (ARI), Linsear Write Formula.

Table 11: Selected features on two datasets.

B Instructions

The following are the instructions for the Ranker and Scorer in RTS method for HSK, Vanilla method for HSK, Vanilla method for ASAP, and the two methods: Scoring in 5 essays and Simultaneous Generation.

System
你是一位经验丰富的中文教师，专门负责批改HSK留学生的中文作文。
我会给你两篇作文，分别叫做作文1和作文2，请你输出你认为更好的那一篇作文。你需要评估以下几个方面：
1.语法和拼写准确性
2.文章结构和逻辑性
3.词汇丰富度和使用恰当性
4.内容表达的清晰度和连贯性
5.是否与题目相关
输出格式：更好的作文是：作文n
User
[Student Essay1]
{{content}}
{{features}}
(end of [Student Essay1])
[Student Essay2]
{{content}}
{{features}}
(end of [Student Essay2])
Assistant
{{better essay id}}

Figure 9: Instruction for the Ranker in RTS for HSK. Contents to be filled are highlighted in red.

System
你是一位经验丰富的中文教师，专门负责批改HSK留学生的中文作文。
我会给你作文的分数候选集，你的任务是仔细阅读这篇作文，并根据HSK官方评分标准给出分数。你需要评估以下几个方面：
1.语法和拼写准确性
2.文章结构和逻辑性
3.词汇丰富度和使用恰当性
4.内容表达的清晰度和连贯性
5.是否符合题目要求和字数限制
请基于以上的评判标准对下面的作文进行评分,作文分数必须为5的倍数，区间为0分至100分。
输出格式：该文章的最终得分为：n分
User
[Prompt]
{{prompt}}
(end of [Prompt])
[Student Essay]
{{content}}
{{candidate score set}}
(end of [Student Essay])
Assistant
{{score}}

Figure 10: Instruction for the Scorer in RTS for HSK. Contents to be filled are highlighted in red.

System
你是一位经验丰富的中文教师，专门负责批改HSK留学生的中文作文。
我会给你作文的分数候选集，你的任务是仔细阅读这篇作文，并根据HSK官方评分标准给出分数。你需要评估以下几个方面：
1.语法和拼写准确性
2.文章结构和逻辑性
3.词汇丰富度和使用恰当性
4.内容表达的清晰度和连贯性
5.是否符合题目要求和字数限制
请基于以上的评判标准对下面的作文进行评分,作文分数必须为5的倍数，区间为0分至100分。
输出格式：该文章的最终得分为：n分
User
[Prompt]
{{prompt}}
(end of [Prompt])
[Student Essay]
{{content}}
(end of [Student Essay])
Assistant
{{score}}

Figure 11: Instruction for Vanilla for HSK. Contents to be filled are highlighted in red.

System
Imagine you are a teacher's assistant in a middle school tasked with reviewing a 7th to 10th grade student's essay. You have been given a student's essay and the prompt the student responded to.
User
[Prompt]
{{prompt}}
(end of [Prompt])
[Student Essay]
{{content}}
(end of [Student Essay])
Assistant
{{score}}

Figure 12: Instruction for Vanilla for ASAP. Contents to be filled are highlighted in red.

System
你是一位经验丰富的中文教师，专门负责批改HSK留学生的中文作文。
现在有5篇同一主题的作文，每一篇作文都有特定的编号，你的任务是仔细阅读每一篇作文，并根据HSK官方评分标准先为它们排序，再给出一个作文的分数。
你需要通过以下几个方面来进行排序和评分：
1.语法和拼写准确性
2.文章结构和逻辑性
3.词汇丰富度和使用恰当性
4.内容表达的清晰度和连贯性
5.是否符合题目要求和字数限制
请基于以上的评判标准先给出5篇作文的排名情况，再给出每一篇作文的分数。
排名情况输出为每篇作文和对应排名的字典，质量越高的作文排名越高；作文分数输出为每篇作文和对应分数的字典，作文分数必须为5的倍数，区间为0分至100分。
输出格式："排名:{'作文1':排名1,'作文2':排名2...'作文5':排名5}
分数:{'作文1':分数1,'作文2':分数2...'作文5':分数5}"
User
[Student Essay1]
{{content}}
(end of [Student Essay1])
[Student Essay2]
{{content}}
(end of [Student Essay2])
[Student Essay3]
{{content}}
(end of [Student Essay3])
[Student Essay4]
{{content}}
(end of [Student Essay4])
[Student Essay5]
{{content}}
(end of [Student Essay5])
Assistant
{{rank result}}

Figure 13: Instruction for the method of scoring in 5 essays. Contents to be filled are highlighted in red.

System
你是一位经验丰富的中文教师，专门负责批改HSK留学生的中文作文。
我会给你作文的分数候选集，你的任务是仔细阅读这篇作文，并根据HSK官方评分标准先判断这一篇作文在之前所有作文中的排名，再给出分数。你需要评估以下几个方面：
1.语法和拼写准确性
2.文章结构和逻辑性
3.词汇丰富度和使用恰当性
4.内容表达的清晰度和连贯性
5.是否符合题目要求和字数限制
请基于以上的评判标准对下面的作文进行评分,作文分数必须为5的倍数，区间为0分至100分。
输出格式：
该文章在之前所有的作文中排名：位次
该文章的最终得分为：n分
User
[Prompt]
{{prompt}}
(end of [Prompt])
[Student Essay]
{{content}}
(end of [Student Essay])
Assistant
{{rank and score}}

Figure 14: Instruction for the method of simultaneous generation. Contents to be filled are highlighted in red.

C Dataset Description

Idx	#Prompt	Num
1	The Impact of Smoking on Personal Health and Public Interest	1220
2	My Views on Gender-Specific Classes	340
3	A Job Application Letter	495
4	Green Food and Hunger	1402
5	Views on "Euthanasia"	655
6	Reflections on "Three Monks Have No Water to Drink"	894
7	The Person Who Influenced Me the Most	643
8	How to Address the "Generation Gap"	778
9	Parents as the First Teachers of Children	822
10	My Views on Pop Music	704
11	A Letter to My Parents	644
12	Athlete Salaries	36
13	The Harm of Silent Environments on the Human Body	92
14	The Joys and Struggles of Learning Chinese	198
15	One of My Holidays	294
16	Views on "Wives Returning Home"	12
17	My Childhood	183
18	The Ideal Way to Make Friends	228
19	My Father	121
20	How to Face Setbacks	267
21	Why I Learn Chinese	107
22	Gum and Environmental Sanitation	15
23	My Views on Divorce	67
24	My Favorite Book	42
25	On Effective Reading	70

Table 12: The prompts of the HSK dataset are displayed as shown above, with the first 11 prompts utilized for experimentation.

Dataset	Prompt	#Essay	Avg Len	Range	Diff
HSK	1	1220	355	40-100	5
	2	340	434	40-100	5
	3	495	353	40-100	5
	4	1402	360	40-100	5
	5	655	366	40-100	5
	6	894	365	40-100	5
	7	643	416	40-100	5
	8	778	391	40-100	5
	9	822	373	40-100	5
	10	704	365	40-100	5
	11	644	403	40-100	5
ASAP	1	1783	427	2-12	1
	2	1800	432	1-6	1
	3	1726	124	0-3	1
	4	1772	106	0-3	1
	5	1805	142	0-4	1
	6	1800	173	0-4	1
	7	1569	206	0-30	1
	8	723	725	0-60	1

Table 13: Statistics of two datasets. **#Essay** represents the number of essays. **Avg Len** represents the average number of words. **Range** represents the score range. **Diff** represents the common difference.

Prompt
吸烟对个人健康和公众利益的影响
Content
据说有一个城市出台一个规定，在公共场所边走边抽烟的人被罚款。这个消息让我不由地想起我的故乡——日本大阪市的一些情况。其实，大阪早就出台同样的规定，现在已经几年了。所以我亲眼看过这种规定的实际操作中会出现的一些问题。
抽烟的人经常自称“爱烟家”，他们不顾社会上有许多专家提醒吸烟的有害性，尤其是对呼吸道病患者和孕妇的影响，却说“我就要抽”，甚至说“我就不相信他们说的所谓‘有害性’”[BQ，]使人实在无可奈何。其实人们并不是要他们完全戒掉，而是要在公共场所内不吸烟。
当然，最好使这些“爱烟家”最后能戒掉烟。理论和事实都证明，吸烟对身体确实有害。{CJ+zhuy我}{CJ+sy认为，}我们应该更重视[D视]的是吸烟对本人的影响，还是对别人的影响？显然是后者更重要。所以一些城市开始着手，首先把吸烟对别人的影响、对环境的影响、对城市形象的影响{CJ-zxy的问题}解决好。个人吸烟不吸烟是另外一回事。
所以我们应该使“爱烟者”充分地理解这一道理。有时候还得耐心地等待他们，尽量地让他们也能下得了台。这样才能开始着手个人吸烟的问题{CD了}。
Score
80.0

Figure 15: The uncleaned sample essay from the HSK Dataset, which contains flags for syntax errors.

Prompt
吸烟对个人健康和公众利益的影响
Content
据说有一个城市出台一个规定，在公共场所边走边抽烟的人被罚款。这个消息让我不由地想起我的故乡——日本大阪市的一些情况。其实，大阪早就出台同样的规定，现在已经几年了。所以我亲眼看过这种规定的实际操作中会出现的一些问题。
抽烟的人经常自称“爱烟家”，他们不顾社会上有许多专家提醒吸烟的有害性，尤其是对呼吸道病患者和孕妇的影响，却说“我就要抽”，甚至说“我就不相信他们说的所谓‘有害性’”使人实在无可奈何。其实人们并不是要他们完全戒掉，而是要在公共场所内不吸烟。
当然，最好使这些“爱烟家”最后能戒掉烟。理论和事实都证明，吸烟对身体确实有害。我认为，我们应该更重视视的是吸烟对本人的影响，还是对别人的影响？显然是后者更重要。所以一些城市开始着手，首先把吸烟对别人的影响、对环境的影响、对城市形象的影响解决好。个人吸烟不吸烟是另外一回事。
所以我们应该使“爱烟者”充分地理解这一道理。有时候还得耐心地等待他们，尽量地让他们也能下得了台。这样才能开始着手个人吸烟的问题了。
Score
80.0

Figure 16: The cleaned sample essay from the HSK Dataset.