RepEval: Effective Text Evaluation with LLM Representation

Shuqian Sheng[1], Yi Xu[1], Tianhang Zhang[1], Zanwei Shen[1], Luoyi Fu[1]*,
Jiaxin Ding[1], Lei Zhou[1], Xiaoying Gan[1], Xinbing Wang[1], Chenghu Zhou[2]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]IGSNRR, Chinese Academy of Sciences, Beijing, China
{susisheng, yixu98, zhangtianhang, yiluofu}@sjtu.edu.cn

Abstract

The era of Large Language Models (LLMs) raises new demands for automatic evaluation metrics, which should be adaptable to various application scenarios while maintaining low cost and effectiveness. Traditional metrics for automatic text evaluation are often tailored to specific scenarios, while LLM-based evaluation metrics are costly, requiring fine-tuning or rely heavily on the generation capabilities of LLMs. Besides, previous LLM-based metrics ignore the fact that, within the space of LLM representations, there exist direction vectors that indicate the estimation of text quality. To this end, we introduce RepEval, a metric that leverages the projection of LLM representations for evaluation. Through simple prompt modifications, RepEval can easily transition to various tasks, requiring only minimal sample pairs for direction vector construction. Results on fourteen datasets across two evaluation tasks demonstrate the high effectiveness of our method, which exhibits a higher correlation with human judgments than previous methods, even in complex evaluation scenarios involving pair-wise selection under nuanced aspects. Our work underscores the richness of information regarding text quality embedded within LLM representations, offering insights for the development of new metrics.[1]

1 Introduction

Text evaluation is widely applied in the era of LLM, such as detecting harmful responses (Sun et al., 2023; Kim et al., 2024), identifying high-quality data for model training (Meta, 2024; Cai et al., 2024) and constructing preference data for model alignment (Nvidia, 2024; Bai et al., 2022; Lee et al., 2023). Such requirements pose significant challenges to automatic text evaluation metrics, as metrics must be adaptive to diverse evaluation tasks and achieve high-quality assessment while maintaining a low cost. However, traditional automatic evaluation metrics, such as BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020), are usually designed for specific tasks or criteria, making them difficult to transfer to new application scenarios. Also, their requirement for references and other inputs makes them infeasible in various evaluation contexts. LLM-based metrics offer a possible solution (Gao et al., 2023; Chiang and Lee), but such metrics may also encounter certain limitations. On the one hand, they rely heavily on the generation ability of LLM to adhere to predefined formats, which typically require more model parameters or fine-tuning, resulting in higher costs for inference and deployment. On the other hand, their assessment is frequently unsatisfactory, which does not align well with human judgments and exhibits unstable performance (Shen et al., 2023).

Fortunately, though language models may struggle to generate appropriate responses, their representations contain rich information related to correct answers, which could be extracted with neural network or other models (Zou et al., 2023). Imagine, when people are assessing a piece of text, they may have a clear sense of its quality yet struggle to quantify their impressions with a precise score. This implies that during evaluation, we can reduce the reliance on the generation capabilities of LLMs and instead focus on the meaningful information contained in their representations. By doing so, we can utilize models with fewer parameters, thereby avoiding excessive computational resource consumption while achieving better performance. The remaining questions are: Do representations of LLM really encapsulate information relevant to text quality? How can we effectively **extract and apply** this information to evaluation tasks?

In this study, we introduce **RepEval**, a metric utilizing the projection of LLM representation for custom evaluation. We explored the performance

* Luoyi Fu is the corresponding author.

[1] The project is publicly available for research purpose https://github.com/susisheng/RepEval

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7019–7033
November 12-16, 2024 ©2024 Association for Computational Linguistics

of RepEval in two scenarios: absolute evaluation and pair-wise evaluation. In **absolute evaluation**, which requires evaluation metrics to output scores as assessment, our intuition is that representations of high-quality and low-quality text exhibit distinct distributions. We validate that, in vector space, their projection in a specific direction characterizes the degree of variation in textual properties. In **pair-wise evaluation**, metrics need to select the better one out of the two inputs. To solve this problem, we construct a projection vector that measures the probability of whether the preceding sentence is better than the latter.

For absolute evaluation, experiments on three criteria with ten datasets show that our method has better correlations with human judgments than previous metrics, which is flexible and easy to extend to other applications. As to pair-wise evaluation, experiments on four tasks with custom criteria demonstrate that our method remains highly feasible in complex application scenarios, achieving excellent classification accuracy. Through visualization, we further demonstrate that a well-designed prompt can transfer the representation to different positions within the semantic space, thus facilitating evaluations based on diverse criteria. We also demonstrate that using PCA can produce nearly optimal projection vectors, and we explore the optimization strategy of RepEval, offering a reasonable scheme for representation creation.

In summary, the key contributions of this work are:

- We introduce the evaluation metric RepEval, surpassing previous metrics on nearly all tasks, even outperforming GPT-4 with much fewer model parameters.

- RepEval can easily adapt to new evaluation scenarios, requiring only a few samples for training, and obviating the need for extensive human annotations and LLM fine-tuning.

- RepEval offers insights for the introduction of new metrics, demonstrating that LLM representations contain decisive information about text quality inherently.

2 Related Work

Automatic evaluation metrics can be categorized into three types, reference-based, reference-free, and LLM-based metrics.

2.1 Reference-based metrics

Reference-based metrics measure the similarity between the hypothesis and one or multiple references, and a hypothesis more similar to the reference is considered to be better (Gehrmann et al., 2023). These metrics can be further classified into two types: n-gram-based and embedding-based. Popular n-gram-based metrics include BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005). Embedding-based metrics include BERTScore (Zhang et al., 2019) and MoverScore (Zhao et al., 2019). However, the requirement of human-written references limits their applications, as the creation of references is always a serious problem.

2.2 Reference-free metrics

Reference-free metrics instead require the source to generate the hypothesis in the Natural Language Generation(NLG) process. Their advantage lies in the independence of human-written references, which costs expensive manual preparation. Polular reference-free metrics include BARTScore (Yuan et al., 2021), UniEval (Zhong et al., 2022) and GPTScore (Fu et al., 2023). Compared to reference-based metrics, they often exhibit better performance and adaptability (Sheng et al., 2024). However, these metrics are mostly designed for specific application scenarios and criteria, making it challenging to effectively apply them to new tasks.

2.3 LLM-based metrics

In recent years, there is a new trend to utilize LLM in text evaluation. Relying on the powerful capabilities of LLM, these studies use few-shot or zero-shot methods to directly generate the assessment results (Gao et al., 2023; Chiang and Lee). To enhance model performance, some studies have trained models specifically for evaluation through fine-tuning (Kim et al., 2024). However, LLMs with better generation capability usually contain more parameters, which is costly for evaluation, while the outputs are often unsatisfactory (Shen et al., 2023). The method of fine-tuning is also time-consuming and expensive as well.

3 Preliminary

3.1 Standard Evaluation

A standard NLG process receives a source text src as input and outputs a text hyp based on certain requirements, which can be seen as a generation

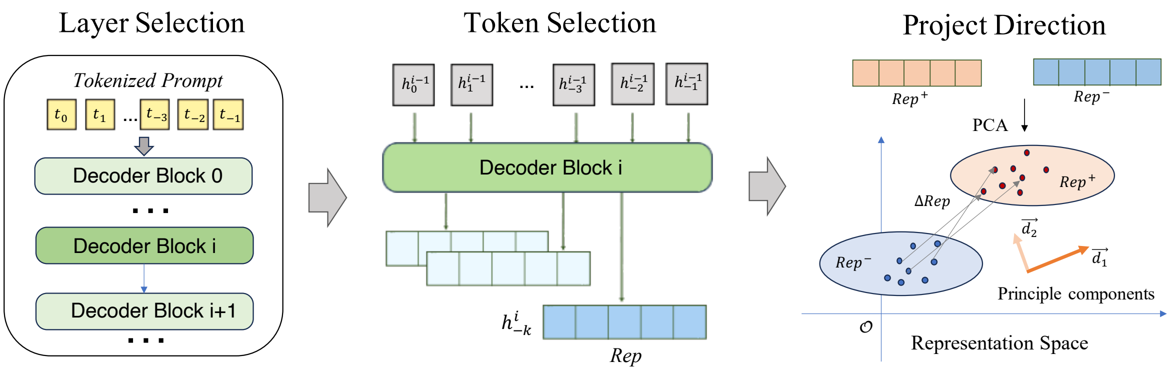

Figure 1: Pipeline of collecting representations with decoder-only LLM and constructing project direction.

function. In the same scenarios, an answer written by human experts can be viewed as a reference ref.

A common evaluation scenario is **absolute evaluation**, where an automatic metric function f is applied to evaluate a single hyp based on the specific criterion and output the evaluation result in the form of a score. This process can be described as Equation 1. We should note that src and ref are not necessary for all metrics. Also, for some metrics, the evaluation scores are irrelevant to the criterion.

$$score = f(criterion, hyp, src, ref) \tag{1}$$

Another scenario is **pair-wise evaluation**. Each time in the evaluation, a pair of hyp is provided, and metrics are required to choose the better one from two hyps based on specific criteria. Datasets in this scenario are all collected from complicated tasks, which have custom evaluation criteria for different samples. This scenario requires the model to clearly understand the evaluation criteria and accurately discern the quality difference between hyp pairs.

3.2 Meta-Evaluation

Human judgment is still the gold-standard approach to text evaluation (Yuan et al., 2021), which is also the basis of meta-evaluation methods used in this study.

In absolute evaluation tasks, the effectiveness of the metric is measured by the correlation between its scores and human judgments. The calculation is shown in 2.

$$\begin{aligned} correlation = \rho([&s_1, s_2, \dots, s_N], \\ &[h_1, h_2, \dots, h_N]) \end{aligned} \tag{2}$$

where s_i is the metric score of the i-th sample in a certain dataset, h_i is the relative human judgment, and ρ is the correlation function. In this study, we use Spearman Correlation (Spearman, 1987).

In pair-wise evaluation scenarios, we use the accuracy of detecting better hyp as the meta-evaluation method, as shown in Equation 3

$$accuracy = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \tag{3}$$

Where N is the number of sample pairs, $\hat{y}_i$ is the predicted index of better hyp, and y_i is the ground truth label.

3.3 Representations of LLM

In this study, representation refers to the hidden states of LLM with specific input texts. LLMs utilized in this study are in decoder-only architecture, typically comprising n decoder layers and a language modeling head with the hidden size of d. As shown in Figure 1, specifically, given a text input with s tokens, denote the output of the ith layer as h^i, where $i \in [0, n-1]$, and $h^i \in \mathbb{R}^{s \times d}$. We further denote the hidden states of kth token on layer i as h_k^i.

Suppose we choose the last kth token on the ith layer as the representation rep, we then have $rep = h_k^i$.

4 Methodology

4.1 Collecting Representation

Though RepEval does not rely on the generation ability of LLM, a good prompt helps integrate representations with more information related to the evaluation tasks. As defined in Section 3.1, to collect the representation rep, we can simply apply hyp as input. However, this is agnostic to the

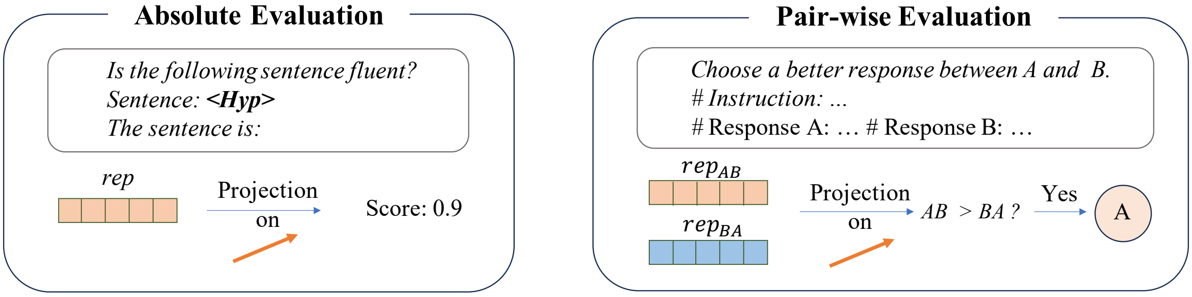

Figure 2: Evaluation process of absolute evaluation and pair-wise evaluation.

evaluation scenarios, and constructing task-related prompt templates helps improve the performance.

For absolute evaluation, we adopt three general criteria: fluency, consistency and coherence. In this scenario, the metric score represents how likely the hyp is a qualified text. We design and utilize the following prompt template.

Is the following Hyp <criterion_description>?
Hyp: <hyp>
Src: <src>
The sentence is

Here, "<hyp>" is filled by hyp to be evaluated, "<src>" is optional and only used in consistency evaluation, while "<criterion_description>" is different for each criterion. Please refer to the Appendix C.5 for more information. We also add a control group without the prompt template, using only hyp as inputs.

For pair-wise tasks, we need to compare the quality of two different hyp_A and hyp_B. Datasets related to the pair-wise evaluation are collected from complicated tasks, adopting different score criteria for each sample, such as harmlessness, honesty, etc. Follows Kim et al. (2024), here, "<instruction>" is the description of the task description, "<response 1>" and "<response 2>" could be filled by hyp_A and hyp_B, and "<score criterion>" is the evaluation requirement. More details could be found in the Appendix C.5

Instruction: <instruction>
Response A: <response 1>
Response B: <response 2>
Score Rubric: <score criterion>
Ans:

By exchanging the position of hyp_A and hyp_B in the prompt, we can obtain two reps, marked as rep_{AB} and rep_{BA}. These reps contain information about the following question: How likely is the previous sentence better than the latter? We will explain how to utilize this information in subsequent sections.

4.2 Project Direction

In the previous steps, we converted both evaluation tasks into **binary classification** problems by constructing proper prompts and obtained the relevant reps. Next, we need to figure out a specific projection direction $\vec{d}$, where the projection of rep on $\vec{d}$ represents the probability of the answer is "Yes".

We utilize Principal Component Analysis (PCA) to accomplish this task. In absolute evaluation, assume we have K high-quality texts, i.e. they receive high scores from human evaluators, and we denote their representations as rep^{+}. Similarly, we collect K low-quality texts and their representations, denoted as rep^{-}. For each pair of (rep^{+}, rep^{-}), their difference is given by $\Delta rep = rep^{+} - rep^{-}$ or $\Delta rep = rep^{-} - rep^{+}$. In pair-wise evaluation, consider K pairs of texts, where one sentence (A) is better than the other (B). According to the process described in section 4.1, since A is better than B, we denote the representation of rep_{AB} as rep^{+} and rep_{BA} as rep^{-}. Here, Δrep indicates the probability that 'A is better than B'."

As shown in Figure 1, Δreps represents the change in the likelihood of the answer being "Yes" instead of "No" in each sample, while their principal components should capture the overall variations. Therefore, with Δrep as inputs, assuming that we collect k main component vectors with PCA, as well as their importance score. Mark the ith vector and its importance as $\vec{d_i}$ and w_i. we can obtain the final $\vec{d}$ following Equation 4:

$$\vec{d} = \sum_{i=1}^{k} w_i \vec{d_i} \tag{4}$$

		RepEval				Baselines						
		Prompt			Hyp-only	LLM			Ref-free			Ref-based
		PCA(20)	PCA(5)	SVM	PCA(20)	GPT-4	GPT-3.5	Mistral-7b	GPTS	BARTS	UniE	BertS
FLU	BAGEL	**0.330**	0.236	**0.358**	0.060	0.325	0.222	0.156	0.152	0.241	0.309	0.247
	Newsroom	0.548	**0.565**	0.515	0.478	0.297	0.218	0.411	0.565	**0.596**	0.443	0.182
	SFHOT	**0.351**	0.345	**0.368**	0.108	0.305	0.178	0.238	0.135	0.164	0.312	0.164
	SFRES	**0.377**	0.370	**0.391**	0.021	0.352	0.289	0.272	0.229	0.226	0.332	0.183
	SummEval	**0.447**	0.424	0.419	0.324	0.245	0.120	0.285	0.288	0.285	**0.451**	0.194
	USR-P	0.360	**0.404**	0.363	0.306	**0.391**	0.310	0.288	-0.030	0.034	0.239	0.322
	USR-T	0.329	**0.368**	0.336	**0.402**	0.324	0.203	0.309	0.087	0.027	0.302	0.292
	WebNLG	**0.587**	0.534	**0.633**	0.268	0.503	0.409	0.401	0.072	0.330	0.521	0.499
CON	QAGS-C	0.541	0.561	0.453	NA	0.505	0.295	0.380	0.583	**0.680**	**0.618**	0.507
	QAGS-X	0.497	**0.550**	**0.524**	NA	0.457	0.315	0.185	0.081	0.159	0.387	-0.057
	SummEval	0.426	0.421	0.342	NA	**0.436**	0.269	0.210	0.355	0.334	**0.435**	0.200
COH	Newsroom	0.444	0.392	0.273	0.373	0.274	0.207	0.421	**0.595**	**0.623**	0.458	0.221
	SummEval	**0.534**	0.516	0.418	0.263	0.347	0.247	0.262	0.412	0.408	**0.592**	0.333

Table 1: **Absolute Evaluation Results.** Each row represents the **Spearman's correlations** of a metric with human judgments on absolute evaluation datasets. The **bold** scores represent the top two highest correlation results for each task on each criterion. Coherence, consistency, and fluency are written in abbreviations COH, CON, and FLU respectively. PCA(n) represents n samples are used in training. Hyp-only can not be used for consistency evaluation.

4.2.1 Collect Evaluation Results

As shown in Figure 2, we obtain the evaluation score following equation 5 in absolute evaluation.

$$score = rep^T \vec{d} \tag{5}$$

where rep is the representation of the hyp, $\vec{d}$ is the project direction vector, marked the probability of hyp been a qualified text.

In pair-wise evaluation, by switching the position of hyp_A and hyp_B, we obtain two reps, noted as rep_{AB} and rep_{BA}, respectively. Then the prediction result is:

$$prediction = \begin{cases} A, & rep_{AB}^T \vec{d} > rep_{BA}^T \vec{d} \\ B, & else \end{cases} \tag{6}$$

where $\vec{d}$ is the project direction, marking the probability of the first hyp better than the latter one.

4.3 Selection of layer and token

As shown in Figure 1, when constructing representations, there are many layers and tokens to choose from, and the optimal layer may depend on specific tasks and input. As we only utilize decoder-only LLM, which predicts the next token from left to right, the reps of the last few tokens contain the semantic information of the entire preceding text and are selected for application.

After projection vectors are collected, we test the performance of different tokens combined with different layers, and select the target token and layer with the best performance, i.e. with the highest human correlations or pair-wise accuracy, on the validation set, and apply it to the test set.

5 Experiments

5.1 Datasets

For absolute evaluation, we focus on three evaluation criteria: fluency, consistency and coherence, which are widely applied in NLG tasks. We utilize datasets from four tasks: Asset (Alva-Manchego et al., 2020) for simplification, SummEval (Fabbri et al., 2021) and Newsroom (Grusky et al., 2018) for summarization, WebNLG (Shimorina et al., 2019), SFRES, and SFHOT (Wen et al., 2015) for data-to-text, and USR-Persona and USR-Topic for dialogue (Mehri and Eskenazi, 2020).

For pair-wise evaluation, according to Kim et al. (2024), we utilize datasets HHH Alignment (Askell et al., 2021), MT Bench Human Judgment, Auto-J Eval (Li et al., 2023), and Preference Bench (Kim et al., 2024). All samples in these datasets contain a pair of hyps, instructions to generate the hyp, human judgments and relevant criteria. For both scenarios, all texts in datasets are written in English.

5.2 Baselines

For absolute evaluation, we utilize three reference-based metrics BERTScore (Zhang et al., 2019), along with three reference-free metrics: GPTScore (Fu et al., 2023), BARTScore (Yuan

Evaluator LM	HHH Alignment					MT Bench	Auto-J	Preference Bench
	Help.	Harm.	Hon.	Other	Total Avg.	w/o TIE	w/o TIE	Instance-wise
Llama2-Chat 7B	55.93	62.07	49.18	62.79	57.01	50.39	45.73	58.60
Llama2-Chat 13B	71.19	77.59	60.66	62.79	68.33	49.61	43.28	63.00
Llama2-Chat 70B	62.71	81.03	65.57	65.12	68.78	60.88	50.64	64.70
Mistral-Instruct-7B	59.32	68.97	63.93	81.40	67.42	63.82	60.94	79.40
Mixtral-Instruct-8x7B	83.05	87.93	67.21	69.77	77.38	71.42	73.50	84.00
Pair RM (0.4B)	84.75	84.48	80.33	90.70	84.62	59.00	59.05	81.80
Ultra RM (13B)	86.44	79.31	81.97	88.37	83.71	56.00	59.85	86.97
Auto-J (13B)	77.97	79.31	70.49	74.42	75.57	69.12	76.64	81.35
Prometheus-2-7B	76.27	87.93	73.77	76.74	78.73	67.25	73.80	**92.45**
Prometheus-2-8x7B	84.75	96.55	81.97	76.74	85.52	71.96	79.98	**90.65**
RepEval(pair5)	89.83	96.55	**95.08**	**100.00**	**95.00**	**79.90**	73.11	87.20
RepEval(pair20)	**93.22**	**100.00**	**98.36**	**100.00**	**97.74**	**80.39**	74.98	87.90
GPT-3.5-Turbo-0613	77.97	81.03	77.05	67.44	76.47	69.41	72.13	75.05
GPT-4-1106-Preview	89.83	96.55	91.80	83.72	90.95	**79.90**	**83.12**	85.50
Claude-3-Opus	**91.53**	**100.00**	91.80	95.35	94.57	77.65	**82.92**	89.85

Table 2: **Pair-wise Evaluation Results.** Each row represents the **accuracy** (%) of a metric on selecting better hyp based on specific criteria. The **bold** scores represent the top two highest accuracy results for each evaluation task. PCA(n) represents n samples are used in training.

et al., 2021), and UniEval (Zhong et al., 2022). Additionally, we employ the Mistral-7b model[2] and the ChatGPT API (gpt-3.5-turbo and gpt-4) provided by OpenAI to establish baselines by prompting LLMs for evaluation, following the approach of Shen et al. (2023). Baseline For pair-wise evaluation are direct generation results of different LLMs, referencing from (Kim et al., 2024). Please refer to Appendix C for more details about datasets and metrics.

5.3 Training Dataset

We utilize Asset and GCDC for absolute evaluation. Asset belongs to the simplification task, while GCDC is a real-world text dataset specifically created for coherence evaluation, both unrelated to other datasets in this work. Please refer to the Appendix for how we select positive samples and negative samples to construct rep^+ and rep^-.

Since the criteria and application scenarios of pair-wise datasets differ greatly from each other, they can be regarded as unrelated external data. Therefore, for the evaluation of MT Bench Human Judgment, Auto-J Eval, and Preference Bench, we utilize HHH Alignment to construct a training set. For the evaluation of HHH Alignment, we utilize the MT Bench as training data.

5.4 Absolute Evaluation

Following the description in previous sections, the correlations between human judgments and scores generated by each metric are presented in Table 1.

We observe that RepEval outperforms existing metrics on almost all datasets, even surpassing the performance of GPT-4. With just five text pairs, the PCA method surpasses all baseline metrics on half of the datasets, and with 20 pairs, it achieves a top-two performance on seven datasets, similar to the results obtained by SVM. Considering that the training of SVM requires much more samples to achieve similarly good results, PCA significantly reduces the manual cost of constructing samples while maintaining relatively good performance. The Hyp-only experiment's outcome indicates that even without the addition of a prompt template, the embeddings in LLM contain information related to evaluation criteria such as fluency and coherence. Another notable point is that RepEval's performance is evidently better than directly prompting Mistral-7b for evaluation, indicating that even when LLM struggles to generate a satisfying response, their representations can still convey valuable information.

In summary, the projection of $reps$ can efficiently extract information related to the text quality on the desired evaluation criterion of hyp with a few samples. Therefore, in most cases, there's no need to employ more complex models like SVM. Additionally, RepEval only requires hyp as input, whereas traditional metrics depend on src or ref.

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

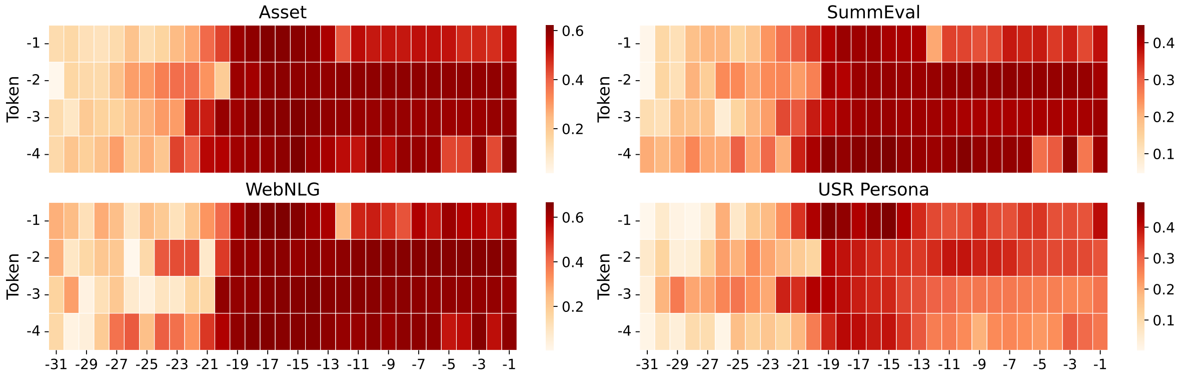

Figure 3: Correlation results for the absolute evaluation of fluency using RepEval with different token and position selections. Layer and token counts are in reverse order, measuring the distance from the output. For instance, layer=-1 represents the last layer closest to the output.

Compared with directly prompting LLMs like GPT-4, it exhibits better performance while maintaining a relatively low computational and time cost.

5.5 Pair-wise Evaluation

The accuracy of each method in pair-wise evaluation is presented in Table 2. We can observe that despite the varying generation tasks and evaluation criteria for each sample, RepEval still achieves high accuracy in selecting the better hyp. Compared to the generation results of vanilla Mistral-7b, the improvement of RepEval in pair-wise evaluation further validates that, failing to generate a good response does not mean that LLM doesn't know the answers, as reps already contain clear directions pointing towards the correct classification within the semantic space. Moreover, RepEval only adopts general LLM that has not been fine-tuned on evaluation tasks. Compared to PROMETHEUS, which is a text evaluation LLM fine-tuned with millions of data, our method saves the expensive cost of training, while maintaining relatively good or better performance. At the same time, by using only a 7b model, RepEval is still comparable to or even surpasses LLMs like GPT-4.

The above experimental results demonstrate that when there is no need to explain the judgment results, RepEval is highly competitive and can accurately make pair-wise selections. By only using a general LLM for inference, RepEval eliminates the high costs associated with pre-training. Additionally, since the optimal layers often reside in the middle layers, it reduces both inference time and computational costs by not requiring the inference of all parameters.

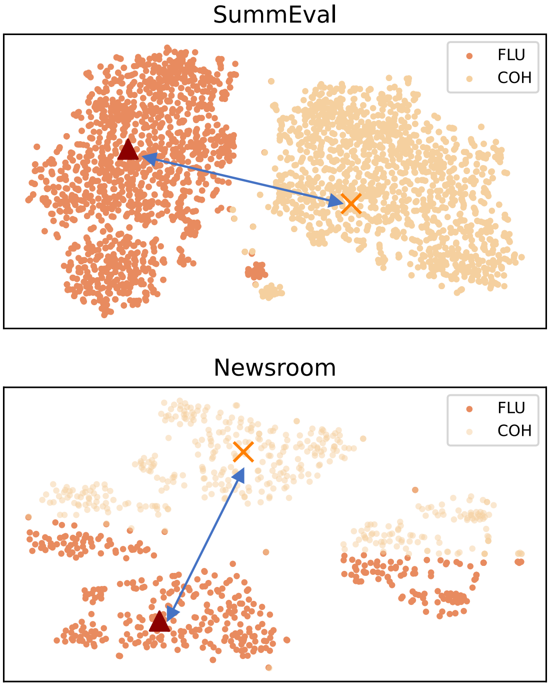

Figure 4: The t-SNE visualization of reps shows the results of dimensionality reduction. The triangles and X on each figure represent the reps of the same sample obtained using different prompts.

5.6 How prompt influence *reps*?

The design of the prompt is an important step when applying RepEval for evaluation. Especially in absolute evaluation, when we need to evaluate different aspects of the same sample, we need to use different prompt templates to obtain the corresponding rep. However, what role do these prompts play? Do they truly distinguish between application scenarios? The previous experiments did not provide an answer to this question.

In this section, we utilize t-SNE for the visualiza-

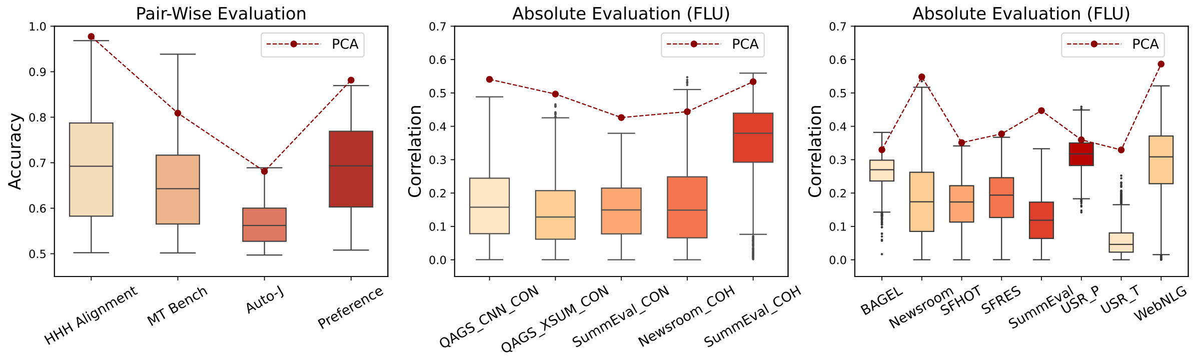

Figure 5: **Random Test Results** Box plots represent meta-evaluation results corresponding to random vectors v, while the scatter points in the figure represent the results corresponding to direction vector d obtained through PCA. For pair-wise evaluation, the y-axis starts at 0.5, which is the expected accuracy of random guessing.

tion of $reps$. We choose SummEval and Newsroom for this experiment, as they include evaluation results for two criteria: fluency and coherence. We collected $reps$ obtained from the two prompt templates and visualized their distribution using t-SNE, which is shown in Figure 4.

It can be seen that representations collected from different prompt templates exhibit different distributions and can be clearly separated from each other. This indicates that the prompts successfully transfer hyp to different positions within the semantic space, enabling the construction of the corresponding project direction in the transformed space and providing relevant assessments of the target criterion.

5.7 Selection of Token and Layer

To better utilize RepEval, in this section, we explore the performance of RepEval with different layers and token selections. Limited by space, we take fluency on absolute evaluation as an example and select four datasets from four tasks. All experiments follow the settings described in Section 5. The results are in Figure 3.

The results show that, surprisingly, the last token is not always the best one. Moreover, the correlation scores increase sharply in the middle layers and achieve the best result. A possible explanation could be that $reps$ collected from middle layers contain more information relevant to the current context. Comparatively, $reps$ from the last layers are more useful to the next token prediction.

This provides us with the following suggestions for improving RepEval. Firstly, we can opt for the token in the last second or third position, instead of the last one token. Secondly, choose embeddings from the second half of the layers. The layer should be far enough from the input to ensure that sufficient information is encoded.

5.8 A Good Projection or Not?

Previous experiments show that PCA works effectively in identifying a suitable projection vector, surpassing other non-linear methods such as SVM. However, it remains uncertain whether PCA identifies the "best" projection. To address this question, we conduct the following experiments.

We randomly generated 2000 vectors $\vec{v}$ with the same shape as the vector $\vec{d}$ obtained by PCA in Section 4.2. We then collect scores using the process outlined in Section 4.2.1, replacing $\vec{d}$ with $\vec{v}$ The selection of token and layer positions followed the settings of PCA outlined in Section 4. The distribution of meta-evaluation results is shown in Figure 5.

We observe that $\vec{d}$ obtained through PCA is a relatively optimal result. Compared to random vectors, it achieves nearly the highest correlation scores in absolute evaluation, as well as the highest accuracy scores in pair-wise evaluation. This indicates that if $reps$ contains related task information and that there exist projection vectors $\vec{d}$ characterizing the direction of variation in text quality, PCA can efficiently help researchers find the target $\vec{d}$, and be applied for evaluation.

6 Conclusion

We introduce RepEval, an evaluation metric utilizing the projection of LLM representations to obtain evaluation results, which exhibits a stronger correlation with human judgments in absolute evaluation, as well as higher accuracy in pair-wise selection than previous methods. RepEval is flexible and is easy to transfer to other evaluation scenarios, requiring only a few sample pairs for training, while avoiding the usage of LLMs with a large number of parameters such as GPT-4. We also provide suggestions on the proper application of RepEval, such as the selection of tokens and layers. Our work provides insights into the development of new metrics.

Limitations

In this study, the language is restricted to English. Further research is necessary to validate the identified performance across a broader spectrum of tasks and languages.

The analysis in this study is primarily driven by experimental data, and we acknowledge the absence of a more comprehensive mathematical explanation of the underlying mechanisms of RepEval. Additionally, our evaluation relies solely on correlation and accuracy as measurement methods. A more detailed analysis is left for future work.

Acknowlegement

This work was supported by NSF China (No. 61960206002,62020106005, 42050105, 62061146002), Shanghai Pilot Program for Basic Research - Shanghai Jiao Tong University

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631. Association for Computational Linguistics.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12. Citeseer.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research*, 77:103–166.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of NAACL-HLT*, pages 708–719.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Alice Lai and Joel Tetreault. 2018. Discourse coherence in the wild: A dataset, evaluation and methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223, Melbourne, Australia. Association for Computational Linguistics.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

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

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1552–1561, Uppsala, Sweden. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Nvidia. 2024. Nemotron-4 340b technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thomas Scialom and Felix Hill. 2021. Beametrics: A benchmark for language generation evaluation evaluation. *ArXiv*, abs/2110.09147.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4215–4233, Singapore. Association for Computational Linguistics.

Shuqian Sheng, Yi Xu, Luoyi Fu, Jiaxin Ding, Lei Zhou, Xinbing Wang, and Chenghu Zhou. 2024. Is reference necessary in the evaluation of NLG systems? when and where? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8580–8596, Mexico City, Mexico. Association for Computational Linguistics.

Anastasia Shimorina, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini. 2019. Webnlg challenge: Human evaluation results.

C. Spearman. 1987. The proof and measurement of association between two things. by c. spearman, 1904. *The American journal of psychology*, 100 3-4:441–71.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models. *arXiv preprint arXiv:2304.10436*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015.

Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Experiment Settings

When evaluating fluency and consistency, we construct the training dataset using Asset. For coherence evaluation, we utilize GCDC. During the training of the PCA model, the number of training pairs is set to 5 and 20. Additionally, we employ the SVM model for comparison with the PCA method, using 100 pairs for SVM training. As SVM needs more training data, during construction, we ensure the distinctiveness of each pair, though some pairs may contain the same good or bad text. No repeated data is contained in the training set of PCA.

We collected representations with Mistral-7b following the process described in Section 4.1. We employ the Sklearn implementation of PCA and SVM. For SVM, the kernel is set as Radial Basis Function (RBF), gamma = $1/d$, and the regularization parameter $C = 1$. We utilized Mistral-7b to generate representations using a single NVIDIA GeForce RTX 3090. The training of PCA and SVM models was performed on a CPU. More experiment details can be found in Appendix C.

B Evaluation Criteria

Coherence In accordance with Dang (2005), coherence evaluates whether models generate a well-structured and organized text body that aligns with the given task, steering clear of a mere compilation of related information.

Consistency Consistency, as per Honovich et al. (2022), assesses whether all factual information in the output text corresponds with the content provided in the input.

Fluency Fluency, as defined by Kann et al. (2018), gauges the natural perception of a sentence by humans. In certain instances, fluency is also referred to as naturalness, grammaticality, or readability.

C Experiments

C.1 Datasets

C.1.1 Absolute Evaluation

ASSET ASSET is a dataset created for the tuning and evaluation of sentence simplification models (Alva-Manchego et al., 2020). In this research, we use the human rating corpus, which contains 100 pairs of original sentences and system simplification as well as the human evaluation results for the system output. For each pair, the rating is done by 15 crowd-sourced workers from 3 aspects: fluency, adequacy, and simplicity.

BAGEL BAGEL features annotations on data-to-text tasks gathered from a dialogue system, with human annotations covering informativeness and naturalness, according to Mairesse et al. (2010). In this context, informativeness is compared with the gold standard, differing from our defined usage. However, for our purposes, we solely utilize the judgment results related to naturalness.

GCDC GCDC is created with real-world texts, which is designed for the development of discourse coherence algorithms (Lai and Tetreault, 2018). Each sample in GCDC contains three evaluation scores of coherence on a 3-point scale from 1 (low coherence) to 3 (high coherence).

NEWSROOM NEWSROOM gathers 60 articles along with summarization outcomes from 7 models, featuring human-written summaries as references, as documented by Grusky et al. (2018). The evaluation encompasses coherence, fluency, relevance, and informativeness.

QAGS QAGS encompasses reference texts and annotation results focused on consistency in the context of the summarization task, as outlined by Wang et al. (2020). The approach involves collecting three annotations for each sentence in a generated summary, utilizing a majority vote strategy to determine a consistency score. The final score is obtained by calculating the mean value across all sentences.

SFHOT and SFRES SFHOT and SFRES deliver evaluation results for the data-to-text task, incorporating annotations of naturalness and informativeness, as detailed by Wen et al. (2015). In this context, informativeness gauges the consistent degree between sources and hypotheses. This dataset is utilized for analyzing consistency, while naturalness serves as a proxy for fluency.

SummEval SummEval offers a compilation of summarization outcomes produced by language models, as detailed by Fabbri et al. (2021). These models undergo training on the CNN/DailyMail datasets, as described by Hermann et al. (2015), along with their corresponding reference texts. Each generated summary in the dataset includes score results from both expert annotators and crowd-workers, covering four dimensions: coherence, consistency, fluency, and informativeness.

USR The USR dataset offers evaluation results for the dialogue task across five aspects: fluency, coherence, engagingness, groundedness, and understandability. In alignment with the rephrasing strategy outlined by Zhong et al. (2022), the original aspects "maintains context" and "natural" are renamed as "coherence" and "fluency," respectively.

WebNLG WebNLG includes human evaluation results from the 2017 WebNLG Challenge, which focuses on the data-to-text task, as described by Shimorina et al. (2019). The candidate text undergoes evaluation based on three aspects: fluency, grammar, and semantics. In this context, fluency assesses whether a text is smooth and natural, and the fluency score is employed for experimentation purposes.

Features contained in each absolute evaluation dataset are listed in Table 3. With the exception of GCDC, all datasets include *src*.

	COH	**CON**	**FLU**	**REF**
summarization				
-Newsroom	✓		✓	✓
-QAGS		✓		✓
-SummEval	✓	✓	✓	✓
data-to-text				
-BAGEL			✓	✓
-SFHOT		✓	✓	✓
-SFRES		✓	✓	✓
-WebNLG			✓	✓
dialogue				
-USR-Persona	✓		✓	✓
-USR-Topical	✓		✓	✓
simplication				
-Asset			✓	
other				
-GCDC	✓			

Table 3: Datasets and available features.

C.1.2 Pair-wise Evaluation

HHH Alignment HHH Alignment contains the evaluation result based on four criteria: helpfulness, harmlessness, honesty, and other, as well as the relevant 221 response pairs judged by human evaluators (Askell et al., 2021).

MT Bench MT-bench consists of a series of open-ended questions that evaluate a chatbot's multi-turn conversational and instruction-following ability, which collect 3,360 response pairs based on 80 prompts, as well as judgment from human evaluators (Zheng et al., 2024).

Auto-J A dataset constructed with massive real-world scenarios with human evaluation judgments, consisting of 58 prompts and 1,392 response pairs (Li et al., 2023).

Preference Bench The preference bench contains 2000 response pairs, which are constructed based on 200 prompts and 200 evaluation criteria, as well as human judgments (Kim et al., 2024).

C.1.3 Resources

The resources of all datasets we used are listed as follows.

- Newsroom, SummEval, QAGS_cnn, QAGS_XSUM, SFHOT, SFRES are downloaded from source provided by Yuan et al. (2021). The related URL is `https://github.com/neulab/BARTScore`.

- Asset and WebNLG is downloaded from source provided by Scialom and Hill (2021). The related URL is `https://github.com/ThomasScialom/BEAMetrics`. We delete empty reference sentences before applying.

- USR_Topical and USR_Persona are created by Mehri and Eskenazi (2020). The related URL is `https://github.com/shikib/usr`.

- GCDC is created by Lai and Tetreault (2018), and the URL is `https://github.com/aylai/GCDC-corpus`.

- HHH Alignment, MT Bench, Auto-J, and Preference Bench are downloaded from source provided by Kim et al. (2024). The related URL is `https://github.com/prometheus-eval/prometheus-eval`.

C.2 Implement of Baselines

- BARTScore is downloaded from `https://github.com/neulab/BARTScore`. We use the faithfulness-based variant based on "facebook/bart-large-cnn"[3] checkpoint (Lewis et al., 2020).

- BERTScore is downloaded from `https://github.com/Tiiiger/bert_score`. We use the F1 score calculated based on checkpoint "deberta-xlarge-mnli"[4] (He et al., 2021).

[3] `https://huggingface.co/facebook/bart-large-cnn`

[4] `https://huggingface.co/microsoft/deberta-xlarge-mnli`

- GPTScore is downloaded from https://github.com/jinlanfu/GPTScore and we use the checkpoint "gpt2-large"[5] (Radford et al., 2019).

- UniEval is downloaded from https://github.com/maszhongming/UniEval. We use the "summarization" variant developed based on checkpoint "MingZhong/unieval-sum"[6] (Zhong et al., 2022).

- For metric BLEU and Meteor, we use the implementation provided by the python package NLTK (Bird et al., 2009).

C.3 SVM

We also add experiments with the Support Vector Machine (SVM) for comparison. With representation rep as inputs, the SVM method involves training a binary classifier on good-bad text pairs, and we use the probability of a text belonging to good text as the score result. To be specific, consider a specific text, denote the predicted probability of being good text as p_1, the predicted probability of being bad text as p_0, and the score satisfies

$$score = p_1/(p_0 + p_1) = p_1 \tag{7}$$

For each pair, we randomly select one from the good text and another from the bad text.

Dataset	Range	Low	High
Asset	[1, 100]	1	90
GCDC	[1,3]	1	3

Table 4: Score range of dataset Asset and GCDC.

C.4 Selection of Token and Layer

Here we present the optimal layer and token selections for different RepEval settings and the SVM method, where k represents the number of components of PCA.

C.5 Prompt Template of RepEval

As described in Section 4.1, the prompt templates of RepEval are listed as follows.

criterion	model	pairs	prompt	k	layer	token
FLU	PCA	20	yes	4	-15	-4
	PCA	5	yes	4	-15	-2
	PCA	20	no	3	-21	-1
	SVM	100	yes	-	-2	-2
CON	PCA	20	yes	3	-16	-2
	PCA	5	yes	3	-15	-2
	SVM	100	yes	-	-2	-1
COH	PCA	20	yes	4	-9	-2
	PCA	5	yes	2	-1	-2
	PCA	20	no	3	-1	-2
	SVM	100	yes	-	-1	-3

Table 5: Selection of token and layer in absolute evaluation. Where k is the number of main components when using PCA.

model	pairs	k	layer	token
PCA	5	1	-13	-1
PCA	20	1	-2	-1

Table 6: Selection of token and layer in pair-wise evaluation. Where k is the number of main components when using PCA.

C.5.1 Absolute Evaluation

For all absolute evaluation, we use the same prompt template.

> *Is the following Hyp <criterion_description>?*
> *Hyp: <hyp>*
> *Src: <src>*
> *The sentence is*

Apart from the inputs of src in consistency evaluation, we only change the <criterion_description> in the template, and please refer to Table 7 for details.

criterion	criterion description
fluency	fluent
coherence	coherent
consistency	consistent with Src

Table 7: Criterion description for each criterion in absolute evaluation.

C.5.2 Pair-wise Evaluation

Refer to the prompt design of Kim et al. (2024), we use the following prompt template for all pair-wise evaluation. Here, for pairs from different datasets, the score rubric should also be chanted to the related one.

[5] https://huggingface.co/gpt2-large

[6] https://huggingface.co/MingZhong/unieval-sum

> ###Task Description: An instruction (might include an Input inside it), a response to evaluate, and a scoring rubric representing evaluation criteria are given.
> Choose a better response between Response A and Response B. You should refer to the scoring rubric.
> ###Instruction: You are a fair judge assistant assigned to deliver insightful feedback that compares individual performances, highlighting how each stands relative to others within the same cohort.
> ###Response A: <hyp_1>
> ###Response B: <hyp_2>
> ###Score Rubric: <score_rubric>
> ###Ans: """

C.6 Prompt of LLM-based Absolute Evaluation

In this study, we use the gpt-3.5-turbo, gpt-4 API, and mistral-7b for a zero-shot baseline. Following the designs of Shen et al. (2023), the prompts we utilized for each criterion are listed as follows.

C.6.1 Absolute Evaluation of Fluency

> Score the following sentence with respect to fluency with one to five stars, where one star means "disfluency" and five stars means "perfect fluency". Note that fluency measures the quality of individual sentences, whether are they well-written and grammatically correct. Consider the quality of individual sentences.
> Summary: <hyp>
> Stars:

C.6.2 Absolute Evaluation of Coherence

> Score the following text with respect to coherence with one to five stars, where one star means "incoherence" and five stars means "perfect coherence". Note that coherence measures the quality of all sentences collectively, to the fit together and sound naturally. Consider the quality of the sentences as a whole and just output an overall score and no more other.
> Summary: <hyp>
> Stars:

C.6.3 Absolute Evaluation of Consistency

> Score the following summarization given the corresponding article with respect to consistency with one to five stars, where one star means "inconsistency" and five stars means "perfect consistency". Note that consistency measures whether the facts in the summary are consistent with the facts in the original article. Consider whether the summary reproduces all facts accurately and does not make up untrue information.
> Article: <src>
> Summary: <hyp>
> Stars: