BatchEval: Towards Human-like Text Evaluation

Anonymous ACL submission

Abstract

Significant progress has been made in automatic text evaluation with the introduction of large language models (LLMs) as evaluators. However, current sample-wise evaluation paradigm suffers from the following issues: (1) Sensitive to prompt design; (2) Poor resistance to context noise; (3) Inferior ensemble performance with static reference. Inspired by the fact that humans treat both criterion definition and inter sample comparison as references for evaluation, we propose BATCHEVAL, a paradigm that conducts batch-wise evaluation iteratively to alleviate the above problems. We explore variants under this paradigm and confirm the optimal settings are two stage procedure with heterogeneous batch composition strategy and decimal scoring format. Comprehensive experiments across 3 LLMs on 4 text evaluation tasks demonstrate that BATCHEVAL outperforms state-of-the-art methods by 10.5% on Pearson correlations with only 64% API cost on average. Further analyses have verified the robustness, generalization, and working mechanism of BATCHEVAL [1].

1 Introduction

Accurately evaluating the text quality in specific criterion (e.g., coherence) can facilitate better understanding, application, and development of large language models (LLMs), which becomes more crucial with their recent rapid progress in text generation capabilities (OpenAI, 2023). Due to the labor-intensive and time-consuming nature of human evaluation, early works have explored automatic evaluation methods, which can be categorized into rule-based (Papineni et al., 2002; Lavie and Denkowski, 2009), embedding-based (Forgues et al., 2014; Zhang et al., 2020), and learning-based (Mehri and Eskénazi, 2020; Zhang et al., 2022) approaches. Continuous progress has been achieved through these methods, but there remains a significant gap in their consistency with human judgments (Sai et al., 2023).

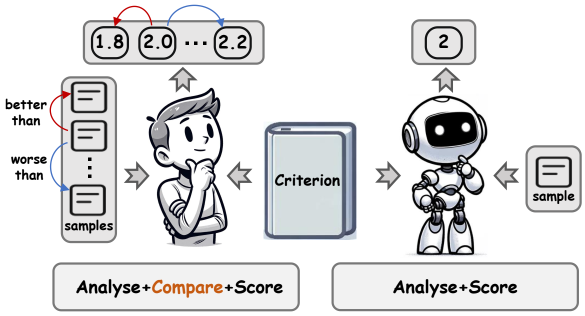

Figure 1: Both humans and LLM-based evaluators assess text based on criterion definition, but humans further conduct sample comparison for better evaluation.

Recently, the revolutionary power of LLMs has been applied across various fields, demonstrating performance that is even on par with humans (OpenAI, 2023; Guo et al., 2023a). In text evaluation filed, LLM-based evaluators (Chiang and Lee, 2023a; Liu et al., 2023; Guo et al., 2023b; Chiang and Lee, 2023b) have also made significant progress compared to traditional methods, but they still lag behind human evaluators. We carefully compare their working procedures and find that the difference in evaluation references might be the reason for the performance disparity (Figure 1). Human evaluators analyze samples based on the criterion definition and provide discriminative scores through comparison between samples. However, LLM-based evaluators assess each sample individually, thus only having criterion as a reference.

We analyze that current sample-wise evaluation paradigm will face problems on three aspects: (1) *Robustness against prompt design?* Since criterion is the sole reference for evaluation, minor changes of prompt may significantly affect the evaluation results (See §4.4 for empirical validation). (2) *Robustness against noise?* Due to the absence of comparison between samples, the evaluation scores

[1] Our code and data have been made public on the internet.

lack discrimination and exhibit a non-uniform distribution (See Figure 3), which can lead to reduced robustness against noise like random deletion or synonym substitution on samples (See Theorem 1). (3) *Better performance under ensemble?* Current LLM-based evaluators average scores from multiple generations as the final rating for given sample. However, generating multiple times from the static reference (criterion) induces a lack of diversity among scores (Figure 4), which can weaken the effect of ensemble according to Theorem 2.

To address the aforementioned problems, we propose BATCHEVAL, a new LLM-based text evaluation paradigm that assesses samples batch-wise, akin to the way of humans. Overall, BATCHEVAL iterates an allocation process where all samples are first split into batches, and then each batch is compiled into a prompt as the input of LLMs. By introducing in-batch samples as an additional reference apart from criterion, the orthogonal and complementary references can not only reduce the dependency on prompt design but also enhance the discrimination of scores between samples through in-batch comparison, leading to improved robustness against noise. Furthermore, the iteratively changing batch composition can provide LLMs with varying evaluation references, thereby enhancing diversity and the ensemble performance.

While the idea of BATCHEVAL is simple, there are many ways it can be realized. We explored variants in evaluation procedure, format of scoring and composition of batch. Some of them work surprisingly well while some do not meet expectations. Experiments and analyses confirm that separate analyzing and scoring evaluation procedure, decimal scoring format, and quality-heterogeneous batch composition strategy yield the optimal results.

We conduct extensive experiments on 4 text evaluation tasks primarily with GPT-4: turn-level response, dialogue, text summarization, and story generation. By allowing in-batch samples to share single prompt and applying a small iteration rounds, BATCHEVAL outperforms best performing LLM-based evaluators by a significant margin (10.5%) in terms of correlation with human evaluations, while incurring only 64% of API costs. We also validate the generalization of BATCHEVAL on more LLMs, robustness to prompt design and noise, and analyze the choice of hyperparameters through further experiments. Finally, we probe into the working mechanism of BATCHEVAL through attention analysis on Llama-2-70b-chat-hf. Our contributions are summarized as follows:

1. We analyzed how the sample-wise evaluation paradigm of LLM-based evaluators, differing from human evaluators, limited their robustness and consistency with human judgment.
2. We proposed BATCHEVAL, a new paradigm that evaluates texts batch-wise, and experimentally validated its optimal settings.
3. We validated through experiments on 4 tasks that BATCHEVAL outperforms public state-of-the-art methods by 10.5% while incurring only 64% of the API cost.
4. We analyzed the generalization, robustness, hyperparameter selection, and probed into the working mechanism of BATCHEVAL.

2 Background

2.1 Automatic Text Evaluation

Automatic text evaluation method has been extensively studied as a supplement to labor-intensive and time-consuming human evaluation, with its correlation to human judgment as the criterion for assessment. Both *rule-based* (Papineni et al., 2002; Lavie and Denkowski, 2009) and *embedding-based* (Zhang et al., 2020; Forgues et al., 2014) evaluation methods rely on the assumption that high-quality generated texts should have a significant word overlap with reference texts. However, this assumption conflicts with the high entropy nature of text generation, restricting its consistency with humans. *Learning-based* methods consider directly assessing text quality through supervised (Lowe et al., 2017; Goyal and Durrett, 2021) and self-supervised (Mehri and Eskénazi, 2020; Zhang et al., 2022) approaches and achieve significant progress. Recently, *LLM-based* evaluators (Guo et al., 2023b; Chiang and Lee, 2023b; Liu et al., 2023) have demonstrated advanced consistency with humans leveraging their incredible knowledge and capabilities. However, typical sample-wise evaluation paradigm of the above methods leads to a lack of inter-sample comparison during scoring process, which serves as an important reference for human evaluators. Therefore, we propose BATCHEVAL to fill this gap for better alignment with humans.

2.2 Supportive Theorems

Theorem 1 *The robustness against noise correlates positively with the uniformity of evaluator* *scoring distribution. (See Appendix A.1 for derivation in details)*

Yuan et al. (2023) proposed this theorem and verified that learning-based evaluators, by adjusting the training loss function to uniformize the score distribution, can achieve better robustness against noise. We have experimentally proven that sample-wise LLM-based evaluators also exhibit an uneven score distribution (Figure 3), which can weaken their robustness against noise (Appendix C). Thus, we propose BATCHEVAL for a more uniform score distribution and better robustness against noise.

Theorem 2 *Given scores from multiple generations of certain LLM $\mathcal{S} = \{s_i | i = 1, .., N\}$ and human evaluation score y for sample x, $\bar{s}$ is the average of $\mathcal{S}$, the following equation holds:*

$$Err(\bar{s}, y) = Err(\mathcal{S}, y) - Var(\mathcal{S}) \tag{1}$$

where:

$$\begin{aligned} Err(\bar{s}, y) &= (\bar{s} - y)^2 \\ Err(\mathcal{S}, y) &= \frac{1}{N}\sum_{i=1}^{N}(s_i - y)^2 \\ Var(\mathcal{S}) &= \frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2 \end{aligned} \tag{2}$$

Eq. (1) (Zhou, 2012) (proof in Appendix A.2) implies that smaller average error in single prediction scores ($Err(\mathcal{S}, y)$) and larger variance among multiple prediction scores ($Var(\mathcal{S})$) induce smaller error in ensemble score ($Err(\bar{s}, y)$). However, current sample-wise LLM evaluators score multiple times based solely on static reference (criterion), resulting in smaller $Var(\mathcal{S})$ (Figure 5). To address this, we propose iterative quality-heterogenized batch composition strategy for LLMs to score with unbiased varying references, thus increasing $Var(\mathcal{S})$ for lower $Err(\bar{s}, y)$.

3 Methodology

The core idea behind BATCHEVAL is to fully use in-batch sample comparison to enhance evaluation accuracy and robustness. Algorithm 1 illustrates the working process of BATCHEVAL, which involves N rounds of iteration: (1) B samples of each batch are compiled with pre-defined (task, criterion, evaluation procedure) into a single prompt for input to the LLM; (2) Based on the LLM's assessment of the samples' quality, we optimize batch allocation according to certain batch composition strategy. The core designs throughout the process are ***how to evaluate*** (evaluation procedure), ***what to input*** (batch composition strategy), and ***what to output*** (scoring format). Below we discuss their potential variants in detail.

Algorithm 1 Workflow of BATCHEVAL.

Require: Samples $x^{1:|\mathcal{D}|}$, LLM $\mathcal{M}$, Evaluation procedure P
Task and criterion T, Iteration rounds N , Batchsize B
Batch composition strategy BATCHSTRATEGY
Ensure: Ensemble evaluation scores $\bar{s}^{1:|\mathcal{D}|}$
Randomly divide $x^{1:|\mathcal{D}|}$ into batches $b^{1:L}$, $L = \lceil \frac{|\mathcal{D}|}{B} \rceil$
$S_{all} \leftarrow \{i : [\,] \text{ for } i \in [1, |\mathcal{D}|]\}$
for $i \leftarrow 1, N$ **do**
  $S_{current} \leftarrow \varnothing$
  for $j \leftarrow 1, L$ **do**:
    $S_{current} \leftarrow S_{current}$.Append($\mathcal{M}(T, P, b^j)$)
  end for
  $S_{all} \leftarrow S_{all}$.Merge($S_{current}$)
  $b^{1:L} \leftarrow$BATCHSTRATEGY($x^{1:|\mathcal{D}|}, S_{all}, B$)
end for
$\bar{s}^{1:|\mathcal{D}|} \leftarrow$Average($S_{all}$)

3.1 *How to Evaluate*

Sample-wise LLM evaluators work through a process of either analyzing followed by scoring (Guo et al., 2023b) or scoring followed by analyzing (Liu et al., 2023), where the former typically performs better (Chiang and Lee, 2023b) possibly due to the effect of chain-of-thought (Wei et al., 2022). On this basis, we further explore what procedures can better facilitate sample comparison for BATCHEVAL (Appendix J for prompts):

One stage As the most intuitive extension of sample-wise evaluation, LLM analyzes and scores each sample of the batch in order. This procedure enables adequate comparison between samples, but insufficient comparison between analyses (the analyses of subsequent samples cannot be referenced by the earlier samples for scoring).

Two stage To enhance the comparison among analyses, the LLM first analyzes all the samples. Based on the full comparisons among samples and analyses, the LLM further scores for each sample.

Three stage From human experience, it can be easier to first rank and then score the samples, as compared to directly scoring them. Therefore, we consider a procedure that sequentially performs analyzing, ranking, and scoring for all samples.

3.2 *What to Input*

The composition of the batch largely determines the efficacy of in-batch comparison as evaluation

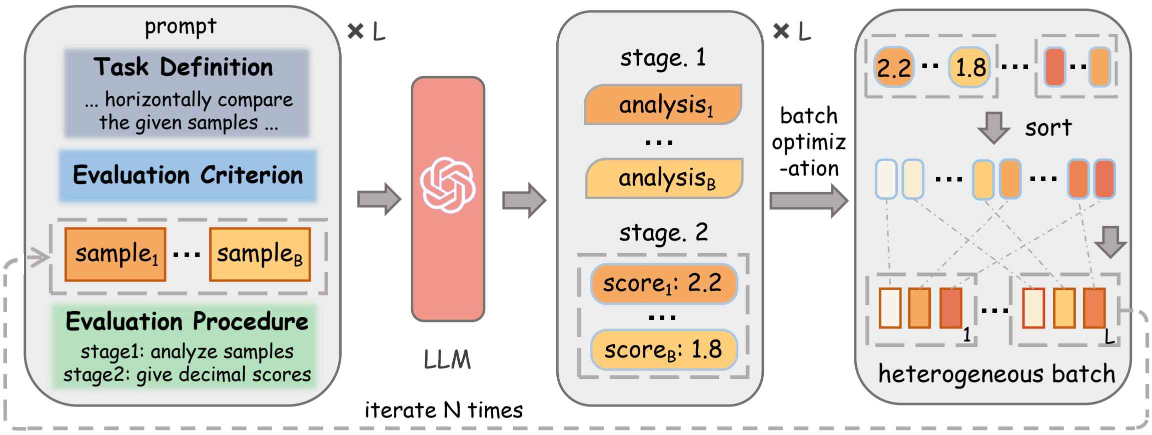

Figure 2: Overall illustration of BATCHEVAL.

reference. According to Theorem 2, we consider redrawing the batch divisions after each round of evaluation to provide the LLM with varying references when assessing a certain sample, thus can improve scoring diversity. Besides diversity, we are curious about what other characteristics the batch should possess to further enhance the effectiveness of BATCHEVAL, for which we explore the following strategies.[2]

Random Batch One base strategy is to reallocate batches randomly after each round of evaluation.

Homogeneous Batch Based on the idea of coarse-to-fine evaluation, we consider forming homogeneous batches in which samples have similar scores from the previous round of evaluation, in the hope that these samples can be further compared by LLM and ultimately attain discriminative scores.

Heterogeneous Batch A contrary idea is to select samples with diversified scores based on the previous round of evaluation results to form a new batch. In this way, LLM develops an unbiased perception of samples with different qualities through batch optimization, thus scoring more accurately.

3.3 *What to Output*

Sample-wise evaluation methods typically apply integers as the format for LLM scoring (Liu et al., 2023; Chiang and Lee, 2023b), and Lin and Chen (2023) proved that using more refined scoring format can not bring additional gains. *Will this trend be similar in* BATCHEVAL*?* Let us consider a concrete example: there are two samples with close but different quality, with human ratings of 2.2 and 1.8, respectively. Due to having only the criterion as reference, sample-wise evaluators may consider them to be close to the 2-point standard and consequently assign a score of 2 regardless of whether decimal is allowed. However, if they appear in the same batch, on the basis of judging that they are all close to 2 points, LLM can further compare their quality directly. Thus, it is possible for LLM to give them differentiated decimal scores if it is allowed, thereby achieving more consistent judgments with humans. Based on the analysis above, we consider trying out two different scoring formats: **integer** and **decimal**.

Our default settings of BATCHEVAL include two stage evaluation procedure, heterogeneous batch composition strategy and decimal scoring format, as shown in Figure 2.

4 Experiments

Centered around BATCHEVAL, we will empirically explore the optimal variants in §4.2, demonstrate its performance on different LLMs and tasks in §4.3, validate the robustness in §4.4, and delve into its working mechanism in §4.5. We also investigate the choice of hyperparameters in Appendix §B.

4.1 Experimental settings

Benchmarks A brief introduction of benchmarks involved are listed as follows:

- **Topical-Chat** (Mehri and Eskénazi, 2020) is a benchmark for evaluating dialogue response generation. To save costs, we exclude knowledge as input to LLM and therefore choose criteria where knowledge is not necessary: `Naturalness`, `Coherence`, `Engaging`, `Naturalness` and `Overall`.

[2] See Appendix F for strategies in detail.

Type	Method	Scheme	Engaging		Understand		Naturalness		Coherence		Overall		Average		
			r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	$\$/item$
Human	Inter-annotator*		.575	.581	.510	.510	.486	.487	.558	.560	.710	.718	.568	.571	-
Rule	BLEU-4*	-	.232	.316	.201	.218	.180	.175	.131	.235	.216	.296	.192	.248	-
	METEOR*	-	.367	.439	.245	.225	.212	.191	.250	.302	.337	.391	.282	.310	-
Embedding	V-Extrema*	-	.210	.205	.156	.132	.101	.076	.184	.184	.203	.209	.171	.161	-
	BERTScore*	-	.317	.335	.256	.226	.226	.209	.214	.233	.298	.325	.262	.266	-
Learning	USR*	-	.456	.465	.293	.315	.276	.304	.416	.377	.422	.419	.373	.376	-
	BCR	-	.460	.463	.297	.325	.260	.298	.425	.391	.437	.421	.376	.380	-
LLM	G-Eval	-	.710	.719	.568	.593	.595	.605	.576	.584	.717	.705	.633	.641	.0614
	CloserLook	-	.651	.688	.649	.699	.656	.665	.675	.687	.778	.772	.682	.702	.0686
	CloserLook	+ ICL	.714	.743	.603	.685	.679	.693	.720	.733	.786	.783	.700	.727	.0856
	BatchEval (Ours)	one stage	.780	.783	.642	.680	.706	.710	.727	.729	.785	.793	.728	.739	.0525
		three stage	.782	.778	.667	.725	.712	704	.712	.714	.797	.798	.734	.744	.0541
		random	.746	.743	.685	.724	.711	.700	.716	.720	.798	.799	.731	.737	.0528
		homogeneous	.654	.663	.639	.607	.671	674	.669	.631	.722	.703	.671	.656	.0537
		integer	.771	.778	.686	**.732**	.726	.727	.722	.727	.790	.783	.739	.749	.0526
		default	**.792**	**.790**	**.694**	.727	**.730**	**.735**	**.740**	**.744**	**.805**	**.800**	**.752**	**.759**	.0529

Table 1: Turn-level Pearson (r_p) / Spearman (r_s) correlations and average API cost per sample ($\$/item$) of different metrics on Topical-Chat benchmark. The results of methods with * come from USR. We reproduced other methods with a unified API (the results were generally better than those reported in the original paper). All results of our replication are statistically significant (p-value < 0.05).

- **FED** (Mehri and Eskenazi, 2020) includes human ratings on 11 criteria to evaluate the quality of dialogue. We choose the top 4 important criteria as claimed in the original paper for evaluation: `Coherent`, `Understanding`, `Likeable` and `Overall`.
- **HANNA** (Chhun et al., 2022) serves as a benchmark for meta-evaluating evaluation methods on story generation, with criteria including: `Coherence`, `Relevance`, `Empathy`, `Surprise`, `Engagement` and `Complexity`.
- **QAGS** (Chhun et al., 2022) is a benchmark for evaluating the `Factual Consistency` of summaries on CNN (Hermann et al., 2015) and XSUM (Narayan et al., 2018).

Baselines We introduce four types of baseline methods in the experiments. Among them, both rule-based and embedding-based methods need reference text, which is unavailable in FED and QAGS. Learning-based methods are typically task-specific. Below we briefly list their categories and snapshots of LLM-based methods. Refer to Appendix G for detailed introductions.

- **Rule-based**: BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009).
- **Embedding-based**: Vector Extrema (Forgues et al., 2014), BERTScore (Zhang et al., 2020)
- **Learning-based**: USR (Mehri and Eskénazi, 2020), BCR (Yuan et al., 2023), FED (Mehri and Eskenazi, 2020), DynaEval (Zhang et al., 2021), QAGS (Wang et al., 2020).
- **LLM-based**[3]: G-Eval (Liu et al., 2023) recommended using LLM to evaluate according to the procedures generated by itself. Chiang and Lee (2023b) tried various evaluation schemes and proved through experiments that *analyze-rate* led to the best performance, which we denote as CloserLook.

Details We explore variants of BatchEval on Topical-Chat for its wide recognition. If not specified, FED serves as our default dataset for exploratory experiments as it only has 125 samples, thus can save API expenses. The other two benchmarks are used to confirm the generalization across tasks of BatchEval. We primarily conduct experiments with GPT-4 (*0613*) and validate the generalization across models of BatchEval with GPT-3.5-turbo (*0613*) and Llama-2-70b-chat-hf. We set iteration rounds as 5, batchsize as 10, decoding temperature as 0.2 for all the experiment. For other

[3]Two latest and well-known LLM evaluators are included. We are unable to reproduce some other methods due to incomplete disclosure of codes or prompts.

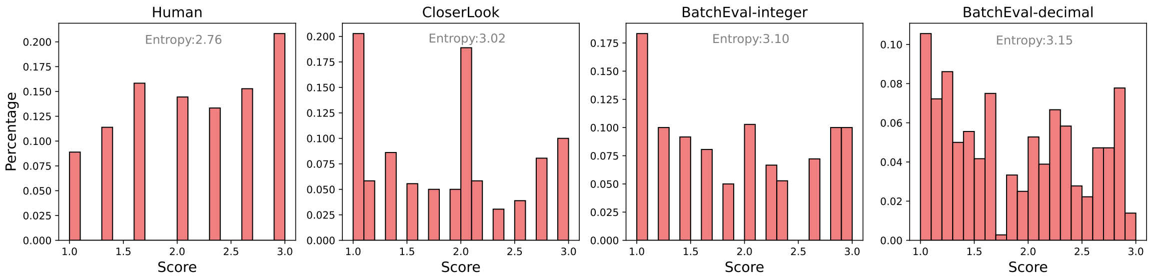

Figure 3: Score distribution and corresponding entropy ($-\sum_s p(s) \log_2 p(s)$) of different methods.

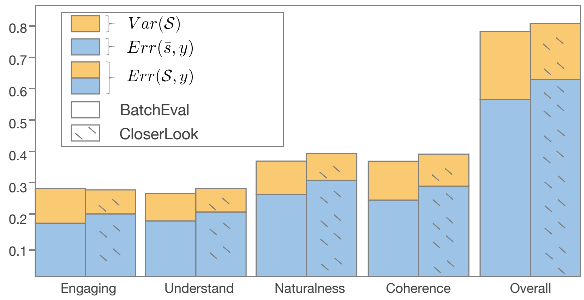

Figure 4: Comparisons between BATCHEVAL and CloserLook from the perspective of Theorem 2.

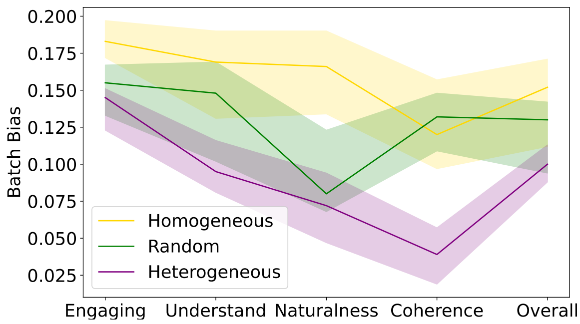

Figure 5: Average batch bias of different strategies.

LLM-based evaluators, we reproduced them according to their default settings (20 generations per sample) with the same API for a fair comparison. We choose Pearson and Spearman correlations to measure consistency with humans and also report API expenses for adequate comparison. We follow (Chiang and Lee, 2023b) to design prompts (See prompts in Appendix J).

4.2 Variants Exploration

As shown in Table 1, based on the default settings shown in Figure 2, we validate the effects of different variants (replacing the default setting with specific scheme) of BATCHEVAL.

Evaluation Procedure Compared to one stage procedure, the two stage procedure (default) achieves higher correlations by enhancing the comparison among analyses during scoring. Surprisingly, however, the three stage procedure does not perform well as expected. We speculate this may be due to the LLM's over-reliance on ranking results while neglecting the analyses and samples during scoring, and valide this in Appendix D.

Batch Composition Strategy As shown in Table 1, the performance of batch composition strategies ranks as follows: heterogeneous (default) > random > homogeneous. To investigate the reasons, we introduce batch bias as follows:

$$Bias(\mathcal{B}) = abs(\sum_{i \in \mathcal{B}} s_i^{\mathcal{B}} - \sum_{i \in \mathcal{B}} \bar{s}_i)/|\mathcal{B}| \tag{3}$$

where $\mathcal{B}$ denotes the set of sample indexes of certain batch, $s_i^{\mathcal{B}}$ denotes score of sample x_i generated with batch $\mathcal{B}$, $\bar{s}_i$ denotes average score of sample x_i across all the iterations. Ideally, we aspire for the batch bias to approach zero. This implies that LLM should not have the overall scores in a batch skewed either high or low compared to the ensemble scores. We evaluate the average $Bias(\mathcal{B})$ of different strategies and find that $Bias(\mathcal{B})$ correlates negatively with correlations $\boldsymbol{r_s}$ and $\boldsymbol{r_p}$ (Figure 5). This indicates that the more varied the quality of samples in a batch, the better they can simulate a real distribution as an unbiased reference to bring smaller batch bias for better correlations.

Scoring Format We observe from Table 1 that decimal scoring format brings around 1 point correlations improvement upon integer. As shown in Figure 3, the decimal scheme brings a more uniform scoring distribution. This implies that LLM indeed assigns more discriminative scores to different samples through in-batch comparison if decimal score is allowed, which verifies our hypothesis in §3.3 and accounts for the progress.

4.3 Overall Performance of BATCHEVAL

As shown in Table 1, 2, 3, 4, BATCHEVAL achieves an average of 6.5 points (10.5%) Pearson and 4.5 points (7.1%) Spearman correlations improvements with humans across four benchmarks compared to

Type	Method	Model	Likeable		Understand		Coherent		Overall		Average		
			r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	$\$/item$
Human	Inter-annotator	-	-	.838	-	.809	-	.809	-	.830	-	.822	-
Learning	USR	-	.245	.226	.182	.178	.170	.185	.284	.302	.220	.223	-
	FED	-	.248	.262	.295	.306	.262	.253	.460	.449	.316	.318	-
	DynalEval	-	.389	.393	.379	.368	.399	.409	.484	.490	.413	.415	-
LLM	CloserLook	Llama-2-70b	.525	.550	.574	**.611**	**.640**	.563	.634	.639	.593	.591	-
	BatchEval	Llama-2-70b	**.537**	**.563**	**.619**	.597	.627	**.648**	**.722**	**.732**	**.626**	**.635**	-
	CloserLook	GPT-3.5-turbo	.681	.666	.691	.605	.726	.724	.687	**.709**	.696	.676	.0022
	BatchEval	GPT-3.5-turbo	**.682**	**.674**	**.704**	**.708**	**.733**	**.730**	**.705**	.699	**.706**	**.703**	.0011
	G-Eval	GPT-4	.638	.692	.670	.625	.707	.721	.689	.652	.676	.673	.0667
	CloserLook *w* human prompt	GPT-4	.658	.680	.701	.614	.739	.751	.715	.684	.703	.682	.0785
	CloserLook *w* GPT-4 prompt	GPT-4	.632	.660	.678	.639	.725	.749	.723	.678	.690	.682	.0827
	BatchEval *w* human prompt	GPT-4	.731	**.741**	.778	.696	.753	**.753**	.738	**.729**	.750	**.730**	.0314
	BatchEval *w* GPT-4 prompt	GPT-4	**.736**	**.741**	**.780**	**.700**	**.784**	.749	**.748**	.727	**.762**	.729	.0314

Table 2: Dialog-level Pearson (r_p) / Spearman (r_s) correlations and average API cost per sample ($\$/item$) on FED-dialog benchmark. We implemented and tested all the methods with p-value < 0.05.

Method	Coherence		Relevance		Empathy		Surprise		Engagement		Complexity		Average		
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	$\$/item$
BLEU-4	.220	.218	.135	.175	.242	.216	.178	.224	.242	.270	.362	.273	.230	.229	-
METEOR	.335	.273	.202	.190	.304	.282	.285	.283	.316	.338	.520	.482	.307	.307	-
BERTScore	.358	.293	.201	.188	.308	.303	.302	.290	.308	.331	.501	.472	.330	.313	-
G-Eval	.572	.578	.582	.584	.453	.461	.311	.347	.562	591	.602	.557	.514	.520	.0772
CloserLook	.595	.591	.579	.597	.498	.478	.280	.339	.605	**.607**	.619	.568	.529	.530	.0835
BatchEval	**.678**	**.625**	**.702**	**.679**	**.546**	**.543**	**.368**	**.381**	**.617**	.605	**.625**	**.575**	**.589**	**.568**	.0538

Table 3: Story-level Pearson (r_p) / Spearman (r_s) correlations and average API cost per sample ($\$/item$) of on HANNA benchmark. We implemented and tested all the methods with p-value < 0.05.

Method	QAGS-C		QAGS-X		Average		
	r_p	r_s	r_p	r_s	r_p	r_s	$\$/item$
BERTScore*	.576	.505	.024	.008	.300	.256	-
QAGS*	.545	-	.175	-	.375	-	-
G-Eval*	.631	**.685**	.558	.537	.599	.611	-
CloserLook	.581	.602	.549	.573	.498	.478	.0691
BatchEval	**.785**	.643	**.618**	**.634**	**.682**	**.639**	.0521

Table 4: Results on QAGS benchmark (QAGS with -C and -X denote subset CNN and XSUM respectively). Results with * come from G-EVAL. we present the original results of G-Eval here as our replication is not good as those reported in the original paper.

the best performing methods. From the perspective of Theorem 2, as shown in Figure 4, we found that the reason BatchEval outperforms CloserLook under score ensemble ($Err(\bar{s}, y)$) is twofold. First, BatchEval attains more accurate single predictions ($Err(\mathcal{S}, y)$) through thorough in-batch comparison. Second, the scoring diversity ($Var(\mathcal{S})$) of BatchEval is significantly improved. This validates that iterative heterogeneous batch composition strategy can provide LLM with unbiased varying evaluation references, thus stably enhancing diversity and ensemble performance.

In terms of cost, BatchEval only consumes 64% API expenses of the best performing baselines. This is because we only use the average scores from 5 iterations and allow in-batch samples to share single prompt, while the LLM-based baselines average scores from 20 generations.[4] Considering that baselines reach ensemble saturation at about 20 generations, BatchEval has broad potential for performance improvement with more iterations.

4.4 Robustness of BatchEval

Robustness against Prompt Design We test BatchEval and CloserLook respectively on prompts written by human and rewritten by GPT-4, with results as shown in Table 2. We calculate the average difference in correlations across metrics under two types of prompts. The standard deviation

[4]Due to changes in the prompt during iteration, the prompt expense needs to be billed 5 times for our method, whereas baselines require only once. Therefore the expenditure ratio (64%) is higher than the proportion of generations (5:20).

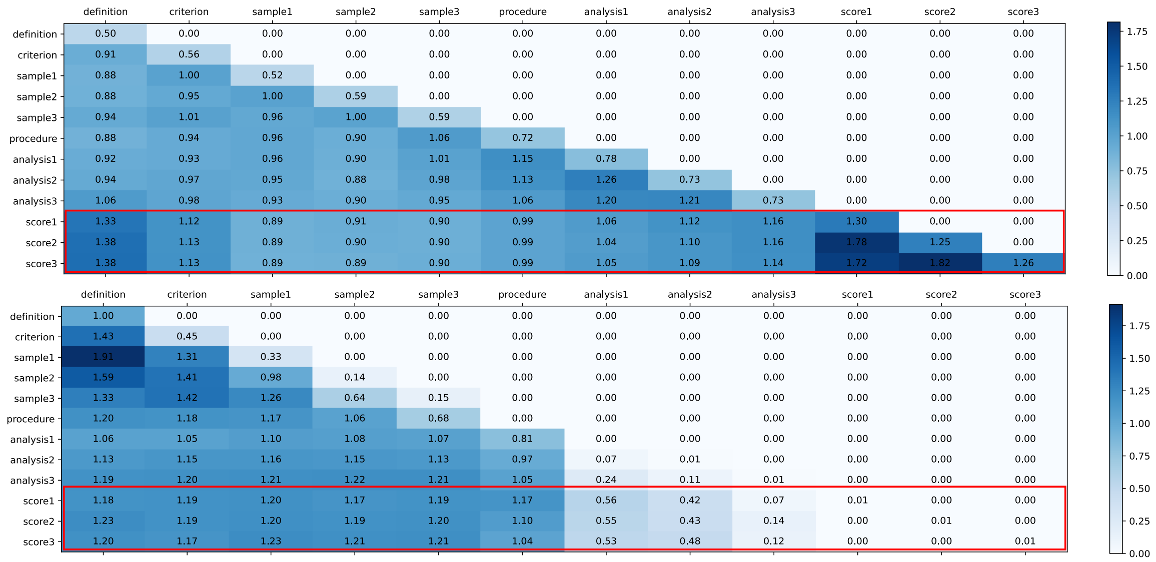

Figure 6: Normalized attention matrices of the first (top figure) and last (bottom figure) transformer layer with Llama-2-70b-chat-hf. We set batchsize as 3 for clear demonstration. See Appendix H for the normalizing process.

of r_p and r_s are 0.009 and 0.007 for CloserLook, while only 0.006 and 0.002 for BATCHEVAL. This verifies that BATCHEVAL attains better robustness against prompt design by introducing in-batch samples as additional references.

Robustness against Noise As shown in Figure 3, the score distribution of BATCHEVAL is more uniform and has lower entropy compared with CloserLook due to in-batch comparison with decimal scoring format, which can theoretically enhance robustness against noise according to Theorem 1. We further experimentally validate this in Appendix C.

4.5 Further Discussion and Analysis

Relationship with In-context-learning ICL (Brown et al., 2020) can also provide sample-side references by incorporating samples and corresponding answers into the prompt. The main differences between ICL and BATCHEVAL are: (1) BATCHEVAL can provide LLM with varying and comprehensive references through iterative heterogeneous batch, while the references provided by ICL are relatively fixed and may bring bias (sensitive to prompt design). (2) BATCHEVAL uses in-batch samples as references to each other, thus saving the costs of demonstrations in ICL prompts. Thanks to the aforementioned advancements, BATCHEVAL outperforms CloserLook with ICL by more than 5 points Pearson correlations while only incurs 61.8% expense (Table 1).

Working Mechanism of BATCHEVAL To further understand how BATCHEVAL benefits from in-batch comparison, we visualized the normalized attention matrices of the first and last layers of Llama-2-70b-chat-hf (Figure 6). The value at (X,Y) represents the average normalized attention of tokens corresponding to X towards tokens corresponding to Y. We observe that in the final scoring phase (red box), LLM first perceives samples with varied qualities based on the already generated scores and analyses at the shallower layers. Afterwards, LLM completes scoring based on criterion and comparison between samples at the deeper layers. This process demonstrates the in-batch comparison mechanism of BATCHEVAL, which we hope can inspire future research.

5 Conclusions

In this paper, we propose BATCHEVAL, a new text evaluation paradigm that evaluate samples batch-wise to alleviate the limitations of sample-wise evaluation paradigm. We explore variants of BATCHEVAL on multiple dimensions and figure out the optimal settings. Following the human evaluation method, BATCHEVAL treats in-batch samples and criterion as complementary references and optimizes the batch composition through iteration to eliminate batch bias. Comprehensive experiments have confirmed that BATCHEVAL can achieve higher consistency with humans at a lower cost, while also demonstrating better robustness to prompt design and noise. We further analyze and reveal the working mechanism of BATCHEVAL, shedding lights on future work.

Limitations

From an objective perspective , we think there are two main limitations of this paper:

1. BATCHEVAL requires LLMs to have a certain capability to handle longer contexts. From Appendix B, we found that as the batchsize increases, LLMs struggle to handle too many samples, leading to a performance decline. We also attempted to test BATCHEVAL's performance on Llama-2-13b-chat-hf and found that the batchsize must be set to 2 or 3 to see any benefits. Therefore, when setting the batchsize, we cannot exceed the limit of how many samples an LLM can process in a single context. Fortunately, we discovered that a batchsize of 10 is suitable for current mainstream LLMs. Additionally, as LLMs continue to advance, they can handle increasingly larger contexts. Thus, from this perspective, BATCHEVAL is a scalable method that improves alongside the capabilities of LLMs (increasing the batchsize within the capabilities of the LLM can enhance the evaluation effectiveness of the LLM).

2. We only explored a limited number of schemes of BATCHEVAL. We leave exploring possible schemes of BATCHEVAL for future research.

Ethics Statement

All of the datasets used in this study were publicly available, and no annotators were employed for our data collection. We confirm that the datasets we used did not contain any harmful content and was consistent with their intended use (research). We have cited the datasets and relevant works used in this study.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cyril Chhun, Pierre Colombo, Fabian M Suchanek, and Chloé Clavel. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. In *29th International Conference on Computational Linguistics (COLING 2022)*.

David Cheng-Han Chiang and Hung-yi Lee. 2023a. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15607–15631. Association for Computational Linguistics.

David Cheng-Han Chiang and Hung-yi Lee. 2023b. A closer look into using large language models for automatic evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 8928–8942. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop*, volume 2, page 168.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1449–1462. Association for Computational Linguistics.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. 2023a. Evaluating large language models: A comprehensive survey. *CoRR*, abs/2310.19736.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. 2023b. Evaluating large language models: A comprehensive survey. *CoRR*, abs/2310.19736.

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.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Mach. Transl.*, 23(2-3):105–115.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *CoRR*, abs/2305.13711.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics.

Ryan Lowe, Michael D. Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1116–1126. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. In *21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 225.

Shikib Mehri and Maxine Eskénazi. 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, ACL 2020, Online, July 5-10, 2020*, pages 681–707. Association for Computational Linguistics.

Shashi Narayan, Shay Cohen, and Maria Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2023. A survey of evaluation metrics used for NLG systems. *ACM Comput. Surv.*, 55(2):26:1–26:39.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Peiwen Yuan, Xinglin Wang, Jiayi Shi, Bin Sun, Yiwei Li, and Kan Li. 2023. Better correlation and robustness: A distribution-balanced self-supervised learning framework for automatic dialogue evaluation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee, and Haizhou Li. 2021. Dynaeval: Unifying turn and dialogue level evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5676–5689. Association for Computational Linguistics.

Pengfei Zhang, Xiaohui Hu, Kaidong Yu, Jian Wang, Song Han, Cao Liu, and Chunyang Yuan. 2022. MME-CRS: multi-metric evaluation based on correlation re-scaling for evaluating open-domain dialogue. *CoRR*, abs/2206.09403.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv e-prints*, pages arXiv–2306.

Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. CRC press.

A Proof of Theorems Involved

A.1 Theorem 1

For any $f(x)$, the probability density function of score distribution, the Spearman correlation $\mathbb{E}(r_s)$ between the original scores and scores adding a small disturbance has an upper bound:

$$\mathbb{E}(r_s) \leq 1 - \frac{6\mathbb{E}(\lambda)^2}{n^2 - 1}, \tag{4}$$

and the equality condition is $f(x) \equiv 1, \forall x \in [0, 1]$.

Proof 1 *The ranking difference* $d(x)$ *before and after disturbance is :*

$$d(x) = \int_x^{x+\lambda} f(x)dx \tag{5}$$

According to the definition of Spearman correlations, $E(r_s)$ *can be written as:*

$$\mathbb{E}(r_s) = \mathbb{E}(1 - \frac{6\sum_{i=1}^{n} d(x_i)^2}{n\,(n^2 - 1)}), \tag{6}$$

we derive the lower bound of $\mathbb{E}(d(x)^2)$ *as follows:*

$$\begin{aligned}
&\mathbb{E}(d(x)^2) \\
&= \int_0^1 \left(\int_x^{x+\mathbb{E}(\lambda)} f(u)du\right)^2 f(x)dx \\
&= \int_0^1 \left(\int_x^{x+\mathbb{E}(\lambda)} f(u)du\sqrt{f(x)}\right)^2 dx \\
&= \int_0^1 \left(\int_x^{x+\mathbb{E}(\lambda)} f(u)du\sqrt{f(x)}\right)^2 dx \\
&\quad \cdot \int_0^1 f(x)dx \\
&\geq \left(\int_0^1 \int_x^{x+\mathbb{E}(\lambda)} f(u)duf(x)dx\right)^2 \\
&\quad (Cauchy's\ Inequality) \\
&= \left(\int_0^1 \mathbb{E}(\lambda) \cdot f(x) \cdot f(x)dx\right)^2 (\mathbb{E}(\lambda) \to 0) \\
&= \mathbb{E}(\lambda)^2 \left(\int_0^1 f(x) \cdot f(x)dx\right)^2 \\
&= \mathbb{E}(\lambda)^2 \left(\int_0^1 f(x)^2dx \cdot \int_0^1 1^2dx\right)^2 \\
&\geq \mathbb{E}(\lambda)^2 \left(\left(\int_0^1 f(x)dx\right)^2\right)^2 \\
&\quad (Cauchy's\ Inequality) \\
&= \mathbb{E}(\lambda)^2
\end{aligned} \tag{7}$$

The equality condition is $f(x) \equiv 1\ for\ x \in [0, 1]$. *Taking the lower bound of* $\mathbb{E}(d(x)^2)$ *into Eq. (6), we conclude the proof. Note that higher* $\mathbb{E}(r_s)$ *denotes better robustness against noise. Hence, we can derive that the robustness against noise correlates positively with the uniformity of score distribution.*

A.2 Theorem 2

Given scores from multiple generations of certain LLM $\mathcal{S} = \{s_i | i = 1, .., N\}$ and human evaluation score y for sample x, $\bar{s}$ is the average of $\mathcal{S}$, the following equation holds:

$$Err(\bar{s}, y) = Err(\mathcal{S}, y) - Var(\mathcal{S}) \tag{8}$$

where:

$$\begin{aligned}
Err(\bar{s}, y) &= (\bar{s} - y)^2 \\
Err(\mathcal{S}, y) &= \frac{1}{N}\sum_{i=1}^{N}(s_i - y)^2 \\
Var(\mathcal{S}) &= \frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2
\end{aligned} \tag{9}$$

Proof 2

$$\begin{aligned}
&Err(\mathcal{S}, y) - Var(\mathcal{S}) \\
&= \frac{1}{N}\sum_{i=1}^{N}(s_i - y)^2 - \frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2 \\
&= \frac{1}{N}(\sum_{i=1}^{N}(s_i^2 + y^2 - 2s_iy - s_i^2 - \bar{s}^2 + 2s_i\bar{s})) \\
&= y^2 - \bar{s}^2 - \frac{1}{N}(\sum_{i=1}^{N} 2s_iy) + \frac{1}{N}(\sum_{i=1}^{N} 2s_i\bar{s}) \\
&= y^2 - \bar{s}^2 - 2\bar{s}y + 2\bar{s}^2 \\
&= y^2 + \bar{s}^2 - 2\bar{s}y \\
&= (\bar{s} - y)^2 \\
&= Err(\bar{s}, y)
\end{aligned} \tag{10}$$

B Hyperparameter Analysis

In the experiments of the main text, we set the batch size to 10 and the temperature to 0.2. In this section, we explore the impact of different hyperparameter choices on performance.

B.1 Effect of Batchsize

On FED dataset, we test BATCHEVAL with batch-size among [1, 2, 5, 10]. As shown in Figure 7, we found that as the batch size increases, the performance generally undergoes a process of initial improvement followed by a decline. Similar observations were made on other datasets as well. We further discovered that the performance turning point of the ensemble results from five iterations is slightly delayed compared to a single prediction. Considering that increasing the batchsize will make

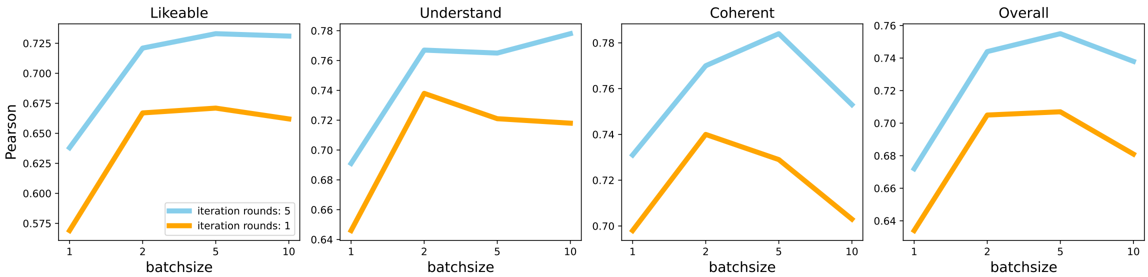

Figure 7: Dialog-level Pearson correlations on FED-dialog dataset of BATCHEVAL with different batchsize.

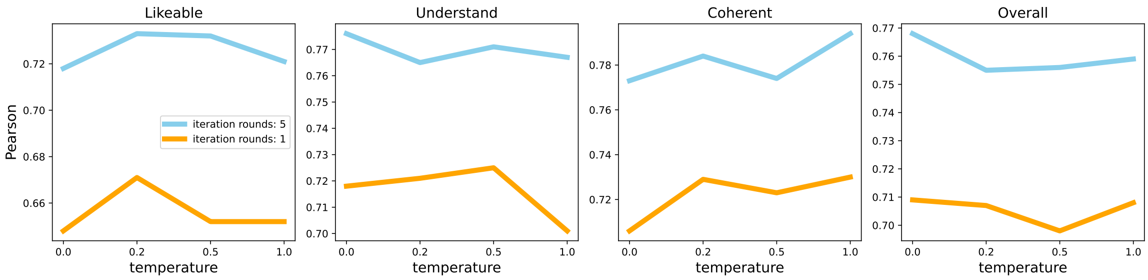

Figure 8: Dialog-level Pearson correlations on FED-dialog dataset of BATCHEVAL with different temperature.

the combination of in-batch samples more diverse, thereby increasing scoring diversity, we have the following conjecture about Figure 7: When the batchsize starts to increase from 1, due to the effect of in-batch comparison and the increase in diversity, the performance of both 1-round score and ensemble score increase a lot. However, as the batchsize continues to increase, LLM finds it difficult to handle too many samples simultaneously, resulting in a decrease in 1-round score performance. When the rate of decrease in 1-round score performance gets greater than the rate of increase in diversity, ensemble score performance also begins to decrease according to Theorem 2. Therefore, the batchsize should not be too large or too small. We found that setting the batchsize to 10 can achieve superior performance on different tasks. We also believe that for LLMs with weaker ability to handle longer context, the batchsize should be set to be smaller. Fortunately, we have noticed that current LLMs are continually improving in processing long contextual texts, which illuminates further development prospects for BATCHEVAL in the future.

B.2 Effect of Temperature

We also test BATCHEVAL with temperature among [0, 0.2, 0.5, 1]. We found that as the temperature rises in Figure 8, the performance of BATCHEVAL does not exhibit a uniform trend of change. Overall, the performance of 5 iterations is relatively stable along the temperature dimension, suggesting that BATCHEVAL is quite robust to temperature variations.

C Robustness against Noise

To test the robustness against noise of BATCHEVAL, we use an external tool[5] to add noise to the input and calculate the changes in performance before and after the noise is added. For the sake of noise balance, we randomly replace 5% of tokens with synonyms and randomly delete 5% of tokens. As shown in Table 5, CloserLook experiences a decrease of 0.109 in Pearson correlation and 0.081 in Spearman correlation, respectively. In contrast, BATCHEVAL only shows a decrease of 0.003 and 0.009, respectively. This indicates that BATCHEVAL has much better robustness to noise.

D Inferior Performance of Three Stage Procedure

As shown in Table 1, we observe a performance drop of BATCHEVAL with three stage procedure, though it may be closer to human evaluation procedure. We speculate this may be due to the LLM's over-reliance on ranking results while neglecting the analyses and samples during scoring. To validate this, we delete the ranking and scoring contents of LLM's three stage procedure response and ask LLM to score based on the remaining con-

[5]nlpaug(https://github.com/makcedward/nlpaug)

Method	Likeable		Understand		Coherent		Overall		Average		
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	$\$/item$
CloserLook *w/o* noise	.658	.680	.701	.614	.739	.755	.715	.684	.703	.683	.0785
CloserLook *w* noise	.509	.580	.626	.606	.608	.605	.632	.616	.594 (-.109)	.602 (-.081)	.0866
BATCHEVAL *w/o* noise	.731	.741	.778	.696	.753	.757	.738	.729	.750	.731	.0314
BATCHEVAL *w* noise	.729	718	.775	.700	.764	.754	.720	.724	.747 (-.003)	.724 (-.007)	.0344

Table 5: Story-level Pearson (r_p) / Spearman (r_s) correlations and average API cost per sample ($\$/item$) of on HANNA benchmark. We tested all the methods for a fair comparison with p-value < 0.05.

Method	Scheme	Engaging		Understand		Naturalness		Coherence		Overall		Average		
		r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	$\$/item$
BATCHEVAL	default	**.792**	**.790**	.694	.727	**.730**	**.735**	**.740**	.744	.805	.800	**.752**	.759	.0529
	3 stage	.782	.778	.667	.725	.712	704	.712	.714	.797	.798	.734	.744	.0541
	3 stage w/o rank results	.789	.785	**.701**	**.733**	.721	.727	.735	**.747**	**.810**	**.808**	.751	**.760**	-

Table 6: Comparison of BATCHEVAL with different scheme. *3 stage w/o ranking results* means results of deleting the ranking and scoring contents of LLM's three stage procedure response and asking LLM to score based on the remaining contents (samples and analyses)

tents (samples and analyses). If the new scoring results perform similarly to BATCHEVAL with two stage procedure, the inferior performance of BATCHEVAL with three stage procedure can be attributed to its excessive focus on ranking results. Otherwise, the reason lies in the decrease in the quality of analyses. As shown in Table 6, the performance of three stage w/o rank results is on par with that of two stage procedure. This validates our conjecture that the over-reliance on ranking results causes the performance drop of BATCHEVAL with three stage procedure.

E Relationship with Pair-wise Evaluation

The current mainstream text evaluation approach adopts sample-wise assessment. Alternatively, an LLM evaluator is presented with a question and two answers, and is tasked with determining which one is better or declaring a tie (Zheng et al., 2023; Dubois et al., 2023). However, as the number of models to be evaluated grows, the scalability of pairwise comparison becomes a challenge, due to the quadratic increase in the potential number of pairs. Therefore, this pair-wise paradigm has not been as extensively studied as sample-wise evaluation. Zheng et al. (2023) validates that this method performs slightly better than a sample-wise evaluator, potentially due to its ability to discern subtle differences between specific pairs.

Similarly, we have enhanced the evaluation capabilities of the LLM evaluator through in-batch sample comparison. The main difference lies in the composition of our batches, which consist of different samples rather than responses from different models to the same sample, thereby offering good scalability.

F Batch Composition Strategies

F.1 Homogenized Batch

Given scores $s^{1:|\mathcal{D}|}$ for samples $x^{1:|\mathcal{D}|}$ predicted by LLM in the previous round, we first sort the scores and attain the corresponding indexes $index^{1:|\mathcal{D}|}$. Based on this, we get indexes of homogenized batch $b^i = index^{1+(i-1)*10:i*10}$.

F.2 Heterogenized Batch

Given scores $s^{1:|\mathcal{D}|}$ for samples $x^{1:|\mathcal{D}|}$ predicted by LLM in the previous round, we first sort the scores and attain the corresponding indexes $index^{1:|\mathcal{D}|}$. Considering that our default batchsize is 10, we group the indexes into 10 splits $split^{1:10}$, where $split^i = index^{1+(i-1)\times\lceil\frac{|\mathcal{D}|}{10}\rceil:i\times\lceil\frac{|\mathcal{D}|}{10}\rceil}$. Based on this, we get indexes of heterogenized batch $b^i = \{split^{j,i} | j \in [1, 10]\}$.

G Introduction of Baselines

G.1 Rule-based Methods

BLEU (Papineni et al., 2002) BLEU is a renowned metric for measuring word overlap, which evaluates n-gram precision in a generated sequence against a reference. It includes a brevity penalty to counteract its inherent preference for shorter sentences, ensuring a more comprehensive assessment.

METEOR (Lavie and Denkowski, 2009) is an advancement over BLEU, utilizing a harmonic mean of precision and recall, and also incorporating stemming and synonym use in its evaluation.

G.2 Embedding-based Methods

Vector Extrema (Forgues et al., 2014) is a scoring method that uses cosine similarity between sentence embeddings, identifying the highest value in each dimension of the word embedding for evaluation.

BERTScore (Zhang et al., 2020) is a method that utilizes a pretrained BERT (Devlin et al., 2019) model to optimally align each word in a reference response with a single word in the generated sequence. By doing so, BERTScore computes the recall of the generated sequence.

G.3 Learning-based Methods

USR (Mehri and Eskénazi, 2020) is a dialogue response evaluation method that uses one masked language model and two dialogue retrieval models to assess various sub-qualities of a sample and then integrates these evaluations into a comprehensive overall score.

BCR (Yuan et al., 2023) is a dialogue response evaluation method that use a dynamic loss function to train a BERT model with uniform score distribution.

FED (Mehri and Eskenazi, 2020) is a unified dialogue evaluation method that uses pretrained language models to calculate scores based on the difference in the probability of generating positive and negative evaluation words for a certain criterion.

DynaEval (Zhang et al., 2021) is also a unified dialogue evaluation method that leverages graph convolutional network to model the sentences among a dialogue for accurate evaluation.

QAGS (Wang et al., 2020) is a method that based on question-answering, which creates questions from a summary and then verifies whether their answers are present in the original source document.

H Details of Normalizing Process

We will introduce how to normalize the attention matrix to make it more visually appealing like in Figure 6. Due to the autoregressive generation mode of mainstream LLMs, the expected values of attention between token pairs at different positions vary. If we use $Att(x, y)$ to represent the attention of the x^{th} token to the y^{th} token, then its expected value is $\frac{1}{x}$. Since tokens at different positions will be visualized into the same graph, we first multiply each $Att(x, y)$ by x to make its expected value 1. On this basis, we determine the token intervals corresponding to different strings through word matching, and calculate $Att(string1, string2)$ as follows:

$$Att(s1, s2) = Avg(\{Att(x, y)|x \in s1, y \in s2\}) \tag{11}$$

according to which we plot our attention matrices.

I Case Study

Below is an example response from GPT-4 regarding FED-coherence:

> *I will do my best to provide individual analysis for each sample. Analysis:*
>
> *Sample1: The system maintains a coherent conversation with the user, but there are some instances where the system's responses are not directly related to the user's questions. For example, when asked about the technology that powers it, the system fails to provide a clear answer.*
>
> *Sample2: The system maintains a good flow of conversation and responds appropriately to the user's questions. However, there is a slight misunderstanding about the release date of a movie.*
>
> *Sample3: The system maintains a coherent conversation with the user, responding appropriately to the user's statements and questions.*
>
> *Sample4: The system's responses are not coherent with the user's statements. For example, the user talks about a ski trip and the system responds with a question about French.*
>
> *Sample5: The system maintains a coherent conversation with the user, but there are some instances where the system's responses are not directly*

related to the user's questions. For example, when asked about its age, the system provides a confusing response.

Sample6: The system maintains a coherent conversation with the user, responding appropriately to the user's statements and questions.

Sample7: The system maintains a coherent conversation with the user, but there are some instances where the system's responses are not directly related to the user's statements. For example, the user talks about Taylor Swift and the system responds with a statement about Leonardo da Vinci.

Sample8: The system's responses are not coherent with the user's statements. For example, the user asks to see the system's humanlike part and the system responds with unrelated statements.

Sample9: The system maintains a coherent conversation with the user, responding appropriately to the user's statements and questions.

Sample10: The system maintains a coherent conversation with the user, responding appropriately to the user's statements and questions.

Float Scores: [Sample1:2.5, Sample2:2.8, Sample3:3, Sample4:1.5, Sample5:2.5, Sample6:3, Sample7:2.5, Sample8:1.5, Sample9:3, Sample10:3]

J Example Prompts

J.1 Evaluate Coherence for Topical-Chat

default prompt

You will be given a batch of {{number}} samples. Each sample contains a conversation between Speaker A and Speaker B and one potential response for the next turn.

Your task is to assign a float score to the response on one metric.

You should carefully horizontally compare the given samples in order to assign a suitable float score to each sample.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Coherence (floating point numbers within the interval [1,3]): Does the response serve as a valid continuation of the conversation history?

- A float score near 1 (no) means that the response drastically changes topic or ignores the conversation history.

- A float score near 2 (somewhat) means the response refers to the conversation history in a limited capacity (e.g., in a generic way) and shifts the conversation topic.

- A float score near 3 (yes) means the response is on topic and strongly acknowledges the conversation history.

Conversations and corresponding potential response to be evaluated:

{{Data}}

Evaluation Form (Answer by starting with "I will do my best to provide individual analysis for each sample. Analysis:" to analyze the given samples regarding the evaluation criteria as concise as possible (Attention: Don't give your scores during this step). After analysing all the samples, please give all the float scores in order following the template "Float Scores: [Sample1:score of Sample1,...,Sample{{number}}:score of Sample{{number}}]".

- Coherence:

one stage prompt

You will be given a batch of {{number}} samples. Each sample contains a conversation between Speaker A and Speaker B and one potential response for the next turn.

Your task is to assign a float score to the response on one metric.

You should carefully horizontally compare the given samples in order to assign a suitable float score to the given samples one by one.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Coherence (floating point numbers within the interval [1,3]): Does the response serve as a valid continuation of the conversation history?

- A float score near 1 (no) means that the response drastically changes topic or ignores the conversation history.

- A float score near 2 (somewhat) means the response refers to the conversation history in a limited capacity (e.g., in a generic way) and shifts the conversation topic.

- A float score near 3 (yes) means the response is on topic and strongly acknowledges the conversation history.

Conversations and corresponding potential response to be evaluated:

{{Data}}

Evaluation Form (Answer by starting with "I will do my best to provide individual analysis and give a suitable float score for each sample in order". When rating for each sample, please follow the template "Score of SampleX:[float score]").

- Coherence:

three stage prompt

You will be given a batch of {{number}} samples. Each sample contains a conversation between Speaker A and Speaker B and one potential response for the next turn.

You will be introduced to a metric to be evaluated.

You should carefully horizontally compare the given samples in order to assign a suitable float score to each sample.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Coherence (floating point numbers within the interval [1,3]): Does the response serve as a valid continuation of the conversation history?

- A float score near 1 (no) means that the response drastically changes topic or ignores the conversation history.

- A float score near 2 (somewhat) means the response refers to the conversation history in a limited capacity (e.g., in a generic way) and shifts the conversation topic.

- A float score near 3 (yes) means the response is on topic and strongly acknowledges the conversation history.

Conversations and corresponding potential response to be evaluated:

{{Data}}

Answer by starting with "I will do my best to provide individual analysis for each sample. Analysis:" to analyze the given samples regarding the evaluation criteria as concise as possible (Attention: Don't give your scores during this step). After analysing all the samples, please horizontally compare the given samples, rank all the samples according to the analysis of the response and give the corresponding reasons. After ranking, according to the analysis and rank, please give all the float scores in order following the template "Float Scores: [Sample1:score of Sample1,...,Sample{{number}}:score of Sample{{number}}]".

- Coherence:

Integer prompt

You will be given a batch of {{number}} samples. Each sample contains a conversation between Speaker A and Speaker B and one potential response for the next turn.

Your task is to rate the responses on one metric.

You should carefully horizontally compare the given samples in order to assign a score to each sample.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Crieteria:

Coherence (1-3): Does the response serve as a valid continuation of the conversation history?

- A score of 1 (no) means that the response drastically changes topic or ignores the conversation history.

- A score of 2 (somewhat) means the response refers to the conversation history in a limited capacity (e.g., in a generic way) and shifts the conversation topic.

- A score of 3 (yes) means the response is on topic and strongly acknowledges the conversation history.

Conversations and corresponding potential response to be evaluated:

{{Data}}

Evaluation Form (Answer by starting with "I will do my best to provide individual analysis for each sample. Analysis:" to analyze the given samples regarding the evaluation criteria as concise as possible (Attention: Don't give your scores during this step). After analysing all the samples, please give all the scores in order following the template "Scores: [Sample1:score of Sample1,...,Sample{{number}}:score of Sample{{number}}]".

- Coherence:

J.2 Evaluate Coherent for FED-Dialogue default prompt

You will be given a batch of {{number}} samples. Each sample contains a conversation between User and a dialogue System.

Your task is to assign a float score to the sample on one metric.

You should carefully horizontally compare the given samples in order to assign a suitable float score to each sample.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Coherent (floating point numbers within the interval [1,3]): Does System maintain coherence and a good flow of conversation throughout the dialogue?

- A float score near 1 (not coherent) means that System's responses are unrelated to the conversation topic and may disrupt or confuse the flow of the dialogue.

- A float score near 2 (somewhat coherent) means that System's responses are partially related to the conversation topic but may not be clear or direct.

- A float score near 3 (very coherent) means that System's responses are closely related to the conversation topic and contribute to maintaining a smooth dialogue.

Conversations to be evaluated:

{{Data}}

Evaluation Form (Answer by starting with "I will do my best to provide individual analysis for each sample. Analysis:" to analyze the given samples regarding the evaluation criteria as concise as possible (Attention: Don't give your scores during this step). After analysing all the samples, please give all the float scores in order following the *template "Float Scores: [Sample1:score of Sample1,...,Sample{{number}}:score of Sample{{number}}]".*

- Coherent:

J.3 Evaluate Coherence for HANNA

default prompt

You will be given a batch of {{number}} samples. Each sample contains a prompt and a story generated following the prompt.

Your task is to assign a float score to the story according to the prompt on one metric.

You should carefully horizontally compare the given samples in order to assign a suitable float score to each sample.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Coherence (floating point numbers within the interval [1,5]) Measures whether the story makes sense?

- A float score near 1 means the story does not make sense at all. For instance, the setting and/or characters keep changing, and/or there is no understandable plot.

- A float score near 2 means most of the story does not make sense.

- A float score near 3 means the story mostly makes sense but has some incoherences.

- A float score near 4 means the story almost makes sense overall, except for one or two small incoherences.

- A float score near 5 means the story makes sense from beginning to end.

Prompts and corresponding stories to be evaluated:

{{Data}}

Evaluation Form (Answer by starting with "I will do my best to provide individual analysis for each sample. Analysis:" to analyze the given samples regarding the evaluation criteria as concise as possible (Attention: Don't give your scores during this step). After analysing all the samples, please give all the float scores in order following the template "Float Scores: [Sample1:score of Sample1,...,Sample{{number}}:score of Sample{{number}}]".

- Coherence:

J.4 Evaluate Factual Consistency for QAGS

default prompt

You will be given a batch of {{number}} samples. Each sample contains an article and a sentence.

Your task is to determine if the sentence is factually correct given the contents of the article.

You should carefully horizontally compare the given samples in order to assign a suitable float score to each sample.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Consistency ([1,3]) - Is the sentence supported by the article? (consistent with the article)

- A float score near 1 (not) means that the sentence is totally not supported by the article.

- A float score near 2 (somewhat) means that the sentence is partially supported by the article.

- A float score near 3 (very) means that the sentence is completely supported by the article.

Articles and corresponding sentences to be evaluated:

{{Data}}

Evaluation Form (Answer by starting with "I will do my best to provide *individual analysis for each sample. Analysis:" to analyze the given samples regarding the evaluation criteria as concise as possible (Attention: Don't give your scores during this step). After analysing all the samples, please give all the float scores in order following the template "Float Scores: [Sample1:score of Sample1,...,Sample{{number}}:score of Sample{{number}}]".*

- Consistency: