CheckEval: A reliable LLM-as-a-Judge framework for evaluating text generation using checklists

Anonymous ACL submission

Abstract

Existing LLM-as-a-Judge approaches for evaluating text generation suffer from rating inconsistencies, with low agreement and high rating variance across different evaluator models. We attribute this to subjective evaluation criteria combined with Likert scale scoring in existing protocols. To address this issue, we introduce CheckEval, a checklist-based evaluation framework that improves rating reliability via decomposed binary questions. Through experiments with 12 evaluator models across multiple datasets, we first demonstrate that CheckEval strongly correlates with human judgments. More importantly, CheckEval dramatically improves the average agreement across evaluator models by 0.45 and reduces the score variance. CheckEval scores furthermore have the benefit of being more interpretable because it decomposes evaluation criteria into traceable binary decisions, allowing analyses of specific attributes driving quality judgments.

1 Introduction

Evaluating text generation quality remains a major challenge in Natural Language Generation (NLG), particularly as Large Language Models (LLMs) continue to advance in their generative capabilities (Brown et al., 2020; Chowdhery et al., 2023; Achiam et al., 2023). This is especially evident in tasks such as summarization, dialogue, and creative writing (Liu et al., 2023d; Kim et al., 2023; Liu et al., 2023a), where qualitative dimensions of the output are crucial yet difficult to measure systematically. Consequently, there is growing interest in developing evaluation methods that can effectively capture these aspects. These methods will ideally involve well-defined protocols that ensure reliability across different raters and tasks. In obtaining actual scores from such protocols, human evaluation remains the gold standard, but it is costly, time-consuming, and difficult to scale (Novikova et al., 2017; Belz et al., 2020). While lexical overlap-based metrics such as ROUGE and BLEU (Lin, 2004; Papineni et al., 2002) have been widely adopted for ease of automation, they align poorly with human judgments, calling for alternatives that better approximate human evaluation.

Recent work has explored the use of LLM-as-a-Judge as a scalable alternative, leveraging LLMs to assess generated text directly (Zheng et al., 2023). This paradigm has evolved through various approaches: single-turn prompting (Liu et al., 2023b; Fu et al., 2023), meta-evaluator training (Kim et al., 2023; Wu et al., 2024b), and even more sophisticated methods like multi-agent debate (Chan et al., 2024; Kim et al., 2024). However, these methods often rely on subjective evaluation protocols that require evaluators to assign holistic scores without clear decision criteria. For example, evaluators are typically asked to rate text on a Likert scale from 1 to 5 (higher is better) across evaluation dimensions, such as coherence, consistency, fluency, and relevance. While Likert scales are effective for capturing ordinal relationships in human evaluation, they face two key challenges when applied to LLM-based evaluator models. First, current LLMs are known to struggle with subjective criteria in Likert-scale evaluations, in particular showing difficulty in differentiating between high-quality outputs (Li et al., 2019; Stureborg et al., 2024). Second, evaluation results are highly sensitive to the choice of evaluator models. These lead to low *inter-evaluator agreement* (IEA),[1] which we define as the agreement among evaluator models (of similar capacity), as well as high variance in evaluation results (Stureborg et al., 2024). Yet, previous LLM-as-a-Judge approaches have overlooked these

[1]This is equivalent to Inter-Annotator Agreement (IAA) in human evaluation (Artstein, 2017), but we use the term IEA in this paper to make it clear that the agreement we are aiming to improve is agreement between evaluator models, rather than between human raters providing the gold evaluation.

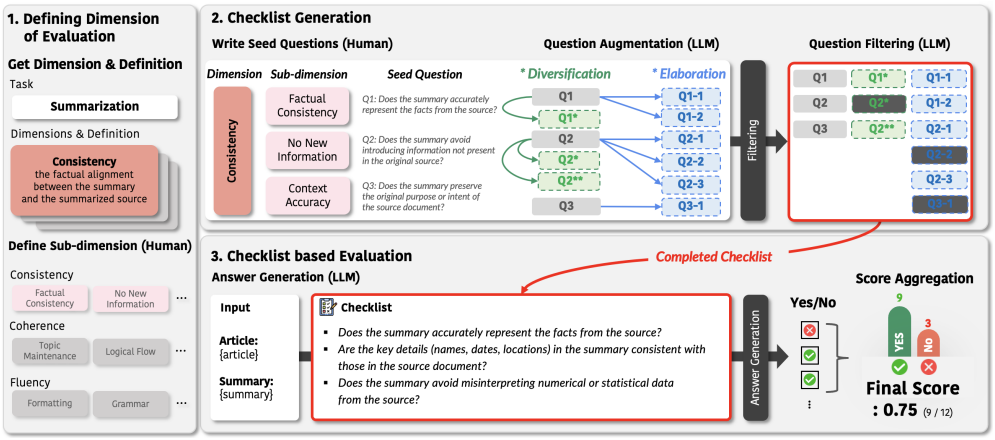


Figure 1: Overall process of CheckEval. CheckEval consists of three stages: (1) Defining Dimensions of Evaluation, where humans select specific dimensions and define sub-dimensions; (2) Checklist Generation, which incorporates two augmentation methods—question diversification (green) and elaboration (blue); and (3) Checklist-based Evaluation, where the model responds to the checklist with yes/no answers.

issues (Gao et al., 2024a).

To address these challenges, we introduce **CheckEval**, a reliable evaluation framework that decomposes evaluation criteria to target fine-grained qualitative dimensions and turns them into a checklist.[2] Inspired by recent advances in fine-grained decomposition of evaluation (Liu et al., 2023c; Min et al., 2023), our framework breaks down evaluation into discrete Boolean questions. This decomposition simplifies each individual evaluation question and clarifies the rationale behind evaluation decisions. CheckEval addresses key limitations of existing methods in two ways. First, it improves explainability by tracking how specific criteria are met, making evaluation decisions more explicit and reducing ambiguity. Second, it enhances consistency through structured binary responses, which improve IEA and reduce variability. Importantly, CheckEval maintains competitive correlation with human evaluation while achieving these improvements. These improvements are verified through comprehensive experiments across 12 different LLM-based evaluator models of varying sizes, including both open and closed-source models, on multiple datasets. The main contributions of this study can be summarized as follows:

- We introduce CheckEval, a fine-grained evaluation framework leveraging a Boolean QA checklist to address the rating consistency issues with existing LLM-as-a-Judge methods for NLG evaluation.
- Experiments across 12 LLMs and multiple datasets demonstrate significant improvements in correlation with human evaluation compared to Likert-based approaches like G-Eval (Liu et al., 2023b).
- CheckEval shows reduced sensitivity to the choice of evaluator models, leading to more consistent evaluation results with lower variance and higher IEA.

2 Related Work

2.1 LLM-as-a-Judge

Traditional NLG evaluation metrics like ROUGE and BLEU show clear limitations due to their reliance on reference texts (Gu et al., 2021). With advances in LLMs, researchers have explored LLM-as-a-Judge, where an LLM evaluates texts based on specified criteria, formalized as $F(\text{subject}, \text{criteria}) \rightarrow \text{result}$ (Li et al., 2024). LLM-as-a-Judge can be categorized into pairwise and pointwise evaluation approaches (Gu et al., 2024). Pairwise evaluation (Zheng et al., 2023; Qin et al., 2024) compares two outputs to determine relative preference but is computationally expensive as comparisons scale exponentially. In contrast, pointwise evaluation (Liu et al., 2023b; Fu et al., 2023) assigns scores to individual outputs, allowing for absolute scaling and continuous assessment. However, existing pointwise evaluation protocols often lack granularity, assigning a single numeric score to each dimension of evaluation. If the specified dimensions of evaluation are too broad (e.g., `fluency`), this may lead to inconsis-

[2] Our checklist concept is inspired by Ribeiro et al. (2020), who proposed checklist-based testing for NLP models.

tencies in judgments because many factors could influence the quality along the target dimension. CheckEval builds on the pointwise evaluation but addresses its limitations by adopting a finer-grained Boolean QA Checklist.[3]

2.2 Decomposition Strategy

Decomposing complex information into minimal units to simplify tasks have been explored in various areas of NLP (Kamoi et al., 2023; Chen et al., 2022; Wright et al., 2022; Krishna et al., 2023; Nenkova and Passonneau, 2004; Liu et al., 2024). Recent studies have shown that breaking down content into atomic units reduces subjectivity in factual consistency judgment (Liu et al., 2023c; Min et al., 2023). Atomic units represent elementary information that cannot be further divided. Similarly, CheckEval decomposes evaluation criteria into fine-grained Boolean QA Checklists to enhance clarity and minimize ambiguity in the evaluation process.

2.3 Reliability of Evaluation

Reliability is an important yet often overlooked component of evaluation. Many LLM-as-a-Judge methods focus only on correlation with human scores, often neglecting consistency and stability across different LLMs. Recent studies have highlighted several reliability concerns. Xiao et al. (2023) demonstrate that LLMs fail to reliably assess subtle quality differences in text. Similarly, Bavaresco et al. (2024) find these models often assign highly variable ratings to identical inputs. Furthermore, IEA remains low across models, compromising evaluation reliability (Stureborg et al., 2024). CheckEval addresses these issues by evaluating not only correlation but also IEA and score variance across evaluator models, improving reliability across diverse LLMs.

3 Method

CheckEval consists of three stages, (1) Defining Dimensions of Evaluation, (2) Checklist Generation, and (3) Checklist-Based Evaluation, as shown in Figure 1. The framework translates the evaluation criteria into a Boolean QA checklist, each question in the checklist expecting a binary (yes/no) response. This format improves clarity and alleviates ambiguity compared to Likert-scale scoring (discussed further in Section 6.2).

3.1 Defining Dimensions of Evaluation

The first stage defines the evaluation dimensions of text quality (e.g., `consistency`, `fluency`) by either adopting predefined dimensions from benchmarks or specifying custom dimensions for the task. For each dimension, we then define sub-dimensions that break down the high-level dimensions further into distinct and detailed components. The sub-dimensions are grounded in the original definitions of the dimensions from benchmark datasets and can also also informed by related work (Liu et al., 2023c; Laban et al., 2023; Tang et al., 2019). For instance, `fluency` in summarization can include sub-dimensions such as `formatting`, `grammar`, `completeness`, and `readability`.

Sub-dimensions must be carefully designed to align with benchmark definitions and to prevent inconsistencies in evaluation. While LLMs can be used to automate the generation of sub-dimensions and questions, we found that fully relying on them often led to misalignment with the criteria defined by the benchmark. This leads to evaluation that is not grounded on the benchmark design, potentially producing incorrect assessments. To address this, we only allow human-selected sub-dimensions in our work, following prior work that recommends human oversight as an effective way to maintain alignment with benchmark objectives (Szymanski et al., 2024; Pan et al., 2024).

3.2 Checklist Generation

Seed Question Writing We create Boolean questions that correspond to the sub-dimensions defined in the first step. Each question requires a 'yes' or 'no' answer, where 'yes' indicates adherence to the evaluation criterion corresponding to the target sub-dimension. This binary format simplifies the judgment process, ensuring that evaluation criteria are explicitly defined and consistently applied (Laban et al., 2023; Liu et al., 2023c). This format also helps LLMs generate more reliable responses by constraining the answer space, minimizing response variability, and reducing ambiguity. For example, the question *"Are all words in the sentence spelled correctly?"* elicits a clearer and more direct response than an open-ended alternative like *"How well does the sentence adhere to or deviate from standard grammar rules?"*.

[3] Recent studies (Wu et al., 2024a; Wang et al., 2024) use LLM-as-a-Judge as a reward signal in alignment training with RLHF (Ouyang et al., 2022). However, this approach primarily aims to optimize model training rather than enhance evaluation robustness and explainability. Our work focuses on improving evaluation frameworks, and integrating evaluation signals into model training is beyond our scope.

Question Augmentation Manually designing a comprehensive set of evaluation questions would be ideal for ensuring a high-quality checklist. However, this approach faces scalability limitations, making it impractical to generate a sufficiently large and diverse set of questions for evaluation. This challenge becomes even more significant when extending to individual application scenarios, each requiring its own comprehensive set of questions. To this end, we expand the seed questions using LLMs, enhancing both the diversity and granularity of evaluation. Augmentation enables broader coverage while refining questions to capture a wider range of lexical and semantic variations. This process follows two strategies, each extending the coverage of seed questions. (1) *Question Diversification* expands evaluation diversity by introducing variations that explore different perspectives of sub-dimensions and contexts of the seed question. (2) *Question Elaboration* increases granularity by expanding the seed questions into more specific and detailed questions. To ensure that the augmented questions remain grounded in the seed questions, Question Diversification and Elaboration are performed independently rather than sequentially. For example, the seed question *"Are all words in the sentence spelled correctly?"* can be expanded into *"Are all sentences complete, with no fragments or missing components?"* (diversification) or specified into *"Are proper nouns (names of people, places, etc.) spelled correctly?"* (elaboration). By integrating both approaches, the checklist maintains a structured and scalable evaluation framework.

Question Filtering LLM-based augmentation expands the question set, but it can also generate questions that do not fully align with the intended evaluation criteria. Some questions may reflect misinterpretations of dimension definitions or add unnecessary redundancy, which can affect evaluation reliability. To filter out such questions, we apply an LLM-based minimal filtering process that evaluates a combined pool of seed and augmented questions for each dimension. This filtering step applies three main criteria for retaining relevant questions: (1) alignment, verifying that a 'yes' response to the question indicates higher quality; (2) dimension consistency, confirming that the question adheres to the original definition of the evaluation dimension; and (3) redundancy removal, eliminating semantically overlapping questions to avoid unnecessary repetition. While there is no direct metric to measure filtering effectiveness, we observe improved correlation with human judgments after filtering, suggesting that the filtering is functioning as intended. We further validated the quality of the checklist via human study, where annotators scored the augmented and filtered questions (See Appendix B.1.)

3.3 Checklist-Based Evaluation

In the final stage, LLMs evaluate the target text using the completed checklist (see Table 12 and 13 for the number of checklist questions and Table 22 and 23 for the dimensions, sub-dimensions, and corresponding seed question for each dataset). To improve efficiency, we ask multiple questions simultaneously rather than asking each question separately. We compared single-question and multi-question inference in our pilot experiments and found no noticeable difference in performance. Therefore, we evaluated multiple questions together to reduce the computational cost. The questions are grouped by sub-dimensions, ensuring that related questions are presented together to aid model comprehension. For each question in the checklist, the LLM generates a 'yes' or 'no' response. The final quality score is computed as the proportion of 'yes' responses among all questions (e.g., 15 'yes' out of 20 questions yields 0.75). We note that the final score is computed by treating all checklist questions equally, each 'yes' response contributes the same weight. We discuss an alternative weighting strategy in Appendix C.2. More implementation details about the evaluation process are described in Section 4.4. This approach enhances explainability by explicitly tracking how specific criteria are met, making evaluation decisions more interpretable without requiring additional rationale generation. Unlike existing LLM-as-a-Judge approaches, such as G-Eval (our main comparison point) that generate only numerical scores without explanation (e.g., *"Based on the conversation history, the corresponding context, and the response, here is the evaluation: 'Naturalness': 2"*), the reasoning behind the evaluation score is easily traceable from the checklist responses.

4 Experimental Setup

4.1 Datasets and Metrics

Following Liu et al. (2023b), we use three meta-evaluation benchmarks for various tasks to measure

the effectiveness of CheckEval: **SummEval** (Fabbri et al., 2021), **Topical-Chat** (Zhong et al., 2022) and **QAGS** (Wang et al., 2020). Our main evaluation focuses on SummEval and Topical-Chat, which cover a broad range of evaluation dimensions. QAGS, which focuses solely on factual consistency in summarization, is included in Appendix B.2. We report Pearson's r, Spearman's ρ, Kendall's τ on each benchmark.

4.2 Baselines

We selected **G-Eval** (Liu et al., 2023b) as our main baseline for comparison. G-Eval adopts chain-of-thought prompting (Wei et al., 2022) and a form-filling paradigm to generate evaluation scores on a Likert scale. We selected it based on three factors: (1) its widespread adoption, (2) the availability of publicly released prompts that facilitate reproducibility, and (3) its relatively simple setup that avoids confounding performance-enhancing techniques—such as prompt optimization (e.g., self-correction), training meta-evaluators, preference learning, or multi-agent frameworks. Like G-Eval, CheckEval is also designed to rely solely on a binary checklist mechanism, without introducing additional optimization techniques beyond standard prompting. Although they are not apples-to-apples comparisons, we also include comparisons to methods surveyed in Gu et al. (2024); Gao et al. (2024b) showing that CheckEval remains competitive even in light of more recent methods. Further details on the baseline implementations are provided in Appendix A.2.

4.3 Models

We test both open-source models of varying sizes and closed-source GPT models as evaluators. The models included in each category are as follows:[4] (1) **Large models** (70–123B): `LLama3.1-70B`, `Mistral-Large` (123B), `Qwen2.5-72B`. (2) **Medium models** (22–32B): `Mistral-Small` (22B), `Gemma2-27B`, `Qwen2.5-32B`. (3) **Small models** (7–9B): `LLama3.1-8B`, `Gemma2-9B`, `Qwen2.5-7B`, (4) **GPT models**: `GPT-4-Turbo`, `GPT-4o`, `GPT-4o-mini` (Achiam et al., 2023; Dubey et al., 2024; Jiang et al., 2023; Yang et al., 2024; Riviere et al., 2024).

4.4 Implementation Details

Following prior work (Liu et al., 2023b), we set `temperature` = 0, `n` = 1, and fix the random seed for both G-Eval and CheckEval. Additionally, We set `max_length` to 20 for G-Eval as it generates a single score, and 200 for CheckEval as it needs to generate responses to multiple checklist questions. We use the original prompts provided by the authors of G-Eval without any modifications. Example prompts for CheckEval are provided in the Appendix F. We evaluated multiple questions in the checklist within a single prompt to enhance efficiency and practicality rather than evaluating each question individually, as discussed in Section 3.3.

We use GPT-4o for both the question augmentation and filtering steps in the checklist generation stage. The total number of generated questions at each step is provided in Appendix D. For experiments, we use vLLM 0.6.3 (Kwon et al., 2023) with four A100 GPUs. The costs of evaluating all 1,600 samples from SummEval with GPT-4o are approximately $22.

[4]The links for each model are provided in Appendix E.

Model	Evaluation Methods	SummEval (Avg.)		Topical-Chat (Avg.)	
		ρ	τ	ρ	r
non-LLM-as-a-Judge					
	ROUGE-L	0.17	0.13	0.24	0.24
	BERTScore	0.23	0.18	0.25	0.24
	MOVERScore	0.47	0.38	0.22	0.24
	BARTScore	0.19	0.15	0.29	0.29
	UniEval	0.39	0.31	0.28	0.26
LLM-as-a-Judge					
Llama3.1-70B	G-Eval	0.40	0.36	0.45	0.39
	CheckEval	**0.46**	**0.40**	**0.57**	**0.57**
Mistral-Large	G-Eval	0.52	0.47	0.64	0.62
	CheckEval	**0.55**	**0.48**	**0.65**	**0.65**
Qwen2.5-72B	G-Eval	0.43	0.39	**0.62**	**0.61**
	CheckEval	**0.50**	**0.44**	0.59	0.60
Mistral-Small	G-Eval	0.18	0.16	**0.58**	**0.52**
	CheckEval	**0.45**	**0.39**	0.47	0.49
Gemma2-27B	G-Eval	0.44	0.39	0.31	0.29
	CheckEval	**0.51**	**0.44**	**0.53**	**0.52**
Qwen2.5-32B	G-Eval	0.50	**0.45**	0.46	0.38
	CheckEval	**0.52**	0.44	**0.56**	**0.56**
Llama3.1-8B	G-Eval	0.24	0.21	0.11	0.09
	CheckEval	**0.41**	**0.34**	**0.46**	**0.45**
Gemma2-9B	G-Eval	0.38	0.34	0.46	0.35
	CheckEval	**0.43**	**0.37**	**0.49**	**0.50**
Qwen2.5-7B	G-Eval	0.41	**0.38**	0.45	0.39
	CheckEval	**0.42**	0.37	**0.48**	**0.47**
GPT-4 Turbo	G-Eval	0.51	**0.46**	0.59	0.58
	CheckEval	**0.52**	**0.46**	**0.63**	**0.64**
GPT-4o	G-Eval	0.32	0.29	0.52	0.43
	CheckEval	**0.50**	**0.44**	**0.64**	**0.63**
GPT-4o-mini	G-Eval	0.45	0.40	0.58	0.56
	CheckEval	**0.49**	**0.42**	**0.59**	**0.59**

Table 1: Average correlation scores across dimensions on the benchmarks. For SummEval, we report sample-level ρ and τ. For Topical-Chat, we report turn-level ρ and r. Colors indicate model groups: large (pink), medium (blue), small (green) and GPT (purple). The best score per model category is bolded, and the highest overall score is marked with an underline.

Dataset	Correlation	p-value
SummEval	Spearman	0.005^{**}
	Kendall	0.043^{*}
Topical-Chat	Spearman	0.003^{**}
	Pearson	0.036^{*}

Table 2: Wilcoxon test p-values for different datasets and metrics after FDR correction. (*: $p < .05$, **: $p < .01$)

Evaluation Methods	Model	SummEval (Avg.)		Topical-Chat (Avg.)	
		ρ	τ	ρ	r
TIGERScore	LLaMA 2–13B†	0.39	0.31	0.28	0.26
GPTScore	GPT-4	0.39	0.34	0.36	0.34
G-Eval	Mistral-large	0.52	0.47	0.64	0.62
Analyze-Rate	Claude 3 Sonnet	0.53	0.44	0.64	0.64
HD-EVAL	GPT-4	0.53	–	0.62	0.63
SEEval	Claude 3 Sonnet	0.52	0.47	0.65	0.64
CheckEval	**Mistral-large**	**0.55**	**0.48**	**0.65**	**0.65**

Table 3: Comparative performance of various LLM-as-a-Judge methods. Models marked with † are fine-tuned.

Model Group	Evaluation Methods	SummEval (Avg.)		Topical-Chat (Avg.)	
		α	κ	α	κ
All	G-Eval	0.09	0.19	0.06	0.34
	CheckEval	**0.48**	**0.48**	**0.45**	**0.45**
Large	G-Eval	0.05	0.16	0.01	0.51
	CheckEval	**0.67**	**0.67**	**0.67**	**0.67**
Medium	G-Eval	0.04	0.14	0.07	0.22
	CheckEval	**0.56**	**0.56**	**0.50**	**0.50**
Small	G-Eval	0.06	0.10	0.04	0.16
	CheckEval	**0.24**	**0.24**	**0.17**	**0.17**
GPT	G-Eval	0.08	0.20	0.04	0.50
	CheckEval	**0.56**	**0.56**	**0.54**	**0.54**
Top-3	G-Eval	0.07	0.23	0.03	0.56
	CheckEval	**0.65**	**0.65**	**0.57**	**0.57**

Table 4: Inter-evaluator agreement (IEA) results for SummEval and Topical-Chat, comparing G-Eval and CheckEval across different model groups. Top-3 refers to the three models with the highest correlation to human judgments. The best score per model category is bolded.

5 Results

5.1 Correlation with Human Evaluation

Table 1 shows the correlation between various evaluation methods and human judgments on the SummEval and Topical-Chat datasets (detailed correlation results for all dimensions are shown in Table 18, 20 and 9 in the Appendix). We compare both non-LLM-as-a-Judge and LLM-as-a-Judge, with an emphasis on how CheckEval compares against G-Eval across 12 LLMs.

Excluding MOVERScore, most non-LLM-as-a-Judge metrics exhibit very low correlation with humans. Among LLM-as-a-Judge methods, CheckEval consistently achieves higher correlation with human judgments than G-Eval, with only a few exceptions of Qwen2.5 and Mistral-Small. These results suggest that CheckEval's fine-grained, checklist-based design more effectively captures subtle differences in text quality, leading to improved correlation with human judgments. When analyzing model sizes, large open-source models show strong performance, with Mistral-Large combined with CheckEval achieving the highest correlation among all models. Even in medium- and small-sized models–where evaluation capacity tends to be weaker–CheckEval maintains its advantage over G-Eval. Notably, some medium-sized models perform particularly well on SummEval, achieving correlations comparable to larger models. For GPT models, CheckEval consistently yields stronger correlations than G-Eval, particularly with GPT-4o.

To assess the statistical significance of the performance difference between CheckEval and G-Eval, we conducted Wilcoxon Rank-Sum Tests with False Discovery Rate adjusted using the Benjamini-Hochberg correction (Table 2). The results show that the distributions of average correlation scores derived from CheckEval and G-Eval are significantly different for both datasets.

We also included a broader comparison with recent evaluation methods surveyed in Gu et al. (2024); Gao et al. (2024b). For CheckEval and G-Eval, we use scores using the best-performing evaluator in our experiments (Mistral-large). Table 3 shows that CheckEval performs well overall on both datasets, and remains competitive even compared to more recent approaches. However, we would like to emphasize again that our main goal is not to propose the best-performing LLM-as-a-judge method. Instead, our focus is on building a more reliable evaluation process and analyzing its consistency across different LLMs, and that is why comparison to G-Eval is the most directly relevant result.

5.2 Inter-evaluator Agreement (IEA)

Table 4 compares the IEA of G-Eval and CheckEval on the SummEval and Topical-Chat datasets. We measure IEA using Krippendorff's α and Fleiss' κ, treating different LLMs within the same group (large, medium, small, GPT) as annotators. While correlation with human judgments is a main metric in LLM-as-a-Judge, **high correlation alone does not guarantee reliability**. Reliability is a desirable property for evaluation methods, as it ensures that different evaluator models assign similar

Correlation	ρ	τ
Mistral-large (C) vs. Humans (C)	0.73^{**}	0.58^{**}
Qwen2.5-72B (C) vs. Humans (C)	0.72^{**}	0.59^{***}
Llama3.1-70B (C) vs. Humans (C)	0.73^{**}	0.58^{**}
Humans (L) vs. Humans (C)	0.69^{**}	0.54^{***}
***Agreement** (dim: Relevance)*	# Annotators	κ
Humans	3	0.53
LLMs (Large) & Humans	6	0.49

Table 5: Human evaluation on SummEval. **C** denotes CheckEval, **L** denotes Likert (original Summeval Score). We use the LLM results from the large model group. (**: $p < .01$, ***: $p < .001$)

scores/rating to the same input. This reliability is critical yet overlooked in existing frameworks.

G-Eval demonstrates this limitation. It achieves fairly good correlation with human judgments but shows much lower IEA in general. Table 4 is evident when looking at G-Eval scores for Large and Top-3 best models,[5] and contrasting them with CheckEval's IEA. This indicates inconsistent scoring across different LLM evaluator models (of similar general capacity). We speculate that existing protocols like G-Eval's mainly lend themselves to inconsistencies in the following two ways: (1) the evaluation dimensions adopted encompass multiple distinct fine-grained criteria, making it difficult for LLMs to generate a consistent holistic score, and (2) adjacent Likert scale scores lack clear distinctions (e.g., 3 vs. 4) and are not calibrated well across models (Laban et al., 2023).

CheckEval's fine-grained checklist approach improves upon this limitation greatly. For the large models, CheckEval achieves best IEA scores of 0.67 (α and κ), on SummEval, which is comparable to IEA among human raters ($\kappa \approx 0.7$) (Fabbri et al., 2021), and 0.67 (α and κ) on Topical-Chat. Crucially, CheckEval maintains both high correlation and IEA across different LLMs and tasks. These results demonstrate that CheckEval provides a more reliable evaluation than G-Eval (See Table 19 and 21 for a detailed per-dimension IEA).

5.3 Human Validation

To further assess the validity of CheckEval, we asked human annotators to manually apply the same checklist and compared their scores with those generated by LLMs (Table 5). Details of the human validation setup are provided in Appendix A.4.

Correlation We sampled 20 summaries from the SummEval dataset. The random sampling was stratified based on original annotation scores to ensure balanced coverage of a wide range of quality levels. Three annotators evaluated each summary using the checklist, which contains approximately 25 binary (yes/no) questions per evaluation dimension. This resulted in roughly 2,000 binary annotations per annotator. For each summary, we aggregated checklist scores by summing the number of 'yes' responses per dimension, following the same method used for LLM outputs. We then computed correlation between these aggregated human scores and those from three large LLMs: Mistral-Large, Qwen2.5-72B, and Llama3.1-70B. In addition, we calculated correlation between the original Likert-scale scores from SummEval and the checklist-based human scores. All correlations are statistically significant, indicating that CheckEval scores successfully capture human judgments.

Agreement Due to the high annotation cost, we focused on `relevance` for agreement analysis. We collected binary annotations on 100 summaries (sample size selected based on a power analysis targeting 95% confidence interval width of ≤ 0.2 for IEA scores). Each annotator answered approximately 10,000 questions. We report inter-annotator agreement among the three human annotators, as well as agreement between the human group and the large LLM group. We observe high agreement between humans as well as between humans and LLMs, showing that CheckEval elicits consistent scores across both human and LLM raters.

6 Analysis

6.1 Stability Analysis of Evaluation Methods

We further analyze the stability of evaluation methods by examining the distribution of correlations with human judgments across different evaluator models. While agreement analysis (Section 5.2) focuses on how consistently models assess the same samples, stability evaluates whether an evaluation method maintains reliable alignment with human annotations across all evaluator models. As shown in Figure 2, CheckEval achieves higher mean correlations and lower variance than G-Eval on both datasets, demonstrating more stable evaluation across different models. Detailed correlation

[5] **CheckEval** (SummEval: `GPT-4-Turbo`, `Mistral-Large`, `Gemma2-27B`, Topical-Chat: `GPT-4-Turbo`, `GPT-4o`, `Mistral-Large`), and **G-Eval** (SummEval: `GPT-4-Turbo`, `GPT-4o-mini`, `Mistral-Large`, Topical-Chat: `GPT-4-Turbo`, `Mistral-Large`, `Qwen2.5-72B`

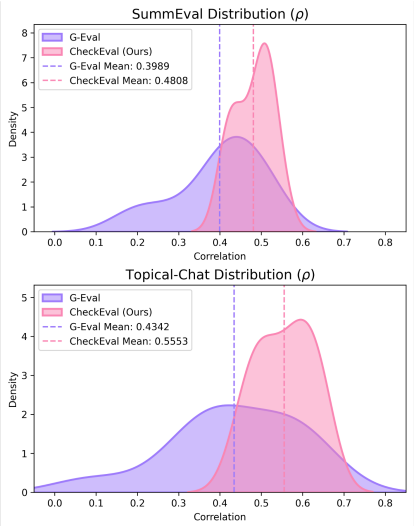


Figure 2: Kernel density estimation (KDE) of correlations with human judgments for G-Eval (purple) and CheckEval (pink) across different evaluator models on SummEval and Topical-Chat. Dashed lines indicate mean correlation values.

statistics, including full mean and variance values, are available in Table 14.

6.2 Case Study

We conduct a case study on the `naturalness` dimension in the Topical-Chat dataset to illustrate how CheckEval enhances explainability by explicitly showing which evaluation criteria contribute to the final score (see Table 6). We evaluate system responses generated by Mistral-large, the model with the strongest correlation with human judgments. For this case study, we normalize all scores to a 0–1 scale for direct comparison. On evaluating the given text on naturalness, CheckEval (0.88) aligns more closely with human judgments (1.0), rating the response as natural. In contrast, G-Eval (0.25) assigned a much lower naturalness score. More importantly, while G-Eval provides only a score without explanation, CheckEval's systematic decomposition into specific sub-questions helps us attribute the high score to individual questions with a 'yes' answer (e.g., the response is natural because it avoids repetition, the message is clear, etc.). An additional case study on low-quality samples from benchmark datasets is presented in Appendix C.1, further demonstrating how CheckEval operates across a wider range of text qualities.

6.3 Ablation Study

We conducted an ablation study to assess the contribution of each component in the CheckEval pipeline. Table 7 reports results when removing filtering and augmentation step. Both components contribute to overall performance, with the augmentation stage showing a slightly larger impact.

Conversation history (source)	
A: Hello, how are you today? Do you like to go to concerts? B: Not as much as I used to, but I do. A: Yeah, same here! Creed gave a concert so bad there were lawsuits against the band. ... B: I have no idea. I'm sure that someone has video of it. Do you enjoy the music of the Foo Fighters? A: Oh yes, I love them. I love the video of all the drummers and other instruments playing at the same time. People came from all over the world to be in that. B: They are pretty amazing. They performed a concert so loud that it showed up on New Zealand's seismic monitors!	
Context - *In 2002, a Creed concert was so bad that four concertgoers filed a lawsuit against the band.*	
System Response - *I know, I think I have heard that before, I think it was really cool.*	
Checklist	
Questions	**Answer**
Does the response avoid unnecessary repetition of the same content between sentences?	Yes
Does each sentence directly relate to the topic being discussed?	No
Is the overall message clear and easy to understand?	Yes
Does each sentence in the response convey a clear meaning?	Yes
Is the tone consistent throughout?	Yes
Does the response avoid using jargon or overly complex words that might confuse the listener?	Yes
Are there no major grammatical errors?	Yes
Are there no ambiguous terms or phrases that could confuse the reader?	Yes
Raw Scores - Human: **3** (1-3), G-Eval: **2** (1-5), CheckEval: **0.88** (0-1)	
Normalized Scores - Human: **1** (0-1), G-Eval: **0.25** (0-1), CheckEval: **0.88** (0-1)	

Table 6: Case study on the `naturalness` dimension in the Topical-Chat.

	SummEval	**Topical-Chat**
CheckEval	0.48	0.55
w/o filtering	0.48	0.54
w/o augmentation	0.46	0.53

Table 7: checkeval-ablation

7 Conclusion

We propose CheckEval, a fine-grained Boolean QA Checklist framework that addresses key limitations in existing LLM-as-Judge approaches for evaluating text generation. By decomposing evaluation criteria into structured binary questions, CheckEval enables reliable evaluation of (open-ended) text. Our experiments across various models and datasets demonstrate that CheckEval outperforms widely-adopted Likert scale-based methods like G-Eval, achieving higher correlation to human evaluation and IEA across different LLM evaluators. The framework shows particular strength in evaluating high-quality texts by effectively capturing subtle qualitative differences while maintaining explainability. Additionally, CheckEval enhances evaluation stability through reduced variance across LLMs. This shows that our framework offers a promising solution for constructing more reliable evaluation benchmarks across diverse NLG tasks.

8 Limitations

CheckEval improves the reliability of LLM-as-a-Judge evaluation, but it has several limitations. First, while automating checklist generation is a promising direction for improving scalability, it introduces challenges that are common to many automatic evaluation methods. CheckEval uses task-specific, human-written seed questions, which helps ground the evaluation in task-relevant criteria. However, as an automatic evaluation method, there may be factors beyond our control that lead to potential misalignment. In such cases, human involvement may be necessary to ensure alignment with task-specific goals. This is not a limitation of CheckEval specifically, but a broader challenge inherent to automatic evaluation approaches.

Second, this study focused on analyzing model-wise evaluation trends and comparing Likert-scale evaluation with Boolean QA checklist-based evaluation. However, recent LLM-as-a-Judge studies have introduced various techniques to enhance human alignment. Methods such as prompt optimization (e.g. chain-of-thought (Wei et al., 2022), self-correction (Xu et al., 2023)), multi-agent debate (Chan et al., 2024; Kim et al., 2024), and meta-evaluator training (Kim et al., 2023; Wu et al., 2024b; Zhu et al., 2025) enable LLMs to make more enhanced judgments. Therefore, future work should compare it against these approaches and analyze how it differs in terms of reliability. This would also help determine whether CheckEval can be combined with such techniques to build a more robust evaluation framework.

Third, while CheckEval's boolean-style decision improves evaluation reliability, not all NLG tasks and evaluation criteria can be strictly answered with a yes/no response. This limitation becomes more apparent when considering evaluation scenarios involving texts two to three times longer than those in the current benchmarks. As text length increases, some parts of a response may be strong while others are weak. For example, the first half of a response may be well-written and coherent, while the latter half is unclear or contains errors. This makes binary decisions insufficient for capturing subtle quality differences. The constraints of a yes/no format may become more pronounced in long-form evaluations, suggesting that future research should explore ways to mitigate this limitation while preserving the strengths of CheckEval.

Fourth, CheckEval's efficacy should be tested on a wider range of NLG tasks. While this study primarily focused on summarization and dialogue response generation, additional experiments are needed to validate CheckEval's applicability to tasks such as story generation, long-form question answering, machine translation, and dialogue generation. Given that evaluation criteria vary by domain, it is important to examine how well CheckEval generalizes across different task settings. In recent works, CheckEval has been used for tasks such as essay scoring, creative writing evaluation, and healthcare-related text assessment. Although we do not detail these applications here due to anonymity constraints, they demonstrate that CheckEval is already in active use beyond the scope of this paper.

Finally, improving the automation of checklist design and evaluation processes would enhance CheckEval's usability. Currently, checklist construction is a manual process tailored to specific tasks, making it difficult to predict the time and effort required for new evaluation domains. One potential solution is to pre-build a large-scale question database for NLG tasks and develop a system that automatically assembles relevant checklists based on task requirements. Future research should explore LLM-assisted checklist generation and reconfiguration methods to ensure that CheckEval can be efficiently applied to a broader range of tasks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ron Artstein. 2017. Inter-annotator agreement. *Handbook of linguistic annotation*, pages 297–313.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K. Surikuchi, Ece Takmaz, and Alberto Testoni. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *CoRR*, abs/2406.18403.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference*

on Natural Language Generation, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. *arXiv preprint arXiv:2205.06938*.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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, Xinyu Hu, Li Lin, and Xiaojun Wan. 2024a. Analyzing and evaluating correlation measures in nlg meta-evaluation. *arXiv preprint arXiv:2410.16834*.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024b. Llm-based nlg evaluation: Current status and challenges. *ArXiv*, abs/2402.01383.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Jing Gu, Qingyang Wu, and Zhou Yu. 2021. Perception score: A learned metric for open-ended text generation evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12902–12910.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2024. TIGERScore: Towards building explainable metric for all text generation tasks. *Transactions on Machine Learning Research*.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *arXiv preprint arXiv:2303.01432*.

Alex Kim, Keonwoo Kim, and Sangwon Yoon. 2024. DEBATE: Devil‘s advocate-based assessment and text evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1885–1897, Bangkok, Thailand. Association for Computational Linguistics.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. Longeval: Guidelines for human evaluation of faithfulness in long-form summarization. *arXiv preprint arXiv:2301.13298*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring

LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9662–9676, Singapore. Association for Computational Linguistics.

Bo Li, Irina Sigler, and Yuan Xue. 2024. Evaluating large language models - principles, approaches, and applications. Neurips 2024 Tutorial.

Margaret Li, Jason Weston, and Stephen Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.

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

Minqian Liu, Ying Shen, Zhiyang Xu, Yixin Cao, Eunah Cho, Vaibhav Kumar, Reza Ghanadan, and Lifu Huang. 2023a. X-eval: Generalizable multi-aspect text evaluation via augmented instruction tuning with auxiliary evaluation aspects. *arXiv preprint arXiv:2311.08788*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023c. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023d. Calibrating llm-based evaluator. *arXiv preprint arXiv:2309.13308*.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. HD-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7641–7660, Bangkok, Thailand. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Ani Nenkova and Rebecca J Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*, pages 145–152.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Qian Pan, Zahra Ashktorab, Michael Desmond, Martín Santillán Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. 2024. Human-centered design recommendations for LLM-as-a-judge. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, TBD. ACL.

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*, pages 311–318.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*.

Annalisa Szymanski, Simret Araya Gebreegziabher, Oghenemaro Anuyah, Ronald A Metoyer, and Toby

Jia-Jun Li. 2024. Comparing criteria development across domain experts, lay users, and models in large language model evaluation. *arXiv preprint arXiv:2410.02054*.

Hongyin Tang, Miao Li, and Beihong Jin. 2019. A topic augmented text generation model: Joint learning of semantics and structural features. 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 5090–5099, Hong Kong, China. Association for Computational Linguistics.

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

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. *arXiv preprint arXiv:2203.12990*.

Meng-Chen Wu, Md Mosharaf Hossain, Tess Wood, Shayan Ali Akbar, Si-Chi Chin, and Erwin Cornejo. 2025. SEEval: Advancing LLM text evaluation efficiency and accuracy through self-explanation prompting. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7357–7368, Albuquerque, New Mexico. Association for Computational Linguistics.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024a. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, and Sujian Li. 2024b. InstructEval: Instruction-tuned text evaluator from human preference. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13462–13474, Bangkok, Thailand. Association for Computational Linguistics.

Ziang Xiao, Susu Zhang, Vivian Lai, and Q. Vera Liao. 2023. Evaluating evaluation metrics: A framework for analyzing NLG evaluation metrics using measurement theory. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10967–10982, Singapore. Association for Computational Linguistics.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. Instructscore: Towards explainable text generation evaluation with automatic feedback. *arXiv preprint arXiv:2305.14282*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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. *arXiv preprint arXiv:1909.02622*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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. *arXiv preprint arXiv:2210.07197*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. JudgeLM: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations*.

A Detailed Experimental Setup

A.1 Datasets

We use three meta-evaluation benchmarks for various tasks to measure the effectiveness of CheckEval. **SummEval** (Fabbri et al., 2021) is a benchmark designed for the meta-evaluation of summarization. SummEval includes human evaluations for each generated summary across four dimensions: `coherence`, `consistency`, `fluency`, and `relevance`. **Topical-Chat** (Gopalakrishnan et al., 2019) serves as a benchmark for meta-evaluating evaluation methods for knowledge-grounded dialogue systems. Following Zhong et al. (2022), we evaluate our method using human ratings across four dimensions: `naturalness`, `coherence`, `engagingness`, and `groundedness`. **QAGS** (Wang et al., 2020) is a benchmark designed to assess hallucinations in summarization. It focuses on evaluating the factual consistency of summaries.

A.2 Baselines

Baselines for main comparison (Table 1) (1) **BERTScore** (Zhang et al., 2019) calculates text similarity by contextual embeddings of BERT (Devlin et al., 2018). (2) **MoverScore** (Zhao et al., 2019) extends BERTScore by incorporating soft alignments, allowing words to be dynamically matched across texts. It refines similarity computation through an improved aggregation strategy that accounts for word importance and semantic shifts. (3) **BARTScore** (Yuan et al., 2021) evaluates text quality by computing the average likelihood of a generated output under a BART-based conditional probability model. (4) **UniEval** (Zhong et al., 2022) is a multi-dimensional evaluation framework that assesses various dimensions of text generation by leveraging both reference-based and reference-free evaluation. (5) **G-Eval** (Liu et al., 2023b) is an LLM-based method, using chain-of-thought (Wei et al., 2022) and a form-filling paradigm to generate evaluation scores on a Likert scale. We select G-Eval as the main comparison point due to its widespread adoption (Liu et al., 2023a, 2024), as well as considering the similarity between G-Eval and CheckEval that neither approach involves complex prompt engineering, additional model training or multi-agent evaluation.

Baselines for Comparative Analysis (Table 3) (1) **TIGERScore** (Jiang et al., 2024) is a Llama 2 fine-tuned evaluation method that uses LLM to perform an explainable text evaluation. (2) **GPTScore** (Fu et al., 2023) evaluates text by computing the conditional log-likelihood of reference or output generated under LLM. (3) **Analyze-Rate** (Chiang and Lee, 2023) analyzes how specific design choices in LLM-based evaluation, such as explanation prompting and output format, affect alignment with human judgment and finds that encouraging explanation improves correlation. (4) **HD-EVAL** (Liu et al., 2024) decomposes the evaluation into fine-grained criteria and trains a regression model to aggregate them in alignment with human preferences through iterative preference-based optimization. (5) **SEEval** (Wu et al., 2025) is a prompt-based evaluator that incorporates self-explanation, guiding the model to justify its rating decisions without additional training.

A.3 Detailed Process of Seed Question Writing

We constructed seed questions based on predefined evaluation criteria (e.g., coherence, consistency), aiming for atomic, conceptually clear, and non-overlapping formulations. Each evaluation dimension was first decomposed into finer-grained sub-dimensions, and a set of seed questions was written to cover each sub-dimension. This ensured both conceptual coverage and balance across dimensions. To guide this process, we consulted prior task-specific literature (e.g., summarization evaluation papers) and followed established guidelines where available. We observed that overly fine-grained seed questions often led LLMs to generate augmented variants that deviated from the original intent. Therefore, we intentionally maintained an appropriate granularity level to preserve alignment throughout augmentation. All seed questions were cross-validated by our team to ensure clarity, consistency, and relevance across different evaluation dimensions.

A.4 Human Validation

To validate CheckEval, we conducted two human evaluation studies (correlation and agreement study: Section 5.3). For these studies, summaries were randomly sampled from the SummEval dataset using stratification based on original human annotation scores to ensure balanced coverage across quality levels. Each study involves three Ph.D student-level evaluators. We recruited three human evaluators with Ph.D. student-level qualifications or above in Computer Science, all

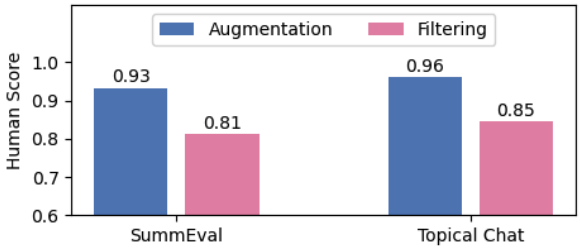


Figure 3: Average checklist validation scores for all dimensions from SummEval and Topical-chat. The 'Augmentation' represents the percentage of augmented questions that fulfilled the specified quality criteria, and 'Filtering' represents the percentage for filtered questions.

Model	Evaluation Method	SummEval (Avg.)	
		ρ	τ
Mistral-Large	CheckEval	0.5486	0.4797
	CheckEval #	0.5486	0.4797
		Topical-Chat (Avg.)	
		ρ	r
	CheckEval	0.6451	0.6453
	CheckEval #	0.6443	0.6412
		QAGS (Avg.)	
		r	ρ
	CheckEval	0.6681	0.6558
	CheckEval #	0.6680	0.6558

Table 8: Effect of applying additional human filtering to Mistral-Large. # indicates that filtering was applied.

of whom had a background in evaluation research and summarization/dialogue tasks. Each participant was compensated with a gift card equivalent to approximately 10,000 KRW (≈ 7 USD) per hour.[6]

For the correlation study (Table 5 - Correlation), 20 summaries are randomly sampled from the SummEval dataset. These summaries are subsequently evaluated on a binary (yes/no) basis against a checklist comprising four dimensions: `coherence`, `consistency`, `fluency`, and `relevance`.

For the agreement study (Table 5 - Agreement), 100 summaries are sampled from the SummEval dataset. These summaries are then evaluated on a binary (yes/no) basis concerning only `relevance` due to practical cost constraints (evaluation this dimension alone already requires each annotator to answer approximately 10K questions). The sample size of 100 was calculated from a power analysis based on a pilot study.

B Additional Results

B.1 Checklist Validation

To veirfy the each stage of the checklist generation process worked as intended, we conducted an additional human evaluation focused on checklist quality. This evaluation validates the augmentation stage (seed questions, augmented questions), and filtering stage (seed questions, filtered questions) on both the SummEval and Topical-Chat datasets. Human evaluators are tasked with assessing each question on a binary (yes/no) basis, determining whether it satisfies the augmentation and filtering criteria. Figure 3 shows the average scores derived from the checklist validation evaluation for both the SummEval and Topical-Chat datasets. The augmentation stage consistently achieves very high average scores across both datasets (above 90%), which suggests that the question augmentation process of CheckEval is highly effective. The filtering stage yields slightly lower scores but remains competitive. We observed that annotators often expected 1–2 additional questions per dimensions to be filtered. Comments from annotators suggest that these questions were mostly semantically overlapping questions that the filter failed to capture.

To test whether removing these remaining questions would affect evaluation results, we conducted a follow-up experiment by applying an additional human-curated filtering step. We used Mistral-Large, the best-performing model, for this experiment. As shown in As shown in Table 8, the correlation scores after applying the additional filtering were extremely similar to the original results, with only minor drops. This indicates that removing one or two additional questions per evaluation dimension does not meaningfully impact the evaluation behavior, suggesting that CheckEval's automatic filtering is functioning effectively in practice.

B.2 Additional experiments with QAGS

Table 9 shows the correlation between various evaluation methods and human judgments on the QAGS dataset. The results show that CheckEval outperforms G-Eval for 9 out of the 12 LLMs (comparable to results on the other two datasets reported in the main text), indicating its effectiveness as an evaluation Framework. Furthermore, Table 10 compares the IEA of G-Eval and CheckEval on the QAGS dataset. Across all model groups, CheckEval consistently achieves a higher IEA than G-Eval,

[6] Note that the annotation was conducted in South Korea, where the compensation level is slightly above the local minimum wage.

Model	Evaluation Methods	CNN			Xsum			Average		
		r	ρ	τ	r	ρ	τ	r	ρ	τ
Llama3.1-70B	G-Eval	0.5097	0.4559	0.4261	0.2317	0.2317	0.2317	0.3707	0.3438	0.3289
	CheckEval	0.7002	0.6747	0.5683	0.5394	0.5018	0.4355	**0.6198**	**0.5883**	**0.5019**
Mistral-Large	G-Eval	0.5617	0.6104	0.5705	0.5834	0.5834	0.5834	0.5726	0.5969	0.5770
	CheckEval	0.7472	0.7291	0.6277	0.5889	0.5825	0.5352	**0.6681**	**0.6558**	**0.5815**
Qwen2.5-72B	G-Eval	0.6830	0.7154	0.6686	0.5236	0.5236	0.5236	0.6033	**0.6195**	**0.5961**
	CheckEval	0.7312	0.7013	0.6078	0.4931	0.4898	0.4197	**0.6122**	0.5956	0.5138
Mistral-Small	G-Eval	0.5656	0.5425	0.5070	0.4833	0.4833	0.4833	0.5245	0.5129	**0.4952**
	CheckEval	0.6563	0.6211	0.5239	0.4950	0.4496	0.3890	**0.5757**	**0.5354**	0.4565
Gemma2-27B	G-Eval	0.6124	0.6543	0.6115	0.5644	0.5644	0.5644	**0.5884**	**0.6094**	**0.5880**
	CheckEval	0.6975	0.6493	0.5397	0.4547	0.4040	0.3482	0.5761	0.5267	0.4440
Qwen2.5-32B	G-Eval	0.6487	0.6357	0.5941	0.4290	0.4290	0.4290	0.5389	0.5324	0.5116
	CheckEval	0.7286	0.7132	0.6145	0.5532	0.5231	0.4547	**0.6409**	**0.6182**	**0.5346**
Llama3.1-8B	G-Eval	0.2785	0.2228	0.2082	0.0614	0.0614	0.0614	0.1700	0.1421	0.1348
	CheckEval	0.6100	0.5995	0.4924	0.4244	0.4292	0.3669	**0.5172**	**0.5144**	**0.4297**
Gemma2-9B	G-Eval	0.6599	0.7002	0.6544	0.5546	0.5546	0.5546	**0.6073**	**0.6274**	**0.6045**
	CheckEval	0.5353	0.5713	0.4597	0.4502	0.4529	0.3875	0.4928	0.5121	0.4236
Qwen2.5-7B	G-Eval	0.4688	0.4307	0.4025	0.2137	0.2137	0.2137	0.3413	0.3222	0.3081
	CheckEval	0.6157	0.5672	0.4775	0.4419	0.4681	0.4063	**0.5288**	**0.5177**	**0.4419**
GPT-4 Turbo	G-Eval	0.4941	0.5402	0.5049	0.5560	0.5560	0.5560	0.5251	0.5481	0.5305
	CheckEval	0.7155	0.7211	0.6363	0.5922	0.5658	0.4961	**0.6539**	**0.6435**	**0.5662**
GPT-4o	G-Eval	0.2864	0.3100	0.2897	0.0582	0.0582	0.0582	0.1723	0.1841	0.1740
	CheckEval	0.6724	0.6601	0.5452	0.5448	0.5282	0.4564	**0.6086**	**0.5942**	**0.5008**
GPT-4o-mini	G-Eval	0.5424	0.5833	0.5136	0.4591	0.4591	0.4212	0.5008	0.5212	0.4674
	CheckEval	0.6175	0.6340	0.5451	0.4394	0.4831	0.4591	**0.5285**	**0.5586**	**0.5021**

Table 9: Average correlation scores across dimensions on the QAGS-CNN and QAGS-Xsum. we report r, ρ and τ. Colors indicate model groups: large (pink), medium (blue), small (green) and GPT (purple).

Model Group	Evaluation Methods	CNN		Xsum	
		α	κ	α	κ
All	G-Eval	0.2215	0.3624	0.2873	0.2853
	CheckEval	**0.4149**	**0.4149**	**0.3416**	**0.3416**
Large	G-Eval	0.1595	0.3345	0.1166	0.3772
	CheckEval	**0.6420**	**0.6420**	**0.5189**	**0.5189**
Medium	G-Eval	0.0526	0.5612	0.0546	0.3458
	CheckEval	**0.5971**	**0.5970**	**0.4074**	**0.4074**
Small	G-Eval	0.0805	0.0761	0.1796	0.0440
	CheckEval	**0.0846**	**0.0846**	**0.1881**	**0.1880**
GPT	G-Eval	0.0625	0.3920	0.1674	0.2156
	CheckEval	**0.4720**	**0.4719**	**0.2998**	**0.2997**
Top-3	G-Eval	0.0489	0.4845	0.0349	0.4381
	CheckEval	**0.5234**	**0.5234**	**0.5066**	**0.5066**

Table 10: IEA - QAGS

demonstrating its advantage in robustness.

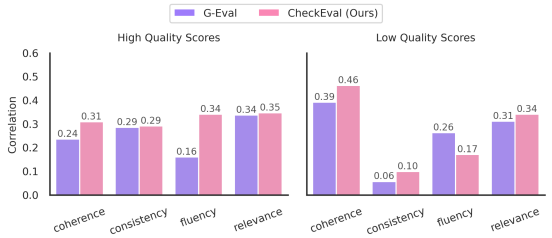
C Discussion

C.1 Analysis of Performance on High and Low-Quality Texts

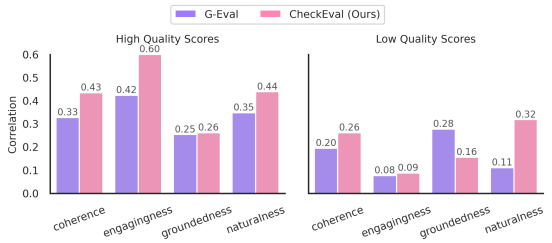
As LLMs improve, their high-quality outputs become more fluent and coherent, making it increasingly difficult for evaluation methods to differentiate subtle quality differences. Meanwhile, low-quality text poses a different challenge, as its overall readability is low, obscuring distinctions between evaluation criteria and making it harder to properly assess all target dimensions of quality. Given these differences, it is important to assess how evaluation methods handle varying levels of text quality. To this end, we conduct a detailed dimension-wise analysis by dividing the data into high-quality and low-quality groups based on human annotation scores (e.g., on a 1–5 scale, treat scores $\geq$3 as High, $<$3 as Low). We compute the average correlation across 12 LLMs to analyze how CheckEval and G-Eval align with human judgments for different levels of text quality.

As shown in Figure 4, CheckEval consistently achieves higher correlations with human judgments than G-Eval in high-quality texts across all dimensions. Notably, for SummEval, CheckEval shows much stronger alignment in `fluency` (0.34 vs. 0.16). For Topical-Chat, it outperforms G-Eval in `engagingness` (0.60 vs. 0.42) and `naturalness` (0.44 vs. 0.35) by a large margin.

However, for low-quality texts, while CheckEval generally maintains stronger correlations com-



(a) SummEval



(b) Topical-Chat

Figure 4: dimension-wise correlation analysis of G-Eval (purple) and CheckEval (pink), with samples divided based on human annotator ratings into High-Quality (human ratings $\geq$3) and Low-Quality (human ratings $<$3) groups. Each bar represents correlation with human judgments across different quality dimensions.

Model	Aggregation Strategy	SummEval (Avg.)	
		ρ	τ
Llama3.1-70B	original	0.4628	0.4037
	weighted	0.4674 ($\pm$0.015)	0.4046 ($\pm$0.016)
Mistral-Large	original	0.5486	0.4797
	weighted	0.5320 ($\pm$0.021)	0.4622 ($\pm$0.021)
Qwen2.5-72B	original	0.5024	0.4413
	weighted	0.5002 ($\pm$0.0130)	0.4360 ($\pm$0.006)
Mistral-Small	original	0.4473	0.3938
	weighted	0.4424 ($\pm$0.029)	0.3920 ($\pm$0.029)
Gemma2-27B	original	0.5108	0.4426
	weighted	0.5063 ($\pm$0.008)	0.4361 ($\pm$0.006)
Qwen2.5-32B	original	0.5193	0.4566
	weighted	0.5093 ($\pm$0.006)	0.4422 ($\pm$0.005)
Llama3.1-8B	original	0.4342	0.3654
	weighted	0.3752 ($\pm$0.009)	0.3191 ($\pm$0.008)
Gemma2-9B	original	0.4186	0.3607
	weighted	0.4561 ($\pm$0.005)	0.3920 ($\pm$0.003)
Qwen2.5-7B	original	0.4162	0.3652
	weighted	0.4026 ($\pm$0.023)	0.3545 ($\pm$0.019)
GPT-4 Turbo	original	0.5212	0.4633
	weighted	0.5182 ($\pm$0.003)	0.4563 ($\pm$0.001)
GPT-4o	original	0.5042	0.4377
	weighted	0.4771 ($\pm$0.026)	0.4113 ($\pm$0.023)
GPT-4o-mini	original	0.4913	0.4157
	weighted	0.4817 ($\pm$0.013)	0.4032 ($\pm$0.008)

Table 11: Effect of question weighting strategy on SummEval.

pared to G-Eval, it exhibits performance drops in a small number of cases, notably in `fluency` (SummEval) and `groundedness` (Topical-Chat). From our additional analysis of the results, one possible explanation is that discrepancies between benchmark definitions and actual human annotations of these dimensions may have contributed to the observed performance drop in CheckEval. For example, while SummEval defines `fluency` as the absence of formatting issues, capitalization errors, or ungrammatical sentence structures that hinder readability, human annotators often prioritized overall readability over strict grammatical correctness. Since CheckEval relies on fine-grained Boolean QA decisions aligned with predefined criteria, the correlation with human scores may be impacted when human annotation practices deviate from the exact evaluation guidelines. In the `groundedness` dimension of Topical-Chat, a different issue arises. For low-quality texts, CheckEval's strict yes/no framework often results in uniformly low scores, making it difficult to distinguish between varying degrees of poor responses. In contrast, G-Eval, which allows for more gradient judgments, showed advantages in these cases. This suggests potential refinements to the Boolean QA framework to better handle annotation inconsistencies while preserving its fine-grained evaluation capability.

C.2 Does CheckEval need question weighting?

We conducted an additional analysis to investigate whether incorporating question-specific weights improves the reliability of CheckEval scores. Motivated by HD-Eval (Liu et al., 2024), we trained a linear regression model using 20% of the SummEval data to estimate the relative importance (i.e., weights) of each checklist question. These weights were then used to compute a weighted CheckEval score. To assess robustness, the process was repeated across five random seeds, each sampling a different 20% subset of the data. Table 11 reports the average results and standard deviation across seeds. "original" denotes the unweighted CheckEval score, while "weighted" denotes the score after applying the learned question-specific weights. The overall results were mixed. A couple of evaluator models benefited from learning the weights, but most others did not. Since there were no reliable gains from weighting the questions, we ultimately chose not to incorporate weighted aggregation into our results. While we only experimented with a simple linear weighting strategy here, we could explore more sophisticated methods of estimating question importance as well as learning weights that are generalizable across different evaluator models in future work.

	Coherence	Consistency	Fluency	Relevance
Seed Questions	3	3	4	5
Diversification	7	12	11	5
Elaboration	13	14	24	21
Filtered Questions	0	0	4	5
Final Checklist	23	29	35	26

Table 12: The number of questions - SummEval

	Naturalness	Coherence	Engagingness	Groundedness
Seed Questions	5	4	4	5
Diversification	9	6	10	6
Elaboration	14	11	17	15
Filtered Questions	0	1	0	0
Final Checklist	28	20	31	26

Table 13: The number of questions - Topical-Chat

D The number of questions at each stage

We provide a step-by-step breakdown of the number of questions, from the initial seed questions through the augmentation and filtering stages to the final checklist, with the number of questions varying across different dimensions. Before and after filtering, the correlation shows slight variations. For the SummEval, Spearman's ρ changed from 0.4790 to 0.4816, while Kendall's τ changed from 0.4143 to 0.4163. In the Topical-Chat, Pearson's r remained unchanged at 0.5553, whereas Spearman's ρ increased from 0.5446 to 0.5546. The number of questions for each dataset is reported in Table 12 and 13, respectively.

E Information of open-source models

Table 15 provides links to all open-source models used in our experiments. Table 16 lists each model along with its corresponding license. Table 17 summarizes the datasets used and their associated licenses. If a dataset is publicly available but no explicit license is provided, we denote the license as '–' in the table.

F Prompts

Figure 5 and 6 shows the detailed evaluation prompt. Figure 7 and 8 shows the detailed augmentation prompt. Figure 9 shows the filtering prompt.

Dataset	Correlation	Method	Mean	Variance
SummEval	Spearman	G-Eval	0.3989	0.0100
		CheckEval	0.4808	0.0019
	Kendall	G-Eval	0.3647	0.0084
		CheckEval	0.4163	0.0016
Topical-Chat	Spearman	G-Eval	0.4342	0.0220
		CheckEval	0.5553	0.0043
	Pearson	G-Eval	0.4797	0.0205
		CheckEval	0.5546	0.0042

Table 14: Mean and variance for each dataset and correlation method

Model	Link
Llama3.1-70B	https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
Mistral-large (123B)	https://huggingface.co/mistralai/Mistral-Large-Instruct-2411
Qwen2.5-72B	https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
Mistral-Small (22B)	https://huggingface.co/mistralai/Mistral-Small-Instruct-2409
Gemma2-27B	https://huggingface.co/google/gemma-2-27b
Qwen2.5-32B	https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
Llama3.1-8B	https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
Gemma2-9B	https://huggingface.co/google/gemma-2-9b-it
Qwen2.5-7B	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Table 15: Model Links

Models	License
meta-llama/Llama-3.1-70B-Instruct	llama3.1
mistralai/Mistral-Large-Instruct-2411	mrl
Qwen/Qwen2.5-72B-Instruct	qwen
mistralai/Mistral-Small-Instruct-2409	mrl
google/gemma-2-27b	gemma
Qwen/Qwen2.5-32B-Instruct	Apache license 2.0
meta-llama/Llama-3.1-8B-Instruct	llama3.1
google/gemma-2-9b-it	gemma
Qwen/Qwen2.5-7B-Instruct	Apache license 2.0
GPT-4 Turbo	Proprietary
GPT-4o	Proprietary
GPT-4o-mini	Proprietary

Table 16: List of models and their corresponding licenses.

Datasets	License
SummEval	MIT license
Topical-chat	CDLA-Sharing-1.0
QAGS	-

Table 17: List of datasets and their corresponding licenses.

Model	Evaluation Methods	Coherence		Consistency		Fluency		Relevance		Average	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
non-LLM-as-a-judge											
	ROUGE-L	0.0990	0.1150	0.1280	0.0920	0.1050	0.0840	0.2840	0.2370	0.1650	0.1280
	BERTScore	0.2840	0.2110	0.1100	0.0900	0.1930	0.1580	0.3120	0.2430	0.2250	0.1750
	MOVERScore	0.5750	0.4420	0.4800	0.3710	0.4490	0.3710	0.5620	0.3250	0.4740	0.3770
	BARTScore	0.1590	0.1180	0.1570	0.1270	0.1290	0.1050	0.3180	0.2440	0.1910	0.1480
	UniEval	0.4480	0.3520	0.3820	0.3150	0.3560	0.2920	0.3560	0.2730	0.3850	0.3050
LLM-as-a-judge											
Llama3.1-70B	G-Eval	0.5206	0.4459	0.3513	0.3306	0.3104	0.2924	0.4371	0.3800	0.4048	0.3622
	CheckEval	0.6222	0.5264	0.5406	0.4913	0.2637	0.2288	0.4248	0.3682	**0.4628**	**0.4037**
Mistral-Large	G-Eval	0.5892	0.5078	0.6153	0.5824	0.3611	0.3435	0.5026	0.4368	0.5171	0.4676
	CheckEval	0.6439	0.5424	0.6132	0.5668	0.4563	0.3926	0.4811	0.4169	**0.5486***	**0.4797***
Qwen2.5-72B	G-Eval	0.3937	0.3420	0.5248	0.4903	0.3202	0.3050	0.4762	0.4178	0.4287	0.3888
	CheckEval	0.5778	0.4932	0.5490	0.5047	0.4113	0.3582	0.4717	0.4092	**0.5025**	**0.4413**
Mistral-Small	G-Eval	0.2885	0.2463	0.2748	0.2532	0.0134	0.0126	0.1629	0.1343	0.1849	0.1616
	CheckEval	0.5297	0.4531	0.5113	0.4712	0.3098	0.2670	0.4381	0.3837	**0.4472**	**0.3937**
Gemma2-27B	G-Eval	0.5731	0.4951	0.5111	0.4684	0.1596	0.1520	0.5239	0.4515	0.4419	0.3917
	CheckEval	0.6199	0.5244	0.4924	0.4485	0.4402	0.3756	0.4906	0.4220	**0.5108**	**0.4426**
Qwen2.5-32B	G-Eval	0.5361	0.4682	0.5550	0.5199	0.3606	0.3420	0.5363	0.4703	0.4970	**0.4501**
	CheckEval	0.6056	0.4938	0.5311	0.4767	0.4879	0.4157	0.4605	0.3797	**0.5213**	0.4415
Llama3.1-8B	G-Eval	0.2689	0.2253	0.2988	0.2763	0.0088	0.0087	0.3644	0.3139	0.2352	0.2060
	CheckEval	0.5045	0.4048	0.4561	0.3887	0.3040	0.2654	0.3933	0.3168	**0.4145**	**0.3439**
Gemma2-9B	G-Eval	0.5649	0.4895	0.4555	0.4206	-0.0252	-0.0221	0.5272	0.4602	0.3806	0.3370
	CheckEval	0.5777	0.4876	0.3979	0.3450	0.2798	0.2358	0.4590	0.4003	**0.4286**	**0.3672**
Qwen2.5-7B	G-Eval	0.3785	0.3270	0.5343	0.5020	0.3309	0.3146	0.4154	0.3617	0.4148	**0.3763**
	CheckEval	0.4068	0.3398	0.4214	0.3800	0.4598	0.4226	0.3768	0.3183	**0.4162**	0.3652
GPT-4 Turbo	G-Eval	0.4912	0.4251	0.6498	0.6229	0.3878	0.3668	0.5064	0.4397	0.5088	**0.4636**
	CheckEval	0.5807	0.4901	0.6232	0.5872	0.4611	0.4058	0.4197	0.3713	**0.5212**	**0.4636**
GPT-4o	G-Eval	0.1896	0.1581	0.4219	0.3911	0.2862	0.2676	0.3969	0.3421	0.3237	0.2897
	CheckEval	0.5564	0.4644	0.5304	0.4738	0.4699	0.4125	0.4602	0.4001	**0.5042**	**0.4377**
GPT-4o-mini	G-Eval	0.4826	0.4197	0.5243	0.4837	0.2734	0.2598	0.5192	0.4524	0.4499	0.4039
	CheckEval	0.5854	0.4829	0.4939	0.4286	0.3883	0.3314	0.4975	0.4199	**0.4913**	**0.4157**

Table 18: Sample-level Spearman (ρ) and Kendall tau (τ) correlations on the SummEval benchmark. Colors indicate different model sizes: GPT (purple), large (pink), medium (blue), and small (green). The best score per model category is **bolded**, and the highest overall score is marked with *.

Model Group	Evaluation Methods	Coherence		Consistency		Fluency		Relevance		Average	
		α	κ	α	κ	α	κ	α	κ	α	κ
All	G-Eval	0.0751	0.2706	0.0539	0.1625	0.1626	0.0699	0.0799	0.2407	0.0929	0.1859
	CheckEval	0.4242	0.4242	0.2963	0.2963	0.4422	0.4422	0.7584	0.7584	0.4803	0.4803
Large	G-Eval	0.0448	0.2170	0.0476	0.0057	0.0621	0.2372	0.0502	0.1745	0.0512	0.1586
	CheckEval	0.7154	0.7154	0.5757	0.5757	0.5207	0.5206	0.8806	0.8806	0.6731	0.6731
Medium	G-Eval	0.0096	0.3742	0.0229	0.1306	0.0970	-0.1462	0.0424	0.2057	0.0430	0.1411
	CheckEval	0.6455	0.6455	0.2723	0.2723	0.5851	0.5851	0.7440	0.7440	0.5617	0.5617
Small	G-Eval	0.0704	0.2237	0.0044	0.1351	0.1089	-0.1161	0.0702	0.1564	0.0635	0.0998
	CheckEval	0.0827	0.0826	0.0237	0.0237	0.1746	0.1746	0.6739	0.6739	0.2387	0.2387
GPT	G-Eval	0.1425	0.1513	0.0984	0.0823	0.0064	0.3388	0.0889	0.2347	0.0841	0.2018
	CheckEval	0.5081	0.5081	0.4135	0.4135	0.5473	0.5473	0.7612	0.7612	0.5575	0.5575
Top-3	G-Eval	0.1104	0.2360	0.1002	0.0544	0.0171	0.3751	0.0647	0.2407	0.0731	0.2266
	CheckEval	0.6236	0.6236	0.4836	0.4836	0.6698	0.6698	0.8114	0.8114	0.6471	0.6471

Table 19: IEA - SummEval

Model	Evaluation Methods	Coherence		Engagingness		Groundedness		Naturalness		Average	
		ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
non-LLM-as-a-judge											
	ROUGE-L	0.1930	0.2030	0.2950	0.2840	0.3100	0.3270	0.1760	0.1450	0.2430	0.2440
	BERTScore	0.2140	0.2330	0.5170	0.3350	0.2910	0.3170	0.2560	0.2090	0.2520	0.2370
	MOVERScore	0.2470	0.2590	0.2750	0.2690	0.1980	0.1470	0.1690	0.1700	0.2220	0.2380
	BARTScore	0.2510	0.2250	0.4110	0.4060	0.1920	0.2050	0.2660	0.1560	0.2930	0.2850
	UniEval	0.2020	0.2050	0.5730	0.4300	0.1220	0.1530	0.2340	0.2360	0.2830	0.2620
LLM-as-a-judge											
Llama3.1-70B	G-Eval	0.4089	0.3622	0.3968	0.3501	0.6190	0.5553	0.3684	0.2991	0.4483	0.3917
	CheckEval	0.5517	0.5360	0.6547	0.6551	0.4706	0.4917	0.6065	0.6082	**0.5709**	**0.5727**
Mistral-Large	G-Eval	0.5709	0.5699	0.7135	0.6996	0.6217	0.5703	0.6494	0.6307	0.6389	0.6176
	CheckEval	0.6269	0.6174	0.7215	0.7206	0.5806	0.5766	0.6512	0.6664	**0.6451***	**0.6452***
Qwen2.5-72B	G-Eval	0.5650	0.5507	0.6944	0.6870	0.6122	0.6217	0.5927	0.5812	**0.6161**	**0.6101**
	CheckEval	0.5551	0.5506	0.7204	0.7199	0.4769	0.4873	0.6252	0.6398	0.5944	0.5994
Mistral-Small	G-Eval	0.4439	0.4215	0.6550	0.6411	0.6939	0.5102	0.5103	0.4996	**0.5758**	**0.5181**
	CheckEval	0.3925	0.4225	0.6061	0.5914	0.4789	0.4826	0.4191	0.4777	0.4742	0.4935
Gemma2-27B	G-Eval	0.4086	0.4337	0.3286	0.2928	0.2680	0.2361	0.2173	0.1953	0.3056	0.2895
	CheckEval	0.5036	0.4952	0.6390	0.6323	0.3794	0.3718	0.5825	0.5714	**0.5261**	**0.5177**
Qwen2.5-32B	G-Eval	0.4834	0.4515	0.3663	0.2697	0.4616	0.3082	0.5367	0.4924	0.4620	0.3804
	CheckEval	0.4918	0.4702	0.6914	0.6806	0.4139	0.4363	0.6300	0.6350	**0.5568**	**0.5555**
Llama3.1-8B	G-Eval	0.1109	0.1013	0.1031	0.0813	0.1702	0.0959	0.0667	0.0765	0.1127	0.0887
	CheckEval	0.5046	0.4986	0.5200	0.5069	0.3972	0.3934	0.4050	0.3876	**0.4567**	**0.4466**
Gemma2-9B	G-Eval	0.4357	0.3879	0.5512	0.4123	0.4742	0.3055	0.3681	0.2969	0.4573	0.3507
	CheckEval	0.3943	0.4232	0.6520	0.6588	0.4167	0.4136	0.4971	0.5137	**0.4900**	**0.5023**
Qwen2.5-7B	G-Eval	0.4625	0.4540	0.5496	0.5111	0.3346	0.1429	0.4459	0.4421	0.4481	0.3875
	CheckEval	0.3704	0.3840	0.6329	0.6266	0.4712	0.4247	0.4489	0.4486	**0.4809**	**0.4710**
GPT-4 Turbo	G-Eval	0.4924	0.4719	0.7026	0.6900	0.6112	0.6126	0.5724	0.5512	0.5947	0.5814
	CheckEval	0.5209	0.5232	0.7367	0.7438	0.6292	0.6341	0.6425	0.6476	**0.6323**	**0.6372**
GPT-4o	G-Eval	0.5917	0.5669	0.6111	0.5770	0.3903	0.1655	0.4770	0.4255	0.5175	0.4337
	CheckEval	0.5889	0.5790	0.7362	0.7354	0.5869	0.5761	0.6462	0.6448	**0.6395**	**0.6338**
GPT-4o-mini	G-Eval	0.5424	0.5333	0.6024	0.5623	0.5748	0.5744	0.5977	0.5756	0.5793	0.5614
	CheckEval	0.5140	0.5171	0.5980	0.5984	0.6362	0.6241	0.6038	0.6160	**0.5880**	**0.5889**

Table 20: Turn-level Spearman (ρ) and Pearson (r) correlations on the Topical-Chat. The best score per model category is **bolded**, and the highest overall score is marked with *.

Model Group	Evaluation Methods	Coherence		Consistency		Fluency		Relevance		Average	
		α	κ	α	κ	α	κ	α	κ	α	κ
All	G-Eval	0.0651	0.3051	0.0418	0.3263	0.0825	0.4443	0.0462	0.2871	0.0589	0.3407
	CheckEval	0.4796	0.4796	0.4354	0.4354	0.3995	0.3995	0.4830	0.4830	0.4494	0.4494
Large	G-Eval	0.0070	0.4550	0.0110	0.5134	0.0030	0.7288	0.0371	0.3378	0.0145	0.5088
	CheckEval	0.6486	0.6486	0.6626	0.6626	0.6263	0.6263	0.7569	0.7569	0.6736	0.6736
Medium	G-Eval	0.1680	0.1361	0.0115	0.2581	0.0572	0.2907	0.0384	0.2074	0.0688	0.2231
	CheckEval	0.3635	0.3635	0.5338	0.5338	0.4486	0.4486	0.6715	0.6715	0.5044	0.5043
Small	G-Eval	0.0357	0.1535	0.0287	0.1528	0.0603	0.2139	0.0242	0.1343	0.0372	0.1636
	CheckEval	0.4040	0.4040	0.2127	0.2127	0.0218	0.0218	0.0289	0.0289	0.1669	0.1668
GPT	G-Eval	0.0079	0.4970	0.0698	0.3936	0.0225	0.6910	0.0536	0.4067	0.0385	0.4971
	CheckEval	0.5651	0.5651	0.2452	0.2452	0.6124	0.6124	0.7352	0.7352	0.5395	0.5395
Top-3	G-Eval	0.0234	0.4389	0.0015	0.6510	0.0020	0.7701	0.0752	0.3773	0.0255	0.5593
	CheckEval	0.6215	0.6215	0.2481	0.2480	0.6435	0.6434	0.7813	0.7812	0.5736	0.5736

Table 21: IEA - Topical-Chat

Evaluation Prompt for SummEval

<Task Overview>
Your task is to read a provided news article and its summary, then answer 'yes' or 'no' to specific questions. These questions will relate to a particular dimension of the summary.

<dimension Definition>
<dimension>- <definition>

<Instructions>

1. Read these instructions thoroughly.
2. Carefully read both the Article and the Summary.
3. Understand the given questions and the definition of the <dimension>.
4. Respond to each question with 'yes' or 'no'. Base your answers on a clear rationale.
5. Follow the specified format for your answers.

<Answer Format>
Q1: [Your Answer]
Q2: [Your Answer]
...

 # Article #
<source>

 # Summary #
<summary>

 # Questions #
<questions>

 # Response #
Provide your answers to the given questions, following the specified Answer Format.

Figure 5: Evaluation Prompt - SummEval

Evaluation Prompt for Topical-Chat

<Task Overview>
You will be given a conversation between two individuals. You will then be given one potential response for the next turn in the conversation. The response concerns an interesting fact, which will be provided as well.
Your task is to read a provided conversation history, corresponding fact, and response, then answer 'yes' or 'no' to specific questions. These questions will relate to a particular dimension of the response.

<dimension Definition>
<dimension>- <definition>

<Instructions>

1. Read these instructions thoroughly.
2. Carefully read the Conversation History, the Corresponding Fact, and the Response.
3. Understand the given questions and the definition of the <dimension>.
4. Respond to each question with 'yes' or 'no'. Base your answers on a clear rationale.
5. Follow the specified format for your answers.

<Answer Format>
Q1: [Your Answer]
Q2: [Your Answer]
...
 # Conversation History #
<document>

 # Corresponding Fact #
<fact>

 # Response #
<response>

 # Questions #
<questions>

 # Your Answer #
Provide your answers to the given questions, following the specified Answer Format.

Figure 6: Evaluation Prompt - Topical-Chat

Augmentation - Question Diversification Prompt

<Task Overview>
You will be provided with: 1) Information about the benchmark to be evaluated, 2) The main concept being assessed in the benchmark, and 3) Seed questions that include key components and sub-questions related to this concept.
Your task is to create additional sub-questions for the key components to comprehensively assess the main concept. Each sub-question must meet given conditions to ensure a high-quality question set.

1) Benchmark Information:
{benchmark description}

2) Main Concept in the Benchmark:
{concept}: {description}

3) Key Components and Seed Questions:
{seed questions}

<Conditions for a Good Question List>
{conditions}

<Constraints>
- Each sub-question must be answerable with a simple 'yes' or 'no'.
- A 'yes' answer should indicate that the sentence improves the specified evaluation criterion (e.g., Coherence, Relevance).
- Each question should assess only a single dimension or concept.
- Each question should not ask about more than one topic or concept.

Figure 7: Augmentation - Question Diversification Prompt

Augmentation - Question Elaboration Prompt

<TASK OVERVIEW>

Your task is to generate multiple additional questions to evaluate benchmark performance under specific constraints. You will receive the key component and sub-component evaluating {dimension} and the question related to it. The definition of {dimension} is as follows: {def}. The evaluation for dimension {dimension} will be centered around the key component {key components}.

<TASK>

Your role: You have to break down sub-questions into 3 to 10 sub-sub-questions considering {dimension} when pairs of seed name and question are given.
Benchmark information: {benchmark info}

<CONSTRAINTS>

{constraints}

<Conditions for a Good Question List>

{conditions}

<FORMAT>

```
1. sub_component_name_1:
1-1. q1-1_origin_question
1-1-1. q1-1-1_aug_question
1-1-2. q1-1-2_aug_question
...
1-2. q1-2_origin_question
1-2-1. q1-2-1_aug_question
1-2-2. q1-2-2_aug_question
...

2. sub_component_name_2:
2-1. q2-1_origin_question
2-1-1. q2-1-1_aug_question
...
2-2. q2-2_origin_question
...
```

<EXAMPLE>

{example}

Figure 8: Augmentation - Question Elaboration Prompt

Filtering Prompt

<Task Overview>
Your task is to filter out questions from a list based on the following criteria:

1) dimension Alignment:
- dimension definition: {dimension def}
- Remove questions that deviate from the given dimension's definition.
- Remove questions that are more closely related to other dimensions than the current one.

2) Redundancy:
- Remove questions that:
* Ask for the same or very similar information (even if phrased differently).
* Convey very similar meanings without adding unique insight.

3) Style:
- Remove questions that:
* Use overly exaggerated wording.
* Focus on excessively detailed or minor points that don't meaningfully affect overall quality.

4) Benchmark Context
- Name: Topical-Chat
- Purpose: Evaluation of knowledge-grounded dialogue systems
- Key Metrics: Naturalness, Coherence, Engagingness, Groundedness
- Do not modify any of the remaining questions or generate new ones.
- Keep questions in their original dictionary format.

5) Sub-dimensions and Questions:
{format_sub_dimensions(sub_dimensions)}

6) Output Requirements:
- Output format: JSON only
- Structure:

```
{"Sub-dimension Name": [
    "Filtered Question 1",
    "Filtered Question 2"]}
```

<Important Note>
- Do not modify the content of remaining questions
- Do not generate new questions
- Maintain the original dictionary format
- Only remove questions that fail the above criteria
- Do not remove entire sub-dimensions or their keys unless no valid questions remain.

Figure 9: Filtering Prompt

Dimension	Sub-dimension	Seed Questions
Coherence	Topic Maintenance	Does the summary consistently focus on the central topic without deviating into unrelated areas?
	Logical Flow	Does the summary present information in a logical order?
	Consistent Point of View	Is the point of view or perspective in the summary consistent with the source?
Consistency	Factual Consistency	Does the summary accurately represent the facts from the source?
	No New Information	Does the summary avoid introducing information not present in the original source?
	Contextual Accuracy	Does the summary preserve the original purpose or intent of the source document?
Fluency	Formatting	Is the summary free from formatting issues and correctly capitalized throughout?
	Grammar	Are all sentences grammatically correct and free from errors?
	Completeness	Are all sentences complete, with no fragments or missing components?
	Readability	Is the summary easy to read, without unnecessary complexity?
Relevance	Content Coverage	Does the summary encapsulate all critical points of the source document?
	Topic Consistency	Does the summary maintain the main topic of the source?
	Consistent Use of Terminology	Does the summary use the same terminology or jargon as the source?
	Use of Key Terms and Phrases	Does the summary incorporate key terms and phrases from the source material effectively?
	Importance	Is each point mentioned in the summary important to the overall understanding of the original text?

Table 22: Dimensions, sub-dimensions, and corresponding seed questions for SummEval.

Dimension	Sub-dimension	Seed Questions
Coherence	Logical Flow	Does the response logically follow from the earlier part of the conversation, maintaining a clear flow of ideas?
	Relevance	Is the response directly relevant to the content and context of the previous dialogue?
	Continuity	Does the response stay consistent with the topic discussed in the previous dialogue? Does the response integrate smoothly with the ongoing conversation, ensuring a coherent progression?
Engagingness	Informative	Does the response add meaningful value to the conversation?
	Emotional Engagement	Is the response friendly, polite, and empathetic?
	Interest Level	Does the response capture interest or intrigue, making the conversation more engaging? Does the response actively contribute to keeping the conversation lively and engaging?
Groundedness	Relevance	Does the response appropriately address the preceding question or statement? Does the answer provide new information while maintaining the flow of the conversation? Does it effectively utilize the key information that has been mentioned in the conversation?
	Consistency	Does the response remain consistent with previous utterances? Does it avoid contradicting previously provided information?
Naturalness	Avoid repetition	Does the response avoid unnecessary repetition of the same content between sentences?
	Context relevance	Are all the sentences relevant to the topic of conversation and used naturally within the context?
	Clarity	Is the overall message clear and easy to understand?
	Word choice and tone	Is the tone consistent throughout? Are there no major grammatical errors?

Table 23: Dimensions, sub-dimensions, and corresponding seed questions for Topical-Chat.