SEAL: Structured Evaluation via LLMs for Text Generation

Anonymous ACL submission

Abstract

The rapid progress of large language models (LLMs) has made automatic evaluation of natural language generation an important yet challenging task. While recent LLM-as-a-judge approaches achieve promising performance, they typically rely on monolithic evaluation and produce scalar scores. Consequently, most existing LLM-based evaluators suffer from structural limitations arising from granularity stochasticity, evaluation signal compression, and implicit aggregation. In this work, we propose SEAL, a structured framework which rethinks the role of LLMs in automatic evaluation. Instead of exploiting LLMs as black-box scorers, we conceptualize evaluation as a structured process and treat LLMs as constrained semantic decision modules. Concretely, SEAL addresses existing limitations by decomposing evaluation into task-specific semantic units, formulating quality assessment as verifiable sub-dimension binary decisions, and enforcing deterministic aggregation, ensuring the evaluation process is structurally rigorous and interpretable. We evaluate SEAL across multiple tasks and benchmarks. Experimental results demonstrate that SEAL achieves state-of-the-art correlation with human judgments while providing fine-grained actionable insights. Our findings propose that structured evaluation offers a principled path toward trustworthy and reproducible LLM-based evaluation. Our code is available at https://anonymous.4open.science/r/SEAL-B837

1 Introduction

With the rapid advancement of Large Language Models (LLMs), natural language generation systems have demonstrated revolutionized fluency, coherence, and semantic competence across a wide range of tasks, including summarization, dialogue, and creative writing. As generation quality continues to improve, reliable evaluation has increasingly emerged as a central bottleneck, becoming one of the most important and challenging problems in modern AI. Accurately evaluating generation quality is essential for model development, yet remains a fundamental challenge.

Features	Traditional Metrics	LLM-based	SEAL (ours)
Granularity	N-gram	Whole text	Semantic unit
Semantic Depth	Low	High	High
Robustness	Low	High	High
Reference-free	No	Flexible†	Flexible†
Aggregation	Explicit and fixed	Implicit	Explicit and deterministic
Objectivity	High	Low	High
Decision	Deterministic matching	Holistic scalar judgment	Verifiable binary decisions
Interpretability	High	Low	High

† Depends on the specific evaluation dimension.

Table 1: Comparison of evaluation paradigms.

Traditional evaluation typically employs reference-base metrics such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and METEOR (Banerjee and Lavie, 2005). Although offering high objectivity, structural interpretability, and explicit aggregation schemes, these methods rely on lexical overlap, leading to poor correlation with human judgments on semantic fidelity and conversational coherence. Recent work has increasingly turned to the LLM-as-a-judge paradigm, where large language models are leveraged to directly evaluate generated texts (Zhong et al., 2022; Liu et al., 2023; Wu et al., 2025). These methods have demonstrated promising empirical alignment with human judgments and have offered appealing properties such as reference-free evaluation and semantic sensitivity. Despite the advantages, there remain three critical challenges which hinder the reliability of existing LLM-based evaluators in real-world applications.

(i) Granularity-Stochasticity Bias, where evaluation is performed at an overly coarse granularity under stochastic LLM judgments. In practice, generation errors are often localized to specific

sentences, facts or dialogue turns. However typical LLM-evaluators tend to perform holistic evaluation, amplifying such local noise into unstable global scores. This mismatch leads to substantial variance across runs and limited reproducibility.

(ii) Evaluation Compression Bias, where rich, multi-faceted generation quality is irreversibly compressed into holistic judgments or scalar scores. This compression obscures the underlying causes of evaluation outcomes because different error patterns (*e.g.,* factual inconsistency, discourse breakdown, or stylistic mismatch) may produce indistinguishable final scores. Consequently, the final evaluation signals become difficult to interpret or diagnose.

(iii) Implicit Judgment Bias, where aggregation and trade-offs among evaluation criteria are left to the internal and unobservable reasoning capability of LLMs. Even though multiple dimensions are considered, the relative importance of dimensions is implicitly encoded in the LLM pretraining and prompt phrasing. This lack of transparency leads to preference leakage and unstable evaluation policies, which ultimately limits the controllability and generalizability of the evaluator across different tasks and domains.

Addressing these issues requires rethinking the evaluation structure of LLM-based evaluators. In this work, we propose **SEAL** (Structured Evaluation via LLMs), a new evaluation paradigm that explicitly addresses these biases by restructuring the role of LLMs in automatic evaluation process. SEAL does not treat LLMs as black-box scorers, but employs LLMs as constrained semantic decision modules within a metric-like framework that preserves explicit decomposition and aggregation. To be specific, SEAL assesses generated text with a hierarchical and structured process. Similar to existing methods (Zhong et al., 2022; Liu et al., 2023; Wu et al., 2025), we first define a set of primary dimensions (*e.g.,* coherence, fluency and consistency) for a given task. Within each dimension, SEAL decomposes the generated text into task-specific semantic units such as adjacent sentence pairs or individual dialogue turns. For each semantic unit, SEAL further specifies a set of fine-grained sub-dimensions. Each sub-dimension is instantiated as a verifiable binary question and the LLM is leveraged to answer it. Additionally, the LLM is then employed to dynamically assign weights to these sub-dimensions and all sub-dimension-level judgments are aggregated through a deterministic and transparent scoring function to produce dimension-level evaluation scores.

The design of SEAL framework explicitly mitigates the fundamental biases. The semantic unit decomposition alleviates evaluation compression by preserving fine-grained structure. The verifiable binary decisions replace ambiguous scalar scoring, thereby converting subjective impressions into verifiable statistical evidence. The final explicit aggregation eliminates implicit judgment while smoothing stochastic variability through unit-level averaging. Consequently, SEAL provides a more interpretable and stable evaluation process as traditional metrics, while retaining the semantic sensitivity and annotation-free advantages of LLM-based evaluation.

We evaluate SEAL on three benchmarks, SummEval, TopicalChat, and QAGS, which covers various tasks including summarization, dialogue, and factual evaluation. The experimental results demonstrate that SEAL consistently achieves state-of-the-art correlation with human judgments, outperforming both traditional metrics and prior LLM-based evaluation approaches.

This work makes three main contributions:

- We summarize the fundamental biases underlying existing LLM-based evaluation paradigms.
- We propose SEAL, a structured and metric-like evaluation framework that systematically addresses these biases by rethinking the role of LLMs in evaluation.
- Evaluation on various benchmarks demonstrates that our method achieves new SOTA performance in alignment with human perception.

2 Related Work

Automatic evaluation for text generation tasks has always been an important problem in the natural language community. Traditional evaluation approaches, including lexical metrics (*e.g.,* ROUGE (Lin, 2004), BLEU (Papineni et al., 2002)) and embedding-based methods (*e.g.,* BERTScore (Zhang et al., 2019) and MoverScore (Zhao et al., 2019)) often fail to capture deep semantic coherence or factual correctness (Reiter and Belz, 2009; Fabbri et al., 2021a). To improve diagnostic value, prior work has explored dimension-specific or decomposed evaluation, including NLI-based methods for factual consistency (*e.g.,* FEQA (Durmus et al., 2020) and QAFactEval (Fabbri et al., 2021b)), unified frameworks (*e.g.,* UniEval (Zhong

et al., 2022) and UniSumEval (Lee et al., 2024)), and atomic-unit verification approaches (FActScore (Min et al., 2023), FineSurE (Song et al., 2024), ACUEval (Wan et al., 2024), FineDialFact (Chen et al., 2025)). While these methods demonstrate the benefits of granularity, they often rely on task-specific training or predefined atomic spans, limiting the applicability across diverse scenarios.

Recently, Large Language Models (Wang et al., 2023; Lin and Chen, 2023; Shi et al., 2023) have emerged as powerful evaluators for NLG tasks. Several studies demonstrate that LLMs can directly assess generation quality via prompting and can achieve strong correlation with human judgments, such as GPTScore (Fu et al., 2023) and G-Eval (Liu et al., 2023). Despite their empirical success, these vanilla LLM evaluators rely on holistic and scalar judgments, where aggregation and trade-offs among criteria are handled implicitly by LLMs.

To improve transparency, recent studies have shifted toward decomposition. CheckEval (Lee et al., 2025) breaks down evaluation criteria into a boolean checklist, transforming subjective scoring into binary verification. Similarly, SEEval (Wu et al., 2025) emphasizes the extraction of explicit semantic evidence to justify evaluation outcomes. While effective, their evaluation logic largely rely on fixed templates and the aggregation strategy is implicit, limiting extensibility and principled control. In contrast, we formalize evaluation as a structured decision process rather than a collection of checks. SEAL introduces a unified abstraction across tasks and dimensions and models sub-dimensions as verifiable propositions with explicit weights and aggregation rules. This design enables SEAL to systematically address known biases in LLM-as-a-judge evaluation, including evaluation compression, implicit judgment, and granularity bias, while remaining task-agnostic and extensible beyond specific check templates.

3 Methodology

3.1 Problem Setup

In this work, we consider the task of automatic evaluation for natural language generation problems such as summarization and dialogue. Formally, given an input x and a generated output y, the task is to assign a set of scores that reflect the quality of y across several predefined dimensions such as coherence, consistency, and fluency.

Unlike typical LLM-based evaluators that directly evaluate the whole text and holistically assign scalar scores, our approach is designed to explicitly model the evaluation process as a structured decision procedure and reconstruct the LLMs from black-box scorers to constrained semantic decision modules. Our objective is to preserve the compositional and deterministic properties of classical metrics while leveraging LLMs for semantic understanding and annotation-free judgment.

Algorithm 1: SEAL

Input: Input x, generated output y, and evaluation dimension d
Output: Dimension-level score score_d

Decompose y into semantic units.
$\mathcal{U}_d = \{u_1, u_2, \cdots, u_{|\mathcal{U}_d|}\} \leftarrow c(y, d)$
Define sub-dimensions $\mathcal{S}_d = \{s_1, s_2, \cdots, s_{|\mathcal{S}_d|}\}$
Estimate sub-dimension weights with LLMs
$w(s_j, x, y, d) > 0$,
$w.r.t. \sum_{s_j \in \mathcal{S}_d} w(s_j, x, y, d) = 1$;
foreach $u_i \in \mathcal{U}_d$ **do**
  foreach $s_j \in \mathcal{S}_d$ **do**
    LLM-based binary verification
    $v(u_i, s) \in \{0, 1\}$
  end
  $\text{score}_{d,u_i} \leftarrow \sum_{s_j \in \mathcal{S}_d} w(s_j, x, y, d) v(u_i, s_j)$
end
$\text{score}_d \leftarrow \frac{1}{|\mathcal{U}_d|} \sum_{u_i \in \mathcal{U}_d} \text{score}_{d,u_i}$
return score_d

3.2 Overview of SEAL

As shown in Figure 1, SEAL is a metric-like framework which assesses the generated text with a hierarchical and structured process. At the highest level, SEAL operates over a set of predefined evaluation dimensions $\mathcal{D}$ (*e.g.,* coherence, consistency, fluency, etc.), which follow common practice in text generation evaluation (Fabbri et al., 2021a; Gopalakrishnan et al., 2023; Liu et al., 2023). This work does not introduce any new dimension but focuses on how each dimension is operationalized through structured decomposition and aggregation. The framework is agnostic to the particular choice of dimensions and can be readily applied to new criteria. For each dimension $d \in \mathcal{D}$, SEAL decomposes the text assessment into three stages:

- **Semantic unit decomposition** aligns evaluation granularity with localized generation behavior by decomposing outputs into minimal assessable semantic units.
- **Sub-dimension verification** reduces complex quality assessment to a set of objective and verifiable binary decisions, enabling consistent and interpretable semantic judgments by LLMs.

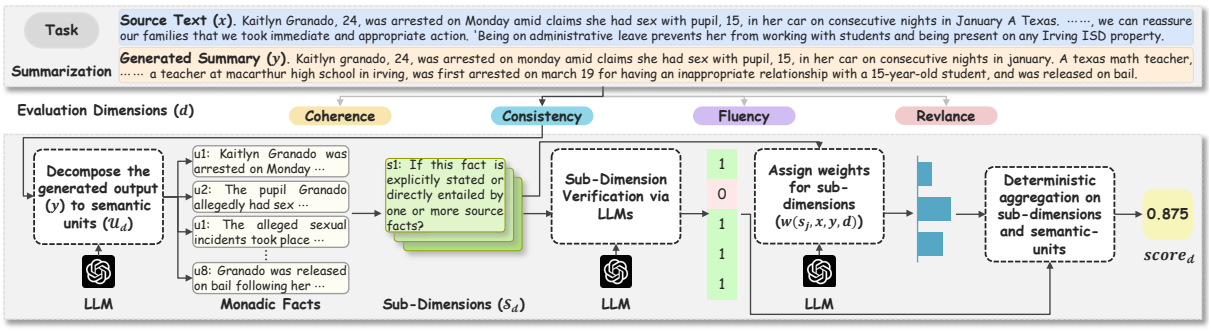

Figure 1: Overview of SEAL. SEAL is a metric-like framework which decomposes text generation evaluation into three stages: semantic unit decomposition, sub-dimension verification, and deterministic aggregation. SEAL reconstruct the LLMs from black-box scorers to constrained semantic decision modules.

Task	Dimension d	Semantic Units $\mathcal{U}_d$	Sub-Dimensions $\mathcal{S}_d$
Summarization	Coherence	Adjacent sentence pairs	Logical continuity;
	Fluency	Individual sentences	Grammatical or formatting errors;
	Consistency	Monadic facts	Entailed by any source fact;
	Relevance	Monadic facts	Reflecting a core idea from the reference;
Dialogue	Naturalness	Response with dialogue context	(1) Fluency; (2) Register match; (3) Turn alignment; (4) Lexical variation; (5) Natural expressiveness;
	Coherence		(1) Referential continuity; (2) Logical consistency; (3) Discourse move alignment; (4) Topical bridging; (5) Temporal/causal order; (6) Presupposition support;
	Engagingness		(1) Interest; (2) Emotional resonance; (3) Initiative; (4) Personalization; (5) Coherence with interest flow;
	Groundedness		(1) Factuality; (2) No Hallucination; (3) Knowledge Integration; (4) Accuracy; (5) Contextual Relevance;
QAGS	Consistency	QA triples	(1) No Hallucination; (2) Relational Accuracy; (3) Numerical/Attribute Integrity; (4) No Contradiction; (5) Contextual Fidelity;

Table 2: Instantiation details on different tasks. For text summarization, the monadic facts are defined as the minimal verifiable claims in the text. For the dialogue task, all evaluation dimensions are instantiated at the level of individual system responses conditioned on the dialogue history, reflecting the localized nature of conversational quality and errors. For QAGS, a QA triple consists of a source document, a reference text and a system output.

- **Deterministic aggregation** composes unit-aware decisions into dimension-level scores through transparent aggregation rules, which ensures reproducibility and interpretability.

3.3 Semantic Unit Decomposition

Semantic unit decomposition determines the granularity at which evaluation is performed. Given the evaluation dimension $d \in \mathcal{D}$, SEAL decompose the generated output y into a set of semantic units:

$$\mathcal{U}_d = \{u_1, u_2, \cdots, u_{|\mathcal{U}_d|}\}, \tag{1}$$

where $|\mathcal{U}_d|$ denotes the size of $\mathcal{U}_d$. A semantic unit represents the minimal textual span over which dimension d can be meaningfully and individually assessed. The semantic units are task-specifically defined as the minimal span for independent dimension-aware verification. The semantic unit is manually predefined and the decomposition process is operated by LLM, denoted by:

$$\mathcal{U}_d = c(y, d). \tag{2}$$

This mechanism is introduced to address **Granularity–Stochasticity Bias**. In practice, many generation errors are localized such as a single incoherent sentence in a summary. Conventional holistic evaluation forces the LLM to compress such localized phenomena into a single scalar score, where stochastic variation in one part can disproportionately affect the entire judgment. By evaluating at

the level of semantic units (*e.g.,* adjacent sentence pairs in the summary), SEAL aligns the granularity of evaluation with that of errors. Moreover, aggregating judgments over multiple units naturally smooths stochastic variability, yielding more stable and reliable scores.

3.4 Sub-Dimension Verification

Given the semantic units $\mathcal{U}_d$, the second stage specifies how each unit is evaluated. For each dimension d, we first define a set of sub-dimensions:

$$\mathcal{S}_d = \{s_1, s_2, \cdots, s_{|\mathcal{S}_d|}\}. \tag{3}$$

Instead of generating a scalar score, SEAL formulates the judgment process as a verifiable LLM-based binary proposition. Formally, for each semantic unit $u_i \in \mathcal{U}_d$ and each sub-dimension $s_j \in \mathcal{S}_d$, the LLM is asked to judge:

$$v(u_i, s_j) \in \{0, 1\}. \tag{4}$$

Typically, LLM-based evaluators compress multiple heterogeneous factors into a single number (*e.g.,* rating coherence on 1-5 scale) and obscure the reasons behind an evaluation outcome. Consequently, different error patterns may collapse to the same score, making evaluation signals difficult to interpret or diagnose. SEAL addresses this **Evaluation Compress Bias** by introducing binary verification. Each decision corresponds to a concrete, task-grounded question whose truth value can be independently assessed. This mechanism reduces ambiguity and judgment entropy, and prevents irreversible information loss during evaluation.

3.5 Deterministic Aggregation

The deterministic aggregation stage specifies how individual verification decisions are combined into final scores. While sub-dimensions are explicitly defined, the relative importance may differ across tasks and dimensions. SEAL therefore asks the LLM to associate each sub-dimension $s_j \in \mathcal{S}_d$ with a normalized weight score:

$$w(s_j, x, y, d) > 0, w.r.t. \sum_{s_j \in \mathcal{S}_d} w(s_j, x, y, d) = 1, \tag{5}$$

An LLM is leveraged to estimate the weights based on the input x, generated output y and dimension description d. Given the weights and binary judgments of semantic unit u_i, SEAL calculates a unit-level score:

$$\text{score}_{d,u_i} = \sum_{s_j \in \mathcal{S}_d} w(s_j, x, y, d) v(u_i, s_j), \tag{6}$$

and then aggregates unit scores to obtain the dimension-level score as:

$$\text{score}_d = \frac{1}{|\mathcal{U}_d|} \sum_{u_i \in \mathcal{U}_d} \text{score}_{d,u_i}. \tag{7}$$

Here we treat all semantic units with equally importance and directly calculate the average scores.

Typical LLM-based evaluators may suffer from the **Implicit Judgment Bias**. In the holistic evaluation paradigm, trade-off across different dimensional criteria are implicitly performed inside the LLM internal reasoning mechanism, which makes the evaluation policies unstable and opaque. By contrast, SEAL integrates an explicit and deterministic aggregation. The LLM is used only to provide semantic judgments and relative importance estimates, while the final score is computed by a transparent aggregation rule. This separation ensures interpretability reproducibility, allowing the evaluation behavior to be further improved.

3.6 Task-Specific Implementation

As shown in Algorithm 1, SEAL is a general framework and the components (evaluation dimensions, semantic units, sub-dimensions) are instantiated according to specific tasks and dimensions. In this section, we provide some of the instantiation details for better clarification. The completed implementation details are described in Appendix D. In this work, we apply SEAL to three distinct evaluation tasks: (1) text summarization, (2) topical dialogue responses, and (3) QAGS factual consistency evaluation. The instantiations used in our experiments are summarized in Table 2. Across tasks, SEAL instantiates semantic units and sub-dimensions by aligning evaluation granularity with the minimal verifiable unit of each criterion, while preserving a unified verification and aggregation procedure. This separation of framework and instantiation enables SEAL to generalize across tasks without task-specific training or annotation.

4 Experiments

Datasets and Metrics. We evaluate SEAL on three benchmarks targeting different generation scenarios and evaluation challenges. **SummEval** (Fabbri et al., 2021a) is a summarization evaluation dataset with human annotations on consistency, coherence, fluency, and relevance. **Topical-Chat** (Gopalakrishnan et al., 2019) is a knowledge-grounded open-domain dialogue dataset including

Type	Method	Coherence		Consistency		Fluency		Relevance		Average	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Traditional Metrics	ROUGE-1	0.167	0.126	0.160	0.130	0.115	0.094	0.326	0.252	0.192	0.150
	ROUGE-2	0.184	0.139	0.187	0.155	0.159	0.128	0.290	0.219	0.205	0.161
	ROUGE-L	0.128	0.099	0.115	0.092	0.105	0.084	0.311	0.237	0.165	0.128
Embedding Methods	BERTScore	0.284	0.211	0.110	0.090	0.193	0.158	0.312	0.243	0.225	0.175
	MoverScore	0.159	0.118	0.157	0.127	0.129	0.105	0.318	0.244	0.191	0.148
Pretrained LMs	BARTScore	0.448	0.342	0.382	0.315	0.356	0.292	0.356	0.273	0.385	0.305
	UniEval	0.575	0.442	0.446	0.371	0.449	0.371	0.426	0.325	0.474	0.377
	GPTScore	0.437	-	0.452	-	0.411	-	0.363	-	0.416	-
LLM-based (GPT-4o)	G-Eval	0.189	0.158	0.422	0.391	0.286	0.268	0.397	0.342	0.324	0.289
	SEEval	0.339	0.362	0.442	0.416	0.366	0.351	0.402	0.362	0.387	0.373
	CheckEval	0.556	**0.464**	0.530	0.474	0.469	0.412	0.460	0.400	0.504	0.438
	SEAL	**0.561**	0.437	**0.540**	**0.502**	**0.551**	**0.499**	**0.521**	**0.420**	**0.543**	**0.464**

Table 3: Spearman (ρ) and Kendall tau (τ) correlation of different aspects on SummEval.

four dimensions: naturalness, coherence, engagingness, and groundedness. **QAGS** (Wang et al., 2020) focuses on the factual consistency of summaries. For evaluation, we follow the existing work to report correlations with human judgments using Pearson r, Spearman ρ and Kendall's τ.

Compared Methods. We compare SEAL with 13 approaches from four categories: (1) traditional metrics, including ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and METEOR (Banerjee and Lavie, 2005); (2) embedding-based methods, including BERTScore (Zhang et al., 2019), MoverScore (Zhao et al., 2019), and FactCC (Kryściński et al., 2020); (3) pretrained language models, including BARTScore (Yuan et al., 2021), USR (Mehri and Eskenazi, 2020), UniEval (Zhong et al., 2022), and GPTScore (Fu et al., 2023); (4) LLM-based methods, including G-Eval (Liu et al., 2023), SEEval (Wu et al., 2025), CheckEval (Lee et al., 2025).

Implementation. We use both closed-source models (GPT-4o and GPT-4 (Achiam et al., 2023)) and open-source models (Qwen-235B (Yang et al., 2025) and DeepSeek-V3 (Liu et al., 2024)) as the judge. For all LLM-based methods, temperature is set to 0 for reproducibility and fair comparison.

4.1 Main Results

Tables 3–5 presents the partial results on SummEval, QAGS, and Topical-Chat respectively, and the detailed results are shown Appendix B.1. Across all three benchmarks, SEAL consistently outperforms traditional metrics, learned scorers, and holistic LLM-based judges. Notably, the unit-decomposition mechanism yields consistent gains, especially on fluency and relevance on SummEval. While the structured assessment of individual turns on Topical-Chat delivers substantial improvements in naturalness and engagingness, indicating that binary judgments for each sub-dimension capture fine-grained conversational cues more accurately than global scalar ratings. These results demonstrate that structuring LLM judgments through semantic unit decomposition, sub-dimension verification, and deterministic aggregation yields more reliable and interpretable evaluation, while preserving the semantic coverage of LLM-based approaches.

Method	QAGS-CNN		QAGS-XSUM		Average	
	r	ρ	r	ρ	r	ρ
ROUGE-L	0.357	0.324	0.024	0.011	0.190	0.156
BERTScore	0.576	0.505	0.024	0.008	0.300	0.256
BARTScore	0.735	0.680	0.184	0.159	0.459	0.420
UniEval	0.682	0.662	0.461	0.488	0.571	0.575
G-Eval	0.494	0.540	0.556	0.556	0.525	0.548
CheckEval	0.715	**0.721**	0.592	0.566	0.654	0.643
SEAL	**0.720**	0.716	**0.640**	**0.643**	**0.680**	**0.680**

Table 4: Pearson (r) and Spearman (ρ) of different metrics on QAGS.

4.2 Ablation Study

Table 6 and 8 partially present the ablation results which evaluate the contribution of each component in SEAL, and the full results are provided in Appendix B.2. Concretely, we analyze the effectiveness of **Semantic unit decomposition** (SD), **Sub-dimension Verification** (SV) and **Weighted aggregation** across summarization and dialogue benchrmarks. Removing semantic unit decomposition (w/o SD) leads to the most severe perfor-

Type	Method	Naturalness		Coherence		Engagingness		Groundedness		Average	
		ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
Traditional Metrics	ROUGE-L	0.176	0.146	0.193	0.203	0.295	0.300	0.310	0.327	0.243	0.244
	BLEU-4	0.180	0.175	0.131	0.235	0.232	0.316	0.213	0.310	0.189	0.259
	METEOR	0.212	0.191	0.250	0.302	0.367	0.439	0.333	0.391	0.290	0.331
Embedding	BERTScore	0.226	0.209	0.214	0.233	0.317	0.335	0.291	0.317	0.262	0.273
Pretrained LMs	USR	0.337	0.325	0.416	0.377	0.456	0.465	0.222	0.447	0.358	0.403
	UniEval	0.455	0.330	0.602	0.455	0.573	0.430	0.577*	0.453	0.552	0.417
LLM-based (GPT-4o)	G-Eval	0.592	0.567	0.611	0.577	0.390	0.165	0.477	0.425	0.517	0.434
	CheckEval	0.589	0.579	0.736	0.735	0.587	0.576	0.646	0.645	0.639	0.634
	SEAL	0.635	0.633	0.693	0.686	0.675	0.671	0.598	0.597	**0.650**	**0.647**

Table 5: Spearman (ρ) and Pearson (r) correlation of different aspects on Topical-Chat.

Method	Coherence		Consistency		Fluency		Relevance		Average	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
SEAL	**0.5613**	**0.4373**	**0.5396**	**0.5018**	**0.5514**	**0.4986**	**0.5205**	**0.4195**	**0.5432**	**0.4643**
-w/o SD	0.3773	0.2678	0.3543	0.3083	0.2223	0.2003	0.3563	0.2827	0.3275	0.2660
-w/o SV	0.4662	0.3180	0.4755	0.4064	0.4706	0.4068	0.3190	0.2579	0.4328	0.3472

Table 6: Spearman (ρ) and Kendall's Tau (τ) correlation of ablation variants on SummEval. SD denotes *Semantic Unit Decomposition* and SV denotes *Sub-Dimension Verification*.

	Nat	Coh	Eng	Grd	Avg
Acc	96.1	94.4	97.2	92.2	95.0

Table 7: Manual annotated accuracy of sub-dimension judgement over different dimension (including Naturalness (Nat), Coherence (Coh), Engaginingness (Eng), and Groundedness (Grd) of Topical-Chat.

mance degradation on SummEval, where the average Spearman correlation drops sharply with particularly large declines in fluency and coherence. This indicates that evaluating signals at the semantic unit level is critical for capturing localized errors. Removing sub-dimension verification (w/o SV) also results in consistent degradation. Notably, the largest drop is observed on relevance, suggesting that explicitly verifying atomic criteria is essential for dimensions that rely on multiple latent factors. On TopicalChat, discarding aggregate weighting results in a consistent decline in average correlation, suggesting the importance of weighted aggregation in reflecting the relative salience of sub-dimensions, rather than assuming uniform importance across criteria.

To further validate the reliability of the sub-dimension verification process, Table 7 reports the manually annotated accuracy of LLM-based sub-dimension judgments on TopicalChat. The consistently high accuracy across all dimensions indicates that the model can reliably perform fine-grained binary judgments at the sub-dimension level.

Overall, the ablation results demonstrate that semantic unit decomposition is the primary driver of performance gains, while sub-dimension verification and weighted aggregation provide complementary improvements. Together, these components enable SEAL to achieve robust and interpretable evaluation across various tasks.

4.3 Additional Analysis

Token and Time Efficiency. As shown in Table 9, SEAL incurs higher token usage and latency than single-pass LLM judges such as GEval and CheckEval. The overhead stems from two design decisions. First, every semantic unit is explicitly materialised in the prompt, yielding quadratic growth with summary length. Second, each sub-dimension is elicited separately before aggregation, which requires multiplying calls. Importantly, latency scales predictably with token count, suggesting that the overhead primarily stems from increased reasoning depth rather than inefficiencies in system design. Despite requiring more costs, the modular structure of SEAL enables natural parallelization across semantic units and sub-dimensions, which offers a clear pathway to further reduce wall-clock latency in practical deployments.

Impact of LLM Choice. Table 10 compares the performance of different evaluation methods when

Method	Naturalness		Coherence		Engagingness		Groundedness		Average	
	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
SEAL	0.6351	0.6332	0.6932	0.6857	0.6747	0.6706	0.5980	0.5972	**0.6503**	**0.6467**
-w/o weighting	0.6259	0.6306	0.6606	0.6300	0.6475	0.6415	0.6122	0.6175	0.6365	0.6299

Table 8: Spearman (ρ) and Pearson (r) correlations of ablation variants on Topical-chat.

	Coherence		Consistency		Fluency		Relevance		Average	
	Tokens	Latency (s)	Tokens	Latency (s)	Tokens	Latency (s)	tokens	Latency (s)	Tokens	Latency (s)
GEval	807.46	1.419	764.55	1.375	288.35	1.358	756.40	1.376	654.19	1.382
CheckEval	912.87	1.192	950.52	1.980	916.31	1.276	871.51	1.132	912.80	1.395
SEAL (stage 1)	776.58	2.021	1561.86	4.249	306.04	2.376	1561.86	4.249	1,051.58	3.268
SEAL (stage 2 & 3)	758.30	1.874	927.68	2.052	668.74	2.015	910.97	1.909	816.42	1.962
SEAL (Total)	1,534.88	3.895	2,489.54	6.301	974.78	4.391	2,472.83	6.158	1,868.00	5.230

Table 9: Average number of tokens and latency (in seconds) on SummEval.

LLM	Method	Average	
		ρ	τ
Qwen3-235B	G-Eval	0.426	0.352
	CheckEval	0.470	0.394
	SEAL	0.496	0.420
Deepseek-V3	G-Eval	0.435	0.293
	CheckEval	0.403	0.346
	SEAL	0.498	0.423
GPT-4o	G-Eval	0.324	0.290
	CheckEval	0.504	0.438
	SEAL	0.543	0.464
GPT-4	G-Eval	0.509	0.463
	CheckEval	0.521	0.463
	SEAL	0.546	0.464

Table 10: Correlation comparison using different LLMs on SummEval.

instantiated with diverse LLMs. Notably, the relative improvement of SEAL over compared methods is most pronounced when using weaker or noisier judges. This suggests that SEAL is less sensitive to the intrinsic reliability of the underlying LLM, as its structured decomposition and deterministic aggregation effectively constrain judgment variance.

Stability Analysis. Table 11 reports the mean and standard deviation of correlation scores over three independent runs on SummEval. Notably, SEAL demonstrates substantially lower standard deviation. The high stability is attributed by two components. First, sub-dimension verification decomposes evaluation into low-entropy binary judgments. Second, deterministic aggregation explicitly defines how unit-level decisions are composed. These mechanisms together mitigate randomness introduced by sampling and reduce sensitivity.

Hyperparameter Analysis. We also provide hyperparameter analysis in Appendix B.3.

LLM	Method	Average	
		ρ	τ
GPT-4o	G-Eval	0.329 ± 0.017	0.290 ± 0.015
	CheckEval	0.490 ± 0.012	0.433 ± 0.010
	SEAL	$\mathbf{0.546 \pm 0.006}$	$\mathbf{0.465 \pm 0.005}$
GPT-4	G-Eval	0.477 ± 0.028	0.451 ± 0.035
	CheckEval	0.507 ± 0.040	0.443 ± 0.021
	SEAL	$\mathbf{0.540 \pm 0.015}$	$\mathbf{0.468 \pm 0.010}$

Table 11: Spearman (ρ) and Kendall's Tau (τ) correlatiosn on SummEval over three runs (mean $\pm$ std).

4.4 Case Study

We also provide case studies to illustrate the practical advantages of SEAL in Appendix C.

5 Conclusion and Future Work

We introduced SEAL, a structured evaluation framework that recasts large language models as constrained semantic decision modules instead of opaque scorers. By decomposing generated texts into task-specific semantic units, reducing each unit to a set of verifiable binary sub-judgments, and aggregating the results through deterministic, interpretable rules, SEAL systematically mitigates granularity stochasticity, evaluation compression, and implicit judgment biases which trouble existing LLM-as-a-judge approaches. Experiments on SummEval, Topical-Chat and QAGS demonstrate that the same parameter-free pipeline delivers state-of-the-art correlation with human ratings using various LLM judge models, while remaining improved stability, interpretable and reproducible.

For future work, we are interested in automatic sub-dimension definition and extending the framework to broader scenarios such as multilingual and multimodal generation.

Limitations

Though achieving promising improvements, SEAL has several limitations. First, the framework relies on manually designed semantic units and subdimensions for each task. While this design enables fine-grained control and interpretability, it also introduces additional annotation and engineering effort when adapting to new tasks or domains. Automating these decompositions remains an open challenge. Second, compared to existing LLM-as-a-judge methods, SEAL requires higher computational cost due to its multi-stage evaluation process. While our efficiency analysis shows that the overhead remains manageable, the method may be less suitable for large-scale or real-time evaluation settings without further optimization.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Dianne C Berry. 1983. Metacognitive experience and transfer of logical reasoning. *The Quarterly Journal of Experimental Psychology Section A*, 35(1):39–49.

Xiangyan Chen, Yufeng Li, Yujian Gan, Arkaitz Zubiaga, and Matthew Purver. 2025. Finedialfact: A benchmark for fine-grained dialogue fact verification. *arXiv preprint arXiv:2508.05782*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *arXiv preprint arXiv:2005.03754*.

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

Alexander R Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2021b. Qafacteval: Improved qa-based factual consistency evaluation for summarization. *arXiv preprint arXiv:2112.08542*.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André FT Martins. 2022. Results of wmt22 metrics shared task: Stop using bleu–neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68.

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

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.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2023. Topical-chat: Towards knowledge-grounded open-domain conversations. *arXiv preprint arXiv:2308.11995*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 9332–9346.

Yuho Lee, Taewon Yun, Jason Cai, Hang Su, and Hwanjun Song. 2024. Unisumeval: Towards unified, fine-grained, multi-dimensional summarization evaluation for llms. *arXiv preprint arXiv:2409.19898*.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. Checkeval: A reliable llm-as-a-judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15782–15809.

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

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

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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

Shikib Mehri and Maxine Eskenazi. 2020. Usr: An unsupervised and reference free evaluation metric for dialog generation. *arXiv preprint arXiv:2005.00456*.

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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Xiaoming Shi, Jie Xu, Jinru Ding, Jiali Pang, Sichen Liu, Shuqing Luo, Xingwei Peng, Lu Lu, Haihong Yang, Mingtao Hu, and 1 others. 2023. Llm-mini-cex: Automatic evaluation of large language model for diagnostic conversation. *arXiv preprint arXiv:2308.07635*.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. Finesure: Fine-grained summarization evaluation using llms. *arXiv preprint arXiv:2407.00908*.

David Wan, Koustuv Sinha, Srini Iyer, Asli Celikyilmaz, Mohit Bansal, and Ramakanth Pasunuru. 2024. Acueval: Fine-grained hallucination evaluation and correction for abstractive summarization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10036–10056.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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*.

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*.

A Baselines

We compare against the following approaches:(1) traditional metrics: **ROUGE** (Lin, 2004), **BLEU** (Papineni et al., 2002) and **METEOR** (Banerjee and Lavie, 2005) quantify n-gram or synonym overlap against gold references; they are deterministic, interpretable, but lexically shallow, and thus exhibit suboptimal performance in capturing the semantic quality of generated text (Freitag et al., 2022). (2) embedding-based methods: **BERTScore** (Zhang et al., 2019) leverages contextual embeddings to calculate cosine similarity between candidate and reference tokens, with greedy matching for precision, recall, and F1 scores, which are derived from the pre-trained model BERT (Devlin et al., 2019). **MoverScore** (Zhao et al., 2019) combines contextualized embeddings with Earth Mover's Distance to measure semantic similarity between system outputs and references. **FactCC** (Kryściński et al., 2020) is a weakly-supervised BERT-based model to verify the factual consistency of abstractive summaries against their source documents. It uses synthetic training data generated via rule-based transformations and includes span extraction modules to highlight supporting or conflicting evidence. (3) pretrained language models: **BARTScore** (Yuan et al., 2021) formulates generated text evaluation as a text generation problem using pre-trained seq2seq BART (Lewis et al., 2020), leveraging conditional generation probabilities across source-hypothesis/reference-hypothesis directions (with prompt and fine-tuning variants) for unsupervised assessment from multiple perspectives. **USR** (Mehri and Eskenazi, 2020) is an unsupervised, reference-free dialog evaluation metric using fine-tuned RoBERTa for MLM and dialog retrieval, combining interpretable sub-metrics via configurable regression to align with human judgments. **UniEval** (Zhong et al., 2022) reframes multi-dimensional NLG evaluation as a unified Boolean Question Answering task using a pre-trained T5 model (Raffel et al., 2020), enabling a single model to assess various dimensions. **GPTScore** (Fu et al., 2023) leverages the emergent abilities of 19 pre-trained language models to perform customizable, multi-aspect, and training-free evaluation of generated texts via natural language instructions and conditional generation probability calculation. (4) LLM-based methods: **G-Eval** (Liu et al., 2023) employs large language models with chain-of-thought (Wei et al., 2022) prompting and a structured form-filling format to perform holistic, reference-free evaluation of NLG outputs. Since the prompts on the Topical-Chat dataset were not available, we utilized the G-Eval-related dataset provided by Chiang and Lee (2023). **SEEval** (Wu et al., 2025) is a novel prompting method grounded in the educational psychology principle of self-explanation (Berry, 1983), which enhances large language models' evaluation accuracy by prompting them to articulate task-specific criteria before scoring. **CheckEval** (Lee et al., 2025) improves LLM-as-a-Judge reliability through decomposing evaluation criteria into a checklist of binary questions to boost agreement and reduce variance.

B Complete Experiment Results

B.1 Main Results

Tables 12, 14, and 17 present the full experimental results of SEAL and other baselines on the SummEval, Topical-Chat, and QAGS datasets, respectively. The experiments were conducted using two open-source models(Deepseek-V3, Qwen3-235b) and two closed-source models(GPT-4o, GPT-4 Turbo). The best experimental results have been **bolded**, and the findings validate the effectiveness of SEAL.

B.2 Ablation Analysis

Table 13 presents the detailed results of the ablation studies, specifically showing the performance of removing semantic unit decomposition (w/o SD) and removing sub-dimension verification (w/o SV) when using GPT-4o and GPT-4 Turbo as the judge LLMs. Table 15 further shows the detailed results of discarding aggregate weighting (w/o weights) under the same two judge LLMs.

We also conducted the same ablation study (w/o weights) on the QAGS. As shown in Table 16, this modification consistently degrades correlation with human judgments, aligning with our findings on Topical-Chat and reinforcing the importance of learned weighting in accurately fusing fine-grained sub-dimension scores, particularly when different aspects contribute unequally to overall quality.

B.3 Additional Analysis

Table 18 presents the token consumption and average latency of SEAL on the Topical-Chat dataset and shows that SEAL incurs higher token usage and longer inference time than G-Eval, which employs a single holistic prompt per dimension. This

Type	Evaluation Method	Coherence		Consistency		Fluency		Relevance		Average	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
non-LLM	ROUGE-1	0.167	0.126	0.160	0.130	0.115	0.094	0.326	0.252	0.192	0.150
	ROUGE-2	0.184	0.139	0.187	0.155	0.159	0.128	0.290	0.219	0.205	0.161
	ROUGE-L	0.128	0.099	0.115	0.092	0.105	0.084	0.311	0.237	0.165	0.128
	BERTScore	0.284	0.211	0.110	0.090	0.193	0.158	0.312	0.243	0.225	0.175
	MOVERSScore	0.159	0.118	0.157	0.127	0.129	0.105	0.318	0.244	0.191	0.148
	BARTScore	0.448	0.342	0.382	0.315	0.356	0.292	0.356	0.273	0.385	0.305
	UniEval	0.575	0.442	0.446	0.371	0.449	0.371	0.426	0.325	0.474	0.377
Open-source LLM-Based	***Qwen3***										
	G-Eval	0.4632	0.3597	0.4621	0.3932	0.3946	0.3638	0.3857	0.2937	0.4264	0.3526
	SEEval	0.4922	0.3986	0.4217	0.3804	0.4601	0.4051	0.4442	0.3225	0.4546	0.3767
	CheckEval	0.5598	0.4402	0.4171	0.3723	0.4709	0.4149	0.4740	0.3496	0.4805	0.3943
	SEAL	0.5470	0.4064	0.4956	0.4436	0.4741	0.4410	0.4651	0.3897	**0.4955**	**0.4202**
	Deepseek-v3										
	G-Eval	0.3447	0.2651	0.4307	0.3655	0.2334	0.2198	0.4081	0.3200	0.4352	0.2926
	SEEval	0.5336	0.4424	0.4461	0.4125	0.4526	0.3976	0.4479	0.3261	0.4701	0.3947
	CheckEval	0.5312	0.4389	0.4402	0.4042	0.2266	0.2139	0.4146	0.3266	0.4032	0.3459
	SEAL	0.5157	0.3935	0.4634	0.4336	0.5046	0.4475	0.5064	0.4169	**0.4978**	**0.4229**
closed-source LLM-Based	***GPT-4o***										
	G-Eval	0.1896	0.1581	0.4219	0.3911	0.2862	0.2676	0.3969	0.3421	0.3237	0.2897
	SEEval	0.3391	0.3618	0.4421	0.4162	0.3665	0.3512	0.4021	0.3617	0.3875	0.3727
	CheckEval	0.5564	**0.4644**	0.5304	0.4738	0.4699	0.4125	0.4602	0.4001	0.5042	0.4377
	SEAL	**0.5613**	0.4373	**0.5396**	**0.5018**	**0.5514**	**0.4986**	**0.5205**	**0.4195**	**0.5432**	**0.4643**
	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
	SEEval	0.5292	0.4621	0.6351	0.6031	0.3551	0.3327	0.4728	0.4501	0.4981	0.4620
	CheckEval	0.5807	0.4901	0.6232	0.5872	0.4611	0.4058	0.4197	0.3713	0.5212	0.4636
	SEAL	0.5699	0.4318	0.5738	0.5334	0.5218	0.4717	0.5200	0.4193	**0.5464**	**0.4641**

Table 12: Sample-level Spearman (ρ) and Kendall tau (τ) correlations on the full results of SummEval.

LLM	Metric	Coherence		Consistency		Fluency		Relevance		Average	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
GPT-4o	SEAL	**0.5613**	**0.4373**	**0.5396**	**0.5018**	**0.5514**	**0.4986**	**0.5205**	**0.4195**	**0.5432**	**0.4643**
	- w/o SD	0.3773	0.2678	0.3543	0.3083	0.2223	0.2003	0.3563	0.2827	0.3275	0.2660
	- w/o SV	0.4662	0.3180	0.4755	0.4064	0.4706	0.4068	0.3190	0.2579	0.4328	0.3472
GPT-4	SEAL	**0.5699**	**0.4318**	**0.5738**	**0.5334**	**0.5218**	**0.4717**	**0.5200**	**0.4193**	**0.5464**	**0.4641**
	- w/o SD	0.3593	0.2528	0.3512	0.3030	0.2111	0.1903	0.3456	0.2723	0.3168	0.2546
	- w/o SV	0.4687	0.3210	0.4820	0.4107	0.4708	0.4041	0.3358	0.2723	0.4393	0.3520

Table 13: Spearman (ρ) and Kendall's Tau (τ) correlation of ablation variants on SummEval. SD denotes *Semantic Unit Decomposition* and SV denotes *Sub-Dimension Verification*

.

overhead stems from SEAL 's semantic unit decomposition and multi-step binary evaluation, which enable substantially higher correlation with human judgments. Compared to CheckEval, SEAL exhibits only marginally higher computational costs, yet achieves consistently better evaluation accuracy. These results suggest that SEAL strikes a favorable balance between efficiency and effectiveness, delivering significant quality gains without prohibitive resource demands.

We vary the sentences and monadict-fact threshold n that gates holistic versus fine-grained evaluation. Table 19 shows the full results of our decomposition threshold analysis on SummEval. For sentence-based dimensions (coherence and fluency) we sweep $n \in \{2, 3, 4, 5\}$; for monadic-fact dimensions (consistency and relevance) we sweep $n \in \{5, 6, 7, 8\}$. Performance exhibits only minor fluctuations across different values of n. While the choice of n introduces minor variations in correlation scores, the overall performance remains stable across a reasonable range of values, suggesting that SEAL 's effectiveness is not highly sensitive to this hyperparameter.

C Case Study

To illustrate the advantages of our approach, we present two representative case studies from the SummEval(doc_id: `dm-test-8b51a10370a12191 37e231192e65b9cd52238ea6`, system_id: `M9`)and Topical-Chat datasets(source: `are you a fan of the nfl ? i definitely am . watching the super bowl has become a family tradition . what about you ?`, system_id: `Original`

Type	Evaluation Method	Naturalness		Coherence		Engagingness		Groundedness		Average	
		ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
non-LLM	ROUGE-L	0.176	0.146	0.193	0.203	0.295	0.300	0.310	0.327	0.243	0.244
	BLEU-4	0.180	0.175	0.131	0.235	0.232	0.316	0.213	0.310	0.189	0.259
	METEOR	0.212	0.191	0.250	0.302	0.367	0.439	0.333	0.391	0.290	0.331
	BERTScore	0.226	0.209	0.214	0.233	0.317	0.335	0.291	0.317	0.262	0.273
	USR	0.337	0.325	0.416	0.377	0.456	0.465	0.222	0.447	0.358	0.403
	UniEval	0.455	0.330	0.602	0.455	0.573	0.430	0.577*	0.453	0.552	0.417
Open-source LLM-Based	***Qwen3***										
	G-Eval	0.5437	0.5284	0.4933	0.4860	0.6205	0.5924	0.5352	0.5398	0.5482	0.5367
	SEEval	0.5872	0.5699	0.5110	0.4992	0.6663	0.6460	0.6103	0.6150	0.5937	0.5825
	CheckEval	0.6298	0.6156	0.4973	0.4932	0.6667	0.6519	0.5338	0.5493	0.5819	0.5775
	SEAL	0.5624	0.5436	0.5904	0.5731	0.6799	0.6689	0.5695	0.5704	**0.6006**	**0.5890**
	Deepseek-v3										
	G-Eval	0.5426	0.5290	0.4767	0.4876	0.5826	0.5781	0.5532	0.5640	0.5388	0.5397
	SEEval	0.5690	0.5583	0.4942	0.4988	0.6911	0.6843	0.5458	0.5569	0.5750	0.5746
	CheckEval	0.5615	0.5514	0.6100	0.5981	0.5600	0.5569	0.5599	0.5697	0.5729	0.5690
	SEAL	0.6067	0.6207	0.6402	0.6198	0.6292	0.6206	0.5750	0.5780	**0.6128**	**0.6098**
Closed-source LLM-Based	***GPT-4o***										
	G-Eval	0.5917	0.5669	0.6111	0.5770	0.3903	0.1655	0.4770	0.4255	0.5175	0.4337
	SEEval	0.6011	0.5881	0.6551	0.5822	0.4512	0.2620	0.5331	0.4627	0.5601	0.4738
	CheckEval	0.5889	0.5790	0.7362	0.7354	0.5869	0.5761	0.6462	0.6448	0.6395	0.6338
	SEAL	0.6351	0.6332	0.6932	0.6857	0.6747	0.6706	0.5980	0.5972	**0.6503**	**0.6467**
	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
	SEEval	0.5012	0.5162	0.7123	0.7221	0.6232	0.6231	0.5829	0.5922	0.6049	0.6134
	CheckEval	0.5209	0.5232	0.7367	0.7438	0.6292	0.6341	0.6425	0.6476	0.6323	0.6372
	SEAL	0.6233	0.6315	0.6998	0.7004	0.6953	0.6887	0.5838	0.5762	**0.6506**	**0.6492**

Table 14: Turn-level Spearman (ρ) and Pearson (r) correlations on the full results of Topical-Chat

LLM	Metric	Naturalness		Coherence		Engagingness		Groundedness		Average	
		ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
GPT-4o	SEAL	0.6351	0.6332	0.6932	0.6857	0.6747	0.6706	0.5980	0.5972	**0.6503**	**0.6467**
	- w/o weights	0.6259	0.6306	0.6606	0.6300	0.6475	0.6415	0.6122	0.6175	0.6365	0.6299
GPT-4	SEAL	0.6233	0.6315	0.6998	0.7004	0.6953	0.6887	0.5838	0.5762	**0.6506**	**0.6492**
	- w/o weights	0.6500	0.6415	0.6415	0.6261	0.6771	0.6716	0.5839	0.5941	0.6381	0.6333

Table 15: Spearman (ρ) and Pearson (r) correlations of ablation variants on Topical-chat.

`Ground Truth`)), respectively. Details are shown in Table 20.

In the first case (fluency on SummEval), the system output consists of five grammatically sound and well-formed sentences. SEAL correctly assigns a perfect fluency score by evaluating each sentence individually and confirming their linguistic acceptability. In contrast, G-Eval using a holistic 1–3 scale, consistently rates the same output as "2" across multiple runs, likely due to its coarse-grained prompt failing to distinguish between minor stylistic preferences and actual fluency errors. This highlights how monolithic scoring can conflate subjective tone with objective grammaticality, whereas SEAL 's atomic binary judgments align more closely with human notions of fluency.

In the second case (Topical-Chat, engagingness), G-Eval not only yields a low score of 1.0 but also fails to identify the specific source of the deficit. SEAL decomposes the response into five weighted sub-dimensions and consistently flags Q4 (Personalization) as 0 while keeping the remaining criteria at 1; the resulting score of 0.80, mirrors the human rating of 2.67/3.

These cases demonstrate that SEAL not only yields more accurate scores but also provides transparent, interpretable rationales for its evaluations, offering actionable insights beyond what black-box scoring can deliver.

D Prompts

Table 21–22 shows detailed prompts on SummEval.Table 23–24 shows detailed prompts on Topical-chat.Table 25 shows detailed prompts on QAGS.

E Use of LLMs

We used AI tools only for language polishing, grammar checking, and code debugging. All ideas, analysis, and scientific content were developed solely by the authors.

LLM	Metric	QAGS-CNN			QAGS-XSUM			Average		
		r	ρ	τ	r	ρ	τ	r	ρ	τ
GPT-4o	SEAL	**0.6447**	0.6298	0.5478	**0.5898**	**0.6377**	**0.5529**	**0.6173**	**0.6338**	**0.5504**
	- w/o weights	0.6360	**0.6609**	**0.5760**	0.5849	0.5995	0.5177	0.6105	0.6302	0.5469
GPT-4	SEAL	**0.7202**	0.7159	**0.5990**	**0.6398**	**0.6427**	**0.5323**	**0.6800**	**0.6793**	**0.5657**
	- w/o weights	0.6974	**0.7160**	0.5971	0.5066	0.4971	0.4171	0.6020	0.6066	0.5071

Table 16: Spearman (ρ) and Pearson (r) correlations of ablation variants on QAGS

Type	Evaluation Method	QAGS-CNN			QAGS-XSUM			Average		
		r	ρ	τ	r	ρ	τ	r	ρ	τ
non-LLM	ROUGE-2	0.459	0.418	0.333	0.097	0.083	0.068	0.278	0.250	0.200
	ROUGE-L	0.357	0.324	0.254	0.024	-0.011	-0.009	0.190	0.156	0.122
	BERTScore	0.576	0.505	0.399	0.024	0.008	0.006	0.300	0.256	0.202
	MoverScore	0.414	0.347	0.271	0.054	0.044	0.036	0.234	0.195	0.153
	FactCC	0.416	0.484	0.376	0.297	0.259	0.212	0.356	0.371	0.294
	QAGS	0.545	–	–	0.175	–	–	0.375	–	–
	BARTScore	0.735	0.680	0.557	0.184	0.159	0.130	0.459	0.420	0.343
	UniEval	0.682	0.662	0.532	0.461	0.488	0.399	0.571	0.575	0.465
Open-source LLM-Based	***Qwen3***									
	G-Eval	0.4211	0.5339	0.4999	0.5276	0.5293	0.5282	0.4744	0.5316	0.5141
	CheckEval	0.5955	0.6375	0.5958	0.5797	0.5797	0.5797	0.5876	0.6086	**0.5878**
	SEAL	0.6610	0.6609	0.5841	0.6219	0.6337	0.5700	**0.6415**	**0.6473**	0.5771
	Deepseek-v3									
	G-Eval	0.3988	0.4641	0.3896	0.2662	0.2666	0.2444	0.3325	0.3664	0.3170
	CheckEval	0.6366	0.6474	0.6050	0.4888	0.4888	0.4888	0.5627	0.5681	0.5469
	SEAL	0.6809	0.6589	0.5928	0.5911	0.6015	0.5653	**0.6360**	**0.6302**	**0.5791**
Closed-source LLM-Based	***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
	SEAL	0.6447	0.6298	0.5478	0.5898	0.6377	0.5529	**0.6173**	**0.6338**	**0.5504**
	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**
	SEAL	0.7202	0.7159	0.5990	0.6398	0.6427	0.5323	**0.6800**	**0.6793**	0.5657

Table 17: Pearson (r), Spearman (ρ), and Kendall-Tau (τ) correlations on the full results of QAGS-CNNDM and QAGS-XSUM.

	Naturalness		Coherence		Engagingness		Groundedness		Average	
	tokens	Latency (s)	tokens	Latency (s)	tokens	Latency (s)	tokens	Latency (s)	tokens	Latency (s)
G-Eval	622.51	1.406	656.58	1.353	647.82	1.427	628.84	1.369	638.93	1.388
CheckEval	820.48	1.925	852.06	1.680	849.56	1.267	818.32	1.687	826.52	1.638
SEAL	899.46	1.950	1078.72	2.025	904.74	1.941	899.34	2.014	945.56	1.982

Table 18: Average number of tokens and Latency (in seconds) per method and evaluation dimension on Topical-chat.

n	coherence		fluency		consistency		relevance	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ
2	0.5185	0.3976	0.5137	0.4453	-	-	-	-
3	**0.5613**	**0.4373**	**0.5514**	**0.4986**	-	-	-	-
4	0.5479	0.4123	0.5241	0.4593	-	-	-	-
5	0.5452	0.4070	0.4950	0.4567	0.4719	0.4344	0.5173	0.4103
6	-	-	-	-	**0.5396**	**0.5018**	**0.5205**	**0.4195**
7	-	-	-	-	0.5220	0.4831	0.5050	0.4099
8	-	-	-	-	0.4819	0.4396	0.4961	0.4031

Table 19: Spearman (ρ), and Kendall-Tau (τ) correlations of SEAL under varying decomposition threshold n across four dimensions.

Dataset	Field	Content
SummEval	Source	There would have been no mercy in the dressing room . Nothing is sacred in the team . When Ben Stokes first turned up at the Kensington Oval this week someone would have said : ‘ Mind that locker , Stokesey. ’ It was here , of course , that Stokes broke his wrist punching a locker after being dismissed last year and that will be a reminder to him of the need to channel his aggression in the right way now he is back in the England team . Ben Stokes has forged a reputation as a fiery character in the England cricket team Stokes (left) clashed with Marlon Samuels during the second Test between West Indies and England But Stokes (right) responded positively to the row after talking to England coach Peter Moores Only a slight niggle in his back might stop Stokes playing here in Barbados and I am glad he has fought his way back after what has been a year of ups and downs . I like Stokes and I like his character at a time when the game can be perceived as producing robots and players who can not capture the imagination . He has something about him , as he showed when he was in Mitchell Johnson ’ s face during the Ashes series . He was not worried about the bombardment he would get in return . But I do believe he will need careful man-managing . Peter Moores , England ’ s coach , dragged him to one side after his clash with Marlon Samuels in the second Test and had a long chat with him and if it was to talk about how he should respond then the advice he gave was clearly right . The way he approached the Jamaican after that third-day salute was spot on . England must make sure they do not quash that natural exuberance and desire to take someone on . It will not always come off but we have to accept that there will be lows as well as highs with this lad . What Stokes must not do is try to live up to his reputation and become pigeon-holed . He can be feisty and fiery as long as it ’ s natural but his aggression has to be controlled . You want the character as long as the end product is runs and wickets . Stokes (second left) can use his aggression to bring runs and wickets to the England team Stokes (left) congratulates England captain Alastair Cook after the side won the second Test England want to have Stokes in the team as much as possible to help him hit his full potential Let him go a bit but rein him in occasionally . There is an argument here to play the second spinner in Adil Rashid and if England went down that route in Bridgetown the place of Stokes or Chris Jordan would come into question but they clearly want both to play as much as possible . This summer there will rarely be the need for two spinners so England want these two exciting cricketers to grow into the international game . They can only get better and that ’ s an exciting prospect .
	Reference	Ben stokes forged a reputation as a fiery character in the england cricket team . Stokes broke his wrist punching a locker after being dismissed last year . Stokes clashed with marlon samuels during the second test . Ben stokes first turned up at the kensington oval this week . Stokes responded positively to the row after talking to england coach peter moores .
	System Output	Ben Stokes returns to the ground where he broke his wrist hitting a locker . He clashed with West Indies batsman Marlon Samuel in the second Test . But his aggression , when controlled , is important to England .
	Scores	coherence: 1.67, consistency: 5, fluency: 5, relevance: 4, overall: 3.92
Topical-Chat	Source	are you a fan of the nfl ? i definitely am . watching the super bowl has become a family tradition . what about you ?
	System Output	the same i like watching both college and nfl . do you have a favorite team ?
	Context	new orleans saints cheerleaders are forbidden from eating in the same restaurant as any nfl player and if they are already dining at a restaurant and an nfl player comes in after , the cheerleaders are required to leave .
	Scores	naturalness: 2.67, coherence: 3, engagingness: 2.67, groundedness: 0.33

Table 20: **Case study examples from SummEval and Topical-Chat.**

Step 1	Prompt
Sentence Split	You will be given a source text and a summary. Task (Stage 1): Split the summary into individual sentences in their original order. Rules: Only split the summary; do NOT evaluate quality. Do NOT add, delete, or rewrite content. Keep each sentence as a clean string (trim leading/trailing whitespace). If the summary is empty or only whitespace, output an empty list. If the summary is NOT empty but you are unsure how to split, return a single-item list containing the full summary as one sentence. Output JSON ONLY with the key "sentence". Output format (JSON only): {"sentence": ["sentence 1", "sentence 2"]}
Monadic Fact Generation	You will be provided with a Source text, a Reference text and a Summary. You need to split them into "Monadic facts". A monadic fact refers to a semantically indivisible information unit that satisfies the following criteria: 1. Self-contained proposition: Contains the minimal elements required to form a complete statement (typically a single subject-predicate-object or subject-copula-complement structure) 2. Entity parsimony: Contains no more than three essential entities (e.g., agent + action + patient) 3. Verifiability independence: Can be validated without requiring additional contextual statements 4. Structural atomicity: Maintains propositional integrity when stripped of all modifiers (adverbial phrases, attributive clauses, etc.) 5. Logical explicitness: Contains no implied meanings or inferential components requiring deductive reasoning Instruction: First, read the source text, reference text and summary carefully. Second, split every sentence into monadic facts. Rules: - Do NOT evaluate correctness or support in this stage. - Do NOT add new information. - Keep each fact atomic and concise. - Output JSON ONLY with exactly these keys: - "source text monadic facts" - "reference text monadic facts" - "summary monadic facts" Output format (JSON only): { "source text monadic facts": ["c1:<first source text monadic fact>", "c2:<second source text monadic fact>"], "reference text monadic facts": ["r1:<first reference text monadic fact>", "r2:<second reference text monadic fact>"], "summary monadic facts": ["s1:<first summary monadic fact>", "s2:<second summary monadic fact>"] }

Table 21: **Prompt for Sentences Split and Monadic Fact Generation.**

Dimension	Prompt
Coherence	You will be given a source text and a summary split into sentences (s_1, s_2, ..., s_n). Task: Evaluate the INTERNAL COHERENCE of the summary by judging whether each adjacent transition is coherent:(s_1->s_2), (s_2->s_3), ..., (s_{n-1}->s_n). Definition of "coherent" (adjacent transition): The next sentence follows naturally from the previous one in logic/topic/time, or It is a reasonable elaboration/addition/contrast that is clearly connected, AND the transition does not feel abrupt or missing context. Important: Focus on sentence-to-sentence flow; ignore factual correctness vs the source. Output 1 if the transition is coherent, else 0. Output EXACTLY expected numbers aligned to the adjacent order. Output JSON ONLY: {"adjacent coherence relation": "1,0,1"}
Fluency	You will be provided with a summary split into sentences:s_1, s_2, ..., s_n. Task: For each sentence s_i, determine whether it is fluent. Fluency definition: Fluent if grammatically correct, natural, and easy to read with correct punctuation/spacing. Not fluent if it contains formatting errors or grammatical errors. Note: Ignore capitalization errors except for the first letter of each sentence. Output: For each sentence output 1 if fluent else 0. Output exactly expected numbers aligned to $s_1..s_{expected}$. Output JSON ONLY with the key "fluency relation between sentences". Output format (JSON only): {"fluency relation between sentences": "1,0,1"}

Table 22: **Prompts for SummEval Evaluation Dimensions (Part 1 of 2).**

Consistency	You will be provided with summary monadic facts ($s_1..s_n$) and source monadic facts ($c_1..c_m$). Task: For each summary fact s_i, determine whether it is supported by one or more source facts c_j. If it is supported, list the supporting c_j identifiers. Support rules (what counts as supported): - s_i is supported only if it is explicitly stated or directly entailed by one or more source facts c_j. - Support must cover the full core proposition of s_i (who did what to whom), including any key constraints present in s_i such as: numbers/quantities, dates/times, locations, comparisons/superlatives, causal claims, and attributions ("X said/announced"). Non-support rules (when you must not support; output []): - If s_i introduces any key detail that is not present in the source facts (e.g., a new number/date/place, a new cause, a new comparison, a new attribution), do NOT support. - If s_i is only topically related to the source facts but not factually supported, do NOT support. - Do not use external knowledge, common sense, or multi-hop reasoning to create support. Use only the provided source facts. - If s_i is contradicted by the source facts, do not support. Allowed normalization (only after support is established): - Synonym substitution (same meaning) - Voice conversion (active/passive) - Entity normalization (e.g., "Apple Inc." -> "Apple") Output: - For each s_i, output a list of supporting c_j IDs (e.g., ["c_2","c_5"]). - If s_i is not supported, output an empty list []. - The output must be a list of length n, aligned to $s_1..s_n$. - Output JSON ONLY with the key "supporting source facts for each summary fact". Output format (JSON only): {"supporting source facts for each summary fact": [["c2","c5"], [], ["c1"]]}
Relevance	You will be provided with: - summary monadic facts ($s_1..s_n$) - reference monadic facts ($r_1..r_k$) - source monadic facts ($c_1..c_m$) Your task is to evaluate RELEVANCE of each summary fact s_i. Definition: - s_i is relevant (1) if it expresses a core idea from the reference/source (main topic or key detail). - s_i is not relevant (0) if it is redundant, trivial, off-topic, or does not contribute meaningful aligned content. Instruction: 1. For each s_i, decide 1 or 0. 2. Output exactly {n} numbers aligned to $s_1..s_n$. 3. Output JSON ONLY with key "relevance relation between summary facts". Output format (JSON only): {"relevance relation between summary facts": "1,0,1"}

Table 22: **Prompts for SummEval Evaluation Dimensions (Part 2 of 2).**

Dimension	Prompt
Naturalness	You will be provided with a dialogue history between two people, along with context information for the dialogue, which typically includes background knowledge or related events, and also the next response in the conversation. Evaluate the naturalness of the response by answering these five yes/no questions (1 = Yes, 0 = No): Q1 (Fluency): Is the response grammatically correct and free of awkward or unnatural phrasing? Q2 (Register match): Does the formality level, use of contractions, and conversational style match the dialogue context and speaker role? Q3 (Turn alignment): Does the response appropriately acknowledge or respond to the intent of the previous turn (e.g., answers questions, reacts to emotions)? Q4 (Lexical variation): Does the response avoid unnecessary verbatim repetition of words or phrases from the immediately preceding turn? Q5 (Natural expressiveness): Does the response use sentence forms that suggest natural spoken intonation (e.g., questions, ellipsis, interjections)? Then assign a weight to each criterion based on its importance for naturalness in this specific dialogue. - Weights: non-negative decimals summing to exactly 1.0. Please note that your answer should be a JSON with exactly two fields: -"naturalness": "s1,s2,s3,s4,s5" (each 0 or 1) -"naturalness_weight": "w1,w2,w3,w4,w5" (sum=1.0) Example:{"naturalness": "1,1,0,1,1", "naturalness_weight": "0.25,0.20,0.20,0.15,0.20"}
Coherence	You will be provided with a dialogue history between two people, along with context information for the dialogue, which typically includes background knowledge or related events, and also the next response in the conversation. Your task is to evaluate the coherence of this response by answering the following six yes/no questions. For each, assign: - 1 if "Yes" (the response satisfies the criterion) - 0 if "No" (it violates or fails to satisfy) Q1 (Referential continuity): Do all pronouns, definite noun phrases (e.g., "the book", "that idea"), or anaphoric expressions in the response have a clear antecedent in the dialogue history or context? Q2 (Logical consistency): Does the response contradict any explicitly stated fact in the dialogue history or context? Q3 (Discourse move alignment): Does the response perform an appropriate conversational move given the prior turn (e.g., answers a question, acknowledges a statement, continues a story)? Q4 (Topical bridging): If the response introduces a new topic, does it use a bridging phrase or clearly link back to a concept from earlier turns? Q5 (Temporal/causal order): Are events or actions mentioned in the response temporally or causally consistent with what was said before? Q6 (Presupposition support): Does the response avoid making assumptions (presuppositions) that are not supported by the dialogue history or context? Then, assign a weight to each of the six criteria based on how critical it is for coherence in this specific dialogue. - Weights must be non-negative decimals. - The sum of all six weights must be exactly 1.0. - For example, in a factual QA dialogue, Q2 and Q6 may be more important; in a narrative, Q5 may dominate. Please note that your answer should be a JSON with exactly two fields: - "coherence": "s1,s2,s3,s4,s5,s6" (each si is 0 or 1) - "coherence_weight": "w1,w2,w3,w4,w5,w6" (each wi is a decimal, sum=1.0) Example:{"coherence": "1,1,0,1,1,0", "coherence_weight": "0.20,0.25,0.15,0.10,0.20,0.10"}

Table 23: **Prompts for Topical-Chat Evaluation Dimensions (Part 1 of 2).**

Dimension	Prompt
Engagingness	You will be provided with a dialogue history between two people, along with context information for the dialogue, which typically includes background knowledge or related events, and also the next response in the conversation. Evaluate the engagingness of the response by answering these five yes/no questions (1 = Yes, 0 = No): Q1 (Interest): Does the response introduce interesting or novel content that encourages continued conversation? Q2 (Emotional resonance): Does the response convey appropriate emotion, enthusiasm, or empathy that makes it feel human and relatable? Q3 (Initiative): Does the response proactively ask questions, suggest topics, or invite further interaction rather than giving a flat or terminal reply? Q4 (Personalization): Does the response reflect personal opinions, experiences, or preferences (when appropriate), making it feel unique and not generic? Q5 (Coherence with interest flow): Does the response maintain or enhance the conversational momentum by building on shared interests or prior engagement cues? Then assign a weight to each criterion based on its importance for engagingness in this specific dialogue. Weights: non-negative decimals summing to exactly 1.0. Please note that your answer should be a JSON with exactly two fields: - "engagingness": "s1,s2,s3,s4,s5" (each 0 or 1) - "engagingness_weight": "w1,w2,w3,w4,w5" (sum=1.0) Example:{"engagingness": "1,1,0,1,1", "engagingness_weight": "0.25,0.20,0.20,0.15,0.20"}
Groundedness	You will be provided with a dialogue history between two people, along with context information for the dialogue, which typically includes background knowledge or related events, and also the next response in the conversation. Evaluate the groundedness of the response by answering these five yes/no questions (1 = Yes, 0 = No): Q1 (Factuality): Is the information provided in the response supported by the provided context information? Q2 (No Hallucination): Does the response avoid making up "facts" or external details that are not present in the context or common knowledge? Q3 (Knowledge Integration): Does the response smoothly incorporate the background knowledge into the flow of conversation rather than just "copy-pasting" it? Q4 (Accuracy): If the response mentions specific names, dates, or statistics from the context, are they used correctly? Q5 (Contextual Relevance): Is the specific piece of knowledge chosen from the context actually relevant to the current turn of the dialogue? Then assign a weight to each criterion based on its importance for groundedness in this specific dialogue. Weights: non-negative decimals summing to exactly 1.0. Please note that your answer should be a JSON with exactly two fields: -"groundedness": "s1,s2,s3,s4,s5" (each 0 or 1) -"groundedness_weight": "w1,w2,w3,w4,w5" (sum=1.0) Example:{"groundedness": "1,1,1,1,0", "groundedness_weight": "0.20,0.20,0.15,0.15,0.30"}

Table 24: **Prompts for Topical-Chat Evaluation Dimensions (Part 2 of 2).**

Dimension	Prompt
Consistency	You will be provided with a Source Document and a generated Summary of that document. Evaluate the consistency of the summary by answering these five yes/no questions (1 = Yes, 0 = No): Q1 (No Hallucination): Are all the facts, names, and entities mentioned in the summary explicitly present in the source document? Q2 (Relational Accuracy): Does the summary correctly represent the relationships between entities (e.g., who said what, or the cause-and-effect of events) as described in the source? Q3 (Numerical/Attribute Integrity): Are any numbers, dates, or quantities in the summary identical in meaning to those in the source document? Q4 (No Contradiction): Does the summary avoid making any claims that are directly refuted or contradicted by the information in the source? Q5 (Contextual Fidelity): Does the summary maintain the original intent and tone of the source without twisting the author's meaning? Then assign a weight to each criterion based on its importance for consistency in this specific summary. Weights: non-negative decimals summing to exactly 1.0. Please note that your answer should be a JSON with exactly two fields: - "consistency": "s1,s2,s3,s4,s5" (each 0 or 1) - "consistency_weight": "w1,w2,w3,w4,w5" (sum=1.0) Example: {"consistency": "1,1,1,0,1", "consistency_weight": "0.30,0.20,0.15,0.25,0.10"}

Table 25: **Prompt for QAGS Consistency Evaluation.**