DeepJSONEval: Benchmarking Complex Nested JSON Data Mining for Large Language Models

Zhicheng Zhou*, **Jing Li***, **Suming Qiu, Junjie Huang, Linyuan Qiu, Zhijie Sun**[†]

[1]GTS, Huawei Technologies Co., Ltd
Correspondence to: sunzhijie3@huawei.com

Abstract

The internet is saturated with low-density, high-redundancy information, such as social media comments, repetitive news, and lengthy discussions, making it difficult to extract valuable insights efficiently. Multi-layer nested JSON structures provide an effective solution by compressing such information into semantically rich, hierarchical representations, which organize data into key-value pairs, arrays, and nested objects, preserving contextual relationships and enabling efficient storage, retrieval, and semantic querying. For instance, in news aggregation, a JSON object can nest an article's metadata (title, author, date), content (text, multimedia), and multimedia information (multimedia type, caption) hierarchically. Large Language Models (LLMs) play a transformative role in web data mining by parsing unstructured text and outputting structured results directly into complex JSON schemas. However, current benchmarks for evaluating LLMs' JSON output capabilities overemphasize pure JSON generation rather than assessing data comprehension and extraction abilities, a limitation that lacks relevance to practical web data mining tasks. To address this, we introduce DeepJSONEval, a novel benchmark featuring 2100 multi-domain instances with deep nested structures, categorized by difficulty. Experiments show significant performance gaps among LLMs in handling such complexity. Our benchmark and datasets are open-sourced to advance research in structured JSON generation. (https://github.com/GTS-AI-Infra-Lab-SotaS/DeepJSONEval).

Introduction

The exponential growth of digital content has created an information extraction paradox: while data volume increases dramatically, information density remains critically low due to redundant social media posts, repetitive news coverage, and poorly structured web content. This sparsity-abundance contradiction demands robust information condensation techniques that can transform noisy, heterogeneous data into structured, semantically rich representations suitable for computational analysis.

Multi-layer nested JSON structures have emerged as a powerful solution, enabling hierarchical compression of sparse information through systematic key-value organization and nested object relationships. Unlike flat data for-

mats, nested JSON preserves complex semantic dependencies while maintaining both machine readability and human interpretability—essential for capturing nuanced information relationships in domains ranging from news aggregation (article metadata nested with content analysis and multimedia details) to financial analytics (stock data with nested performance metrics and temporal indicators)(Syafiq, Azri, and Ujang 2025; Chinta 2025).

Recent advancements in artificial intelligence, particularly the emergence of Large Language Models (LLMs), like GPT-4, have become transformative agents in web mining and content analysis, fundamentally altering how people extract, interpret, and structure information from the vast and often chaotic digital corpus (Guo et al. 2025).LLMs' capabilities in natural language understanding, content summarization, entity identification, and relationship inference allow them to convert raw, noisy web data, such as news articles, forum discussions, and product reviews into structured, actionable JSON (see Figure 1) (Xu, Ding, and Wang 2025).

However, evaluating these capabilities presents significant challenges, as traditional benchmarks often fail to capture the complexity and nuanced requirements of extracting key information from information-saprse web data and convert the key information into multi-lavel nested JSON structrue. Current benchmarks lack standardized evaluation of multi-layer JSON generation quality. This gap necessitates a comprehensive evaluation dataset that can systematically measure the fidelity, completeness, and structural appropriateness of JSON-based information extraction systems, moving beyond traditional metrics that fail to capture the full spectrum of challenges in web data mining applications.

While existing evaluation frameworks have made notable advances in assessing LLM capabilities across various dimensions, including contamination-resistant evaluation(White et al. 2024), instruction-following verification(Zhou et al. 2023), mobile function calling(Wang et al. 2024), schema-based information extraction(Gui et al. 2024; Geng et al. 2025; Yang et al. 2025). Systematic analysis reveals fundamental gaps that limit their applicability to real-world JSON-based information extraction.

A fundamental conceptual distinction emerges: existing benchmarks (Yang et al. 2025; Liu et al. 2024; Xia et al. 2024; Geng et al. 2025) predominantly frame JSON generation as schema-adherence tasks, wherein models synthesize

---

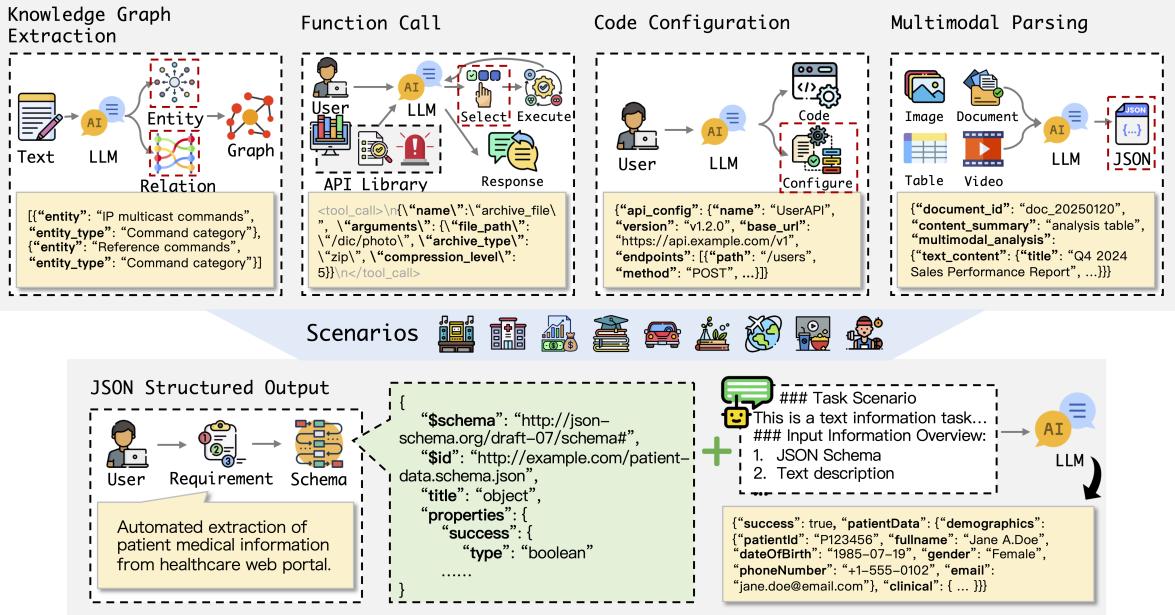*These authors contributed equally.

[†]Corresponding author.

Figure 1: The representative application scenarios for multi-layer nested JSON with LLM.

structures from templates without source content, providing limited evaluation of information extraction from unstructured text—essential for web data mining applications.

Current frameworks exhibit constraining characteristics: monolingual focus (predominantly English), limited domain diversity (exceptions: SoEval (Liu et al. 2024), Schema-Guided Dialogue (Rastogi et al. 2020)), variable structural complexity (shallow nesting in json-mode-eval (NousResearch 2024), STRUCTUREDRAG (Shorten et al. 2024), NESTFUL (Basu et al. 2025) versus deeper but domain-agnostic structures in JSONSchemaBench), and emphasis on format generation over context-based extraction.

To address these limitations, we propose DeepJSONEval, a pioneering multilingual deep-nested JSON evaluation benchmark and framework. Our approach focuses on comprehension and extraction tasks, designed to comprehensively assess LLMs' ability to map raw text to given JSON schemas and return syntactically and semantically correct multi-layer nested JSON objects in web data mining contexts. The comparison of DeepJSONEval and other JSON-related benchmarks is illustrated in Table 1.

The workflow for constructing this benchmark dataset involves systematic data collection from diverse web sources, followed by schema conceptualization through hierarchical organization of extracted concepts, automated schema generation using our novel algorithm, and ground truth compilation from refined text corpora. Our Real-time Path-Value Updating Beam Exploration for Constrained Schema Subtree Construction algorithm ensures effective construction of complex nested structures required for comprehensive evaluation.

Our approach introduces several innovations:

- An innovative **Real-time Path-Value Updating Beam**

Exploration for Constrained Schema Subtree Construction is introduced that effectively supports the construction of complex nested structures.

- The evaluation framework is designed specifically targeting **deep nesting structures**, featuring JSON schemas with 3 to 7 levels of nesting depth with 17.5 properties in average, significantly enhancing evaluation complexity and real-world applicability in web data mining. Each field includes detailed descriptions to rigorously examine instruction-following capabilities of LLMs and precise information extraction under complex format constraints and semantic understanding requirements.

- **Comprehensive data type coverage** is implemented, incorporating strings, numbers, boolean values, string enumerations, and lists to systematically evaluate LLM robustness across different data types.

- A **multi-dimensional fine-grained evaluation** framework is developed encompassing format matching accuracy, field correctness, and complete structural correctness, providing multi-perspective and detailed assessment of JSON generation quality.

DeepJSONEval establishes a new standard for objective and comprehensive evaluation of LLM structured output capabilities through its innovative evaluation dimensions, rigorous difficulty classification, detailed field descriptions, and large-scale multi-domain coverage. Our benchmark comprises 2100 high-quality data instances spanning ten diverse domains in web applications including Tourist Attraction Promotion, Electronic Devices Introduction, Patient Information, etc. We implement systematic difficulty grading based on nesting depth, categorizing 3-4 level structures as *Medium* and 5-7 level structures as *Hard*, providing progres-

Table 1: Comparative Analysis of JSON Generation and Extraction Benchmarks: Multilingual Support, Task Design, Scale, Text Provided, Structural Complexity, Domain Label Provided, and Constrained Decoding Support (CDS).

Benchmark	Multilingual	Task	Scale	Text	Depth	Label	CDS
StructEval (Yang et al. 2025)	No	Structured Generation	2035 (50 for JSON)	No	2-4	No	No
json-mode-eval (NousResearch 2024)	No	JSON Extraction	100	Yes	1-2	No	Yes
STRUCTUREDRAG (Shorten et al. 2024)	No	Format Following	112	Yes	1-2	No	No
SoEval (Liu et al. 2024)	Yes	JSON/XML Generation	3700 (200 for JSON)	No	Not Specified	Yes	Yes
FoFo (Xia et al. 2024)	No	Format Following	493	No	Not Specified	No	No
NESTFULl (Basu et al. 2025)	No	Function Calling	1800	Yes	1	No	Yes
StrucText-Eval (Gu et al. 2024)	No	Structure Interpretation	5800	No	1-3	No	No
JSONSchemaBench (Geng et al. 2025)	No	JSON Generation	6,000	No	2-23	No	Yes
Schema-Guided Dialogue (Rastogi et al. 2020)	No	Function Calling	Not Specified	No	1	Yes	No
DeepJSONEval (Ours)	**Yes**	**JSON Extractive**	**2,100**	**Yes**	**3-7**	**Yes**	**Yes**

sive evaluation benchmarks for model capabilities. Through this framework, DeepJSONEval enables systematic assessment of LLMs' structured output capabilities and advances their reliability in real-world applications.

Related Work

Recent research has witnessed growing interest in evaluating and benchmarking LLMs across diverse capabilities and domains (Mitchener et al. 2025). RocketEval (Wei et al. 2025) demonstrates that lightweight LLMs can achieve comparable evaluation accuracy through structured checklist-based assessment, though it does not specifically address structured output validation. ThinkJSON (Agarwal, Joshi, and Rojkova 2025) tackled schema adherence through reinforcement learning, focusing primarily on model training rather than comprehensive evaluation.

Critical limitations in existing benchmarks have been identified (Liu et al. 2025), emphasizing the need for qualitative attributes such as diversity, redundancy, and difficulty assessment. Data contamination significantly impacts evaluation validity, particularly in larger models (Kocyigit et al. 2025). Domain-specific evaluation frameworks like IberoBench (Baucells et al. 2025) and Evalita-LLM (Magnini et al. 2025) have developed comprehensive multilingual benchmarks, while Language Ranker (Li et al. 2025) proposed metrics for quantifying LLM performance across diverse languages.

Automated evaluation approaches have emerged to address scalability challenges. CodeArena (Du et al. 2025) introduced collective evaluation mechanisms for code generation, while annotation costs have been reduced by combining human and synthetic feedback (Zhou, Song, and Zanette 2025). However, systematic JSON evaluation benchmarks remain absent, creating a gap in assessing structured output capabilities essential for web data mining applications.

Method

The construction of DeepJSONEval follows a systematic four-stage workflow designed to generate high-quality, domain-diverse evaluation instances with complex nested structures. Figure 2 illustrates our comprehensive pipeline, which transforms raw web content into a rigorous benchmark through automated processing and careful validation.

Web Text Collection and Multi-Document Aggregation

To address the information dispersion and redundancy inherent in web corpora across heterogeneous sources, we implement a systematic multi-text aggregation and rewriting strategy that synthesizes complementary content while eliminating cross-source redundancy. We employ LLMs to execute multi-text aggregation and rewriting by processing raw web text using the following prompt template:

Role
You are an expert editor specializing in multi-document synthesis. Read multiple input texts and produce one concise, information-dense summary that merges overlapping content, resolves contradictions when possible, and preserves key facts.

Guide for Inputs
DOCS: [A list/array of documents (plain text). Each item may be a paragraph, article, or note.]

Rules
1. Aggregation: Merge redundant points; group related ideas; eliminate repetition.
2. Compression: Prefer shorter phrasing and high signal-to-noise ratio; avoid filler, hedging, and rhetorical questions.
3. Faithfulness: Do not invent facts. Only use information present in the inputs. If sources disagree, briefly note the disagreement.
4. Clarity & Flow: Use precise vocabulary, active voice, and logical order (problem → evidence → implications or theme → key points → takeaway).
5. Style: Neutral, objective, and professional. Avoid bullet lists unless the content clearly benefits from this format.
6. Language: Use the same language as the source DOCS for output.
7. Length: Generate at least 1500 words.

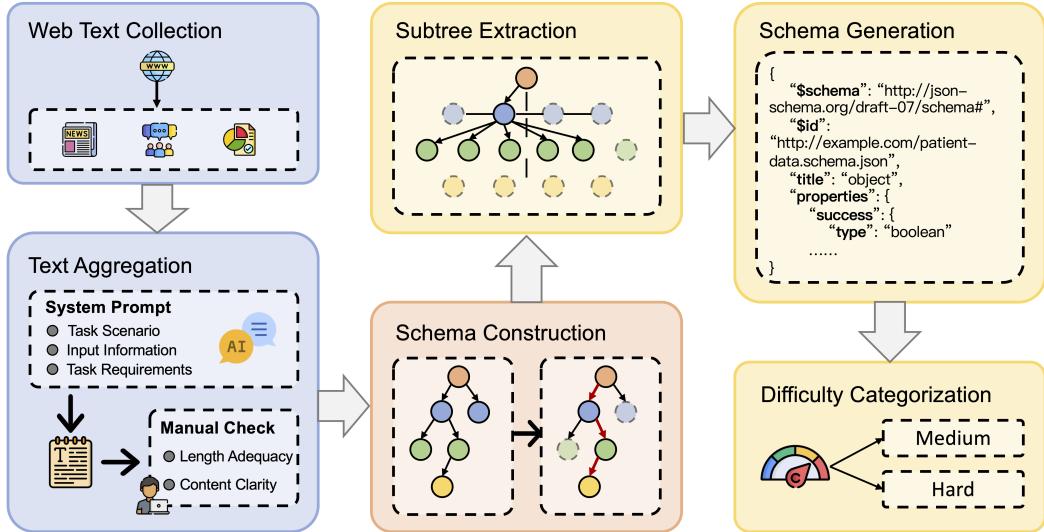**DOCS**
[INPUT DOCUMENTS]

Figure 2: The workflow of benchmark construction.

Through this prompt template, we instruct the LLM to integrate complementary aspects (methods, datasets, effect sizes) that rarely co-occur in single sources while removing inter-source redundancy and prioritizing data-bearing propositions. This approach enables the synthesis of coherent, comprehensive summaries that preserve factual accuracy while achieving significant compression ratios.

After the text-schema pair constructed, we apply a lightweight human-in-the-loop protocol (seed Apendix) to (i) verify semantic faithfulness of LLM-constructed items, (ii) reduce teacher bias/leakage risk, and (iii) release a reliable *Gold* set.

Schema Tree Construction

Following text aggregation, we systematically identify and extract key terms and concepts from the collected corpus. These elements are organized into hierarchical tree structures where nodes represent distinct properties and edges capture inheritance relationships between parent and child properties (e.g., *author → author.name → author.name.first*).

The schema tree construction process leverages domain expertise and linguistic analysis to ensure that the resulting hierarchical structures accurately reflect the semantic relationships inherent in the source content. Each tree serves as a comprehensive representation of the information space within a specific domain, providing the foundational structure for subsequent subtree extraction and schema generation.

Real-time Path-Value Updating Beam Search for Constrained Schema Subtree Construction

Algorithm Overview   Given a rooted property tree $\mathcal{T} = (V, E)$, where V represents properties and E contains directed edges from parent properties to their children (e.g.,

author → author.name), we aim to extract a set of subtrees $\mathcal{S} = \{S_1, S_2, \ldots\}$ by iteratively expanding paths from the current subtree frontier while updating path values in real time. Each expansion must satisfy constraints on schema depth and the number of properties.

Algorithm Inputs:

- Tree $\mathcal{T} = (V, E)$ with $|V| = n$ nodes.
- Depth bounds: minimum target schema depth $d_{\min}$ and maximum target schema depth $d_{\max}$.
- Size bounds: minimum number of properties $n_{\min}$ and maximum number of properties $n_{\max}$.
- Beam width `top_k` $\in \mathbb{N}^+$: the maximum number of candidate paths to retain per expansion round.

Algorithm Outputs: A collection of high-value subtrees $\mathcal{S}$ such that every $S \in \mathcal{S}$ satisfies $d(S) \in [d_{\min}, d_{\max}]$ and $|V_S| \in [n_{\min}, n_{\max}]$, where $d(S)$ denotes the depth of S.

Real-time Path-Value Updating Mechanism   As shown in Algorithm 1, let the current subtree be $S = (V_S, E_S)$ and define the frontier $\mathcal{F}(S) = \{u \in V \setminus V_S \mid \exists v \in V_S, \ (v, u) \in E\}$. A *candidate path* is $p = (u_0, \ldots, u_\ell)$, where $u_0 \in \mathcal{F}(S)$, each $(u_{i-1}, u_i) \in E$, and all $u_i \notin V_S$. We denote by $S \oplus p$ the subtree after augmenting S with all nodes and edges of p.

Given an association score function $\text{Assoc} : V \times V \to [0, 1]$ (see Appendix), the correlation of a node u to the current subtree is defined as:

$$\text{Corr}(u \mid S) = \max_{v \in V_S} \text{Assoc}(u, v). \quad (1)$$

For a new node $u \notin V_S$, we define its marginal contribution as:

$$\Delta(u \mid S) = \alpha \cdot \text{Corr}(u \mid S), \quad \alpha > 0. \quad (2)$$

Let $d(S)$ denote the current depth and $|V_S|$ the current size. After augmenting with path p, we incorporate *soft win-*

Table 2: The leaderboard of DeepJSONEval with leading LLMs

Model	Overall			Medium			Hard		
	Syntax	Key	Strict	Syntax	Key	Strict	Syntax	Key	Strict
Claude Sonnet 4	**99.05**	**90.73**	57.90	**100.00**	94.52	69.51	**98.61**	**89.01**	52.63
Magistral Medium 2506	98.10	90.36	**59.81**	**100.00**	**95.50**	**71.34**	97.22	88.04	**54.57**
DeepSeek R1 0528	97.90	89.57	57.33	**100.00**	94.56	68.29	96.95	87.30	52.35
Gemini 2.5 Pro	97.52	89.00	56.19	**100.00**	94.78	70.73	96.40	86.37	49.58
Qwen3 235B A22B	97.14	88.33	56.19	**100.00**	94.95	67.07	95.84	85.33	51.25
DeepSeek R1 Distill Llama 70B	95.44	88.15	58.29	99.39	93.80	67.68	93.63	85.58	54.02
Magistral Small 2506	93.71	85.75	53.52	99.39	93.46	66.46	91.14	82.25	47.65
Qwen3 30B A3B	93.71	84.66	52.76	**100.00**	93.30	66.46	90.86	80.74	46.54
Llama 4 Maverick	92.19	84.17	54.48	95.73	89.69	64.63	90.58	81.67	49.86
Qwen3 14B	91.81	83.39	51.24	96.95	91.04	62.20	89.47	79.92	46.26
Hunyuan A13B	93.33	83.11	48.38	98.17	89.96	54.88	91.14	79.99	45.43
Qwen3 32B	87.81	80.18	50.67	96.95	91.87	68.29	83.66	74.87	42.66

dow rewards that peak within the target intervals:

$$R_{\text{depth}}(S \oplus p) = 1 - \frac{\left| \text{clip}\big(d(S \oplus p), d_{\min}, d_{\max}\big) - d(S \oplus p) \right|}{d_{\max} - d_{\min} + \varepsilon},$$
(3)

$$R_{\text{size}}(S \oplus p) = 1 - \frac{\left| \text{clip}\big(|V_{S \oplus p}|, n_{\min}, n_{\max}\big) - |V_{S \oplus p}| \right|}{n_{\max} - n_{\min} + \varepsilon},$$
(4)

where $\text{clip}(x, a, b) = \min\{\max\{x, a\}, b\}$ and $\varepsilon > 0$ prevents division by zero.

To ensure structural validity, we penalize infeasible configurations:

$$\text{Penalty}(p \mid S) = \mathbb{I}[u_0 \notin \mathcal{F}(S)] + \sum_{i=1}^{\ell} \mathbb{I}\big[(u_{i-1}, u_i) \notin E\big].$$
(5)

Both rewards achieve a maximum value of 1 within their respective target intervals and decrease linearly outside these bounds.

Let $\gamma \in (0, 1]$ be a discount factor for deeper steps along the path. For $p = (u_0, \ldots, u_\ell)$, we define the path-value function as:

$$
\begin{aligned}
\text{Val}(p \mid S) = &\sum_{i=0}^{\ell} \gamma^i \, \Delta(u_i \mid S \cup \{u_0, \ldots, u_{i-1}\}) \\
&+ \lambda_d \, R_{\text{depth}}(S \oplus p) + \lambda_n \, R_{\text{size}}(S \oplus p) \\
&- \eta \, \text{Penalty}(p \mid S)
\end{aligned}
$$
(6)

with non-negative weights $\lambda_d, \lambda_n, \eta \geq 0$.

To reject infeasible augmentations, we define the feasibility function:

$$\text{Feas}(S \oplus p) = \begin{cases} 0, & \text{if } d(S \oplus p) > d_{\max} \text{ or } |V_{S \oplus p}| > n_{\max}, \\ 1, & \text{otherwise.} \end{cases}$$
(7)

When $\text{Feas}(S \oplus p) = 0$, we set $\text{Val}(p \mid S) = -\infty$ and discard path p.
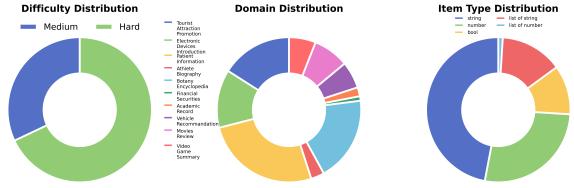

Figure 3: The distribution overview of DeepJSONEval across difficulty levels, domains, and categories.

In practice, we expand subtrees by iteratively generating candidate paths, updating their values using Equation (6), and selecting up to `top_k` best paths to augment the current subtree.

Schema Generation and Benchmark Ground Truth Construction

We systematically convert extracted subtrees into formal JSON schemas with detailed specifications and implement difficulty categorization based on nesting depth: *Medium* (3-4 levels) and *Hard* (5-7 levels). Domain experts annotate ground truth data through systematic mapping and validation, with human-in-the-loop quality control (see Appendix) ensuring semantic faithfulness and establishing a reliable *Gold* standard.

Data Statistics

Using this methodology, we construct DeepJSONEval comprising 2100 data instances across ten web domains: Tourist Attraction Promotion, Electronic Devices Introduction, Patient Information, Athlete Biography, Botany Encyclopedia, Financial Securities, Academic Record, Vehicle Recommendation, Movie Review, and Video Game Summary. Instances are categorized by structural complexity: *Medium* difficulty (3-4 nesting levels) and *Hard* difficulty (5-7 nesting levels), with comprehensive distributions shown in Figure 3.

Algorithm 1 Real-time Path-Value Updating Exploration for Subtree Extraction

---

Require: Tree $\mathcal{T} = (V, E)$; $d_{\min}, d_{\max}$; $n_{\min}, n_{\max}$; `top_k`
Ensure: A set of subtrees $\mathcal{S}$
1: $\mathcal{S} \leftarrow \varnothing$
2: **for** root $r \in V$ **do**
3:     $S \leftarrow (\{r\}, \varnothing)$
4:     **while** $d(S) < d_{\max}$ **and** $|V_S| < n_{\max}$ **do**
5:         Generate candidate path set $\mathcal{P}(S)$ with simple paths starting at $\mathcal{F}(S)$ and length $\leq L_{\max}$ where $L_{\max} = \min\{d_{\max} - d(S), n_{\max} - |V_S|\}$
6:         **for all** $p \in \mathcal{P}(S)$ **do**
7:             Compute $\mathrm{Val}(p \mid S)$ via (6)
8:         **end for**
9:         $\mathcal{B} \leftarrow \mathrm{TopK}(\mathcal{P}(S), \mathrm{Val}, \texttt{top\_k})$
10:        **for all** $p \in \mathcal{B}$ **do**
11:           **if** $\mathrm{Feas}(S \oplus p) = 1$ **then**
12:              $S \leftarrow S \oplus p$
13:           **end if**
14:        **end for**
15:        **if** $\mathcal{B} = \varnothing$ **then**
16:          **break**
17:        **end if**
18:     **end while**
19:     **if** $d(S) \geq d_{\min}$ **and** $|V_S| \geq n_{\min}$ **then**
20:         $\mathcal{S} \leftarrow \mathcal{S} \cup \{S\}$
21:     **end if**
22: **end for**
23: **return** $\mathcal{S}$

The prompt length distribution demonstrates a reasonable progression in complexity, with Hard samples exhibiting longer average lengths and greater variability, which appropriately reflects the increased structural complexity associated with deeper JSON nesting levels (see Figure 4). The concentrated distribution around 2000-4000 tokens provides an optimal range for evaluating JSON parsing capabilities without introducing excessive computational overhead, while the substantial sample sizes (164 Medium, 361 Hard) ensure robust statistical evaluation across difficulty tiers.

Experiments

Experimental Setup

The benchmark generation experiments were conducted using two distinct computational environments optimized for different generation tasks. The text aggregation task utilized the `DeepSeek-R1` model running on `Ascend 910B2` processors, configured with an inference temperature of 1 to promote diverse and creative text rewriting. This dual-environment setup was designed to leverage the specific computational strengths of each hardware platform while optimizing model performance for distinct generation requirements. The evaluation metrics are detailed in Appendix.
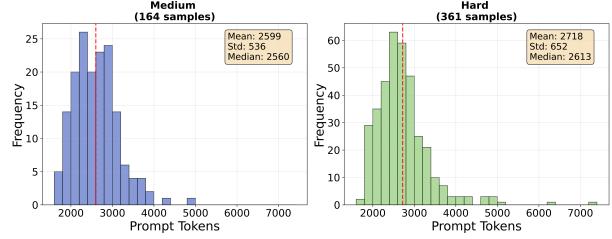

Figure 4: The prompt length statistics for Medium and Hard samples in DeepJSONEval.

Evaluation on Leading LLMs

Overall. Table 2 presents the performance of 12 representative LLMs with leading capabilities, selected from the OpenCompass LLM Leaderboard (OpenCompass 2025), on the DeepJSONEval benchmark. The models are ranked in descending order by their overall detailed scores. The detailed score measures the comprehensive extraction capabilities of models, with higher scores indicating stronger extraction performance. However, in specific agentic systems where complete accuracy is required, the strict score provides a more accurate reflection of model capabilities.

Response Length Analysis In Figure 5, analysis of response length versus evaluation scores across 12 LLMs reveals minimal correlations (r ranging from -0.335 to -0.040), with most models showing $R^2 < 0.05$, indicating that response length accounts for less than 5% of score variation. This demonstrates that DeepJSONEval effectively evaluates structural accuracy and semantic correctness rather than verbosity, validating our multi-dimensional evaluation framework's ability to assess genuine JSON generation capabilities independent of output length.

Performance versus difficulties Performance analysis across medium (3-4 levels) and hard (5-7 levels) difficulty tiers reveals systematic degradation in all LLMs, as shown in Figure 6, with strict evaluation scores exhibiting the most substantial declines (17.22%-37.53%) compared to format scores (1.39%-13.71%). Hard-level tasks consistently challenge all models, with strict scores remaining below 60%, demonstrating that deep nesting structures effectively differentiate model capabilities. These findings validate DeepJSONEval's discriminative power and difficulty stratification, establishing a robust framework for evaluating structured output generation across varying complexity levels.

Performance in diffrent domains Cross-domain performance analysis reveals consistent model behavior across 10 domains, with medium difficulty scores ranging 0.776-0.867 and hard difficulty scores spanning 0.474-0.540, indicating minimal domain-specific variance (see Figure 7). The selective distribution of hard difficulty samples across domains reflects realistic JSON application scenarios where complex nesting requirements are domain-dependent. This balanced performance distribution validates DeepJSONEval's ecological validity and confirms that the benchmark accurately represents practical deployment challenges.
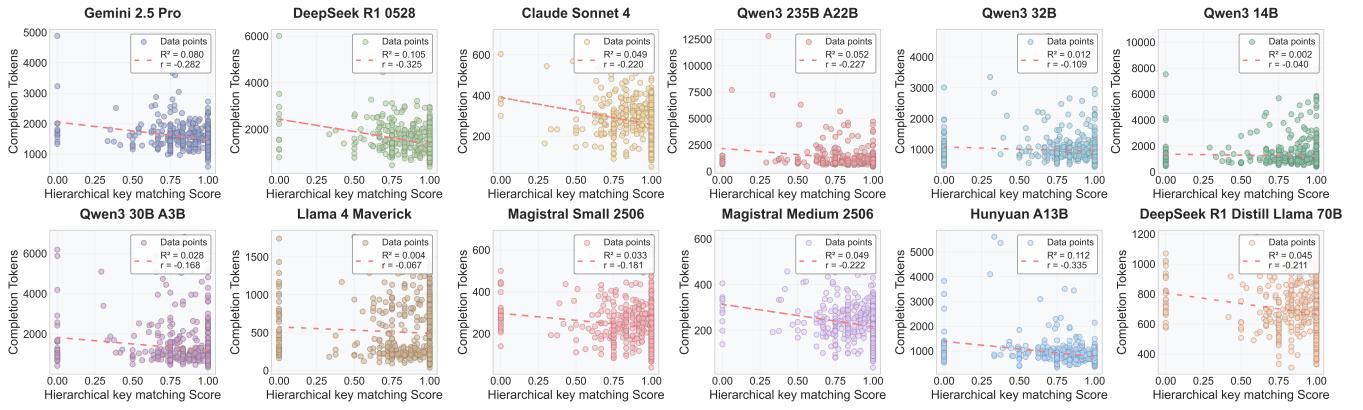
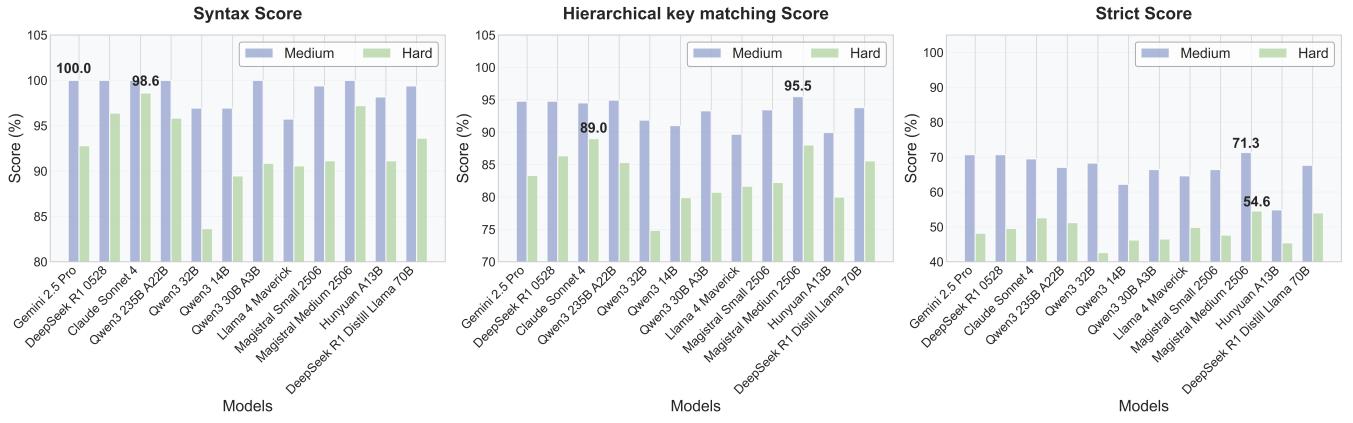Figure 5: The response length distribution across evaluation scores for multiple LLMs.

Figure 6: The performance comparison across format, detailed, and strict evaluation metrics for Medium and Hard difficulty levels.

Performance in different data types   Figure 8 demonstrates systematic performance variation across JSON element types, with LLMs achieving highest accuracy on numeric lists (0.90-1.00) and lowest on string lists (0.576-0.722). This consistent hierarchy across all 12 models reveals fundamental limitations in processing complex nested list structures compared to primitive data types. The pronounced performance degradation on hierarchical list elements indicates architectural challenges in structured output generation, highlighting the need for improved training strategies targeting nested JSON relationships.

Qualitative Analysis   see Appendix.

External Validity via a Small End-to-End Web Pipeline

We examine whether performance on DEEPJSONEVAL predicts real-world utility in a practical Web pipeline: *crawl → extract → JSON → query/analytics*. We quantify the association between benchmark scores and real world pipeline scores.

Experiment Setting
Pipeline
1. **Crawl**: domain-allowlisted seed URL.
2. **Preprocess**: boilerplate removal, language detection, schema construction.
3. **Extract**: run candidate LLMs to produce schema-conformant JSON per document.
4. **Query/Analytics**: calculate the correlation of **DeepJSONEval** score and pipeline scores.

Evaluation Criteria   See Appendix.

Models & Domains Selection   We select 3 LLMs that represent a wide range of **DeepJSONEval** scores (low→high): Qwen3 32B, DeepSeek R1 Distill Llama 70B and Claude Sonnet 4 with 3 domains: Athlete Biography, Vehicle Recommendation and Video Game Summary, each domain containing 2 data units with hard-level schemas.

Experiment Result & Analysis
The correlation between 3 model key matching scores in Table 3 and those in Table 2 is 0.987. This provides external validity: **DeepJSONEval** gains predict concrete im-
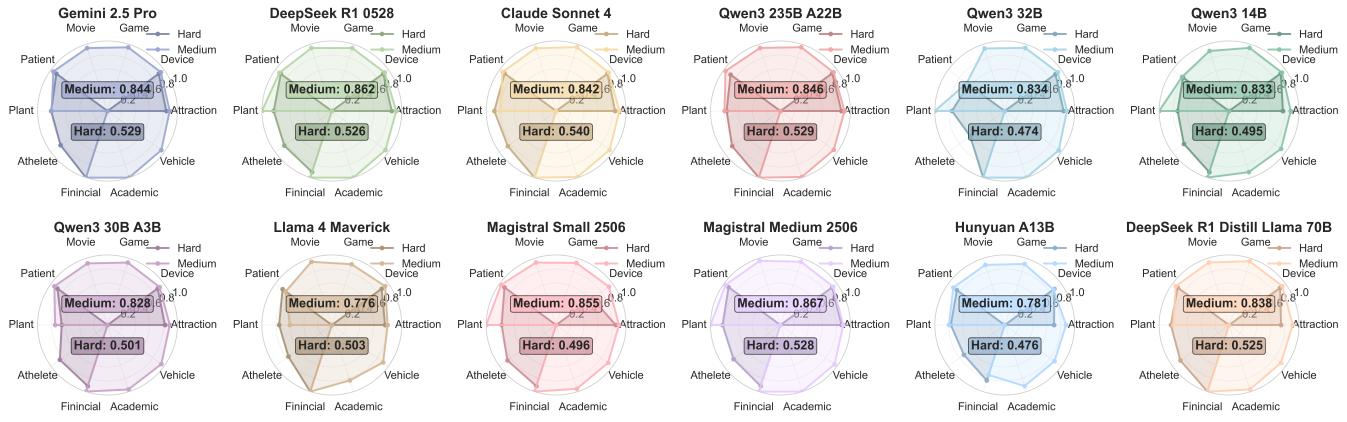
Figure 7: The Hierarchical key matching Score across domains and difficulty levels for multiple LLMs.
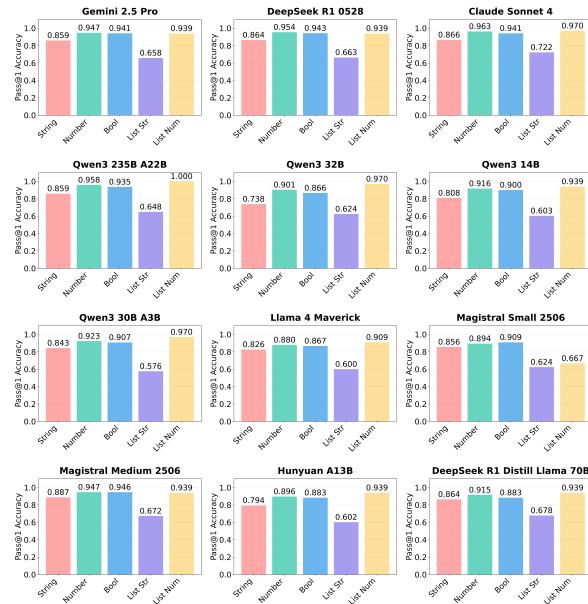

Figure 8: The performance comparison across JSON element types for multiple LLMs.

provements in end-to-end Web extraction and analytics. We also note that the models in the end-to-end pipeline achieve higher scores than those in the benchmark. This discrepancy arises because the evaluation set consists of aggregated original texts, resulting in high information density that may cause models to confuse or overlook information. In contrast, the pipeline employs raw web pages with lower information density, which facilitates more accurate information extraction by the models.

Conclusion

This work addresses the critical limitations in evaluating multi-layer nested JSON generation and information extraction capabilities of LLMs by introducing **DeepJSONEval**, a pioneering benchmark framework for deep nested JSON

Table 3: Result of End-to-End Pipeline

Model	key matching score
Claude Sonnet 4	97.36
DeepSeek R1 Distill Llama 70B	91.87
Qwen3 32B	83.26

structure evaluation. Our innovative tree-based schema generation algorithm successfully constructs complex nested structures ranging from 3 to 7 layers deep, establishing the first graded difficulty assessment framework that transitions from pseudo schema to standard schema with exceptional scalability. It enables extensible synthesis of domain-specific question-answer pairs with customizable nesting depths, allowing for the generation of standardized JSON structured extraction datasets across diverse application scenarios. The comprehensive evaluation framework achieves multi-dimensional assessment through format matching, field correctness, and structural integrity metrics, systematically evaluating model performance across 2100 high-quality instances spanning 10 diverse domains including tourism, healthcare, entertainment, etc. Systematic difficulty grading of DeepJSONEval distinguishes medium complexity from hard scenarios, providing fine-grained differentiation suitable for current leading models and emerging development trends. Future work will focus on extending the framework to support even deeper nesting levels and incorporating dynamic schema adaptation to further enhance the applicability of benchmark to evolving LLM architectures.

Future Work

Time Complexity Optimization of Algorithm

In Section , we outline algorithmic directions that explicitly target the dominant factors in the current complexity $T_{\text{total}} = O(R \cdot L \cdot b^L)$, where b is the (effective) branching factor, L is the candidate path length cap, and R is the number of expansion rounds. The optimization process will focus on the following aspects:

- **Lazy Valuation with Prefix Reuse**: maintain prefix accumulators $\sum_{i \leq \ell} \gamma^i \Delta(u_i \mid \cdot)$ and an incremental cache for $\max_{v \in V_S} \text{sim}(u, v)$; evaluate window rewards only at interval crossing checkpoints
- **Tree-DP for Pure Trees**: bottom-up DP to precompute, for each node, top-t outward path scores for lengths $1..L$ (with discount and cached correlations)
- **Parallel Valuation and Top-k**: SIMD/GPU batching for correlations and rewards; thread-local heaps with periodic k-way merges; 64-bit hashing for deduplication

These optimization strategies target the exponential growth in candidate path evaluation by implementing prefix reuse and dynamic programming techniques, reducing the effective branching factor from b^L to approximately $b \log L$ for practical tree structures. The parallel processing approach enables efficient utilization of multi-core architectures, achieving near-linear speedup for correlation computations across independent subtree branches.

References

Agarwal, B.; Joshi, I.; and Rojkova, V. 2025. Think Inside the JSON: Reinforcement Strategy for Strict LLM Schema Adherence. arXiv:2502.14905.

Basu, K.; Abdelaziz, I.; Kate, K.; Agarwal, M.; Crouse, M.; Rizk, Y.; Bradford, K.; Munawar, A.; Kumaravel, S.; Goyal, S.; Wang, X.; Lastras, L. A.; and Kapanipathi, P. 2025. NESTFUL: A Benchmark for Evaluating LLMs on Nested Sequences of API Calls. arXiv:2409.03797.

Baucells, I.; Aula-Blasco, J.; de Dios-Flores, I.; Paniagua Suárez, S.; Perez, N.; Salles, A.; Sotelo Docio, S.; Falcão, J.; Saiz, J. J.; Sepulveda Torres, R.; Barnes, J.; Gamallo, P.; Gonzalez-Agirre, A.; Rigau, G.; and Villegas, M. 2025. IberoBench: A Benchmark for LLM Evaluation in Iberian Languages. In Rambow, O.; Wanner, L.; Apidianaki, M.; Al-Khalifa, H.; Eugenio, B. D.; and Schockaert, S., eds., *Proceedings of the 31st International Conference on Computational Linguistics*, 10491–10519. Abu Dhabi, UAE: Association for Computational Linguistics.

Chinta, R. C. 2025. JSON-BASED PATIENT DATA ARCHITECTURE: A NOVEL APPROACH TO HEALTHCARE INFORMATION STORAGE IN SALESFORCE CRM. *Technology (IJRCAIT)*, 8(1).

Du, M.; Luu, A. T.; Ji, B.; Wu, X.; Huang, D.; Zhuo, T. Y.; Liu, Q.; and Ng, S.-K. 2025. CodeArena: A Collective Evaluation Platform for LLM Code Generation. arXiv:2503.01295.

Geng, S.; Cooper, H.; Moskal, M.; Jenkins, S.; Berman, J.; Ranchin, N.; West, R.; Horvitz, E.; and Nori, H. 2025. JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models. arXiv:2501.10868.

Gu, Z.; Ye, H.; Chen, X.; Zhou, Z.; Feng, H.; and Xiao, Y. 2024. StrucText-Eval: Evaluating Large Language Model's Reasoning Ability in Structure-Rich Text. arXiv:2406.10621.

Gui, H.; Yuan, L.; Ye, H.; Zhang, N.; Sun, M.; Liang, L.; and Chen, H. 2024. IEPile: Unearthing Large-Scale Schema-Based Information Extraction Corpus. arXiv:2402.14710.

Guo, G.; Zhang, K.; Hoo, B.; Cai, Y.; Lu, X.; Peng, N.; and Wang, Y. 2025. Structured Outputs Enable General-Purpose LLMs to be Medical Experts. arXiv:2503.03194.

Kocyigit, M. Y.; Briakou, E.; Deutsch, D.; Luo, J.; Cherry, C.; and Freitag, M. 2025. Overestimation in LLM Evaluation: A Controlled Large-Scale Study on Data Contamination's Impact on Machine Translation. arXiv:2501.18771.

Li, Z.; Shi, Y.; Liu, Z.; Yang, F.; Payani, A.; Liu, N.; and Du, M. 2025. Language Ranker: A Metric for Quantifying LLM Performance Across High and Low-Resource Languages. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27): 28186–28194.

Liu, C.; Jin, R.; Yao, Z.; Li, T.; Cheng, L.; Steedman, M.; and Xiong, D. 2025. Empirical Study on Data Attributes Insufficiency of Evaluation Benchmarks for LLMs. In Rambow, O.; Wanner, L.; Apidianaki, M.; Al-Khalifa, H.; Eugenio, B. D.; and Schockaert, S., eds., *Proceedings of the 31st International Conference on Computational Linguistics*, 6024–6038. Abu Dhabi, UAE: Association for Computational Linguistics.

Liu, Y.; Li, D.; Wang, K.; Xiong, Z.; Shi, F.; Wang, J.; Li, B.; and Hang, B. 2024. Are LLMs good at structured outputs? A benchmark for evaluating structured output capabilities in LLMs. *Information Processing & Management*, 61(5): 103809.

Magnini, B.; Zanoli, R.; Resta, M.; Cimmino, M.; Albano, P.; Madeddu, M.; and Patti, V. 2025. Evalita-LLM: Benchmarking Large Language Models on Italian. arXiv:2502.02289.

Mitchener, L.; Laurent, J. M.; Tenmann, B.; Narayanan, S.; Wellawatte, G. P.; White, A.; Sani, L.; and Rodriques, S. G. 2025. BixBench: a Comprehensive Benchmark for LLM-based Agents in Computational Biology. arXiv:2503.00096.

NousResearch. 2024. json-mode-eval.

OpenCompass. 2025. OpenCompass LLM Leaderboard - a Hugging Face Space by opencompass.

Rastogi, A.; Zang, X.; Sunkara, S.; Gupta, R.; and Khaitan, P. 2020. Towards Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset. arXiv:1909.05855.

Shorten, C.; Pierse, C.; Smith, T. B.; Cardenas, E.; Sharma, A.; Trengrove, J.; and van Luijt, B. 2024. StructuredRAG: JSON Response Formatting with Large Language Models. arXiv:2408.11061.

Syafiq, M.; Azri, S.; and Ujang, U. 2025. CityJSON Management Using Multi-Model Graph Database to Support 3D Urban Data Management. In *13th International Conference on Geographic Information Science (GIScience 2025)*, 2–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Wang, J.; Zhou, J.; Wen, M.; Mo, X.; Zhang, H.; Lin, Q.; Jin, C.; Wang, X.; Zhang, W.; Peng, Q.; and Wang, J. 2024. HammerBench: Fine-Grained Function-Calling Evaluation in Real Mobile Device Scenarios. arXiv:2412.16516.

Wei, T.; Wen, W.; Qiao, R.; Sun, X.; and Ma, J. 2025. RocketEval: Efficient Automated LLM Evaluation via Grading Checklist. arXiv:2503.05142.

White, C.; Dooley, S.; Roberts, M.; Pal, A.; Feuer, B.; Jain, S.; Shwartz-Ziv, R.; Jain, N.; Saifullah, K.; Dey, S.; Shubh-Agrawal; Sandha, S. S.; Naidu, S.; Hegde, C.; LeCun, Y.; Goldstein, T.; Neiswanger, W.; and Goldblum, M. 2024. LiveBench: A Challenging, Contamination-Limited LLM Benchmark. arXiv:2406.19314.

Xia, C.; Xing, C.; Du, J.; Yang, X.; Feng, Y.; Xu, R.; Yin, W.; and Xiong, C. 2024. FOFO: A Benchmark to Evaluate LLMs' Format-Following Capability. arXiv:2402.18667.

Xu, A.; Ding, R.; and Wang, L. 2025. ChatPD: An LLM-driven Paper-Dataset Networking System. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 5106–5116.

Yang, J.; Jiang, D.; He, L.; Siu, S.; Zhang, Y.; Liao, D.; Li, Z.; Zeng, H.; Jia, Y.; Wang, H.; et al. 2025. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. *arXiv preprint arXiv:2505.20139*.

Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; and Hou, L. 2023. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911.

Zhou, Z.; Song, Y.; and Zanette, A. 2025. Accelerating Unbiased LLM Evaluation via Synthetic Feedback. arXiv:2502.10563.
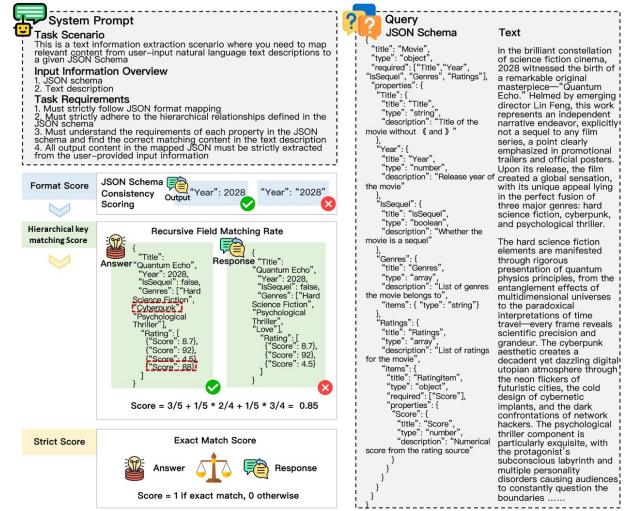
Evaluation Criteria

Figure 9: The illustrative example of the DeepJSONEval multi-dimensional assessment criteria framework.

Upon completion of the benchmark dataset construction, we establish a comprehensive evaluation framework, which is displayed in Figure 9, comprising four distinct criteria to systematically assess LLM performance on our synthesized benchmark. These evaluation metrics are designed to provide a multi-dimensional analysis of model capabilities across different aspects of the structured data generation and comprehension tasks.

- **Criterion 1: Syntax Score**
  This criterion evaluates LLMs' ability to generate syntactically valid JSON outputs through sequential validation of parsing and schema conformance, with scoring defined as:

$$S_{syntax} = \mathbb{1}[\text{valid}(\text{output}) \wedge \text{match}(\text{output}, \text{schema})] \tag{8}$$

- **Criterion 2: Hierarchical key matching Score**
  Based on the *key_mathcing_score* in JSON-Based Reward of ThinkJSON (Agarwal, Joshi, and Rojkova 2025), considering the multi-layer nested structrue, this criterion performs comprehensive property-wise comparison with uniform weighting across all hierarchical levels based on Jaccard similarity, computing weighted differences through systematic structural traversal:

$$S_{key} = \frac{|p_{out} \cap p_{truth}|}{|p_{out} \cup p_{truth}|} \tag{9}$$

- **Criterion 3: Strict Score**
  This criterion implements binary exact-match evaluation through strict equality verification between LLM output and ground truth JSON:

$$S_{strict} = \mathbb{1}[\text{output} \equiv \text{truth}] \tag{10}$$

Node Association Score

Inputs: The discription of node u and node v, desc_u & desc_v

Outputs: the *association* score $\mathrm{Assoc}(u,v) \in [0,1]$ between 2 nodes u and v

Let $d(u,v) \in \{1, 2, \dots\}$ be the shortest-path distance between u and v in the property node tree, the penalty of distance is

$$\kappa_{\mathrm{dist}}(u,v) = \exp\big(-\alpha\, d(u,v)\big) \quad (11)$$

The association of node description is:

$$s_{\mathrm{desc}}(u,v) = \frac{\langle \phi(\mathrm{desc}_u),\, \phi(\mathrm{desc}_v)\rangle}{\|\phi(\mathrm{desc}_u)\|\, \|\phi(\mathrm{desc}_v)\|} \quad (12)$$

where $\phi(\cdot)$ is normalized text embedding.

The Association function is finally:

$$\mathrm{Assoc}(u,v) = \kappa_{\mathrm{dist}}(u,v) \cdot \Big(\lambda_{\mathrm{name}}\, s_{\mathrm{name}}(u,v) + \lambda_{\mathrm{desc}}\, s_{\mathrm{desc}}(u,v)\Big) \quad (13)$$

with $\lambda_{\mathrm{name}}, \lambda_{\mathrm{desc}} \geq 0$, $\lambda_{\mathrm{name}} + \lambda_{\mathrm{desc}} = 1$.

Qualitative Analysis Examples

This section presents two representative examples across different difficulty levels. Comparative analysis reveals that increased nesting depth in hard-level tasks correlates with higher error prevalence and diversity compared to medium-level tasks. The hard-level case exhibits multiple error categories: type mismatches (e.g., returning string values such as "4 GB" instead of required numerical data), semantic hallucinations (e.g., property repetition or introduction of non-existent entities like "personal wellness hub"), and extraction inaccuracies (e.g., incomplete proper noun extraction). Conversely, the medium-level case demonstrates only isolated type errors. These findings substantiate the hypothesis that increased nesting complexity introduces systematic challenges including field omissions, hierarchical misalignments, type inconsistencies, cross-field contradictions, and semantic fabrications.

Difficulty Level: Hard

Text to be extracted

> ...
> (Omission of some content.)
> In summary, the HarmonyFit Smart Band isn't just another wearable; it's a meticulously crafted fusion of advanced hardware, intelligent software, and thoughtful design. From its octa-core CPU and Mali-G52 MC1 GPU delivering raw power to its HarmonyOS 3.0 platform fostering an interconnected digital life, every element is designed to elevate your experience. With 4 GB of RAM for seamless operation and versatile storage choices of 32 GB, 64 GB, or 128 GB, it adapts to your evolving needs effortlessly. Preinstalled apps like the Heart Rate Monitor, Sleep Tracker, Workout Coach, and Weather App turn it into a personal wellness hub, all while its 46.7-gram weight ensures it stays lightweight and comfortable. The precision of its three

> sensors—calibrated at 1.05, 0.92, and 1.10—locks in data integrity, making it an indispensable tool for anyone seeking to harness technology for a healthier, more informed lifestyle.

JSON Schema Given

```
{
 (Omission of some content.)
 "Memory": {
  "title": "Memory",
  "type": "number",
  "description": "Amount of RAM in GB"
 },
 (Omission of some content.)
 "Software": {
  (Omission of some content.)
  "properties": {
   "OperatingSystem": {
    "title": "OperatingSystem",
    "type": "string",
    "description": "Name of the operating system"
   },
   (Omission of some content.)
   "Dimensions": {
    "title": "Dimensions",
    "type": "object",
    "required": [
    "Weight",
    "Sensors"
    ],
    "properties": {
     "Weight": {
      "title": "Weight",
      "type": "number",
      "description": "Weight of the device in grams"
     },
     "Sensors": {
      "title": "Sensors",
      "type": "array",
      "items": {
       "type": "object",
       "required": [
       "SensorType"
       ],
       (Omission of some content.)
      }
```

The ground truth JSON

```
{
 (Omission of some content.)
 "Memory": 4,
 (Omission of some content.)
 "Software": {
  "OperatingSystem": "HarmonyOS 3.0",
  "PreinstalledApps": [
  "Heart Rate Monitor",
  "Sleep Tracker",
  "Workout Coach",
  "Weather App"
  ]
 },
 "Dimensions": {
  "Weight": 46.7,
  "Sensors": [
   {
    "SensorType": {
     "SensorSubtype": {
      "CalibrationData": 1.05
     }
    }
   },
   {
    "SensorType": {
     "SensorSubtype": {
      "CalibrationData": 0.92
     }
    }
   },
   {
    "SensorType": {
     "SensorSubtype": {
      "CalibrationData": 1.1
     }
    }
   }
```

```
            }
          ]
        }
      }
    }
```

Model Output

```
{
  (Omission of some content.)
  "Memory": "4 GB",
  (Omission of some content.)
  "Software": {
   "OperatingSystem": "HarmonyOS 3",
   "PreinstalledApps": ["Heart Rate Monitor", "Sleep Tracker",
   "Workout Coach", "Weather App", "personal wellness hub"]
  },
  "Dimensions": {
   "Weight": 46.7,
   "Sensors": [
    {
     "SensorType": {
      "SensorSubtype": {
       "CalibrationData": 1.05
      }
     }
    },
    {
     "SensorType": {
      "SensorSubtype": {
       "CalibrationData": 0.92
      }
     }
    }
   ]
  },
  "Software": {
   "OperatingSystem": "HarmonyOS 3",
   "PreinstalledApps": ["Heart Rate Monitor", "Sleep Tracker",
   "Workout Coach", "Weather App", "personal wellness hub"]
  },
  "Dimensions": {
   "Weight": 46.7,
   "Sensors": [
    {
     "SensorType": {
      "SensorSubtype": {
       "CalibrationData": 1.05
      }
     }
    },
    {
     "SensorType": {
      "SensorSubtype": {
       "CalibrationData": 0.92
      }
     }
    }
   ]
  }
}
```

For this example in hard difficulty, compared with the ground truth, the model has sevral errors:

- **Ignore the requirement of schema**: for property *Memory*, the type is *number* and description is **numebr in GB**, model ignore the requirement and return **4 GB**;

- **String extraced error**: for property *OperatingSystem*, **HarmonyOS 3.0** is a proper noun for operating system, while model skip **.0**;

- **Property Repeat**: model repeat the property **Software** and **Dimensions** for once;

- **Identification in error**: model add **personal wellness hub** in PreinstalledApps, which is not an application.

Difficulty level: Medium

Text to be extracted

Prepare to be captivated by **Eclipse of the Stars**, an original cinematic spectacle that premiered in **2023**. This film stands as a wholly independent creation—emphatically **not a sequel**—that masterfully fuses the high-stakes tension of a **Thriller** with the visionary grandeur of **Sci-Fi**. Its narrative ingenuity unfolds against a canvas of cosmic wonder and psychological intensity, carving a unique niche within contemporary cinema. The acclaim surrounding this feature is far from superficial, as evidenced by its extraordinary critical reception across major review platforms. On the globally recognized **IMDb**, it boasts an impressive **8.2/10**, reflecting widespread audience admiration for its intricate storytelling and technical execution. Over at **Rotten Tomatoes**, the film achieves a remarkable **92% (Omission of some content.)

JSON Schema Given

```
{
  (Omission of some content.)
  "Ratings": {
   "title": "Ratings",
   "type": "array",
   "items": {
    "title": "RatingItem",
    "type": "object",
    "required": [
    "Source",
    "Score"
    ],
    "properties": {
     "Source": {
      "title": "Source",
      "type": "string",
      "description": "Name of the rating source",
      "enum": [
      "IMDb",
      "Rotten Tomatoes",
      "Metacritic"
      ]
     },
     "Score": {
      "title": "Score",
      "type": "number",
      "description": "Numerical score from the rating source"
     }
    }
   }
  },
  (Omission of some content.)
}
```

Ground Truth

```
{
  (Omission of some content.)
  "Ratings": [
   {
    "Source": "IMDb",
    "Score": 8.2
   },
   {
    "Source": "Rotten Tomatoes",
    "Score": 92
   },
   {
    "Source": "Metacritic",
    "Score": 88
   }
  ],
  (Omission of some content.)
}
```

Model Output

```
{
  (Omission of some content.)
  "Ratings": [
   {
```

```
     "Source": "IMDb",
     "Score": 8.2
   },
   {
     "Source": "Rotten Tomatoes",
     "Score": "92%"
   },
   {
     "Source": "Metacritic",
     "Score": 88
   }
 ],
 (Omission of some content.)
}
```

For this example in midium difficulty, compared with the ground truth, the model has only **Ignore the requirement of schema** error, for property *score*, the type is *number* and descrition is **Numerical score from the rating source**, model ignore the requirement and return **92%** for Rotten Tomatoes score.

Human Quality Control and Correction

Dual Review and Adjudication

Two trained annotators independently review each item (blind to model identity). Disagreements are resolved by a brief adjudication pass. **Checks (yes/no unless noted).**

- **Schema conformance**: keys, nesting, and types match the provided schema.

- **Content faithfulness**: each filled field is supported by the source text (no hallucination).

- **Cross-field consistency**: simple logical or unit constraints hold (for example, date ranges, ID links).

- **Ambiguity rating (3-point)**: *Low* (clear), *Medium* (minor vagueness), *High* (unclear or conflicting). Items rated *High* are marked `Fix` or `Reject`.

- **Difficulty tag**: *Medium* if nesting depth 3–4 without cross-field constraints; *Hard* if depth 5–7 and/or includes cross-field constraints or multi-entity linking. Annotators confirm or correct the tag.

Items are labeled as `Pass`/`Fix`/`Reject`; fixes are minimal (edit JSON value or move key), and changes are logged.

Lightweight Semantic Guards

We run:

- **Schema validator**: JSON Schema parsing and type checks.

- **Constraint checker**: a small set of domain-agnostic rules (ranges, units, equality);

- **Text grounding probe**: simple evidence search; if no supporting span is found, the item is marked for review;

- **Length compliance**: source text (or evidence bundle) must have at least 1500 words; otherwise flag for `Fix`/`Reject`.

Leakage and Bias Quick Audit

We perform a one-pass similarity scan (n-gram overlap and MinHash) of sources against public corpora and our own prompts. High-similarity cases are flagged for human spot-check; flagged items are lightly rewritten or removed.

Acceptance and Release

Items that (i) pass dual review (after adjudication), (ii) pass automatic checks (including length), and (iii) have *Low/Medium* ambiguity are released as *Gold*. We report overall pass/fix/reject rates, ambiguity distribution, corrected difficulty tags, and representative failure cases. We report main results on the full set and replicate key claims on the *Gold* set.

Notes This protocol keeps costs low while adding practical QC dimensions (length sufficiency, ambiguity, difficulty verification) to ensure semantic fidelity and guard against leakage, addressing reviewers' concerns without making QC the focal point of the paper.

Rubric Summary

Dimension	Decision Rule
Schema conformance	Yes/No (No → Fix/Reject)
Content faithfulness	Yes/No (No → Fix/Reject)
Cross-field consistency	Yes/No (No → Fix/Reject)
Length compliance	≥1500 words (No → Fix/Reject)
Ambiguity	Low / Medium / High (High → Fix/Reject)
Difficulty tag	Medium or Hard (confirm/correct)