The Mighty ToRR: A Benchmark for Table Reasoning and Robustness in LLMs

Shir Ashury-Tahan[♠♡], Yifan Mai[♣], Rajmohan C[♠], Ariel Gera[♠], Yotam Perlitz[♠], Asaf Yehudai[♠], Elron Bandel[♠], Leshem Choshen[♠♢], Eyal Shnarch[♠], Percy Liang[♣] and Michal Shmueli-Scheuer[♠]

[♠]IBM Research, [♡]Bar-Ilan University, [♣]Stanford University, [♢]MIT

shir.ashury.tahan@ibm.com, shmueli@il.ibm.com

Abstract

Despite its real-world significance, model performance on tabular data remains underexplored, leaving uncertainty about which model to rely on and which prompt configuration to adopt. To address this gap, we create ToRR, a benchmark for Table Reasoning and Robustness, measuring model performance and robustness on table-related tasks. The benchmark includes 10 datasets that cover different types of table reasoning capabilities across varied domains. ToRR goes beyond model performance rankings, and is designed to reflect whether models can handle tabular data consistently and robustly, across a variety of common table representation formats. We present a leaderboard as well as comprehensive analyses of the results of leading models over ToRR. Our results reveal a striking pattern of brittle model behavior, where even strong models are unable to perform robustly on tabular data tasks. We further find that no single table format consistently yields superior performance. However, evaluating models across multiple formats is essential for a reliable assessment of their capabilities. Moreover, we show that the reliability boost from testing multiple prompts can be equivalent to adding more test examples. Overall, our findings show that reasoning over table tasks remains a significant challenge. The leaderboard, data and code are publicly available.

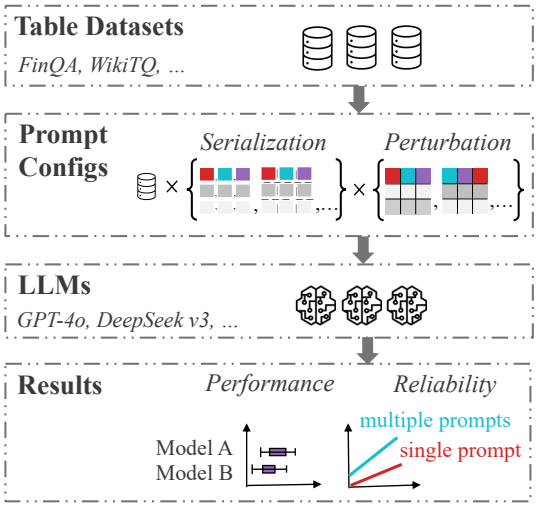


Figure 1: Overview of ToRR. We evaluate LLMs on several tabular reasoning datasets using a variety of prompt configurations. Each configuration includes a table serialization — a method for converting the table into a string format (e.g., HTML serialization) — and may include a perturbation to the table structure (e.g., shuffling row order). Our results explore *model performance* and the effects of *prompt variability*. Our analysis demonstrates that for any number of examples, testing more prompt configurations increases the evaluation reliability.

1 Introduction

Tabular data are ubiquitous across real-world use cases and tasks. Hence, the ability to understand and process tables is a crucial skill for Large Language Models (LLMs). Tabular processing capabilities can manifest in a wide range of NLP tasks, including table-to-text (Moosavi et al., 2021; Suadaa et al., 2021), table question answering (Pasupat and Liang, 2015; Wu et al., 2024) and table fact verification (Chen et al., 2020; Gu et al., 2022). In order to solve such problems, LLMs must correctly parse tabular structures, but must also apply various levels of textual and numerical reasoning over the table contents. Thus, tabular tasks are a challenging test of LLMs' capabilities and practical utility.

Although prior studies have investigated LLM performance on tabular tasks (Ruan et al., 2024; Fang et al., 2024; Lu et al., 2025; Chen, 2022), their scope remains limited, typically constrained to specific tasks, narrow domains, or a small subset of models. Consequently, these evaluations fail to provide a comprehensive understanding of LLM capabilities in structured data reasoning. Furthermore, they fail to account for the complexities of real-world applications, as they do not encompass the full range of skills required for working with

Proceedings of the First Workshop on Structured Understanding, Retrieval, and Generation in the LLM Era (SURGeLLM 2026), pages 16–51

Dataset		Task	Domain	Metric	Knowledge Extraction	Textual Reasoning	Numerical Reasoning
FinQA	(Chen et al., 2022)	Table QA	Finance	Program Accuracy	✓	✓	✓
TableBench DA	(Wu et al., 2024)	Data Analysis	Diverse	Rouge	✓	✓	✓
TableBench NR	(Wu et al., 2024)	Table QA	Diverse	Rouge	✓	✓	✓
TableBench FC	(Wu et al., 2024)	Table QA	Diverse	Rouge	✓	✓	~
WikiTQ	(Pasupat and Liang, 2015)	Table QA	Wikipedia	F1 Strings	✓	✓	~
TabFact	(Chen et al., 2020)	Fact Verification	Wikipedia	Accuracy	✓	✓	~
QTSumm	(Zhao et al., 2023a)	Table-to-Text QA	Wikipedia	Rouge	✓	~	~
SciGen	(Moosavi et al., 2021)	Table-to-Text	Science	Rouge	~	×	~
NumericNLG	(Suadaa et al., 2021)	Table-to-Text	Science	Rouge	~	×	~
TURL CTA	(Deng et al., 2020)	Classification	Wikipedia	Exact Match	~	×	×

Legend: Required ✓ Partially Required ~ Not Required ×

Table 1: The selected datasets for ToRR along with their properties. The 3 columns on the right reflect the required skills to solve each dataset, based on our analysis (§2). Further details in Appendix A.

tables. Importantly, they do not systematically assess the *robustness* of LLMs to variations in table formatting.

Given that real-world tables frequently appear in diverse yet semantically equivalent textual formats (Singha et al., 2023; Sui et al., 2024; Zhao et al., 2023c; Bhandari et al., 2025), it is crucial to assess LLM capabilities across formats. Such evaluation further provides insight into the models' internal representations and their ability to generalize beyond specific formatting styles.

In this work, we paint a comprehensive picture of the ability of LLMs on downstream table understanding and table reasoning tasks. To this end, we design a pipeline for evaluating LLMs on tabular tasks, as illustrated in Figure 1, and collect 10 datasets belonging to 6 diverse tabular tasks, from multiple domains. Those together amount to our benchmark, ToRR, a broad coverage benchmark, testing different levels of table skills.

The focus on robustness is inherent to the design of ToRR. Our benchmark examines how models respond to differing prompt configurations – table serialization formats, as well as table perturbations. Thus, going beyond bottom-line model rankings and performance, we are able to conduct an in-depth analysis of LLM behavior on tabular tasks.

We evaluate the performance and robustness of 14 leading LLMs spanning 7 model families on ToRR, which highlights significant gaps in model capabilities, even for industry-leading LLMs.

Our results demonstrate that existing models struggle with tasks that require table understanding and reasoning, and exhibit brittle and inconsistent response patterns. At the same time, we find that no single table format consistently yields superior performance.

We further analyze the broader implications of our benchmark results, focusing on the aspect of accounting for model robustness. Our meta-evaluation analysis highlights the importance of incorporating a large number of prompt configurations into the evaluation process, demonstrating the benefits of the ToRR benchmark design for evaluation reliability compared to existing benchmarks.

We start by describing the ToRR benchmark design (§2), followed by the results and model analysis (§3). We then discuss the validity of the benchmark in Section 4. Finally, Section 5 analyzes robustness using ToRR data, highlighting behaviors with practical implications beyond tabular reasoning.

Our main contributions are as follows:

1. We introduce a comprehensive benchmark of downstream tabular data tasks, encompassing diverse tasks and incorporating model robustness measurements (§2).

2. We present evaluation results over 14 LLMs. Our study uncovers substantial limitations in the tabular capabilities of existing LLMs, challenging assumptions about their generalization power in structured data contexts (§3.1, §3.2).

3. We demonstrate that no single table format consistently yields better model performance – an observation with implications for both model evaluation and practical use (§3.3).

4. In a meta-evaluation analysis, we show that incorporating prompt configuration into the evaluation process *consistently* enhances model evaluation reliability and can help compensate for smaller test sets (§5).

5. We release the leaderboard, complete model inference outputs and the code.

2 ToRR Construction

We reviewed numerous existing datasets for downstream tabular data tasks, prioritizing *challenging* ones based on the required reasoning abilities. Also, we opted for datasets where textual tables can be *directly incorporated* into the prompt[1], eliminating the need for external tools (e.g., retrieval, SQL queries, agents). Table 1 presents the selected datasets and their attributes, which were analyzed manually. As can be seen, the datasets are diverse in both the target task and domain. Further details on the selected and additional datasets, as well as evaluation metrics, are provided in Appendix A.

For a better understanding of the skills needed to solve each dataset, we performed a qualitative manual analysis of the nature of the tasks. Specifically, we identify 3 *key skills* ranked from easiest to hardest:

1. **Knowledge Extraction** – Extraction of relevant information from the table, such as specific fields, relations, or entities (e.g., *"What was the only year Keene won Class AA?"*; WikiTQ).

2. **Textual Reasoning** – Deducing conclusions by combining the accompanying text with the data contained in the table (e.g., *"Considering the weighted average fair value of options, what was the change of shares vested from 2005 to 2006?"*; FinQA).

3. **Numerical Reasoning** – Performing calculations on the table, such as aggregating information from multiple cells (e.g., *"What is the total GDP of all South American countries listed in the table according to the 2011 IMF estimates?"*; TableBench).

As shown in Table 1, knowledge extraction ability is a key requirement across all datasets. The level of textual reasoning and numerical reasoning required over the tables varies across datasets and tasks. It also demonstrates that our selection of datasets in ToRR covers a range of challenge levels. This will be further supported by a correlation with our robustness analysis (see §5).

[1]We use tables that can be serialized into text and fit within the context of five demos.

2.1 Prompt Configurations

In real-world tabular tasks, the tables provided as input to an LLM can be represented in different formats; for instance, in JSON or HTML format. Although these formats encode the same content (column names, cell values etc.), LLMs may perform differently depending on the input format. Thus, these formats offer insight into how models represent tables and, specifically, on the extent to which they are able to generalize table understanding and reasoning across different formats.

To this end, we manipulate the table format across 2 dimensions. First, for each input table, we examine 7 **serializations** i.e., methods to represent the contents of the table as a string (e.g., using *JSON* or *HTML*). In addition, we explore 4 structural **perturbations** which are applied to the tables; for example, shuffling the order of rows, or transposing the rows and columns. An example of the resulting prompts is shown in Figure 2. For details on all prompt configurations, see Appendix A.5, and for an explanation of why this helps in addressing contamination is provided, refer to Appendix A.6.

2.2 Metrics

ToRR consists of datasets denoted as D, and each dataset $d \in D$ contains examples $\{(x_i, y_i)\}_{i \in d}$, where x_i represents the input, and y_i is the ground-truth response. Each input can be represented using one of $c \in C$ prompt configurations (§2.1), denoted as x_i^c. Each dataset in ToRR is associated with a specific evaluation metric (see Table 1). All metrics fall within the range $[0.0, 1.0]$, ensuring comparability of aggregated scores across datasets. We denote the score function for an example, as defined by the dataset, as S.

Model Performance The performance of a model M is its ability to solve table-related tasks, regardless of the table format. Let $M(x_i^c)$ denote the output of model M for input x_i^c. The performance score, $\mathcal{P}$, is the average across prompts and is defined as:

$$\mathcal{P}_M = \frac{1}{|D|} \sum_{d \in D} \frac{1}{|d|} \sum_{i \in d} \frac{1}{|C|} \sum_{c \in C} S(M(x_i^c), y_i)$$

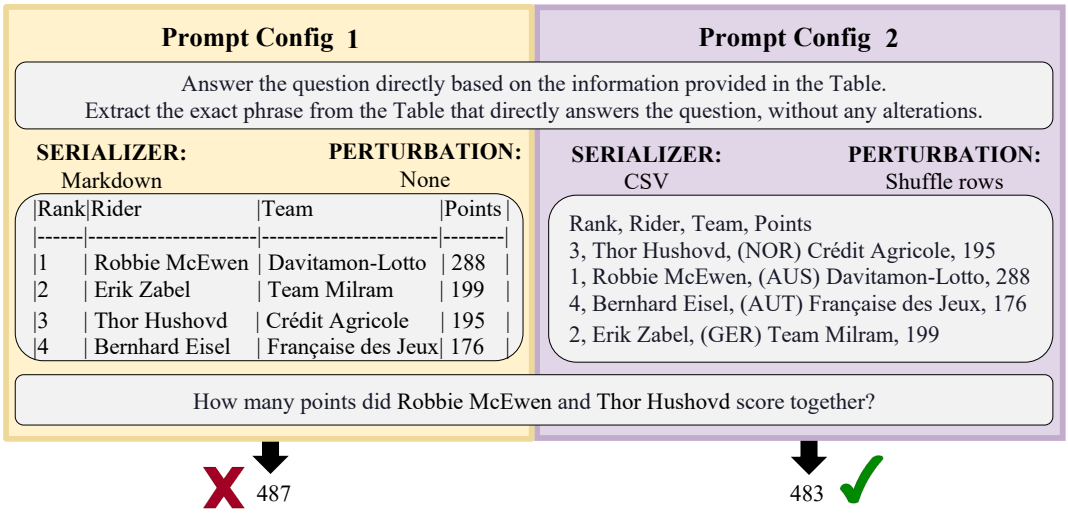


Figure 2: ToRR prompt examples with identical instructions but different table formats: markdown (left) and shuffled CSV (right) (§2.1). While all prompts convey the same information and require the exact same answer, even state-of-the-art models struggle with solving them consistently (§3).

Model Robustness A robust model is expected to perform similarly on different prompt configurations of the same example, i.e., to have a low variance over the example performance scores. Thus, we define the robustness score, $\mathcal{R}$, as the complement of the average score range per example:

$$\mathcal{R}_M = 1 - \frac{1}{|D|}\sum_{d\in D}\frac{1}{|d|}\sum_{i\in d}\Big[\max_{c\in C} S(M(x_i^c), y_i) - \min_{c\in C} S(M(x_i^c), y_i)\Big]$$

2.3 Setup

For each dataset, we sample 100 examples from the test set. For each example, as mentioned in §2.1, we represent its table using 7 different serializations. We also apply 4 different structural perturbations. As the perturbations are orthogonal to the chosen serialization, this yields a total of 35 $p \in P$ prompt configurations (7 serializations $\times$ 4 perturbations + 7 without perturbation).

We run a total of 14 models over ToRR, utilizing the Together AI[2] inference engine. We use 5-shot prompting[3], where each set of 5 shots is randomly sampled for each example, with greedy decoding and limit the maximum token output to 512. Each model was run on the same set of 100 examples per dataset[4] $\times$ 35 prompt configurations.

We utilize the `Unitxt` library (Bandel et al., 2024) to ensure that ToRR is shareable and reproducible, as also highlighted by Reuel et al. (2024). The modular customization of the library allowed us to manipulate the choice of table serialization and apply perturbations while keeping other aspects of prompt design (e.g., few-shot examples) constant. Further details about usage can be found in Appendix B.

3 Results & Analysis

ToRR offers multiple insights into the capabilities and performance of models on tabular tasks.

We present the high-level results of ToRR in §3.1. Next, we analyze advanced aspects of model performance in §3.2. Finally, we examine how prompt configurations affect performance (§3.3).

3.1 Model Capabilities

Table 2 showcases the main results of 14 open and closed models on ToRR. *claude-3-5-sonnet, gpt-4o* and *deepseek-v3* (full model names are in Appendix C.1) outperform others in most of the datasets, however models within the larger size range demonstrate similar performance. These models also have higher robustness scores, and overall, it appears that robustness scores correlate with performance scores, e.g., models with better performance tend

[2] https://www.together.ai/

[3] Except WikiTQ, which we truncated to one-shot due to length.

[4] TableBench FC has 96 examples; we used 79 for ToRR, the rest as demonstrations.

Model	$\mathcal{P}$ (↑)	$\mathcal{R}$ (↑)	FinQA	Numeric-NLG	QTSumm	SciGen	Tab Fact	TableBench DA	TableBench FC	TableBench NR	TURL CTA	WikiTQ
claude-3-5-sonnet	**.50**	**.70**	.43	.17	.37	.15	**.86**	.31	.70	**.42**	**.68**	**.92**
claude-3-5-haiku	.43	.61	.35	.17	.34	.15	.79	.23	.63	.20	.55	.85
gpt-4o	**.50**	.69	.40	**.19**	**.43**	**.17**	.83	**.33**	**.73**	.40	.60	.91
gpt-4o-mini	.43	.62	.34	.18	.40	.16	.65	.27	.64	.24	.54	.88
deepseek-v3	**.50**	.67	.46	**.19**	.41	.16	.81	.32	.71	.35	.66	.91
gemini-1.5-pro	.48	.65	**.47**	**.19**	.38	.16	.80	.29	.70	.32	.62	.89
gemini-1.5-flash	.45	.64	.43	**.19**	.34	.16	.77	.28	.70	.21	.57	.88
qwen2-72b-i	.45	.63	.38	.17	.42	.15	.78	.27	.68	.23	.60	.86
llama-3-1-405b-i	.46	.60	.36	.12	.42	.11	.82	.30	.65	.31	.62	.90
llama-3-1-70b-i	.44	.59	.37	.12	.42	.10	.73	.30	.63	.27	.60	.91
llama-3-1-8b-i	.29	.53	.04	.11	.35	.09	.55	.16	.52	.10	.18	.80
mixtral-8x22b-i	.41	.58	.28	**.19**	.41	**.17**	.74	.24	.68	.20	.54	.67
mixtral-8x7b-i	.35	.49	.20	.18	.36	**.17**	.65	.23	.59	.12	.35	.62
mistral-7b-i	.32	.54	.19	.14	.38	.13	.56	.22	.55	.10	.30	.68

Table 2: Main results of LLMs on ToRR: overall performance ($\mathcal{P}$) and robustness ($\mathcal{R}$) scores, along with performance scores for each dataset. Best scores are marked with bold, second best are underlined.

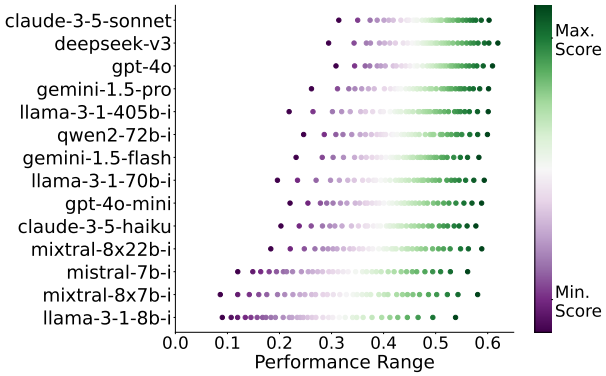


Figure 3: For each example we obtained 35 performance scores using different prompt configurations. The example scores are assigned an index in the range $[1, 35]$, ordered from lowest to highest performance. The plot depicts an average aggregation of each index across all examples. Models exhibit a wide range of scores, reflecting low robustness.

to be more robust. Two main trends that ToRR show are:

Current models exhibit limitations in processing tabular data The absolute scores of all models are medium-low, at best reaching 0.50 (see Table 2); but also, the gap between lower-performing models and higher-performing models is narrow. This indicates that better models are only moderately better at table understanding. Moreover, paired Cohen's d, an effect size metric, shows most model comparisons have small, often non-significant, practical differences. See Appendix D for more details.

All models are not robust. For each model and example from the selected datasets, we obtain a set of 35 scores (§2.3) that should all reflect how well the model handles this example and hence expected to mainly agree with each other. Yet in practice, these scores vary widely.

Figure 3 illustrates this range of scores for each model, aggregated across examples. As can be seen, the minimal score and the maximal score yield entirely different estimates of model performance. Thus, we see that the models exhibit strikingly brittle behavior, and are highly influenced by the choice of configuration. This suggests that current models do not have a stable and generalizable table representation that persists across table formats.

3.2 Model Performance Trends

As can be seen in Table 2, performance within model families is directly correlated to the model size. However, the differences in scores within families tend to be relatively small, and are 0.07 on average[5]. Across model families, size does not always indicate performance; for example, *qwen2-72b-instruct* outperforms *llama-3.1-405b-instruct* in both performance and robustness.

Table 2 compares model behavior across datasets, highlighting the strengths and weaknesses of each model. For example, while *claude-3-5-sonnet* outperforms others in classification tasks (e.g., TabFact), *mixtral-8x22b-instruct* performs the worst in them. However, *mixtral-8x22b-instruct* shows better capabilities than *claude-3-5-sonnet* in Table-to-Text tasks (e.g., QTSumm).

The order of models in Table 2 reflects an overall advantage of closed models over open models. Delving deeper, it appears they outperform in all

[5]Averaging over the differences between pairs of models within the same family.

tasks we examined. Additional analysis can be found in Appendix E.

3.3 Performance by Prompt Configuration

The notable lack of model robustness observed above raises the question of whether it results from certain configurations outperforming others.

No serializer leads to superior performance. To evaluate whether specific serializations give rise to better model performance, we calculate the win-rate of serializers at the example level. Then, we aggregate the results for all models and serializers and find that no serializer consistently outperforms others.

When breaking down the results by model, we do find some weak effects of preferred serializations for specific models, with a maximum difference of 0.06 in overall model performance across serializers. Additional details and figures can be found in Appendix F.1.

Perturbations have no consistent effect. We find that the perturbations outlined in §2.1 do not consistently affect performance, neither decreasing nor increasing model performance. For further details, refer to Appendix F.2.

4 Properties of ToRR

The reliability and validity of a benchmark are critical. We assess ToRR properties through statistical testing, separability analysis, and dataset agreement.

Statistical Testing Tied results in Section 3 raise concerns about the benchmark's ability to distinguish between models. Following Ackerman et al. (2025), we perform significance tests on all pairwise model comparisons across datasets. Many comparisons show statistically significant differences.[6]

Benchmark and Dataset Separability We adopt the "Separability with Confidence" metric from Li et al. (2024), measuring the percentage of model pairs with non-overlapping confidence intervals.[7] Separability varies across datasets (see App. Figure 19), with *WikiTQ* scoring over 71% and *TableBench FC* only 38%. Aggregated ToRR achieves 79%, supporting the need for diverse datasets. This strengthens the need for evaluating models on tables with multiple varied datasets in order to get a reliable result. Indeed, the separability score for the aggregated ToRR is significantly higher, accounting for 79% by applying a similar calculation across all benchmark examples. Further interpretation is in Appendix G.3.

Dataset Agreements To validate that ToRR captures varied skills and difficulty levels, we compute model rankings per dataset and measure agreement using Kendall's tau. Most datasets show medium to high agreement; however, Table-to-Text datasets diverge, with *SciGen* and *NumericNLG* aligning closely (see App. Figure 20).

5 Implications for Reliable Evaluation

A key design choice in ToRR is evaluating performance across prompt configurations (§2.1), which vary in table structure but preserve semantics.

While average performance is not consistently affected by prompt configuration choice (§3.3), the variability it introduces has major implications for evaluation reliability.

We use ToRR to examine how prompt variation impacts the stability of model rankings, a critical aspect of benchmark reliability. Prompt configuration is one of many arbitrary design decisions that can influence evaluation outcomes (Perlitz et al., 2024; Reuel et al., 2024).

Following Perlitz et al. (2024), we define reliability as ranking consistency across experimental choices. We measure this using **Kendall's W** (Kendall and Smith, 1939), which quantifies agreement among rankings in the range $[0.0, 1.0]$.

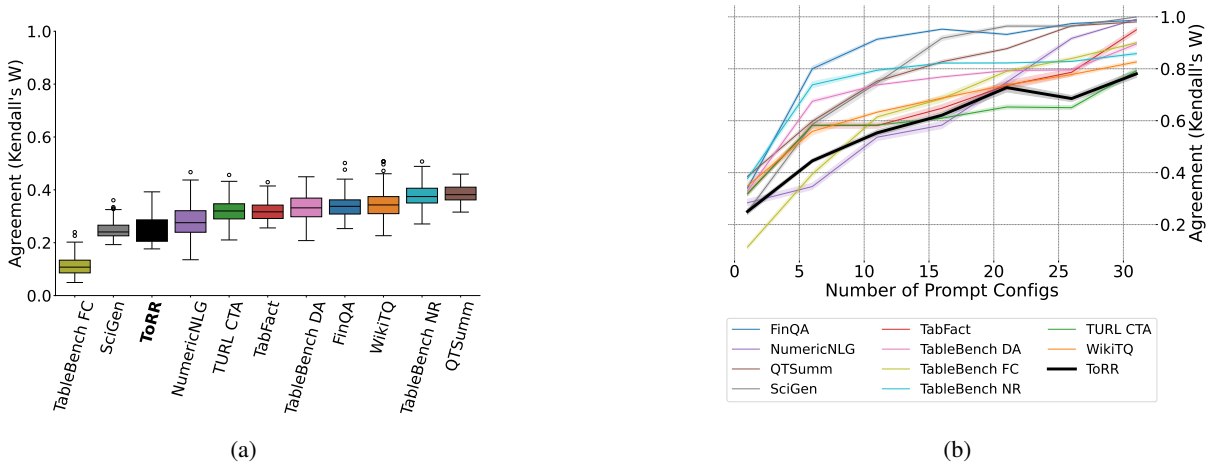
5.1 Using a Single Prompt Config is Unreliable

Existing benchmarks usually select one prompt format (for example, serialize the table using JSON). We ask to what extent this choice influences the model ranking. Thus, we calculate model rankings based on each of our prompt configurations, and test the similarity between them.

The result is depicted in Figure 4a. The agreement scores are generally low, demonstrating that model ranking order changes dramatically based on the choice of prompt. In other words, if a benchmark uses only a single configuration, the resulting model ranking would not be reliable.

[6] Differences between p-values and effect sizes (see §3.1) may stem from the large sample size in ToRR. Details in Appendix D.

[7] We employed bootstrapping with 1K randomly selected seeds to sample 100 examples from each dataset.



(a) (b)

Figure 4: Agreement between model rankings. We sample 30 prompt configuration sets, calculate the model ranking of each set, and report the agreement (Kendall's W) between rankings. (a) *Agreement between rankings based on a single prompt configuration*. Overall we see low agreement, indicating an unreliable model ranking (i.e., choosing a different prompt would lead to a different ranking). (b) *Agreement between rankings based on multiple prompt configurations.* Adding prompts increases the ranking consistency, with the largest gain between $\sim$ 2-8 prompts. Note that *(a)* is a zoomed-in view of *(b)* where the number of prompt configurations is 1.

5.2 Multiple Prompt Configs Increase Reliability

Figure 4b depicts the effect of evaluating with multiple prompts. Unsurprisingly, we see that basing the model ranking on more prompts increases the agreement the resulting rankings have with each other. The plot also shows that even a relatively small number of prompt configurations can make a large difference and contribute to a more reliable model ranking. For example, increasing the number of prompts from 1 to 10 increases Kendall's W score by more than 0.35 on average.

Another observation from Figure 4b is that the datasets exhibit differing patterns of the increase in agreement. For example, the agreement for *FinQA* increases from 0.35 to 0.93 using 11 prompts, while for *NumericNLG* it increases from 0.29 to 0.54, suggesting that the former is more robust to prompt configurations than the latter. The full ToRR benchmark (black line in Fig. 4b) is roughly a lower bound on the prompt robustness of the model ranking across datasets. Moreover, there appears to be a positive correlation between the task complexity suggested by our reasoning analysis (Table 1) and the robustness of the datasets: FinQA, TableBench DA, and TableBench NR generally rank at the top; WikiTQ, TabFact, and QTSumm fall in the mid-range; while NumericNLG and TURL tend to show lower robustness compared to the others.

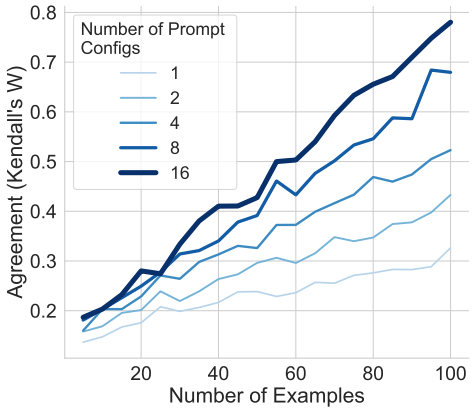


Figure 5: The improvement in model ranking consistency using different numbers of examples and prompt configurations. We calculate Kendall's W agreement over 30 sets of model rankings, where each model ranking is a result of randomly selected examples and prompt configurations. For example, for 2 prompt configurations and 50 examples, we repeat the following process 30 times: randomly sample 2 prompt configurations and 50 examples, then calculate sample model ranking.

5.3 Prompt Configs can Substitute Examples

Figure 4b shows that more prompts can help increase reliability. A simple explanation for this is that *adding more data points* helps capture model behavior. A common approach for this would be to add more test instances; however, increasing the size of labeled data is not always feasible.

Thus, we also look at the relation between adding examples and adding prompts. Figure 5

depicts the ranking agreement as a function of the number of test examples, averaged over each dataset. It is evident that both the number of examples and the number of prompts consistently increase reliability. Strikingly, adding prompt variations can match the effect of doubling the test set; for instance, 50 examples with 2 prompts yield similar reliability to 100 examples with one.

To conclude, evaluating multiple prompt configurations adds a valuable dimension to model assessment. While demonstrated on table formats, this approach may generalize to other tasks (Liang et al., 2023; Alzahrani et al., 2024; Mizrahi et al., 2023), improving reliability and helping mitigate small test set limitations.

6 Related Work

Several recent works systematically evaluate LLMs on tabular data tasks, primarily focusing on question-answering type tasks. TableBench (Wu et al., 2024) tests LLMs over four major categories of QA tasks, namely fact-checking, numerical reasoning, data analysis, and visualization. DataBench (Grijalba et al., 2024) examines the reasoning capabilities of LLMs in a tabular context. TQA-Bench (Qiu et al., 2024), a multi-table benchmark, evaluates complex question answering over relational data. TabIS (Pang et al., 2024) evaluates the table information-seeking capabilities of LLMs. Zhao et al. (2023b) investigate Table-to-text capabilities in several real-world information-seeking scenarios. Compared to ToRR, these benchmarks are limited in the tasks they cover and the model capabilities they reflect. A detailed comparison to other table benchmarks is provided in App. Table 8.

More recently, there has been an increased focus on also analyzing the robustness of LLMs across table formats and perturbations. Singha et al. (2023) explore the impact of table representation formats and noise operations on self-supervised table structure understanding tasks. Specifically, they consider a set of simple fact-finding and transformation tasks on tables to analyze how GPT-3 model performance varies. Similarly, Sui et al. (2024) analyzes the capabilities of GPT-3.5 and GPT-4 in understanding tables by designing a specific set of table structure understanding tasks using structured data from various public datasets. Several recent works (Bhandari et al., 2025; Zhao et al., 2023c; Liu et al., 2023) explore structural variance and adversarial perturbations on tables, and their impact on Table Question Answering performance. However, existing efforts are often restricted to simple synthetic table understanding tasks or a narrow set of table QA datasets and target models, thus overlooking broader challenges posed by complex table reasoning tasks.

7 Discussion

In this work, we introduce ToRR, the first comprehensive benchmark for table reasoning and robustness. ToRR provides a crucial resource for model developers and users seeking a more realistic and nuanced understanding of how LLMs perform in real-world tabular data scenarios.

Our results over a variety of state-of-the-art LLMs reveal a consistent pattern of relatively low performance on table reasoning tasks. Furthermore, and perhaps more strikingly, we demonstrate that models exhibit extreme sensitivity to seemingly minor variations in table formatting. As we show, this sensitivity does not reflect LLM preference for particular table formats; rather, the response to format variations reflects a more general phenomenon of LLM sensitivity to prompts.

This finding resonates with a growing body of recent research demonstrating the brittleness of LLMs to variations in input formatting, even in seemingly non-structured aspects of textual inputs (Alzahrani et al., 2024). These studies, while focusing on different input modalities, converge on a similar conclusion: LLMs' performance can be surprisingly brittle and inconsistent when faced with even minor input variations. This brittleness presents a real issue for reliably evaluating LLM performance (Mizrahi et al., 2023).

ToRR directly addresses these reliability concerns by systematically incorporating multiple prompt configurations into the evaluation protocol. Thus, ToRR offers a significant step towards mitigating the evaluation reliability issues, and provides a more reliable assessment of LLM capabilities. Moreover, utilizing multiple prompt configurations can help address the limitations of smaller test sets.

To conclude, ToRR serves as a crucial benchmark for orienting future advancements in LLM table reasoning. At the same time, it sets an example for robust evaluation methodologies that are more reflective of real-world performance. We encourage the AI community to adopt this benchmark and refine the practices it introduces when developing new benchmarks.

8 Limitations

Selected Datasets ToRR includes datasets in which input tables are directly embedded within the prompt and can be accurately parsed into standard table formats. However, this setup does not encompass all real-world scenarios, such as cases where models must extract or search for table data independently, or handle non-standard formats like hierarchical tables.

Dataset Metrics ToRR relies on datasets and evaluation metrics created by external sources, which may introduce biases or inconsistencies that do not fully align with the intended evaluation goals.

Perturbation Design We selected four perturbations that primarily alter the order in which table content is presented in the prompt. While no semantic changes were introduced, a small number of cases may have been unintentionally affected, potentially leading to incorrect gold answers. However, limited human evaluation and additional analysis (see Appendix F.2) suggest that such instances are rare and do not significantly impact the overall results.

Acknowledgments

We thank Ella Rabinovich for her valuable contributions in shaping the metrics used in this work. Her insights and feedback significantly improved the design of ToRR and strengthened the analysis presented in this paper.

We also thank Sam Ackerman for his contributions to the statistical aspects of this work. His analysis and input helped strengthen the results of the benchmark.

References

Samuel Ackerman, Eitan Farchi, Orna Raz, and Assaf Toledo. 2025. Statistical multi-metric evaluation and visualization of llm system predictive performance. *Preprint*, arXiv:2501.18243.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *Preprint*, arXiv:2402.01781.

Elron Bandel, Yotam Perlitz, Elad Venezian, Roni Friedman, Ofir Arviv, Matan Orbach, Shachar Don-Yehiya, Dafna Sheinwald, Ariel Gera, Leshem Choshen, Michal Shmueli-Scheuer, and Yoav Katz. 2024. Unitxt: Flexible, shareable and reusable data preparation and evaluation for generative AI. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 207–215, Mexico City, Mexico. Association for Computational Linguistics.

Yoav Benjamini and Daniel Yekutieli. 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188.

Kushal Raj Bhandari, Sixue Xing, Soham Dan, and Jianxi Gao. 2025. Exploring the robustness of language models for tabular question answering via attention analysis. *Preprint*, arXiv:2406.12719.

Wenhu Chen. 2022. Large language models are few (1)-shot table reasoners. *arXiv preprint arXiv:2210.06710*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2022. FinQA: A dataset of numerical reasoning over financial data. *Preprint*, arXiv:2109.00122.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding through representation learning. *Preprint*, arXiv:2006.14806.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large language models(llms) on tabular data: Prediction, generation, and understanding – a survey. *Preprint*, arXiv:2402.17944.

Jorge Osés Grijalba, L Alfonso Urena Lopez, Eugenio Martínez-Cámara, and Jose Camacho-Collados. 2024. Question answering over tabular data with databench: A large-scale empirical evaluation of LLMs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13471–13488.

Zihui Gu, Ju Fan, Nan Tang, Preslav Nakov, Xiaoman Zhao, and Xiaoyong Du. 2022. PASTA: Table-operations aware fact verification via sentence-table cloze pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4971–4983, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maurice G Kendall and B Babington Smith. 1939. The problem of m rankings. *The annals of mathematical statistics*, 10(3):275–287.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *Preprint*, arXiv:2406.11939.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Preprint*, arXiv:2211.09110.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023. Rethinking tabular data understanding with large language models. *arXiv preprint arXiv:2312.16702*.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: a survey. *Frontiers of Computer Science*, 19(2).

Moran Mizrahi, Guy Kaplan, Daniel Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2023. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Scigen: a dataset for reasoning-aware text generation from scientific tables. In *NeurIPS Datasets and Benchmarks*.

Chaoxu Pang, Yixuan Cao, Chunhao Yang, and Ping Luo. 2024. Uncovering limitations of large language models in information seeking from tables. *arXiv preprint arXiv:2406.04113*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics*.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2024. Efficient benchmarking of language models. *Preprint*, arXiv:2308.11696.

Zipeng Qiu, You Peng, Guangxin He, Binhang Yuan, and Chen Wang. 2024. TQA-Bench: Evaluating LLMs for multi-table question answering with scalable context and symbolic extension. *arXiv preprint arXiv:2411.19504*.

Anka Reuel, Amelia Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, and Mykel J. Kochenderfer. 2024. Betterbench: Assessing ai benchmarks, uncovering issues, and establishing best practices. *Preprint*, arXiv:2411.12990.

Yucheng Ruan, Xiang Lan, Jingying Ma, Yizhi Dong, Kai He, and Mengling Feng. 2024. Language modeling on tabular data: A survey of foundations, techniques and evolution. *arXiv preprint arXiv:2408.10548*.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023. Tabular representation, noisy operators, and impacts on table structure understanding tasks in LLMs. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards table-to-text generation with numerical reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1451–1465, Online. Association for Computational Linguistics.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. *Preprint*, arXiv:2305.13062.

III Turner, Herbert M and Robert M Bernard. 2006. Calculating and synthesizing effect sizes. *Contemporary issues in communication science and disorders*, 33(Spring):42–55.

Daniel J Wilson. 2019. The harmonic mean p-value for combining dependent tests. *Proceedings of the National Academy of Sciences*, 116(4):1195–1200.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xinrun Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Guanglin Niu, Tongliang Li, and Zhoujun Li. 2024. Tablebench: A comprehensive and complex benchmark for table question answering. *Preprint*, arXiv:2408.09174.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Ruizhe Chen, Xiangru Tang, Yumo Xu, Dragomir Radev, and Arman Cohan. 2023a. Qtsumm: Query-focused summarization over tabular data. *Preprint*, arXiv:2305.14303.

Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. 2023b. Investigating table-to-text generation capabilities of LLMs in real-world information seeking scenarios. *arXiv preprint arXiv:2305.14987*.

Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. 2023c. Robut: A systematic study of table QA robustness against human-annotated adversarial perturbations. *arXiv preprint arXiv:2306.14321*.

A ToRR Benchmark

In this section, we provide more details about the benchmark and the decisions made as part of its development.

A.1 Overview of Datasets in ToRR

1. **FinQA** (Chen et al., 2022) – An expert-annotated question answering (QA) dataset designed to tackle numerical reasoning in real-world financial data. FinQA contains questions that require models to perform complex operations, such as multi-step calculations and logical reasoning over financial reports containing both text and table. *License: Creative Commons Attribution 4.0 International (CC BY 4.0).*

2. **TableBench** (Wu et al., 2024) – A dataset designed to evaluate a model's table question answering capabilities across various tasks. It includes 18 distinct sub-tasks grouped into four major categories: Fact Verification (FV), Numerical Reasoning (NR), Data Analysis (DA), and visualizations. *License: Apache License 2.0.*

3. **WikiTableQuestions** (Pasupat and Liang, 2015) – A dataset designed for question answering over tables sourced from Wikipedia. It often requires reasoning over table data, including operations like aggregation, comparisons, and filtering, to derive accurate answers. *License: Creative Commons Attribution 4.0 International (CC BY 4.0).*

4. **TabFact** (Chen et al., 2020) – A large dataset focused on fact verification using tables, containing Wikipedia tables paired with human-annotated statements. The task is to determine whether a given statement is supported, refuted, or unverifiable based on the information in the table often requiring logical and numerical reasoning over table data. *License: Creative Commons Attribution 4.0 International (CC BY 4.0).*

5. **QTSumm** (Zhao et al., 2023a) - A summarization dataset focused on query-based summarization, where summaries are generated based on specific user queries to retrieve relevant information from a table. *License: MIT License*.

6. **Scigen** (Moosavi et al., 2021) - A dataset designed for reasoning-aware data-to-text generation, featuring tables from scientific articles along with their corresponding descriptions. *License: Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0).*

7. **NumericNLG** (Suadaa et al., 2021) - A dataset for table-to-text generation that pairs tables with their corresponding descriptions from scientific papers, with a focus on numerical-reasoning texts. *License: Creative Commons 4.0 Attribution-ShareAlike (CC BY-SA 4.0)..*

8. **TURL (Table Understanding through Representation Learning)** (Deng et al., 2020) – The TURL Column Type Annotation (CTA) dataset, derived from Wikipedia tables, supports semantic type assignment to table columns from a given list of Freebase types. It tests models' ability to understand the meaning of table columns in context. *License: Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).*

A.2 Other Table Dataset Types

While we selected datasets that can be used directly in the prompt to measure the model's ability to handle tables in the cleanest way possible, there also exist table datasets that require tool use. Our impression is that such datasets do involve the skills analyzed in this work, but they additionally depend on capabilities like tool-use formats and agentic behaviors such as search or code execution. We intentionally excluded these elements to keep the benchmark focused and interpretable, specifically targeting the model's core tabular understanding. Moreover, many models cannot operate in such settings, and even when they can, most researchers would find the setup challenging, as evaluation frameworks are typically not designed for tool use. This design choice enables us to isolate reasoning abilities without the confounding influence of other skills. Thus, while broader coverage would allow averaging over yet another dataset, it would come at the cost of usability.

A.3 Evaluation metrics

We evaluated each of the datasets using the same metrics as reported in the original papers, with

some exceptions: (1) For **WikiTableQuestions**, while the original paper focuses on exact-match for scoring the prediction, we assessed models using F1 score[8], to provide a better normalization of model performance. (2) In the case of **FinQA**, we calculated both execution accuracy and program accuracy, but used the latter as the main metric since execution accuracy tends to overestimate the performance. (3) As for **Scigen**, since the base paper states that none of the evaluated metrics align with human judgment, we adopted the primary metric of NumericNLG, as they correspond to the exact same task.

A.4 Dataset Properties Evaluation

To conduct a manual skill analysis across datasets, we sampled 15 examples from each dataset and engaged three research scientists as annotators. Each annotator independently assessed the skills required to solve each example. A skill was considered "required" if it appeared consistently across all 15 examples, "not required" if it was absent in all examples, and "partially required" if it was present in at least 7 examples. Final labels were determined based on consensus, with agreement from at least two annotators deemed sufficient.

A.5 Prompt Configurations

A.5.1 Serializations

As tables can be presented in various formats during pre-training, models may have biases in how they interpret them as demonstrated by (Sui et al., 2024). Several standard methods exist for converting structured data into text formats that can be embedded in a prompt. We select seven of the most commonly used serializations: *HTML, CSV, JSON, Markdown, Indexed Row Major, Data Frame and Concatenation*. An example representation of each serialization format is provided in Table 3.

A.5.2 Structural Perturbations

These perturbations present a slightly different structure of the table from the original one, without any change to the content or relations between the table cells. Since the information is preserved but presented in a different form, there should be no significant change in the model performance ideally. The perturbations considered are as follows:

- **Row Swapping**: The process of exchanging the positions of rows within a table.
- **Column Swapping**: The act of changing the positions of columns in a table.
- **Transpose**: The action of switching the rows and columns of a standard table, transforming rows into columns and vice versa.
- **Add Empty Rows**: An addition of some empty rows to the table.

Refer to Table 4 for an example of these structural perturbations with CSV as a base serializer.

A.6 Data Contamination

A general concern in benchmarking is the issue of data contamination, where the benchmarked models have memorized some of the data in the benchmark. Our approach in ToRR is likely to mitigate some of the effects of contamination due to the use of 35 distinct prompt variations per example. Specifically, we expect the effects of memorization to be tied to the exact phrasing of an input example; thus, using an unseen prompt format the model is less likely to utilize memorized data. Hence, over a diverse set of prompts we do not expect a model to succeed based solely on memorized data. In other words, our derived performance metrics implicitly penalize shallow memorization, offering a more reliable evaluation of true generalization capabilities (e.g., if a model succeeds on one specific format of an example due to memorization, it will get a performance score of 2.8% on that example).

Moreover, the diversity of datasets in terms of domains, creators, and other attributes further contributes to reducing the likelihood of contamination.

Nevertheless, as is the case for other benchmarks, we cannot rule out some effects of data contamination on the results.

A.7 Challenges

Building a robust benchmark for table reasoning like ToRR presented several challenges, particularly related to dataset quality and evaluation consistency.

[8]We tokenized both the reference and predicted tokens, where true positives are determined by the intersection of these token sets, false positives are the tokens present in the reference set but absent from the predicted set, and false negatives are the tokens in the predicted set that are missing from the reference set.

Format	Example Representation
HTML	<table><thead> <tr><th>Name</th><th>Age</th><th>Sex</th></tr> </thead><tbody> <tr><td>Sophia</td><td>26</td><td>F</td></tr> <tr><td>Aarav</td><td>34</td><td>M</td></tr> <tr><td>Oliver</td><td>30</td><td>M</td></tr> </tbody> </table>
CSV	Name, Age, Sex Sophia, 26, F Aarav, 34, M Oliver, 30, M
JSON	{“0”: {“Name”: “Sophia”, “Age”: “26”, “Sex”: “F”}, “1”: {“Name”: “Aarav”, “Age”: “34”, “Sex”: “M”}, “2”: {“Name”: “Oliver”, “Age”: “30”, “Sex”: “M”}}
Markdown	|Name|Age|Sex| |–|–|–| |Sophia|26|F| |Aarav|34|M| |Oliver|30|M|
Indexed Row Major	col : Name | Age | Sex row 1 : Sophia | 26 | F row 2 : Aarav | 34 | M row 3 : Oliver | 30 | M
DataFrame	pd.DataFrame({“Name”: [“Sophia”, “Aarav”, “Oliver”], “Age”: [26, 34, 30], “Sex”: [“F”, “M”, “M”]}, index=[0, 1, 2])
Concatenation	Name Age Sex Sophia 26 F Aarav 34 M Oliver 30 M

Table 3: Example representations of serialization formats used in ToRR.

Memorized Data. Many existing table datasets are derived from the same sources, particularly Wikipedia tables, which large language models have likely seen during pre-training. As a result, these datasets may be less challenging, as models can rely on memorization rather than genuine reasoning.

Evaluation Rigor. The evaluation metrics used in prior work do not always accurately reflect model performance. Several studies indicate that automatic metrics often fail to align with human evaluations, leading to misleading conclusions about model capabilities. This inconsistency makes it difficult to assess true reasoning ability and robustness.

Unreliable Data. Some datasets rely on automatically aggregated structured data, which can introduce inconsistencies and errors. For instance, a dataset with high potential for benchmarking included web tables, but the extracted table HTML was broken at times, making it unreliable for structured reasoning tasks as such issues can compromise the integrity of the benchmark.

A.8 Computation Cost

The full ToRR benchmark uses between approximately 110 million and 140 million tokens per model under evaluation. Based on the pricing on the model inference platforms used for this paper (OpenAI, Anthropic, Google Vertex AI and Together AI) as of May 2025, the total cost of model inference API credits for all evaluations was approximately $\$2{,}400$.

Perturbation	Example Transformation
Row Swapping	Name, Age, Sex Aarav, 34, M Oliver, 30, M Sophia, 26, F
Column Swapping	Age, Name, Sex 26, Sophia, F 34, Aarav, M 30, Oliver, M
Transpose	, 0, 1, 2 Name, Sophia, Aarav, Oliver Age, 26, 34, 30 Sex, F, M, M
Add Empty Rows	Name, Age, Sex , , Sophia, 26, F , , Aarav, 34, M Oliver, 30, M

Table 4: Structural perturbations with CSV serializer format.

B Usage

ToRR can be easily run using `unitxt`. After installing the package, users can reproduce our results with the following code snippet:

```
from unitxt import evaluate, load_dataset, settings
from unitxt.inference import (
    HFPipelineBasedInferenceEngine,
)

test_dataset = load_dataset(
    "benchmarks.torr",
    split="test",
    use_cache=True,
)

# Infer using inference engine and LLM of your choice
model = HFPipelineBasedInferenceEngine(     # We used Together AI as inference engine
    model_name="google/flan-t5-base", max_new_tokens=512
)

predictions = model(test_dataset)
results = evaluate(predictions=predictions, data=test_dataset)

print("Global scores:")
print(results.global_scores.summary)
print("Subsets scores:")
print(results.subsets.scores.summary)
```

Figure 6: Example code for running ToRR using `unitxt`.

C Full Names

C.1 Model Names

Full Model Name	Short Name
qwen2-72b-instruct	qwen2-72b-i
mixtral-8x22b-instruct-v0.1	mixtral-8x22b-i
mixtral-8x7b-instruct-v0.1	mixtral-8x7b-i
mistral-7b-instruct-v0.3	mistral-7b-i
llama-3-1-70b-instruct	llama-3-1-70b-i
llama-3.1-70b-instruct	llama-3-1-70b-i
llama-3.1-405b-instruct	llama-3-1-405b-i
llama-3.1-8b-instruct	llama-3-1-8b-i
gpt-4o-mini-2024-07-18	gpt-4o-mini
gpt-4o-2024-11-20	gpt-4o
gemini-1.5-flash-002	gemini-1.5-flash
gemini-1.5-pro-002	gemini-1.5-pro
claude-3-5-sonnet-20241022	claude-3-5-sonnet
claude-3-5-haiku-20241022	claude-3-5-haiku

Table 5: Mapping of model names to short names.

C.2 Dataset Names

Full Dataset Name	Short Name
TURL - Column Type Annotation	TURL CTA
TableBench - Data Analysis	TableBench DA
TableBench - Numerical Reasoning	TableBench NR
TableBench - Fact Verification	TableBench FC

Table 6: Mapping of dataset names to short names.

D Statistical testing of models

Here we present results from a statistical analysis of model performance. We analyze the results of the 14 models listed in Table 2 on the ten datasets listed in Table 1. The analysis follows the procedure in Ackerman et al. (2025).

We define a single run configuration for an LLM as the unique combination of values of the serializer, augmenter, and example index. For each dataset, the average metric score of the model on each configuration is calculated. This yields $3{,}500$ unique configurations for each dataset, except for TableBench FC, which had $2{,}765$. Following the analysis setup in Ackerman et al. (2025), a new dataset is formed for each source dataset in Table 1, where each observation in the dataset corresponds to a particular configuration, and its (single) score is the average metric value for the configuration. Thus, for instance, we form the dataset D_1 of size $3{,}500$ (unique configurations) for source dataset FinQA.

Since for each model we have results for all configurations, we can directly compare the results for a pair of models on a given configuration via a paired (observation-level) hypothesis test. For each dataset indexed j, and pair of models (a, b), we obtain a single p-value $p_{a,b,j}$ and effect size $e_{a,b,j}$, which reflect how significantly different the pair of models are across the configurations, on that dataset j. Effect sizes are often used as an alternative to p-values because they are less influenced by the sample size (i.e., the number of configurations) and aim to reflect whether the difference has a practical meaning. The p-values are the non-parametric paired Wilcoxon, adjusted to control the false discovery rate with $\alpha = 0.05$ (Benjamini and Yekutieli, 2001).

We find that *within each dataset*, some model pairs were not statistically significantly different using the p-value method, after adjusting for multiplicity.

We also calculate *aggregated* p-values ($p_{a,b}$) and effect sizes ($e_{a,b}$) across all the datasets. Following Ackerman et al. (2025), p-values are aggregated using Wilson's harmonic mean method (Wilson, 2019) and effect sizes using the inverse variance-weighted mean (Turner and Bernard, 2006).

After aggregating the results across all datasets, we find that all pairwise model comparisons are significantly different from each other according to the p-value ($p < 0.05$). However, using the effect size criterion (paired Cohen's d), which better reflects practical differences, nearly all pairwise effects are relatively small (smaller than 0.5).

E Performance Analysis

This part of the paper presents additional analyses and graphs illustrating model performance.

E.1 Model Performance across Datasets

Figure 7 compares the performance of all models across each dataset in the benchmark. In general, performance tends to be lower on more challenging datasets- FinQA, TableBench Numerical Reasoning, TableBench Data Analysis as well as Table2Text ones. On the other hand, performance scores are relatively higher on question answering and fact-checking datasets including WikiTQ, TabFact and others.

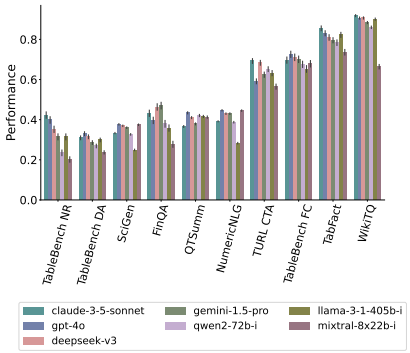


Figure 7: Performance comparison of the strongest model from each family.

E.2 Open vs. Closed models

Figure 8 presents the performance comparison between open-weight and closed-weight models across all the benchmark datasets. As can be seen, closed-weight proprietary models considerably outperform open models across most benchmark datasets.

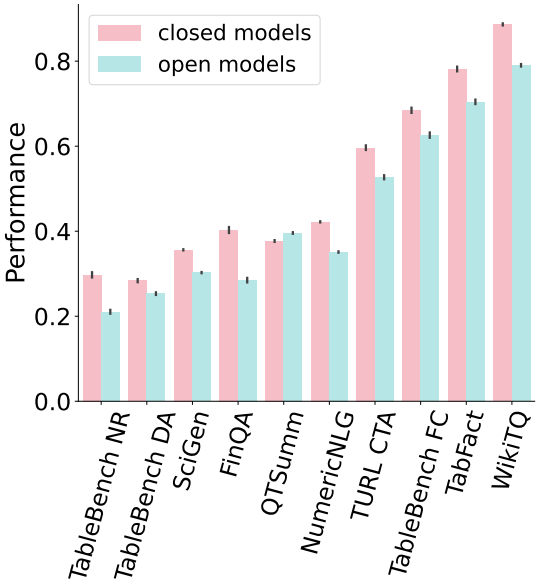


Figure 8: Comparison of open weight and closed models.

F Robustness Analysis

In this section, we present additional analyses of the key components of prompt configurations: serializers and structural perturbations.

We provide both performance scores and win rate graphs. All scores are calculated in a manner similar to that described in 2.2. Win rate calculations were performed at the example level: for each model's score on an example, we counted the number of times it was higher than the others, and then normalized it by the total number of wins over this example. For example, given the following scores of 7 serializers across an example: $[1, 1, 1, 0, 0, 0, 0]$, the counting vector will be $[4, 4, 4, 0, 0, 0, 0]$, and the normalized win rate scores will be $[1/3, 1/3, 1/3, 0, 0, 0, 0]$.

F.1 Serializers Impact

Here, we provide more in-depth analysis of trends concerning serializers. We approach this in steps: starting with the lowest level of aggregation for each model-dataset pair, we then move to model-oriented and dataset-oriented aggregations, and finally conclude with an overall aggregation of the serialization results.

F.1.1 Model-Dataset Score Variability

To measure the extent of variability with respect to serializers, we calculated the score for each model-dataset pair across all serializers and computed the difference between the highest and lowest scores. The result is shown in Table 7. While on average the drop in score is about 0.05, there are some exceptional cases were it can be much worst, e.g. *llama-3-1-8b-i* has a drop in score of 0.22 over *TableBench FC*.

F.1.2 Serializer Preference across Models

To identify trends related to models, we aggregate the results across models, as illustrated in Figure 10. This figure shows the serializer preferences of all models within our benchmark, indicating that different models exhibit varying preferences, with no single serializer consistently ranking highest across all models. Overall, since preferences are highly model-dependent, selecting the most suitable serializer may require case-by-case tuning.

F.1.3 Serializer Preference across Datasets

We explored signals indicating which serializers perform better for different tasks and datasets, as shown in Figure 11. The results show that no single serializer consistently outperforms others across all datasets. In some cases, the ranking differences among serializers is marginal, while in others, a clear performance gap emerges. This emphasizes the need for dataset-specific evaluation rather than a one-size-fits-all approach.

F.1.4 Overall Serializer Preferences

The results presented above show a lack of consistency in preferences. Unsurprisingly, aggregating all the results across models and datasets demonstrates that no serializer consistently outperforms others.

	FinQA	Numeric-NLG	QTSumm	SciGen	TURL CTA	Tab Fact	TableBench DA	TableBench FC	TableBench NR	WikiTQ
claude-3-5-haiku	.04	.01	.01	.01	.02	.06	.02	.07	.11	.05
claude-3-5-sonnet	.02	.01	.01	.00	.02	.03	.01	.08	.07	.02
deepseek-v3	.04	.02	.01	.00	.02	.06	.02	.03	.04	.04
gemini-1.5-flash	.03	.01	.10	.00	.02	.08	.01	.04	.06	.03
gemini-1.5-pro	.04	.01	.03	.01	.04	.05	.01	.04	.05	.05
gpt-4o	.05	.00	.01	.00	.04	.04	.01	.04	.05	.03
gpt-4o-mini	.06	.01	.01	.01	.03	.09	.01	.05	.06	.03
llama-3-1-405b-i	.04	.07	.02	.06	.04	.04	.05	.23	.06	.03
llama-3-1-70b-i	.02	.07	.01	.03	.04	.06	.04	.20	.04	.04
llama-3-1-8b-i	.03	.01	.02	.01	.17	.07	.07	.22	.07	.02
mistral-7b-i	.06	.04	.02	.04	.03	.08	.01	.14	.04	.04
mixtral-8x22b-i	.04	.01	.02	.01	.02	.04	.01	.09	.06	.06
mixtral-8x7b-i	.05	.03	.03	.02	.16	.06	.02	.11	.03	.09
qwen2-72b-i	.06	.07	.01	.05	.02	.03	.01	.06	.03	.02

Table 7: The largest variations in model scores across different serializers, presented for each dataset.

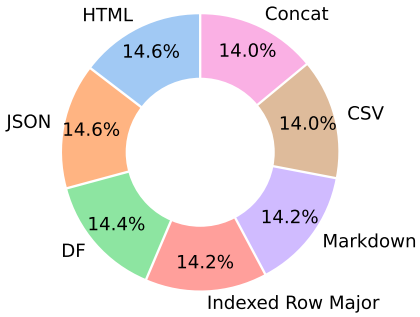


Figure 9: Win rate of serializers averaged across all models and datasets.

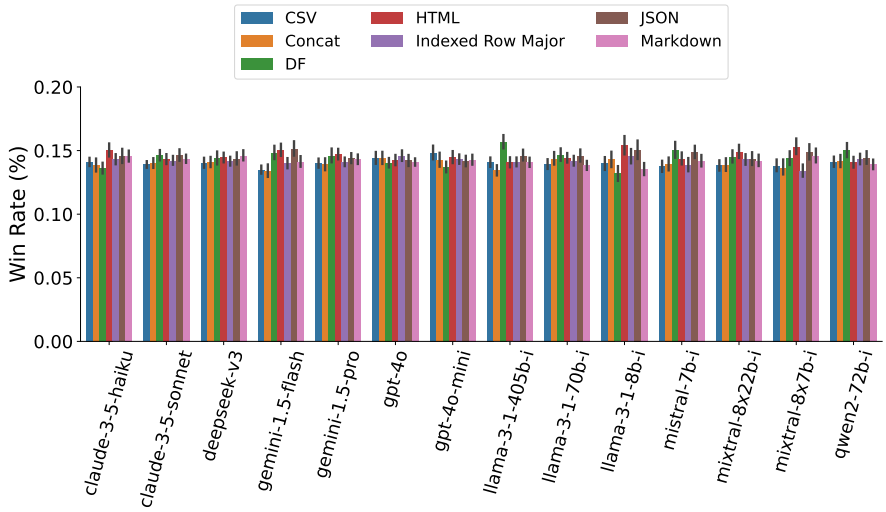


Figure 10: Model-wise win-rate comparison of serializers in ToRR. Some models don't show clear preferences, like *gpt-4o*, while others show a preference for one or more serializers. For example, *llama-3-1-405b-i* prefers DF serialization.

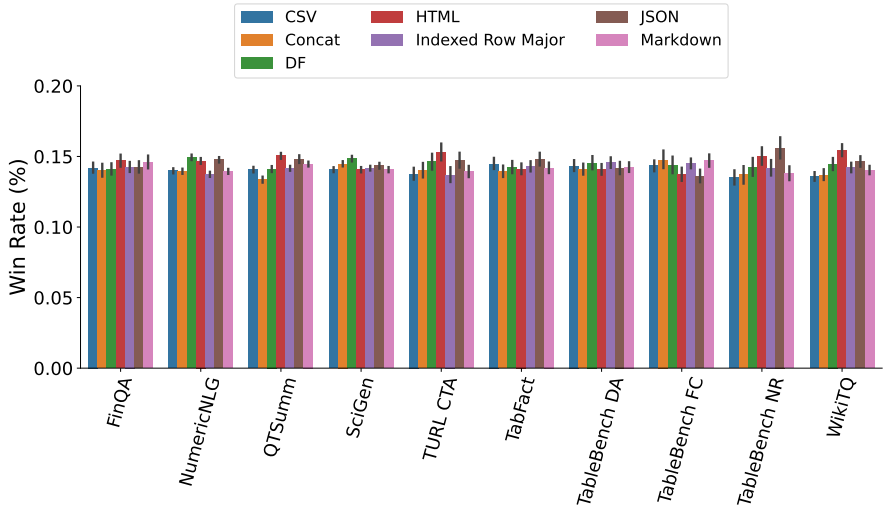


Figure 11: Dataset-wise win-rate comparison of serializers in ToRR. Surprisingly, the *concat* serialziation, which may render the prompt unreadable due to the lack of detailed separation, doesn't present a dramatically low performance across datasets and tasks.

F.2 Perturbations Impact

Here, we present a more detailed analysis of trends associated with the structural perturbations in ToRR. We begin with the low-level aggregations of the differences between scores with and without perturbations, and then we aggregate the overall results.

F.2.1 Perturbations Impact

To gain deep insights into the impact of perturbations on performance, we analyzed the score differences introduced by each perturbation across benchmark models. We established a baseline using prompts with a specific serializer and no perturbations, and then measured the score deviations caused by each perturbation in comparison to this baseline.

The result, depicted in Figure 12, reflects small yet inconsistent variability in model scores. Models exhibit an average difference of 0.03 in overall performance.

F.2.2 Granular Look: Absolute Perturbations Impact

While some perturbations exhibit a strong positive or negative impact on certain examples, their effects vary inconsistently across all datasets and models. A simple average, as we see in Figure 12, can be misleading as effects may cancel out. Here we report Mean Absolute Impact, which quantifies the overall effect using absolute score changes (Δ). Let us denote $i \in N$ the number of examples, and Δ_i the difference between observed and expected scores of perturbation with respect to its baseline. The Mean Absolute Impact is defined as:

$$\text{Mean Absolute Impact} = \frac{1}{N}\sum_{i=1}^{N}|\Delta_i|$$

Figure 13 shows the Mean Absolute Impact on performance scores due to different perturbations for each benchmark dataset, averaged across all models. The impact is higher for Table QA and Fact-checking datasets and lower for Table-to-Text datasets. Similarly Figure 14 displays the Mean

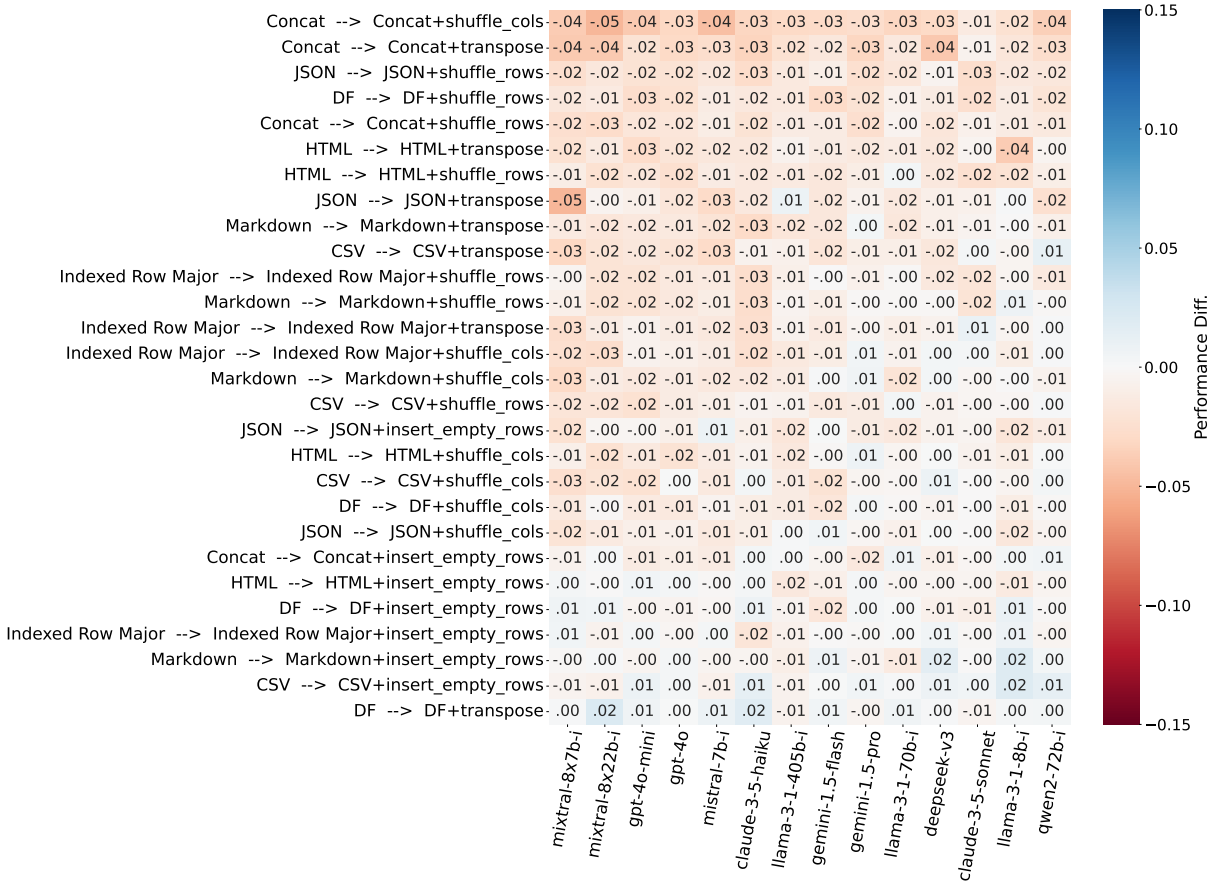


Figure 12: Differences in model score for perturbations with respect to the serializer. Overall, the effect of each perturbation seems to be very low and not consistent.

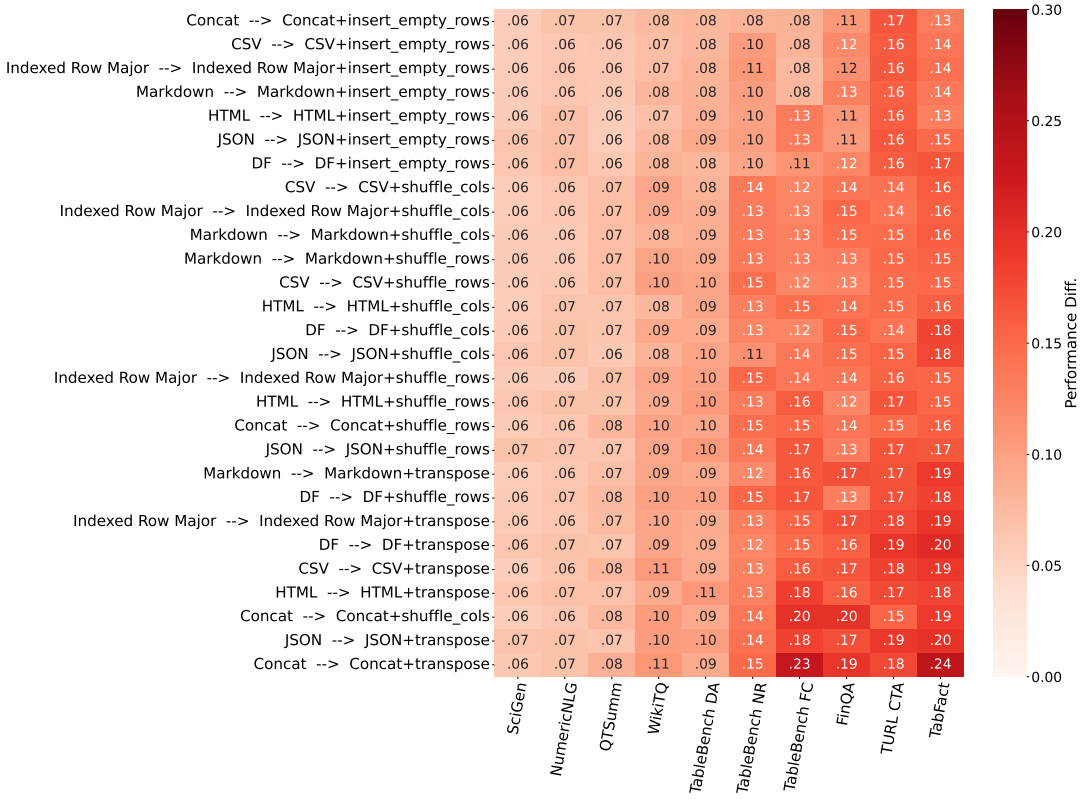


Figure 13: Mean Absolute Impact of different perturbations on benchmark datasets. The signal of differences in score appears to be a part of the datasets rather than related to perturbations.

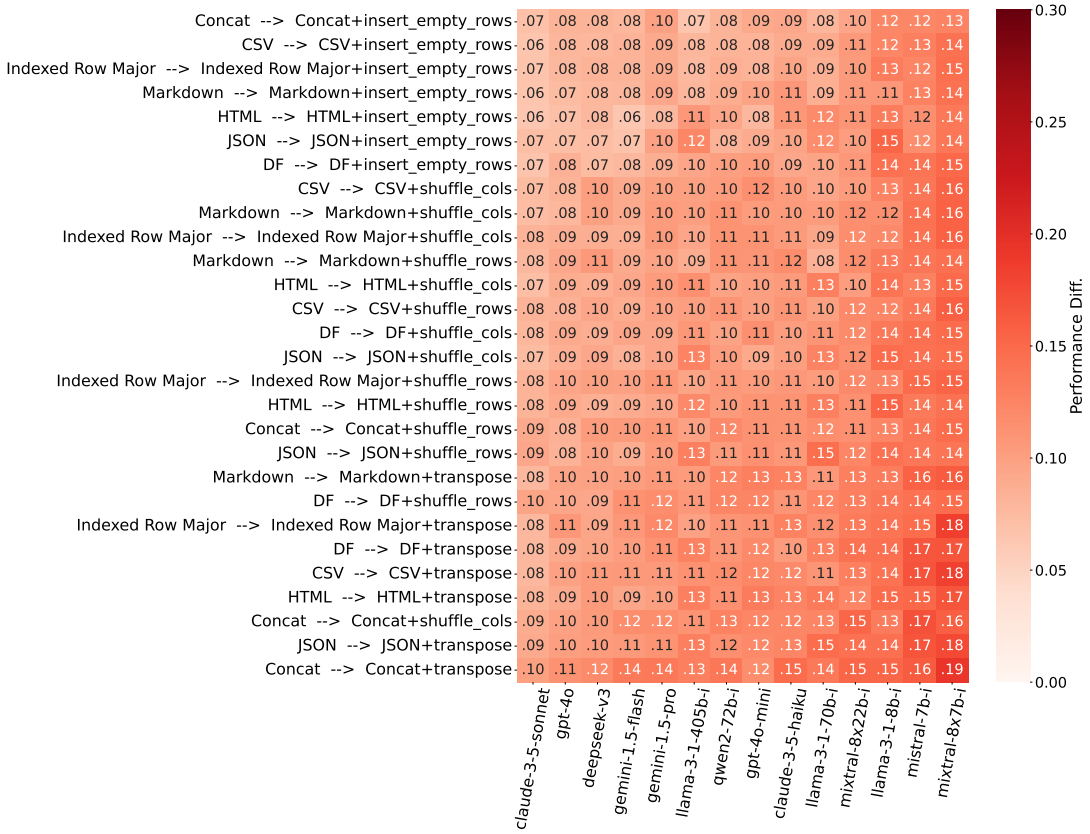


Figure 14: Mean Absolute Impact on each model performance due to perturbations. The average model robustness score (presented in Table 2) appears to correlate with the model's robustness across different perturbations.

Absolute Impact on performance scores for each model, averaged across all benchmark datasets. The impact of perturbations is more pronounced in smaller models compared to larger models in general. Overall, it seems that both datasets and models are key factors that influence robustness.

F.2.3 Overall Perturbations Effect

Figure 15 presents the win-rate of structural perturbations, representing the percentage of times a perturbation outperforms all others aggregated across the benchmark. The win rate results are nearly identical across all perturbations, indicating that no single perturbation consistently leads to under-performance or out-performance of models when aggregated across the benchmark datasets and models.

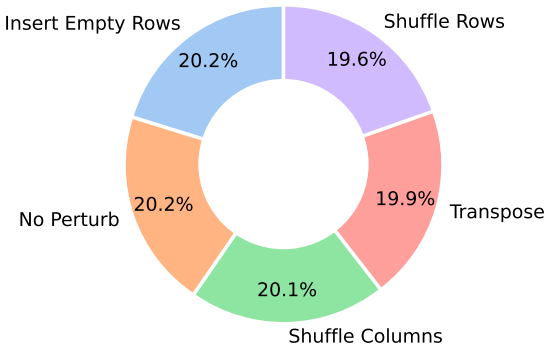


Figure 15: Win rate of structural perturbations across all models and datasets of ToRR.

G Additional Properties of ToRR

Another perspective on benchmark variability can be gained by examining it at the example level. We present two analysis that relate to example difficulty and can serve for indicating better on this variability in ToRR.

G.1 Score Distribution in ToRR

To gain insight into the difficulty of each dataset, we analyze the scores achieved by the models across different prompt configurations, focusing on the score distribution. The distribution patterns show notable differences, even when datasets were evaluated with the same metric, as shown in Figure 16. For example, datasets like TableBench, NumericNLG, and QTSumm, which all use *ROUGE*, exhibited distinct score patterns. This suggests that the dataset itself has a strong and varied impact on model performance.

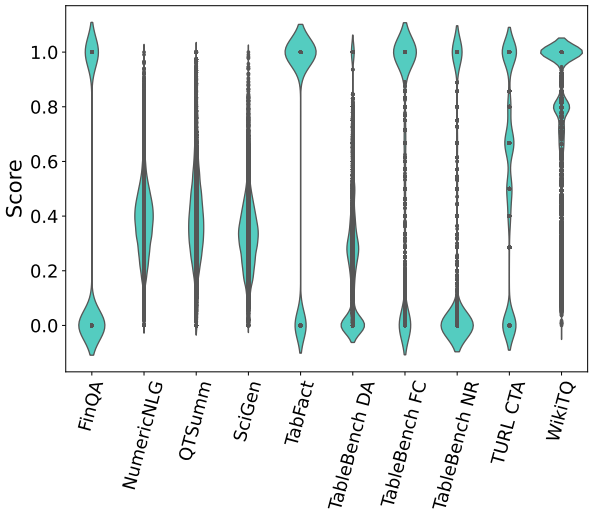


Figure 16: Score distribution across ToRR datasets.

G.2 Example Difficulty in ToRR

We also analyze the difficulty of examples within our benchmark by computing the mean score for each example across all models and prompt configurations. We then examine the distribution of these aggregated mean scores across datasets, as illustrated in Figure 17. The figure indicates that our benchmark encompasses both easy and challenging examples, with the majority falling within a medium difficulty range.

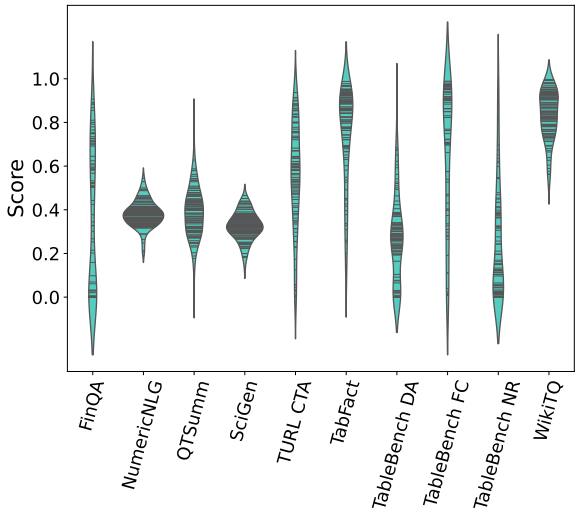


Figure 17: Example difficulty distribution for each dataset in ToRR. Some datasets like *FinQA* and *TableBench* ones include examples with different levels of difficulty for the models, while others like Table-to-Text ones seem to include examples with a similar level.

G.3 An Analysis for Interpreting Dataset Separability

As can be seen in Figure 19, some datasets exhibit higher separability values, while others do not, and this does not strongly correlate with other analyses or results we obtained. To gain intuition about why this occurs, we visualized the behavior of model scores for each dataset using 100 different seeds through bootstrapping — following the same method we used to compute separability, and the results are shown in Figure 18.

The figure suggests that this score is influenced by several factors; The overall score distribution across the dataset, the selected models and the score differences between them, and the variation in difficulty and diversity of the dataset's questions. For example, *NumericNLG* and *SciGen* appear more homogeneous in terms of question types, resulting in stable model scores and, consequently, higher separability.

In contrast, *WikiTQ* shows a more expected trend: weaker models tend to vary more in their scores than stronger ones, and the observed high separability may be a result of the selection of models. *TableBench NR*, on the other hand, seems to be challenging for all models (i.e., the score range is approximately $0.15 - 0.40$) but also contains a wide variety of questions, as indicated by the high variability in model scores.

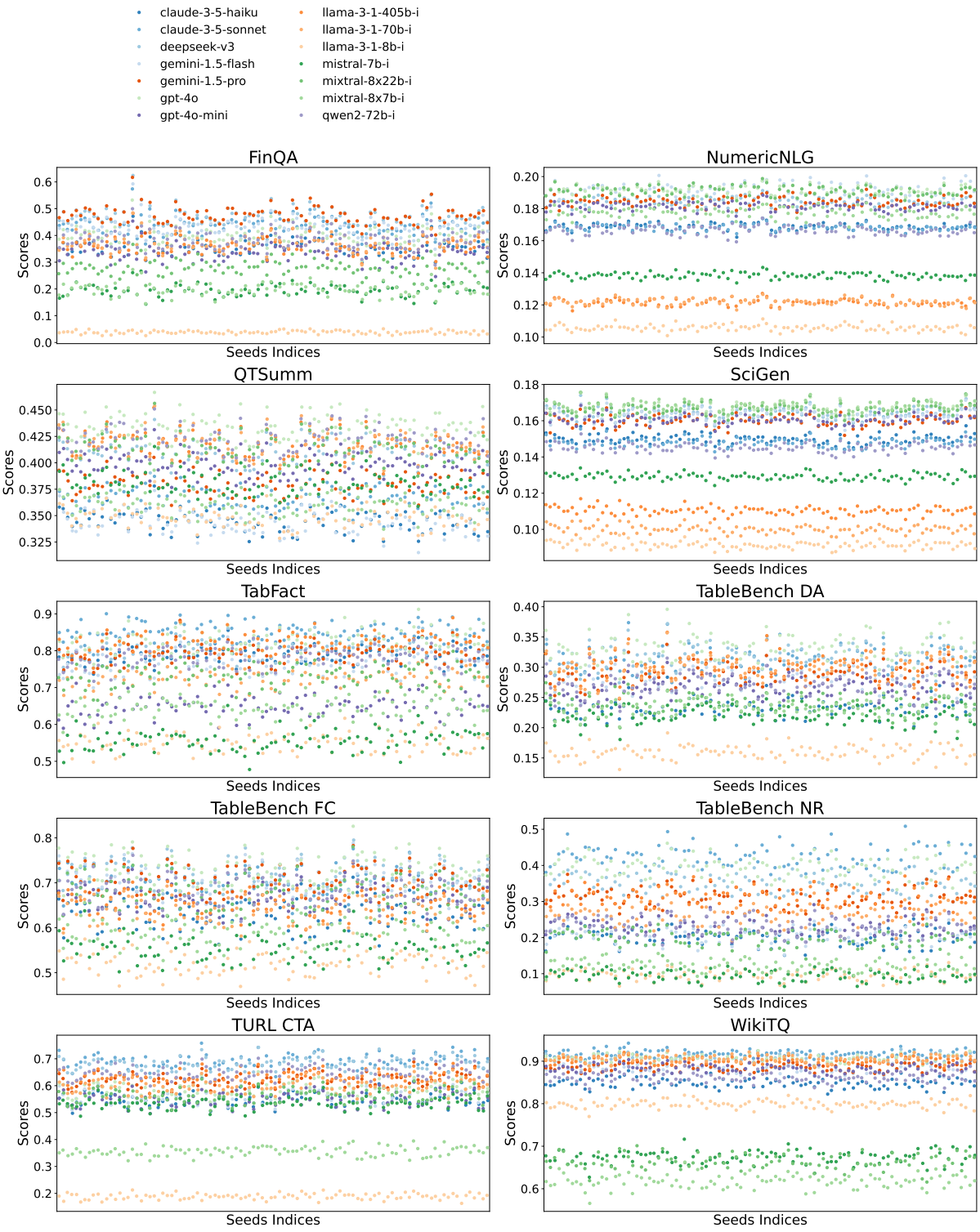


Figure 18: Each graph shows the score distribution and variability across models for a given dataset, based on bootstrapping with different seeds. This visualization supports the separability analysis, as it highlights how distinctly model scores spread within each dataset.

H Related Work

Benchmark	Focus Area	Focus Tasks	Task Diversity	Format Robustness	Model Types	Key Uniqueness
ToRR (ours)	Table Reasoning and Robustness Analysis	Table2Text, Summarization, Column Annotation, QA (Fact Checking, Numerical Reasoning, Data Analysis)	High – 6 tasks, 10 datasets	Yes – Multiple formats, perturbations	General LLMs	Robustness benchmark for diverse table reasoning tasks
TableBench (Wu et al., 2024)	Table Question Answering	QA only (Fact Checking, Numerical Reasoning, Data Analysis, Visualization)	Moderate	No – Fixed formats	General LLMs	Performance ranking on QA
InfiAgent-DABench	Data Analysis with Agents	Data analysis questions	Narrow	No – Agent execution focus	Agents + LLMs with tools	Agent planning for data analysis
TableVQA-Bench	Table Visual Question Answering	Visual table QA (images)	Narrow	No – Image-based format	Vision-Language Models	Vision-language reasoning
DataBench (Grijalba et al., 2024)	Table Question Answering	Table QA	Narrow	No – Fixed representation	General LLMs	QA performance evaluation
TQA-Bench (Qiu et al., 2024)	Multi-table QA over relational data	Table QA	Narrow	Limited – 2 formats	General LLMs	QA over interconnected relational tables

Table 8: Comparison of Table Reasoning Benchmarks.

I ToRR Properties

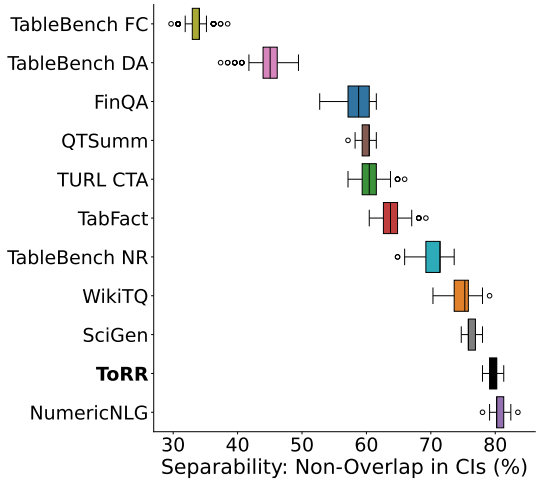


Figure 19: The separability score for each dataset in ToRR. This score represents the proportion of model pairs that can be distinguished with confidence, meaning their confidence intervals (CIs; via bootstrapping over 1K seeds) do not overlap.

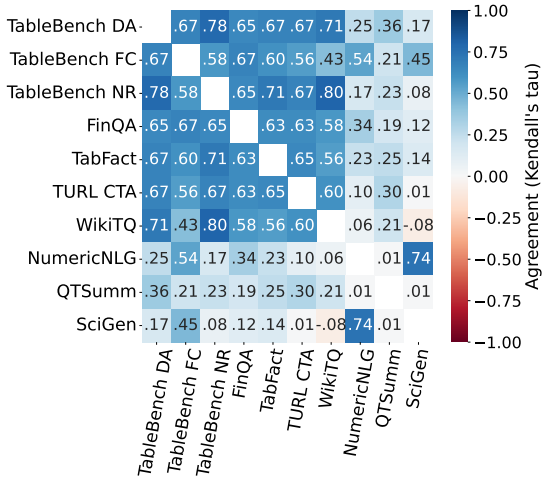


Figure 20: Model ranking agreement between the datasets in ToRR.

J Usage in AI

In this work, we used AI models exclusively for language-related tasks, such as rephrasing and surface-level linguistic transformations.

K Prompt Examples

Here we provide examples of the prompts used in our benchmark. While the structure of each prompt includes the task instructions with highlights, five demonstrations with expected answers, and an input example (as illustrated in Figure 21), we present them with a single demonstration for each task in Figures 22, 23, 24, 25, 26, 27 and 28.

[Instruction]
Given a Table and Statement classify the entailment of the Statement to one of refuted, entailed.
Output only the final answer without any explanations, extra information, or introductory text.
Here are some input-output examples. Read the examples carefully to figure out the mapping. The output of the last example is not given, and your job is to figure out what it is.

[Demos]
Table: date,result,score,brazil scorers,competition
"may 11 , 1919",w,6 – 0,"friedenreich (3) , neco (2) , haroldo",so
uth american championship
"may 18 , 1919",w,3 – 1,"heitor , amílcar , millon",south american
championship
"may 26 , 1919",d,2 – 2,neco (2),south american championship
"may 29 , 1919",w,1 – 0,friedenreich,south american championship
"june 1 , 1919",d,3 – 3,"haroldo , arlindo (2)",taça roberto cherry
Statement: friedenreich be mention as a brazil scorer for 4 different game
refuted

... [4 more demos]

[Example]
Table: tournament,wins,top – 5,top – 10,top – 25,events,cuts made
masters tournament,0,1,2,4,4,4
us open,0,2,3,4,6,5
the open championship,1,2,2,2,3,3
pga championship,0,0,1,2,5,4
totals,1,5,8,12,18,16
Statement: tony lema make it to the top 10 in the pga championship, but do not continue on

Figure 21: A prompt example from our benchmark, demonstrating the template and instructions used for *TabFact*. The same prompt structure was applied across all datasets, with specific instructions tailored for each.

```
    Presented with a financial report consisting of textual contents and a structured table, given a
question, generate the reasoning program in the domain specific language (DSL) that will be executed to
get the answer.
    The DSL consists of mathematical operations and table operations as executable programs. The program
consists of a sequence of operations. Each operation takes a list of arguments.
    There are 6 mathematical operations: add, subtract, multiply, divide, greater, exp, and 4 table
aggregation operations table-max, table-min, table-sum, table-average, that apply aggregation operations
on table rows. The mathemati
    cal operations take arguments of either numbers from the given reports, or a numerical result from a
previous step.
    The table operations take arguments of table row names. We use the special token #n to denote the
result from the nth step.
    For example, in the example "divide(9413, 20.01), divide(8249, 9.48), subtract(#0, #1)", the program
consists of 3 steps; The first and the second division steps take arguments from the table and the text,
respectively, then the
    third step subtracts the results from the two previous steps.
                    Definitions of all operations:
                    [["Name", "Arguments", "Output", "Description"],
                    ["add", "number1, number2", "number", "add two numbers: number1 + number2"],
                    ["subtract", "number1, number2", "number", "subtract two numbers: number1 - number2"],
                    ["multiply", "number1, number2", "number", "multiply two numbers: number1 * number2"],
                    ["divide", "number1, number2", "number", "multiply two numbers: number1 / number2"],
                    ["exp", "number1, number2", "number", "exponential: number1 ^ number2"],
                    ["greater", "number1, number2", "bool", "comparison: number1 > number2"],
                    ["table-sum", "table header", "number", "the summation of one table row"],
                    ["table-average", "table header", "number", "the average of one table row"],
                    ["table-max", "table header", "number", "the maximum number of one table row"],
                    ["table-min", "table header", "number", "the minimum number of one table row"]]

    Answer with only the program, without any additional explanation or introductory text.

    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output
of the last example is not given, and your job is to figure out what it is.

    Pre-table text: notes to consolidated financial statements 2013 ( continued ) ( amounts in millions ,
except per share amounts ) guarantees we have guarantees of certain obligations of our subsidiaries
relating principally to cre
    dit facilities , certain media payables and operating leases of certain subsidiaries .
                    Table: ,2010,2011,2012,2013,2014,thereafter,total
    deferred acquisition payments,$ 20.5,$ 34.8,$ 1.2,$ 1.1,$ 2.1,$ 0.3,$ 60.0
    redeemable noncontrolling interests and call options with
affiliates1,44.4,47.9,40.5,36.3,3.3,2014,172.4
    total contingent acquisition payments,64.9,82.7,41.7,37.4,5.4,0.3,232.4
    less : cash compensation expense included above,1.0,1.0,1.0,0.5,2014,2014,3.5
    total,$ 63.9,$ 81.7,$ 40.7,$ 36.9,$ 5.4,$ 0.3,$ 228.9
                    Post-table text: 1 we have entered into certain acquisitions that contain both
redeemable noncontrolling interests and call options with similar terms and conditions .
                    Question: what percentage decrease occurred from 2011-2012 for deferred acquisition
payments?
                    Program:

    subtract(34.8, 1.2), divide(#0, 34.8), multiply(#1, const_100)

    Pre-table text: performance graph the table below compares the cumulative total shareholder return on
our common stock with the cumulative total return of ( i ) the standard & poor's 500 composite stock index
( "s&p 500 index" )
    , ( ii ) the standard & poor's industrials index ( "s&p industrials index" ) and ( iii ) the standard
& poor's consumer durables & apparel index ( "s&p consumer durables & apparel index" ) , from december 31
, 2012 through decemb
    er 31 , 2017 , when the closing price of our common stock was $ 43.94 .
                    Table: ,2013,2014,2015,2016,2017
    masco,$ 138.48,$ 155.26,$ 200.79,$ 227.08,$ 318.46
    s&p 500 index,$ 132.04,$ 149.89,$ 151.94,$ 169.82,$ 206.49
    s&p industrials index,$ 140.18,$ 153.73,$ 149.83,$ 177.65,$ 214.55
    s&p consumer durables & apparel index,$ 135.84,$ 148.31,$ 147.23,$ 138.82,$ 164.39
                    Post-table text: $ 50.00 $ 100.00 $ 150.00 $ 200.00 $ 250.00 $ 300.00 $ 350.00 masco
s&p 500 index s&p industrials index s&p consumer durables & apparel index .
                    Question: what was the difference in percentage cumulative total shareholder return on
masco common stock versus the s&p 500 index for the five year period ended 2017?
                    Program:
```

Figure 22: Example prompt used in FinQA. This single-shot prompt includes one demonstration that reflects both the input format and the expected output.

```
    You are a table analyst. Your task is to answer questions based on the table content. The answer
should follow the format below:
    [Answer Format]
    Final Answer: AnswerName1, AnswerName2...

    Ensure the final answer format is the last output line and can only be in the "Final Answer:
AnswerName1, AnswerName2..." form, no other form. Ensure the "AnswerName" is a entity name or a impact
description(No clear impact, Negt
    ive impact or Positive impact), as short as possible, without any explanation.
    Output only the final answer without any explanations, extra information, or introductory text.
    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output
of the last example is not given, and your job is to figure out what it is.
    Table: year,number of tropical storms,number of hurricanes,number of major hurricanes,deaths,strongest
storm
    1850,0,3,0,not known,one
    1851,6,3,1,24,four
    1852,5,5,1,100 +,one
    1853,8,4,2,40,three
    1854,5,3,1,30 +,three
    1855,5,4,1,not known,five
    1856,6,4,2,200 +,one
    1857,4,3,0,424,two & four
    1858,6,6,0,none,three & six
    Question: Does an increase in the number of major hurricanes cause an increase in the number of
deaths?
    Final Answer: No, causal analysis indicates a strong negative correlation (-0.84), suggesting an
increase in major hurricanes does not causally lead to an increase in deaths.

    Table: township,county,pop (2010),land ( sqmi ),water (sqmi),latitude,longitude,geo id,ansi code
    oak creek,bottineau,24,35.445,0.0,48.675399,- 100.471642,3800958700,1759286
    oak valley,bottineau,52,36.016,0.087,48.777318,- 100.511814,3800958860,1759287
    oakhill,barnes,51,35.414,0.081,46.679076,- 98.017963,3800358780,1036402
    oakland,mountrail,26,35.167,0.785,48.157497,- 102.109269,3806158820,1036997
    oakville,grand forks,200,35.059,0.047,47.883391,- 97.305536,3803558900,1036604
    oakwood,walsh,228,33.526,0.0,48.412107,- 97.339101,3809958980,1036534
    oberon,benson,67,57.388,0.522,47.925443,- 99.244476,3800559060,2397849
    odessa,hettinger,16,35.766,0.06,46.583226,- 102.104455,3804159100,1759459
    odessa,ramsey,49,37.897,8.314,47.968754,- 98.587529,3807159140,1759587
    odin,mchenry,46,34.424,1.722,47.986751,- 100.637016,3804959180,1759507
    oliver,williams,8,35.987,0.024,48.423293,- 103.320183,3810559260,1037033
    olivia,mchenry,40,35.874,0.035,47.900358,- 100.769959,3804959300,1759508
    olson,towner,19,35.033,0.954,48.505811,- 99.287008,3809559380,1759659
    ontario,ramsey,72,33.923,1.99,48.163172,- 98.601321,3807159460,1759588
    ops,walsh,63,36.015,0.0,48.238231,- 97.578927,3809959540,1036518
    ora,nelson,69,34.414,0.697,47.722982,- 97.946877,3806359580,1036557
    orange,adams,22,35.802,0.133,46.012558,- 102.053893,3800159620,1037214
    oriska,barnes,65,35.082,0.087,46.935397,- 97.752733,3800359700,1036418
    orlien,ward,47,35.645,0.72,47.985154,- 101.796936,3810159740,1036954
    orthell,williams,12,35.894,0.034,48.495353,- 103.728983,3810559860,1759732
    osago,nelson,31,35.4,0.198,47.800898,- 98.328474,3806359900,1036565
    osborn,mountrail,285,30.296,4.988,47.987208,- 102.429987,3806159940,1034001
    osford,cavalier,47,35.803,0.052,48.585234,- 98.115821,3801959980,1759377
    oshkosh,wells,56,34.747,0.065,47.623026,- 99.576942,3810360020,1759708
    osloe,mountrail,41,35.077,0.903,48.146259,- 101.976499,3806160060,1036937
    osnabrock,cavalier,36,35.505,0.439,48.594234,- 98.241946,3801960140,2397851
    ostby,bottineau,45,35.452,0.027,48.581052,- 100.352948,3800960180,1759288
    otis,mclean,41,35.152,0.656,47.799001,- 100.896513,3805560260,1759541
    overland,ramsey,14,35.602,0.4,48.406215,- 98.644574,3807160340,1759589
    ovid,lamoure,46,35.328,0.505,46.318992,- 98.107769,3804560420,1036886
    owego,ransom,21,36.034,0.029,46.50933,- 97.319286,3807360460,1036866
    Question: How does the latitude of a township impact its population density?
    Final Answer:
```

Figure 23: Example prompt used in TableBench datasets (DA, NR and FC). This single-shot prompt includes one demonstration that reflects both the input format and the expected output for TableBench DA.

```
    Answer the question based on the provided table. Extract and output only the final answer—the exact
phrase or data from the table that directly answers the question. Do not include any alterations,
explanations, or introductory text.
    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output
of the last example is not given, and your job is to figure out what it is.

    Question: how many times did an italian cyclist win a round?
    Table: Round,Round,Circuit,Date,Pole Position,Fastest Lap,Winning Rider
    1,R1,Jerez,18 March,Raymond Roche,Stéphane Mertens,Raymond Roche
    1,R2,Jerez,18 March,Raymond Roche,Raymond Roche,Raymond Roche
    2,R1,Donington,16 April,Giancarlo Falappa,Rob Phillis,Fred Merkel
    2,R2,Donington,16 April,Giancarlo Falappa,Raymond Roche,Giancarlo Falappa
    3,R1,Hungaroring,30 April,Malcolm Campbell,Raymond Roche,Fred Merkel
    3,R2,Hungaroring,30 April,Malcolm Campbell,Fred Merkel,Raymond Roche
    4,R1,Hockenheim,6 May,Raymond Roche,Fred Merkel,Fred Merkel
    4,R2,Hockenheim,6 May,Raymond Roche,Raymond Roche,Stéphane Mertens
    5,R1,Mosport,3 June,Giancarlo Falappa,Raymond Roche,Raymond Roche
    5,R2,Mosport,3 June,Giancarlo Falappa,Jamie James,Raymond Roche
    6,R1,Brainerd,10 June,Doug Chandler,Doug Chandler,Stéphane Mertens
    6,R2,Brainerd,10 June,Doug Chandler,Fabrizio Pirovano,Doug Chandler
    7,R1,Österreichring,1 July,Stéphane Mertens,Rob McElnea,Fabrizio Pirovano
    7,R2,Österreichring,1 July,Stéphane Mertens,Stéphane Mertens,Stéphane Mertens
    8,R1,Sugo,26 August,Raymond Roche,Raymond Roche,Raymond Roche
    8,R2,Sugo,26 August,Raymond Roche,Peter Goddard,Doug Chandler
    9,R1,Le Mans,9 September,Baldassarre Monti,Raymond Roche,Raymond Roche
    9,R2,Le Mans,9 September,Baldassarre Monti,Jamie James,Raymond Roche
    10,R1,Monza,7 October,Baldassarre Monti,Rob Phillis,Fabrizio Pirovano
    10,R2,Monza,7 October,Baldassarre Monti,Rob Phillis,Fabrizio Pirovano
    11,R1,Shah Alam,4 November,Rob Phillis,Fabrizio Pirovano,Fabrizio Pirovano
    11,R2,Shah Alam,4 November,Rob Phillis,Raymond Roche,Fabrizio Pirovano
    12,R1,Phillip Island,11 November,Peter Goddard,Fabrizio Pirovano,Peter Goddard
    12,R2,Phillip Island,11 November,Peter Goddard,Malcolm Campbell,Rob Phillis
    13,R1,Manfeild,18 November,Rob Phillis,Brian Morrison,Terry Rymer
    13,R2,Manfeild,18 November,Rob Phillis,Raymond Roche,Rob Phillis
    Answer:
    6


    Question: what is the number of 1st place finishes across all events?
    Table: Date,Competition,Location,Country,Event,Placing,Rider,Nationality
    31 October 2008,2008–09 World Cup,Manchester,United Kingdom,Sprint,1,Victoria Pendleton,GBR
    31 October 2008,2008–09 World Cup,Manchester,United Kingdom,Keirin,2,Jason Kenny,GBR
    1 November 2008,2008–09 World Cup,Manchester,United Kingdom,Sprint,1,Jason Kenny,GBR
    1 November 2008,2008–09 World Cup,Manchester,United Kingdom,500 m time trial,1,Victoria Pendleton,GBR
    2 November 2008,2008–09 World Cup,Manchester,United Kingdom,Team sprint,1,Ross Edgar,GBR
    2 November 2008,2008–09 World Cup,Manchester,United Kingdom,Team sprint,1,Jason Kenny,GBR
    2 November 2008,2008–09 World Cup,Manchester,United Kingdom,Team sprint,1,Jamie Staff,GBR
    2 November 2008,2008–09 World Cup,Manchester,United Kingdom,Keirin,1,Victoria Pendleton,GBR
    2 November 2008,5th International Keirin Event,Manchester,United Kingdom,International keirin,2,Ross
Edgar,GBR
    13 February 2009,2008–09 World Cup,Copenhagen,Denmark,Team sprint,1,Chris Hoy,GBR
    13 February 2009,2008–09 World Cup,Copenhagen,Denmark,Team sprint,1,Jason Kenny,GBR
    13 February 2009,2008–09 World Cup,Copenhagen,Denmark,Team sprint,1,Jamie Staff,GBR
    13 February 2009,2008–09 World Cup,Copenhagen,Denmark,Sprint,1,Victoria Pendleton,GBR
    30 October 2009,2009–10 World Cup,Manchester,United Kingdom,Keirin,1,Chris Hoy,GBR
    30 October 2009,2009–10 World Cup,Manchester,United Kingdom,Sprint,1,Victoria Pendleton,GBR
    30 October 2009,2009–10 World Cup,Manchester,United Kingdom,Sprint,1,Chris Hoy,GBR
    30 October 2009,2009–10 World Cup,Manchester,United Kingdom,500 m time trial,2,Victoria Pendleton,GBR
    1 November 2009,2009–10 World Cup,Manchester,United Kingdom,Team sprint,1,Ross Edgar,GBR
    1 November 2009,2009–10 World Cup,Manchester,United Kingdom,Team sprint,1,Chris Hoy,GBR
    1 November 2009,2009–10 World Cup,Manchester,United Kingdom,Team sprint,1,Jamie Staff,GBR
    Answer:
```

Figure 24: Example prompt used in WikiTQ. This single-shot prompt includes one demonstration that reflects both the input format and the expected output.

```
    Given a Table and Statement classify the entailment of the Statement to one of refuted, entailed.
    Output only the final answer without any explanations, extra information, or introductory text.
    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output
of the last example is not given, and your job is to figure out what it is.
    Table: no in series,no in season,title,directed by,written by,original air date,production code,us
viewers (millions)
    89,1,"4 years , 6 months , 2 days",greg prange,mark schwahn,"january 8 , 2008",3t6801,3.36
    90,2,racing like a pro,paul johansson,mark schwahn,"january 8 , 2008",3t6802,3.57
    91,3,my way home is through you,david jackson,john a norris,"january 15 , 2008",3t6803,2.72
    92,4,"it 's alright , ma (i'm only bleeding)",janice cooke,adele lim,"january 22 , 2008",3t6804,3.04
    93,5,i forgot to remember to forget,liz friedlander,terrence coli,"january 29 , 2008",3t6805,2.79
    94,6,don't dream it 's over,thomas j wright,mark schwahn,"february 5 , 2008",3t6806,2.86
    95,7,in da club,greg prange,mike herro and david strauss,"february 12 , 2008",3t6807,3.16
    96,8,please please please let me get what i want,paul johansson,mike daniels,"february 19 ,
2008",3t6808,2.85
    97,9,for tonight you 're only here to know,joe davola,mark schwahn,"february 26 , 2008",3t6809,3.18
    98,10,running to stand still,clark mathis,william h brown,"march 4 , 2008",3t6810,2.93
    99,11,you 're gonna need someone on your side,michael j leone,zachary haynes,"march 11 ,
2008",3t6811,2.50
    100,12,hundred,les butler,mark schwahn,"march 18 , 2008",3t6812,3.00
    101,13,"echoes , silence , patience , and grace",greg prange,mark schwahn,"april 14 ,
2008",3t6813,2.80
    102,14,what do you go home to,liz friedlander,mark schwahn,"april 21 , 2008",3t6814,2.92
    103,15,life is short,paul johansson,eliza delson,"april 28 , 2008",3t6815,2.57
    104,16,cryin' won't help you now,greg prange,william h brown,"may 5 , 2008",3t6816,2.29
    105,17,hate is safer than love,stuart gillard,mark schwahn,"may 12 , 2008",3t6817,2.72
    Statement: racing like a pro be the most viewed episode
    entailed

    Table: tournament,wins,top - 5,top - 10,top - 25,events,cuts made
    masters tournament,0,1,2,4,4,4
    us open,0,2,3,4,6,5
    the open championship,1,2,2,2,3,3
    pga championship,0,0,1,2,5,4
    totals,1,5,8,12,18,16
    Statement: tournament that tony lema have participate in include the master tournament , the us open ,
the pga championship and the open championship
```

Figure 25: Example prompt used in TabFact. This single-shot prompt includes one demonstration that reflects both the input format and the expected output.

```
    Using the information from the Table given below, summarize a paragraph-long response to the following
user query.
    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output
of the last example is not given, and your job is to figure out what it is.
    Table:
    No.,Event,Date,Venue,Location,Attendance
    45,WEC 45: Cerrone vs. Ratcliff,"December 19, 2009",Pearl at The Palms,"Las Vegas, Nevada","1,741"
    44,WEC 44: Brown vs. Aldo,"November 18, 2009",Pearl at The Palms,"Las Vegas, Nevada","1,835"
    43,WEC 43: Cerrone vs. Henderson,"October 10, 2009",AT&T Center,"San Antonio, Texas","5,176"
    42,WEC 42: Torres vs. Bowles,"August 9, 2009",Hard Rock Hotel and Casino,"Las Vegas, Nevada","2,082"
    41,WEC 41: Brown vs. Faber II,"June 7, 2009",ARCO Arena,"Sacramento, California","13,027"
    40,WEC 40: Torres vs. Mizugaki,"April 5, 2009",UIC Pavilion,"Chicago, Illinois","5,257"
    39,WEC 39: Brown vs. Garcia,"March 1, 2009",American Bank Center,"Corpus Christi, Texas","6,100"
    38,WEC 38: Varner vs. Cerrone,"January 25, 2009",San Diego Sports Arena,"San Diego,
California","10,201"
    Query:
    What was the range of attendances seen at events at The Pearl at The Palms venue in 2009?
    Answer:
    the range of attendances seen at events at The Pearl at The Palms venue in 2009 was from 1,741 to
13,027.

    Table:
    No.,Album,Artist,Released,Chart,Sales
    1,First Love,Hikaru Utada,10 March 1999,1,"7,672,000"
    2,"B'z The Best ""Pleasure""",B'z,20 May 1998,1,"5,136,000"
    3,Review,Glay,1 October 1997,1,"4,876,000"
    4,Distance,Hikaru Utada,28 March 2001,1,"4,472,000"
    5,"B'z The Best ""Treasure""",B'z,20 September 1998,1,"4,439,000"
    6,A Best,Ayumi Hamasaki,28 March 2001,1,"4,312,000"
    7,Globe,Globe,31 March 1996,1,"4,136,000"
    8,Deep River,Hikaru Utada,19 June 2002,1,"3,605,000"
    9,Umi no Yeah!!,Southern All Stars,25 June 1998,1,"3,592,000"
    10,Delicious Way,Mai Kuraki,28 June 2000,1,"3,530,000"
    11,Time to Destination,Every Little Thing,15 April 1998,1,"3,520,000"
    12,Atomic Heart,Mr. Children,1 September 1994,1,"3,430,000"
    13,Sweet 19 Blues,Namie Amuro,22 July 1996,1,"3,359,000"
    14,Bolero,Mr. Children,5 March 1997,1,"3,283,000"
    15,Neue Musik,Yumi Matsutoya,6 November 1998,1,"3,252,000"
    16,Faces Places,Globe,12 March 1997,1,"3,239,000"
    17,The Swinging Star,Dreams Come True,14 November 1992,1,"3,227,000"
    18,Impressions,Mariya Takeuchi,25 July 1994,1,"3,067,000"
    19,Zard Best the Single Collection ～軌跡～,Zard,28 May 1999,1,"3,034,000"
    20,All Singles Best,Kobukuro,27 September 2006,1,"3,018,000"
    Query:
    What are the sales numbers of the top three best-selling albums by domestic acts in Japan and who are
the artists?
    Answer:
```

Figure 26: Example prompt used in QTSumm. This single-shot prompt includes one demonstration that reflects both the input format and the expected output.

```
    Given the following table and caption, generate the corresponding text description.
    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output
of the last example is not given, and your job is to figure out what it is.
    table:
    Cue,[ITALIC] SCOPA,[ITALIC] SB_COPA,Diff.,Prod.
    woman,7.98,4.84,-3.14,0.25
    mother,5.16,3.95,-1.21,0.75
    went,6.00,5.15,-0.85,0.73
    down,5.52,4.93,-0.58,0.71
    into,4.07,3.51,-0.56,0.40
    caption:
    Table 7: Sensitivity of BERT-large to superficial cues identified in §2 (unit: 10-2). Cues with top-5
reduction are shown. SCOPA,SB_COPA indicate the mean contributions of BERT-large trained on COPA, and
BERT-large trained on B-C
    OPA, respectively.
    text description:
    We observe that BERT trained on Balanced COPA is less sensitive to a few highly productive superficial
cues than BERT trained on original COPA. Note the decrease in the sensitivity for cues of productivity
from 0.7 to 0.9. These
    cues are shown in Table 7. However, for cues with lower productivity, the picture is less clear, in
case of RoBERTa, there are no noticeable trends in the change of sensitivity.

    table:
    [BOLD] Concept Input →  [BOLD] Embeddings,[BOLD] Concept Input →  [BOLD] TF,[BOLD] Concept Input
→  [BOLD] IDF,[BOLD] Label  [BOLD] T,[BOLD] Label  [BOLD] P,[BOLD] Label  [BOLD] R,[BOLD] Label  [BOLD]
F,[BOLD] Description  [BOLD]
     T,[BOLD] Description  [BOLD] P,[BOLD] Description  [BOLD] R,[BOLD] Description  [BOLD] F,[BOLD]
Both  [BOLD] T,[BOLD] Both  [BOLD] P,[BOLD] Both  [BOLD] R,[BOLD] Both  [BOLD] F
    [BOLD] GloVe,[BOLD] −,[BOLD] −,.635,.750,.818,.783,.720,.754,.891,.817,.735,.765,.945,.846
    [BOLD] GloVe,[BOLD] +,[BOLD] −,.640,.891,.745,.812,.700,.831,.891,.860,.690,.813,.945,.874
    [BOLD] GloVe,[BOLD] −,[BOLD] +,.600,.738,.873,.800,.670,.746,.909,.820,.755,.865,.818,.841
    [BOLD] GloVe,[BOLD] +,[BOLD] +,.605,.904,.855,.879,.665,.857,.873,.865,.715,.923,.873,.897
    [BOLD] Google,[BOLD] −,[BOLD] −,.440,.813,.945,.874,.515,.701,.982,.818,.635,.920,.836,.876
    [BOLD] Google,[BOLD] +,[BOLD] −,.445,.943,.909,[BOLD] .926,.540,.873,.873,.873,.565,.927,.927,.927
    [BOLD] Google,[BOLD] −,[BOLD] +,.435,.839,.945,.889,.520,.732,.945,.825,.590,.877,.909,.893
    [BOLD] Google,[BOLD] +,[BOLD] +,.430,.943,.909,[BOLD] .926,.530,.889,.873,[BOLD]
.881,.545,.945,.945,[BOLD] .945
    [BOLD] fastText,[BOLD] −,[BOLD] −,.440,.781,.909,.840,.555,.708,.927,.803,.615,.778,.891,.831
    [BOLD] fastText,[BOLD] +,[BOLD] −,.435,.850,.927,.887,.520,.781,.909,.840,.530,.803,.964,.876
    [BOLD] fastText,[BOLD] −,[BOLD] +,.435,.850,.927,.887,.525,.722,.945,.819,.600,.820,.909,.862
    [BOLD] fastText,[BOLD] +,[BOLD] +,.420,.895,.927,.911,.505,.803,.891,.845,.520,.833,.909,.870
    caption:
    Table 1: Tuning Data Results AVG_COS_SIM. Top F per Concept Input Type in Bold.
    text description:
```

Figure 27: Example prompt used in Table-to-Text datasets (NumericNLG and SciGen). This single-shot prompt includes one demonstration that reflects both the input format and the expected output for SciGen.

```
This is a column type annotation task. The goal of this task is to choose the correct types for one selected
column of the given input table from the given candidate types. The Wikipedia page, section and table caption (if
any) provide important information for choosing the correct column types.
     Candidate Types: royalty.noble_person, business.business_operation, protected_sites.listed_site,
music.writer, people.ethnicity, government.government_office_or_title, organization.non_profit_organization,
business.brand, tennis.tennis_tournament, cvg.cvg_genre, ice_hockey.hockey_position, sports.sports_team,
computer.computer, metropolitan_transit.transit_line, award.award_category,
american_football.football_conference, sports.professional_sports_team, soccer.football_world_cup, tv.tv_actor,
business.industry, music.composition, people.person, broadcast.tv_channel, cricket.cricket_player,
internet.website, tennis.tennis_player, music.media_format, tv.tv_personality, film.actor, film.film_genre,
cvg.cvg_developer, business.job_title, chess.chess_player, tv.tv_writer, broadcast.broadcast, soccer.fifa,
cvg.cvg_publisher, film.writer, medicine.anatomical_structure, astronomy.celestial_object, cricket.cricket_team,
sports.golfer, book.periodical_subject, military.rank, spaceflight.astronaut, medicine.disease,
location.province, location.location, amusement_parks.ride, government.general_election, music.musical_
scale, music.lyricist, music.artist, location.capital_of_administrative_division, theater.play,
meteorology.tropical_cyclone, aviation.airport, basketball.basketball_team, education.school,
soccer.football_position, soccer.football_team, cvg.cvg_platform, religion.religious_leader,
business.defunct_company, astronomy.asteroid, sports.pro_athlete, sports.school_sports_team,
baseball.baseball_league, architecture.structure, sports.tournament_event_competition,
sports.multi_event_tournament, music.record_label, travel.accommodation, cricket.cricket_stadium,
ice_hockey.hockey_team, award.competition, business.consumer_company, people.family_member,
biology.organism_classification, business.product_category, book.magazine, royalty.kingdom,
...,
location.australian_local_government_area, theater.theater_actor, music.producer, ice_hockey.hockey_player,
royalty.monarch, sports.sports_championship_event, sports.sports_league_draft, food.food,
military.military_person, geography.island, location.uk_constituent_country, tv.tv_series_episode,
government.u_s_congressperson, amusement_parks.park, book.written_work, geography.body_of_water, tv.tv_genre,
aviation.aircraft_owner, interests.collection_category, astronomy.star_system_body, tv.tv_producer,
medicine.muscle, baseball.baseball_team, government.us_president, location.citytown,
fictional_universe.fictional_organization, biology.organism, tv.tv_program, soccer.football_league_season,
sports.boxer, military.armed_force, location.australian_state, basketball.basketball_conference,
internet.website_owner, medicine.drug, award.award_discipline, location.in_district, business.consumer_product,
broadcast.radio_format, baseball.baseball_position, book.periodical_publisher, government.government_agency,
sports.cyclist, time.event, automotive.model, boats.ship_type, finance.currency, government.legislative_session,
american_football.football_player, royalty.chivalric_order_member, law.invention, martial_arts.martial_artist,
film.film_character, sports.sports_facility, music.group_member, location.region, astronomy.orbital_relationship,
basketball.basketball_player, cvg.computer_videogame, law.legal_case, language.human_language, tv.tv_character,
education.educational_degree, aviation.aircraft_model, business.customer, geography.mountain, location.us_county,
music.album, music.composer, computer.operating_system, religion.religion, organization.membership_organization,
sports.sport, location.uk_statistical_location, location.in_state, film.film_distributor,
basketball.basketball_coach, medicine.medical_treatment, education.fraternity_sorority,
metropolitan_transit.transit_stop, chemistry.chemical_compound, sports.sports_position, music.genre,
award.hall_of_fame_inductee, sports.sports_award_type, exhibitions.exhibition_sponsor, film.film_festival_focus,
film.production_company, location.jp_prefecture, education.field_of_study, award.recurring_competition,
government.election_campaign, sports.sports_award_winner, astronomy.astronomical_discovery,
music.performance_role, soccer.football_league, book.author, film.producer, royalty.noble_title, biology.animal,
american_football.football_team, baseball.baseball_player

    Output only the correct column types from the candidate list for the mentioned columns. Do not include any
explanations, extra information, or introductory text—only the final answer.

    Here are some input-output examples. Read the examples carefully to figure out the mapping. The output of the
last example is not given, and your job is to figure out what it is.

    Column name: athlete
    Page Title: athletics at the 1936 summer olympics – men's high jump
    Section Title: qualifying
    Table caption:
    Table:
    athlete
    Dave Albritton
    Günther Gehmert
    Jerzy Pławczyk
    Kimio Yada
    Mihály Bodosi
    Veikko Peräsalo
    Selected Column: athlete
    people.person


    Column name: election year
    Page Title: kagawa at-large district (house of councillors)
    Section Title: elected councillors
    Table caption:
    Table:
    election year
    1947
    1950
    1953
    1956
    1959
    1962
    1965
    Selected Column: election year
```

Figure 28: Example prompt used in TURL CTA. This single-shot prompt includes one demonstration that reflects both the input format and the expected output.