Uncovering Limitations of Large Language Models in Information Seeking from Tables

Anonymous EMNLP submission

Abstract

Tables are recognized for their high information density and widespread usage, serving as essential sources of information. Seeking information from tables (TIS) is a crucial capability for Large Language Models (LLMs), serving as the foundation of knowledge-based Q&A systems. However, this field presently suffers from an absence of thorough and reliable evaluation. This paper introduces a more reliable benchmark for **Ta**ble **I**nformation **S**eeking (TabIS). To avoid the unreliable evaluation caused by text similarity-based metrics, TabIS adopts a single-choice question format (with two options per question) instead of a text generation format. We establish an effective pipeline for generating options, ensuring their difficulty and quality. Experiments conducted on 12 LLMs reveal that while the performance of GPT-4-turbo is marginally satisfactory, both other proprietary and open-source models perform inadequately. Further analysis shows that LLMs exhibit a poor understanding of table structures, and struggle to balance between TIS performance and robustness against pseudo-relevant tables (common in retrieval-augmented systems). These findings uncover the limitations and potential challenges of LLMs in seeking information from tables. We release our data and code to facilitate further research in this field.

1 Introduction

Tables are widespread and rich sources of information across the web and in various documents. Statistics show that the number of tables on internet web pages has reached several hundred million (Lehmberg et al., 2016); in the corporate environment, the number of tables in Excel-like spreadsheet files has exceeded 115 million (Wang et al., 2020). Precisely seeking relevant information from tables is crucial for a wide array of real-world applications, including financial analysis, scientific research, etc. Recently, the remarkable advancements of Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2023a; Touvron et al., 2023; Google, 2023) have transformed the approach of information retrieval, moving from fetching specific passages to directly providing answers. However, the effectiveness of LLMs in seeking information from tables remains underexplored.

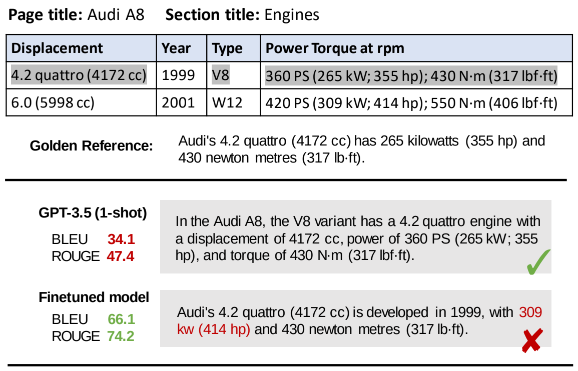

Figure 1: A table-to-text generation example (simplified) to show the unreliable evaluation issue: higher values on surface-level metrics like BLEU and ROUGE do not guarantee better results. Target cells are highlighted.

Some efforts have been made to evaluate the capabilities of LLMs in Table Information Seeking (TIS), but there are unreliable evaluation issues with the used evaluation metrics. Previous studies (Zhao et al., 2023b) mainly use table-to-text generation (TTG) as a test bench to assess the TIS abilities of LLMs. TTG aims at transforming complex tabular data into comprehensible descriptions tailored to users' information seeking needs. The Evaluation relies heavily on surface-level metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), or on metrics based on model predictions such as NLI-Acc (Chen et al., 2020a). Given that LLM responses can greatly differ in style from the reference answers, using these metrics can lead to inconsistent and unreliable evaluations. An example of this issue is illustrated in Figure 1 where a fine-tuned model's incorrect description receives higher BLEU/ROUGE scores than the correct output from GPT-3.5. This discrepancy may occur because GPT-3.5, without being fine-tuned on this specific dataset, might not mimic the style of the reference response.

To provide a more reliable evaluation, this paper introduces a new benchmark for **Ta**ble **I**nformation **S**eeking (TabIS). We design our benchmark using a single-choice question format, motivated by popular benchmarks like MMLU (Hendrycks et al., 2020) and BBH (Suzgun et al., 2022), which utilize this format to offer a reliable and widely accepted evaluation of LLMs. We convert TTG datasets like ToTTo (Parikh et al., 2020) and Hitab (Cheng et al., 2022) into this format so that the results can be simply and reliably evaluated. A challenge during curating this benchmark is to generate high-quality options for single-choice questions. Initially, the original data's answer could serve as the correct option. So we need to generate a *deceptive* wrong option. If the generated option is too simple, e.g. with obvious logical errors or unrelated to the table content, the benchmark will be too easy and fail to test LLMs' capabilities. To address this, we devised three prompting-based methods: Modify-Input, Modify-Output, and Exam-Judge (detailed in Section 2.1) for generating wrong options. These methods together produced a variety of deceptive options. The manually verified accuracy rate of our generated data exceeds 92%. We also noted that the Exam-Judge method we proposed generated more challenging questions, which may be used for future dataset construction.

Leveraging the high-quality options, TabIS encompasses three scenarios with increasing difficulty for table information seeking: (1) basic TIS derived from TTG (B-TIS), (2) TIS that emphasizes structural understanding (SU-TIS), and (3) TIS from multiple tables (M-TIS), i.e. when confronted additional pseudo-relevant tables. These scenarios reflect common challenges in real-world applications, such as retrieval-augmented systems.

While previous studies (Zhao et al., 2023b) that test on the basic TIS setting with unreliable metrics demonstrate the superiority of LLMs, TabIS reveals the limitations and potential challenges of LLMs in table information seeking as follows.

- **L1: Most LLMs perform poorly on our reliable benchmark with complex TIS settings and tables with rich hierarchies.** Experiments on 12 representative LLMs show that only GPT-4 attained an 85.7% accuracy on average (random guess would be 50% accuracy). The top-performing 70B open-source model achieved 74.4%, with the rest falling in the 50-60% range.

- **L2: LLMs exhibit a poor understanding of table structures, with accuracy fluctuating across different cell positions.** Surprisingly, we find that LLMs perform almost at random levels in basic lookup tasks, such as repeating content in a specific row. This highlights the substantial challenges in real-world SU-TIS scenarios, where models struggle to pinpoint the target table area using only positional cues.

- **L3: LLMs struggle to balance between TIS performance and robustness against pseudo-relevant tables, especially for open-source models.** This indicates a great challenge for LLMs in retrieval-augmented generation scenarios.

Finally, we fine-tune *Llama2-13b-chat* on our weakly-supervised training dataset and find that while fine-tuning can significantly improve TIS performance, boosting from 55.5 to 73.2, it still lags behind GPT-4-turbo, which has not been specifically fine-tuned. This indicates that the proposed benchmark is non-trivial, calling for further investigations and improvement in this field.

2 TabIS Benchmark

We curated a benchmark *TabIS* to investigate the table information seeking capabilities of LLMs.

We use table-to-text generation (TTG) datasets as the original data source in our benchmark. The task of TTG is that, given a table and a set of selected cells (T, C), produce a one-sentence description of the cells, and the annotated description is called "reference" R. We transform TTG into a single-choice question with two options for objective and accurate evaluation. The format of a sample in TabIS is (T, Q, R, O) where Q is a question, R, O are correct and wrong options. In TabIS, T and R are the same as the annotation in the TTG task, O is a wrong description of the table that we generate, and Q is a question about the table that can be answered by R. So, the task of TabIS is that, given T and Q, choose an option from $\{R, O\}$ as the answer.

TabIS contains three subsets: basic table information seeking (B-TIS), TIS requiring structure understanding (SU-TIS), and TIS with multiple tables (M-TIS). In the following, we will first introduce how to generate options, and then introduce these subsets respectively.

2.1 Option Generation Method

The option generation has three steps:

1. First, for each TTG sample, we generate one challenging candidate option, expecting that the option is unfaithful to the table but is similar to the golden reference.
2. Second, we perform adversarial filtering (Zeng et al., 2023) to divide all instances into easy and hard categories. Specifically, we use three different LLMs on two different presentation orders of the options (R, O and O, R) to obtain six predicted labels. The instances in which the majority of labels are wrong are hard instances and others are simple instances.
3. Third, for hard instances, we conduct manual checking and modification on generated options to ensure correctness.

In step 1, three strategies to generate options are proposed:

Modify-Input (MI). We directly prompt GPT-4 to first modify the highlighted cells C slightly, resulting in a modified set C', and subsequently perform the TTG task using C' to produce an unfaithful statement O referring to R. The generated O usually has a similar syntactic structure as R but substitutes some entities.

Modify-Output (MO). We directly prompt GPT-4 to refer to the golden reference R and make up a new statement that contains highlighted cells C, but is not faithful to the table fact.

Exam-Judge (EJ). Given the table T and a set of cells C, we first instruct a weak LLM agent to describe the cells in natural language, yielding multiple candidate responses $\{O'_1, O'_2, \dots\}$. Subsequently, a more advanced LLM agent[1] is employed to identify responses that are unfaithful to the table. Among these unfaithful candidates, the one that is most literally similar to the golden reference R is selected as the wrong option. The underlying idea is to automatically obtain incorrect responses from relatively weak agents, thereby producing strong false options that are diverse and deceptive. In the experiments, we find this method is good at generating difficult instances.

In step 3, for hard instances, we instruct annotators to check if the generated option is faithful to the table. If it is faithful, then it needs to be revised to an unfaithful description while ensuring the altered options are convincingly deceptive.

Finally, each instance can be categorized into four classes, **MI, MO, EJ**, and **HA** (Human-Annotation, i.e. modified in step 3) according to how its O is generated. We put more details of the option generation pipeline in Appendix A.

[1]We use gpt-3.5-turbo-16k and gpt-4 as the weak and strong LLM agent, respectively.

2.2 B-TIS Subset

B-TIS mimics situations where the LLM agent is tasked with offering clear statements to users who inquire about specific real-world entities, such as celebrities and sports events, based on a table. This method could markedly diminish the necessity for users to sift through massive table data. We show an example in Figure 2.

We apply the aforementioned option generation pipeline to generate the B-TIS dataset using two public TTG datasets: (1) **ToTTo** (Parikh et al., 2020) is an open-domain English table-to-text dataset with over 120,000 examples. The tables in ToTTo are all semi-structured HTML tables from Wikipedia pages and the reference sentences are mainly descriptive statements over the table fact. (2) **HiTab** (Cheng et al., 2022) is a cross-domain hierarchical table dataset with over 10,000 samples, constructed from a wealth of statistical reports. It contains hierarchical tables and accompanied descriptive sentences collected from StatCan and NSF. Compared to ToTTo, HiTab poses a greater challenge to table information seeking since the tables are with hierarchies and the sentences may involve numerical reasoning (e.g. comparison and simple computation).

2.3 SU-TIS Subset

In LLM-based chat systems like ChatGPT (OpenAI, 2023b), a straightforward way for users to direct the LLM agent to a specific area of a table is by indicating positions (e.g., "row 3"). This requires LLMs to understand table structures. We mimic this scenario by introducing the TIS dataset that emphasizes structural understanding (SU-TIS). For each instance (T, Q, R, O) in B-TIS, we modify question Q by replacing the selected cells with the minimum set of rows or columns covering them, as illustrated in Figure 2.

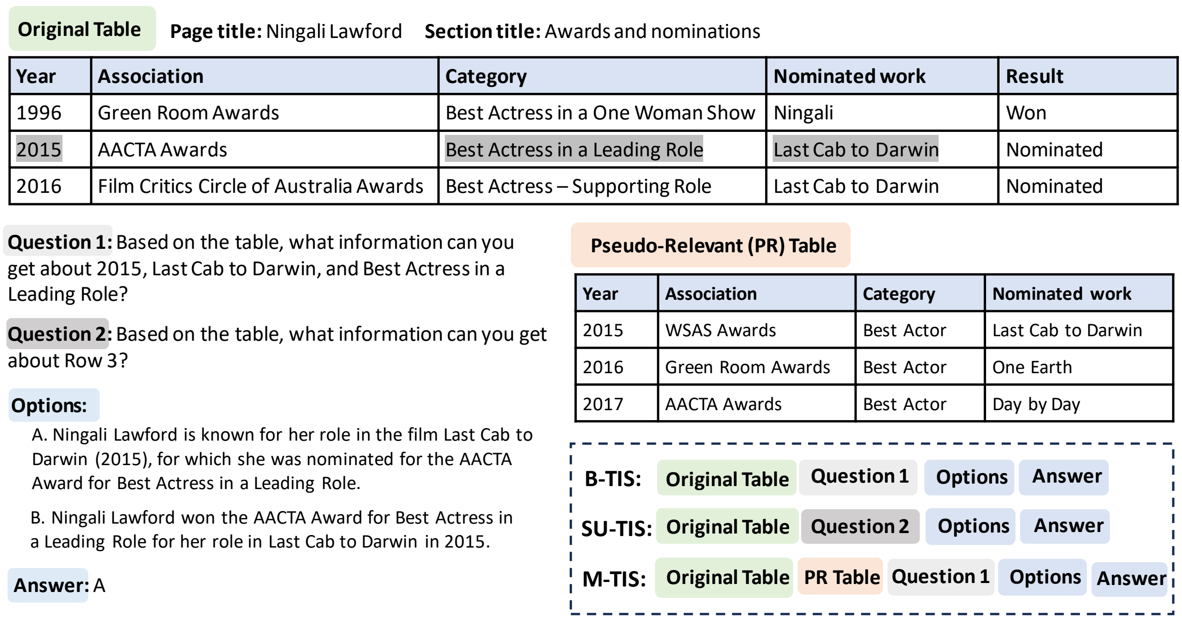

Figure 2: Simplified Examples of B-TIS subset, SU-TIS subset, and M-TIS subset. For each B-TIS sample, we generate one SU-TIS sample and one M-TIS sample with some modifications.

2.4 M-TIS Subset

In real-world scenarios, LLM agents may be presented with additional context that, while superficially related to the golden table (the table that contains the answer), could be misleading and detrimentally affect their information seeking capabilities (Liu et al., 2023). This situation frequently arises in retrieval-augmented LLM systems oriented to documents, where in response to a query, the systems may retrieve several tables that are relevant to the query but not golden.

To mimic this scenario, we investigate the effects of adding one pseudo-relevant table, which appears relevant to the main table but does not provide useful information to answer the question. We show an example in Figure 2. For each instance in B-TIS, we add another table T' to the tuple (T, Q, R, O), resulting in $(\{T, T'\}, Q, R, O)$. T' is generated by prompting *gpt-4-turbo-1106* to create one table mirroring the structure and headers of the golden table, yet contains varied data entries. Refer to Appendix B for more details.

Dataset		# Train	# Test
B-TIS	ToTTo	20,244	1,283
	HiTab	6,943	1,254
SU-TIS	ToTTo	20,054	1,267
	HiTab	6,864	1,215
M-TIS	ToTTo	0	1,217
	HiTab	0	1,139
Total		54,105	7,375

Table 1: Data statistics of TabIS.

	ToTTo	Ratio	Acc.	HiTab	Ratio	Acc.
MI	433	33.7%	93.5%	345	27.5%	90.5%
MO	495	38.6%	95.8%	366	29.2%	97.2%
EJ	267	20.8%	91.7%	438	34.9%	89.2%
HA	88	6.9%	100.0%	105	8.4%	100.0%

Table 2: Statistics of option generation strategies used in B-TIS datasets.

2.5 Dataset Statistics and Quality Assessment

Table 1 illustrates the data statistics of the datasets used in our experiments. We show the statistics of option generation strategies results for the B-TIS dataset in Table 2. We engage 10 sophisticated annotators to meticulously review and revise the hard instances (step 3 in Section 2.1). Out of 410 reviewed samples, the options for 193 samples are manually adjusted. We employ two experts to assess the data quality on 50 samples each from ToTTo-TTG and HiTab-TTG. The accuracy of ToTTo-TTG and HiTab-TTG is 94.1% and 92.5%, respectively, demonstrating the high quality of the proposed TabIS. SU-TIS and M-TIS are generated based on B-TIS, so the statistics and quality are the same as B-TIS.

3 Experiments on TabIS

Based on the curated TabIS benchmark, we evaluate the table information seeking capabilities of 12 representative LLMs.

3.1 Experimental Settings

Problem settings. We evaluate LLMs in a table-based QA setting, where a linearized markdown table is presented in the context, and LLMs are required to answer a question given the context. All the questions are constructed into the single-choice form with two options, as detailed in Section 2. We use a **one-shot example**[2] to familiarize the model with the task description and answering format, similar to previous work (Wang et al., 2023).

We evaluate both proprietary and open-source LLMs. To enhance reproducibility, we set the temperature as 0 for proprietary models, and utilize the maximum probability of the first token as A or B to determine the outputs of open-source models.

Proprietary models. We adopt three representative models: **GPT-3.5** (OpenAI, 2023b), **GPT-4** (OpenAI, 2023a) and **Gemini-pro** (Google, 2023). GPTs[3] is a series of popular and capable LLM systems developed by OpenAI. Recent studies (Akhtar et al., 2023; Sui et al., 2024; Zhao et al., 2023b) have shown the great potential of these models on table-related tasks. Gemini-pro[4] is Google's most capable LLM which operates seamlessly across various modalities.

Open-source models. Using proprietary LLM APIs as agents presents many challenges such as high costs and privacy concerns (Zeng et al., 2023). Therefore, we evaluate several popular open-source models: (1) **Llama2-chat** (Touvron et al., 2023) ranging from 7b to 70b parameters; (2) **Mistral-7b-instruct-v0.2** (Jiang et al., 2023) and **Mixtral-8x7b-instruct** (Jiang et al., 2024), an instruction-tuned sparse mixture of experts language model; (3) **TableLlama-7b** (Zhang et al., 2023), instruction-tuned from Llama2-7b, the first large generalist models for tables; and (4) **Tulu2-70b-DPO** (Ivison et al., 2023), finetuned from Llama2-70b, the first 70b model aligned with DPO (Rafailov et al., 2023). These models represent the highest-quality LLMs of different architectures and alignment strategies available to the community.

3.2 Main Results on TabIS

We show the results of various models on the test set of TabIS in Table 3.

Overall Performance. As shown in the "Avg." column in Table 3, both proprietary models and open-source models perform poorly in TabIS. Proprietary models are generally superior to open-source models, with the highest average accuracy recorded at 85.9 by GPT-4-turbo, compared to 74.1 by Tulu2-70b-DPO. Gemini-pro outperforms GPT-3.5s but falls short of GPT-4-turbo. Regarding open-source models, a trend is observed where larger models within the same series generally outperform their smaller counterparts. For instance, Llama2-chat models with 7b, 13b, and 70b parameters achieve average accuracies of 50.7, 56.7, and 61.9, respectively. However, this trend does not hold across different model series, where a larger model size does not guarantee superior performance. For example, the 7b version of Mistral-instruct even surpasses the 70b Llama2-chat model by 1.3 points. This observation raises an important question about the impact of pre-training and alignment strategies on the TIS capabilities of LLMs which may be an interesting research topic.

Performance on TabIS Subsets. The middle columns in Table 3 show that all models generally perform better in B-TIS compared to SU-TIS and M-TIS, indicating SU-TIS and M-TIS are more challenging. SU-TIS, which only provides the location of highlighted cells as hints, are inherently more difficult than B-TIS. However, models can refer to the cells contained in options to look back at the table to verify each option, therefore the performance drop is not dramatic. M-TIS introduces an extra table that is only seemingly relevant, potentially confusing the judgement of LLMs. In comparisons between datasets, all models show better performance on ToTTo than on HiTab, with improvements ranging from 5.8 to 19.0 points. This discrepancy is likely due to ToTTo predominantly featuring standard tables without merged cells, whereas HiTab includes tables with complex hierarchies, which pose greater challenges for table comprehension.

Comparing option generation strategies. As illustrated in Figure 3, models exhibit the lowest performance with options generated via Exam-

[2] We find that more examples would often surpass the 4,096 token limit commonly used by open-source models.

[3] For GPTs, we investigate *GPT-3.5-turbo-1106* and *GPT-4-turbo-1106* for more consistent evaluation. We also report results on *GPT-3.5-turbo-instruct* and *GPT-3.5-turbo=16k*, since we find their performance varies greatly.

[4] Gemini-pro is currently accessible via the Gemini API.

Model	B-TIS		SU-TIS		M-TIS		Avg.
	ToTTo	HiTab	ToTTo	HiTab	ToTTo	HiTab	
proprietary model							
Gemini-pro	85.6	66.6	81.3	65.1	79.4	64.8	73.8
GPT-3.5-turbo-instruct	75.1	68.3	70.8	65.3	74.5	66.8	70.1
GPT-3.5-turbo-1106	72.1	57.5	66.8	50.4	66.7	53.0	61.1
GPT-3.5-turbo-16k	76.7	61.2	73.3	59.2	73.4	59.2	67.2
GPT-4-turbo-1106	**91.2**	**82.4**	**90.0**	**81.7**	**89.7**	**80.4**	**85.9**
open-source model							
Llama2-7b-chat	53.6	47.8	53.1	48.8	52.3	48.6	50.7
TableLlama-7b	54.3	47.7	54.1	47.8	54.1	47.9	51.0
Mistral-7b-instruct-v0.2	73.2	56.9	69.9	53.5	68.8	57.1	63.2
Llama2-13b-chat	63.3	53.4	57.9	50.5	60.5	54.4	56.7
Mixtral-8*7b-instruct	80.6	65.6	80.8	**62.7**	76.2	57.9	70.6
Llama2-70b-chat	70.0	56.9	67.8	54.3	67.4	54.7	61.9
Tulu2-70b-DPO	**85.7**	**68.2**	**81.9**	61.9	**82.9**	**64.0**	**74.1**

Table 3: Main results on TabIS. Random-guess achieves a 50% accuracy. Details in Appendix D.

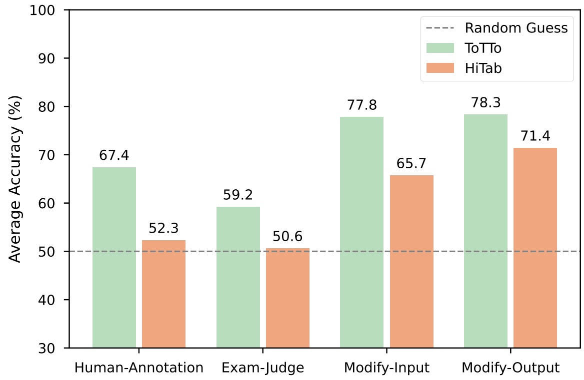

Figure 3: Model performance in different option generation strategies. Averaged over 12 LLMs. Refer to Appendix D for more details.

Judge, with average scores of only 59.2 and 50.6 for ToTTo and HiTab, respectively. This indicates that Exam-Judge is capable of producing options that are even more challenging for LLMs than those annotated by humans. Modify-Input and Modify-Output also present significant hurdles for LLMs, with scores ranging from 65.7 to 78.3 points on average. For options generated by humans, while they are tough enough, they also lead to high expenses. Our option generation pipeline leverages the advanced instruction-following capabilities of potent LLMs, effectively balancing cost-efficiency with scalability.

4 Potential Challenges

In this section, we conduct an in-depth analysis to investigate the LLMs' limitations and potential challenges behind the two complex sub-tasks: SU-TIS and M-TIS.

4.1 Table Structure Understanding

We further investigate the table structure understanding (TSU) capabilities of LLMs, shedding light on future research on the SU-TIS sub-task.

TSU refers to the ability to perceive the two-dimensional layout inherent in tables, such as the positioning of cells, rows, and columns, to access desired content based on the location within the table space. TSU is highly important to our SU-TIS, which involves locating a specific region of the table. While this may seem intuitive to humans, it can be quite challenging for LLMs, especially because tables are fed to these models in a serialized format, such as markdown or HTML. To investigate the TSU capabilities of LLMs, we design six basic lookup tasks, such as "What is the content of cells in row 3/column 3?" and "What is the content of cells within the same row as the cell 'Harry Potter'?" We employ predefined templates to generate test samples from semi-structured HTML tables, transforming them into a single-choice format with two options. Each sample includes one in-context example, similar to TabIS. Refer to Appendix C for more details.

Once humans understand the table structure and the task description, their TSU performance ideally remains excellent and consistent regardless of target locations. However, we find that LLMs work in a totally different manner. Specifically, we report the average accuracy on six tasks and the variation score towards target positions in Figure 4. The variation score for a TSU task is defined as the standard

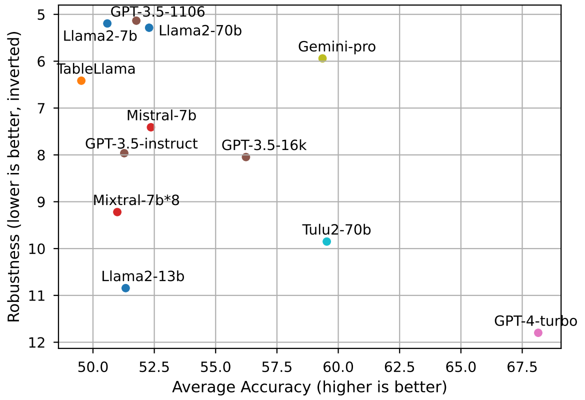

Figure 4: Averaged accuracy and TSU variation score for 12 models, tested and averaged on 6 TSU tasks. Model names are simplified for illustration.

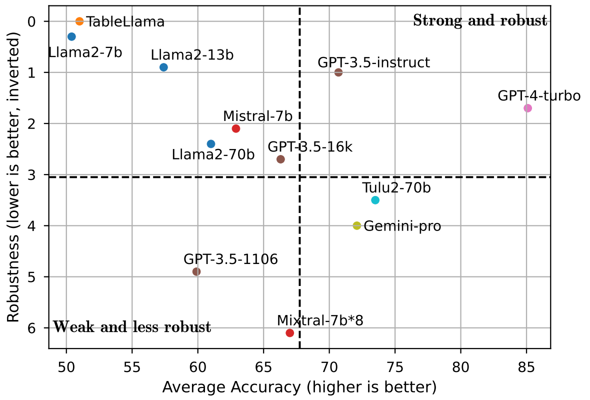

Figure 5: TIS Robustness against pseudo-relevant tables and averaged accuracy for 12 models, tested and averaged on ToTTo and HiTab. Model names are simplified for illustration.

deviation in accuracy across different target locations. Notably, most LLMs achieve near-random performance (50) on TSU tasks. The strongest LLM, GPT-4-turbo, exhibits the lowest stability. No LLMs stand out in both performance and stability. This highlights a common challenge of table structure understanding: **LLMs exhibit poor performance on TSU tasks and the accuracy varies greatly across different positions.** In real-world scenarios of SU-TIS, there are no options for a user query. LLMs can only locate the target region based on the positional information (e.g. row 3). The TIS performance would be largely affected by models' TSU capabilities. We will also release the six TSU datasets to facilitate future research.

4.2 Robustness against Pseudo-Relevant Tables

Based on M-TIS, we further investigate the TIS robustness of various models against pseudo-relevant tables. Specifically, to quantify a model's robustness, we measure the deviation between the accuracy without and with the pseudo-relevant table, averaged on ToTTo and HiTab. The results are shown in Figure 5. Notably, GPT-3.5-instruct and GPT-4-turbo emerge as both effective and robust. However, the two strongest open-source models, Tulu-70b and Mixtral-7b*8, exhibit the lowest robustness levels. Besides, within the same model series, larger models achieve better accuracy scores but worse robustness scores. This phenomenon can be observed in Llama2 series (7b, 13b, 70b) and Mistral series (Mistral-7b, Mixtral-8*7b). M-TIS indicates **great challenges of LLMs in balancing between TIS performance and robustness against pseudo-relevant tables, especially for open-source models**. This finding calls for future research on open-source models to improve TIS robustness against pseudo-relevant tables.

5 Improving Table Information Seeking

In this section, we explore how supervised finetuning enhances table information seeking using weakly-supervised datasets.

We first utilize our proposed data generation pipeline[5] (Section 2) to construct weakly-supervised B-TIS and SU-TIS training datasets without manual checking. The statistics of the training dataset are shown in Table 2. We fully finetune *Llama2-13b-chat* on this training set for 2 epochs to obtain **TISLlama**. We evaluate TISLlama on TabIS[6]. Refer to Appendix E for more training details.

Table 4 demonstrates that TISLlama outperforms both the base model Llama2-13b-chat and the leading open-source model Tulu2-70b-DPO, with margins of 17.7 and 5.4 points, respectively. These results demonstrate the effectiveness of TIS-oriented supervised finetuning. However, its performance does not yet match that of GPT-4-turbo, which has not undergone specialized fine-tuning. This discrepancy highlights the significant challenge TabIS presents to large language models, underscoring the need for further research in this area.

[5]Considering high cost of accessing GPT-4-turbo API, we use GPT-3.5-turbo-16k instead.

[6]Note that training on the weakly-supervised datasets may introduce the spurious correlation between the model-generated options and the wrong options. Thus we only evaluate on human-annotated samples for fair comparision.

Model	B-TIS	SU-TIS	M-TIS	Avg.
Llama2-13b-chat	56.8	53.3	56.5	55.5
Llama2-70b-chat	58.2	58.1	58.5	58.3
Tulu2-70b-DPO ♣	69.7	69.1	64.7	67.8
GPT-4-turbo-1106 ♠	81.2	77.4	79.1	79.2
TISLlama (ours)	73.3	73.7	72.7	73.2

Table 4: Evaluation of TISLlama on TabIS-HA, averaged on ToTTo and HiTab. ♣ and ♠ denote the best open-source and proprietary model in our evaluation.

6 Related Work

6.1 Table-to-Text generation

Table-to-Text generation (TTG) aims at generating natural language statements that faithfully describe the information contained in the provided table region. Given its broad applications like biographical data analysis (Lebret et al., 2016) and sports game summary generation (Wiseman et al., 2017), TTG has been studied extensively in recent years (Wang et al., 2022; Zhao et al., 2023a) with the introduction of several valuable datasets (Parikh et al., 2020; Cheng et al., 2022; Chen et al., 2020a). Previous studies mainly focus on finetuning pre-trained language models on a task-specific dataset (Wang et al., 2022), which are often specialized and lack generalizability. Large Language Models (LLMs) have recently demonstrated remarkable performance on TTG tasks (Yang et al., 2023; Zhao et al., 2023b). However, these evaluations mainly rely on surface-level metrics, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), which may result in unreliable evaluation when the syntactic style of LLMs' response diverges from the golden reference (Dhingra et al., 2019). In this paper, we propose to employ the TTG tasks as a test bench for evaluating table information seeking of LLMs. To ensure a reliable assessment, we construct single-choice questions based on two high-quality TTG datasets, ToTTo (Parikh et al., 2020) and HiTab (Cheng et al., 2022).

6.2 Evaluating Table Information Seeking capabilities of LLMs

Prior research has not fully explored the table information seeking (TIS) abilities of Large Language Models (LLMs). Sui et al. (2024) introduces a benchmark aimed at assessing the structural understanding of LLMs by comparing different input methodologies. This benchmark includes a component designed to evaluate the table structure understanding (TSU), which aligns closely with our TSU dataset, yet it does not specifically address TIS tasks. Zhao et al. (2023b) investigates the potential of applying LLMs in real-world table information seeking scenarios, showcasing their effectiveness in producing faithful statements. Nevertheless, their analysis lacks depth and is significantly influenced by unreliable evaluation metrics.

To the best of our knowledge, we are the first to release a large-scale, comprehensive, reliable benchmark for evaluating TIS capabilities.

7 Conclusion

This paper introduces TabIS, a new benchmark designed to evaluate the table information seeking (TIS) abilities of large language models (LLMs). TabIS is comprised of three typical TIS scenarios and employs a single-question format to ensure reliable evaluation. Extensive experiments on 12 representative LLMs have shown that TabIS presents a significant challenge for current LLMs, with GPT-4-turbo showing only marginal satisfaction. Further analysis points out two main issues: firstly, LLMs perform almost randomly on basic tasks involving comprehension of table structures; secondly, they face difficulties in improving performance and maintaining robustness against pseudo-relevant tables, which could lead to sub-optimal results in real-world TIS tasks. These observations underscore the current limitations and potential challenges in table information seeking, calling for further exploration and advancement in this area.

8 Limitations

In this paper, the benchmark adopts the form of single-choice questions, which ensures the reliability of the evaluation but may deviate from practical applications. We mainly discuss some limitations and potential challenges of LLMs when handling table information seeking tasks, but do not explore how to address these issues or the reasons behind their observations. These will be important future research directions. The templates used for generating TIS questions are relatively simplistic; richer and more diverse questions would enhance the quality of the benchmark.

References

Mubashara Akhtar, Abhilash Reddy Shankarampeta, Vivek Gupta, Arpit Patil, Oana Cocarascu, and Elena Simperl. 2023. Exploring the numerical reasoning capabilities of language models: A comprehensive analysis on tabular data. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*.

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

Wenhu Chen, Jianshu Chen, Yunde Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. *ArXiv*, abs/2004.10404.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. Hitab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*.

Google. 2023. Introducing gemini: our largest and most capable ai model.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing LM adaptation with tulu 2. *CoRR*, abs/2311.10702.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L'elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Conference on Empirical Methods in Natural Language Processing*.

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th international conference companion on world wide web*, pages 75–76.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryscinski, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir R. Radev. 2021. Fetaqa: Free-form table question answering. *Trans. Assoc. Comput. Linguistics*, 10:35–49.

OpenAI. 2023a. Gpt-4 technical report.

OpenAI. 2023b. Introducing chatgpt.

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

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Gpt4table: Can large language models understand structured table data? a benchmark and empirical study. In *WSDM 2024*.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Huai hsin Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Annual Meeting of the Association for Computational Linguistics*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Fei Wang, Zhewei Xu, Pedro A. Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. *ArXiv*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *CoRR*, abs/2306.04751.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2020. Structure-aware pre-training for table understanding with tree-based transformers. *arXiv preprint arXiv:2010.12537*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. *ArXiv*.

Bohao Yang, Chen Tang, Kun Zhao, Chenghao Xiao, and Chenghua Lin. 2023. Effective distillation of table-based reasoning ability from llms. *CoRR*, abs/2309.13182.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following. *CoRR*, abs/2310.07641.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023. Tablellama: Towards open large generalist models for tables. *CoRR*, abs/2311.09206.

Yilun Zhao, Boyu Mi, Zhenting Qi, Linyong Nan, Minghao Guo, Arman Cohan, and Dragomir R. Radev. 2023a. Openrt: An open-source framework for reasoning over tabular data. In *Annual Meeting of the Association for Computational Linguistics*.

Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. 2023b. Investigating table-to-text generation capabilities of large language models in real-world information seeking scenarios. In *Conference on Empirical Methods in Natural Language Processing*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.

A Option Generation Details

We show the prompt of Exam-Judge, Modify-Input, and Modify-Output in Figure 6, Figure 7, and Figure 8, respectively.

B M-TIS Details

We show the prompt for generating pseudo-relevant tables in Figure 9.

C Exploring Table Structure Understanding

In this section, we first introduce the construction of TSU dataset, then we show our additional experiments on TSU.

C.1 Dataset Construction

Understanding the structure of a table is a fundamental ability to navigate among data arranged in a tabular format, interpret the relations among data points, and understand the table. It requires to perceive the two-dimensional spatial layout inherent in tables, such as the positioning of cells, rows, and columns, to access desired content based on the location within the table space.

To examine the table structure understanding capability of LLMs, we propose six probing tasks: positional cell lookup (PCL), relative cell lookup (RCL), positional row lookup (PRL), relative row lookup (RRL), positional column lookup (PLL), relative column lookup (RLL). These tasks require LLMs to acquire certain surface-level table components (cell, row and column) based on relative or absolute position information.

We generate samples for each task by applying predefined templates on high-quality tables. All question templates are shown in Table 5. We collect tables from four public datasets: WikiSQL (Zhong et al., 2017), WikiTableQuestions (Pasupat and Liang, 2015), HybridQA (Chen et al., 2020b) and FeTaQA (Nan et al., 2021). These tables are all semi-structured HTML tables collected from Wikipedia, spaning a wide array of topics such as sports and geography. After deduplicating these tables, we obtain a total of 49,561 high-quality tables. For the test set, we randomly sample 1% tables and generate one sample per table for each task.

For each sample, the options are generated by randomly sampling cells, rows and columns in proximity to the golden answer, employing a gaussian distribution $\mathcal{N}(\mathbf{p}, 1)$, where $\mathbf{p}$ denotes the position of the golden answer.

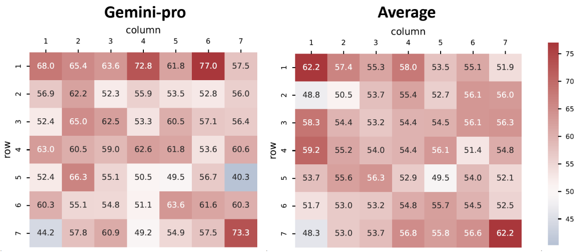

Figure 10: RCL performance with respect to target cell positions. We show a concrete example of Gemini-pro (left) and the averaged results of 12 models (right).

C.2 Experiments

We show the TSU performance of various models on Table 6.

TSU Performance. Unexpectedly, despite TSU being straightforward for humans, all LLMs demonstrate subpar performance. The best performance of proprietary models and open-source models only achieve 66.1 (GPT-4) and 57.6 points (Tulu2-70b), respectively, while most models achieve near-random performance (50). Models do not consistently excel across all types of TSU tasks. Notably, the GPT series (GPT-4 and GPT-3.5) tend to perform better in column-oriented tasks (COL) relative to other tasks, whereas the Llama2 series (Llama2-7b, 13b, 70b) shows greater proficiency in cell-oriented tasks (CELL). This variation in performance could be attributed to the fact that models within the same series likely undergo similar pre-training and alignment processes, resulting in comparable inductive biases.

Case Study on Variations across different positions. We show some results of RCL in Figure 10. Gemini-pro exhibits large variance in different positions,with a disparity exceeding 30 points between its highest and lowest accuracy. On average, the data indicates that LLMs perform more effectively at the beginning (row 1, column 1) and ending (row 7, column 7) of tables. This pattern is likely influenced by the serialization of tables into one-dimension strings, rendering the middle part of the table more challenging to locate accurately.

Effect of Cell Content on TSU. Logically, executing TSU tasks should not depend on the specific content of table cells, as this does not require an understanding of the table's semantics. Thus, the performance across tables with varying content should be consistent. To test this, we altered the cell contents in our TSU test set's real tables to random numbers (ranging from 1 to 8 digits) and random letters (also 1 to 8 characters in length), creating two new synthetic test sets named "letter" and "number."

However, we observe **significant variation in performance across different table contents**. As shown in Figure 11, the performance disparity between the test sets ranges from approximately 2.9 to 19.4 points. Intriguingly, GPT-4 shows markedly improved performance on the "number" set. This may be attributed to the activation of GPT-4's numerical processing capabilities, which are particularly relevant for TSU tasks (e.g. counting rows). This observation warrants further investigation in future studies.

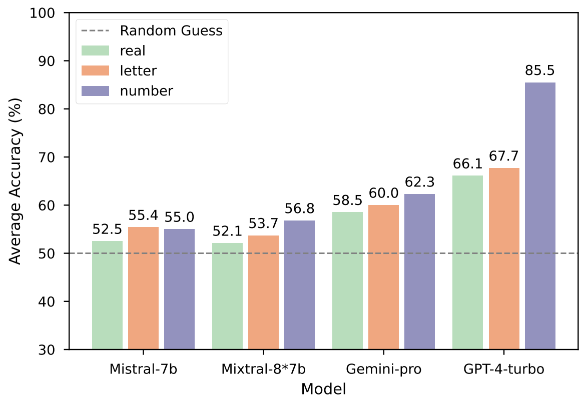

Figure 11: Accuracy on tables of different content, averaged on 6 TSU tasks.

D TabIS Main Results

We show the main results of B-TIS, SU-TIS, and M-TIS in Table 7, Table 8, and Table 9, respectively. We also report the accuracy on each option generation strategies.

E Training Details

We fully fine-tune the model *Llama2-13b-chat*[7] with Huggingface transformers library. We use a learning rate of 2e-5. We train the model on 8 A800 and use a linear scheduler with a 5% warm-up period for 2 epochs. To efficiently train the model, we employ DeepSpeed training with ZeRO-3 stage. For both training and inference, we set the input length as 4096.

[7] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

Task	Question Template (Q)
Positional Cell Lookup	Q: What is the content of the cell located at row {row} and column {col}?
Positional Row Lookup	Q: What are the contents of the cells in row {row}?
Positional Column Lookup	Q: What are the contents of the cells in column {col}?
Relative Cell Lookup	Q1: The anchor cell is {anchor} in row {row} and column {col}. What is the content of the first cell below the anchor cell within the same column? Q2: The anchor cell is {anchor} in row {row} and column {col}. What is the content of the first cell above the anchor cell within the same column? Q3: The anchor cell is {anchor} in row {row} and column {col}. What is the content of the first cell left to the anchor cell within the same row? Q4: The anchor cell is {anchor} in row {row} and column {col}. What is the content of the first cell right to the anchor cell within the same row?
Relative Row Lookup	Q1: The anchor cell is {anchor} in row {row} and column {col}. What are the contents of the cells within the same row as the anchor cell? Q2: The anchor cell is {anchor} in row row and column col. What are the contents of the first row above the anchor cell? Q3: The anchor cell is {anchor} in row {row} and column {col}. What are the contents of the first row below the anchor cell?
Relative Column Lookup	Q1: The anchor cell is {anchor} in row {row} and column {col}. What are the contents of the cells within the same column as the anchor cell? Q2: The anchor cell is {anchor} in row {row} and column {col}. What are the contents of the first column left to the anchor cell? Q3: The anchor cell is {anchor} in row {row} and column {col}. What are the contents of the first column right to the anchor cell?

Table 5: Descriptions of TSU Tasks (T) and Corresponding Question Templates (Q). Placeholders {row}, {col}, and {anchor} represent the row number, column number, and the content of the anchor cell, respectively.

Model	PCL	PRL	PLL	RCL	RRL	RLL	Avg.
proprietary model							
Gemini-pro	50.7	**59.9**	46.2	51.3	70.3	72.9	58.5
GPT-3.5-turbo-16k	**55.1**	53.1	61.5	54.9	55.7	54.7	55.8
GPT-3.5-turbo-instruct	47.5	46.1	56.9	40.0	63.7	56.8	51.8
GPT-3.5-turbo-1106	50.4	50.8	53.6	49.8	53.2	49.9	51.3
GPT-4-turbo-1106	50.2	38.3	**82.4**	**72.7**	**74.7**	**78.3**	**66.1**
open-source model							
Llama2-7b-chat	**53.3**	47.8	50.0	55.7	47.8	50.1	50.8
TableLlama-7b	49.2	**53.7**	53.6	55.1	54.4	54.3	53.4
Mistral-7b-instruct-v0.2	49.0	45.9	52.9	58.0	56.7	52.6	52.5
Llama2-13b-chat	51.6	51.8	51.9	57.8	53.2	52.2	53.1
Mixtral-8*7b-instruct	47.1	48.0	**55.9**	52.7	57.1	52.2	52.1
Llama2-70b-chat	51.6	48.2	47.5	56.5	51.3	47.8	50.5
Tulu2-70b-DPO	50.6	48.6	54.8	**67.1**	**70.0**	**54.5**	**57.6**

Table 6: Main results (accuracy) of various models across TSU tasks.

Model	ToTTo					HiTab				
	EJ	MI	MO	HA	Avg.	EJ	MI	MO	HA	Avg.
proprietary model										
gemini-pro	70.2	93.3	87.9	76.9	85.6	53.1	67.6	79.1	67.4	66.6
GPT-3.5-turbo-instruct	60.7	81.8	80.6	55.7	75.1	62.3	71.9	78.4	45.7	68.3
GPT-3.5-turbo-1106	56.9	77.8	76.6	64.8	72.1	42.5	64.6	71.3	48.6	57.5
GPT-3.5-turbo-16k	58.4	84.5	82.8	59.1	76.7	48.4	67.5	75.4	43.8	61.2
GPT-4-turbo-1106	**79.8**	**93.5**	**96.4**	**85.2**	**91.2**	**73.5**	**85.2**	**91.8**	**77.1**	**82.4**
open-source model										
Llama2-7b-chat	54.3	52.4	53.1	60.2	53.6	44.3	54.8	47.8	39.1	47.8
TableLlama-7b	53.2	54.7	53.9	58.0	54.3	43.8	53.3	48.9	41.0	47.7
Mistral-7b-instruct-v0.2	52.8	77.4	81.0	70.5	73.2	40.9	63.5	72.4	47.6	56.9
Llama2-13b-chat	52.4	66.7	66.7	60.2	63.3	45.0	52.2	64.8	53.3	53.4
Mixtral-8*7b-instruct	55.8	88.7	88.1	73.9	80.6	51.6	**75.1**	77.1	52.4	65.6
Llama2-70b-chat	52.1	70.9	79.6	65.9	70.0	46.8	60.0	68.0	50.5	56.9
Tulu2-70b-DPO	**64.4**	**91.7**	**93.1**	**78.4**	**85.7**	**55.5**	72.5	**81.4**	**61.0**	**68.2**

Table 7: B-TIS Main results on ToTTo and HiTab. We also report accuracy across different option generation strategies: EJ (Exam-Judge), MI (Modify-Input), MO (Modify-Output), HA (Human-Annotation).

Exam Prompt

Instruction

Given a table-related task (Task), an example of the task (Example) and one input (Input), your task is to follow the task instruction and provide a response (Output) to the input. Act like a weak assistant that may generate responses that are not faithful to the table fact. Don't generate incomplete responses or too long responses. Don't explain how you come up with your response.

Task

{task_instruct}

Example

{demo}

New Input

{input}

Answers

Judge Prompt

Instruction

Given a table and a list of statements, your task is to identify which of these statements are unfaithful to the table and its meta information. Please note that the meta information may offer additional context about the table, such as background information about the person, album, or competetion the table pertains to. Your response should in json format: {{"reasoning": your judgement of each statement, "unfaithful statements": the list of the serial number of unfaithful statements}}. Make sure your response can be parsed by json.loads.

Table

Meta Information of the table: **{meta_info}**

{md_table}

Statements

{statements}

Response

Figure 6: Prompt of Exam-Judge.

Modify-Input Prompt

Instruction

You are a helpful assistant in generating one statement that is unfaithful to the table fact. Given a statement generation task, and one input-output pair of the task, you need to (1) slightly modify the input; (2) perform the task on the modified input to get the unfaithful statement. Basically, it is hard for a person to find that your generated statement is actually not faithful. Your response should in json format: {{"reasoning": Your modification of input, "unfaithful statement": the unfaithful statement}}. Make sure your response can be parsed by json.loads.

Task

{task_instruct}

Input

{input}

Standard Answer

{output}

Response

Figure 7: Prompt of Modify-Input.

Modify-Output Prompt

Instruction

You are a helpful assistant in generating one unfaithful statement. You can refer to the given faithful statement and make up a new statement that contains several highlighted cells, but is not faithful to the table fact. Basically, it is hard for a person to find that your generated statement is not faithful. Your response should in json format: {{"reasoning": your reasoning process, "unfaithful statement": the unfaithful statement}}. Make sure your response can be parsed by json.loads.

Table

Meta Information of the table: **{meta_info}**

{md_table}

Highlighted Cells

{highlighted_cells}

Faithful Statement

{output}

Response

Figure 8: Prompt of Modify-Output.

Prompt for Generating Pseudo-Relevant Tables

Instruction

Create a concise table, mirroring the structure of a provided example, but with unique data entries. Ensure specific cell contents are replicated in the new table.

Example Table

{table}

Specific Cell Content

{subset_of_highlighted_cells}

New Table

Note that Limit the table to 5-15 rows, presenting it without additional commentary.

Figure 9: Prompt for generating pseudo-relevant tables.

Model	ToTTo					HiTab				
	EJ	MI	MO	HA	Avg.	EJ	MI	MO	HA	Avg.
proprietary model										
gemini-pro	72.2	81.8	87.6	64.7	81.3	52.6	71.4	75.7	54.0	65.1
GPT-3.5-turbo-instruct	55.9	75.7	78.5	48.9	70.8	57.4	69.9	75.4	48.5	65.3
GPT-3.5-turbo-1106	49.8	72.5	72.5	58.0	66.8	34.9	60.2	62.0	42.7	50.4
GPT-3.5-turbo-16k	56.7	81.3	78.5	55.7	73.3	43.3	69.6	62.0	42.7	59.2
GPT-4	**77.6**	**91.7**	**96.5**	**83.0**	**90.0**	**71.0**	**85.2**	**94.1**	**71.8**	**81.7**
open-source model										
Llama2-chat-7b	52.1	52.1	53.3	60.2	53.1	45.9	55.4	48.4	40.8	48.8
TableLlama-7b	53.2	52.8	54.8	59.1	54.1	44.5	51.8	49.6	42.7	47.8
Mistral-7b-instruct-v0.2	48.3	74.3	78.3	65.9	69.9	34.4	63.0	71.1	41.8	53.5
Llama2-chat-13b	51.3	59.7	61.0	52.3	57.9	42.9	49.1	60.1	54.4	50.5
Mixtral-8*7b-instruct	56.3	**88.0**	88.2	78.4	80.8	**49.0**	**71.1**	73.9	54.4	**62.7**
Llama2-chat-70b	51.0	68.8	75.4	71.6	67.8	44.7	60.5	62.9	44.7	54.3
Tulu2-70b-DPO	**63.1**	85.9	**88.6**	**81.8**	**81.9**	47.8	65.4	**77.3**	**56.3**	61.9

Table 8: SU-TIS Main results on ToTTo and HiTab. We also report accuracy across different option generation strategies: EJ (Exam-Judge), MI (Modify-Input), MO (Modify-Output), HA (Human-Annotation).

Model	ToTTo					HiTab				
	EJ	**MI**	**MO**	**HA**	**Avg.**	**EJ**	**MI**	**MO**	**HA**	**Avg.**
proprietary model										
gemini-pro	63.2	84.6	86.2	62.8	79.4	49.8	67.4	78.3	65.5	64.8
GPT-3.5-turbo-instruct	61.9	81.1	79.7	53.5	74.5	59.3	68.5	79.2	48.4	66.8
GPT-3.5-turbo-1106	55.0	71.0	72.0	53.5	66.7	39.7	59.2	66.1	41.9	53.0
GPT-3.5-turbo-16k	56.5	81.6	79.3	53.5	73.4	45.0	67.2	73.8	39.8	59.2
GPT-4	**74.2**	**93.1**	**96.2**	**86.1**	**89.7**	**71.0**	**83.1**	**91.4**	**72.0**	**80.4**
open-source model										
Llama2-chat-7b	51.2	51.1	52.8	58.1	52.3	45.5	53.8	48.8	43.0	48.6
TableLlama-7b	51.9	53.9	54.9	57.0	54.1	45.0	52.2	49.1	40.9	47.9
Mistral-7b-instruct-v0.2	46.9	74.2	76.7	66.3	68.8	41.9	63.7	71.4	47.3	57.1
Llama2-chat-13b	52.3	63.5	63.0	57.0	60.5	44.2	53.2	67.0	**55.9**	54.4
Mixtral-8*7b-instruct	50.4	84.4	84.6	69.8	76.2	46.5	59.9	72.3	47.3	57.9
Llama2-chat-70b	47.7	69.7	76.3	67.4	67.4	43.9	59.9	64.0	49.5	54.7
Tulu2-70b-DPO	**60.4**	**90.3**	**90.2**	**76.7**	**82.9**	**52.0**	**68.8**	**76.8**	52.7	**64.0**

Table 9: M-TIS Main results on ToTTo and HiTab. We also report accuracy across different option generation strategies: EJ (Exam-Judge), MI (Modify-Input), MO (Modify-Output), HA (Human-Annotation).