Investigating Table-to-Text Generation Capabilities of LLMs in Real-World Information Seeking Scenarios

Yilun Zhao[*1] **Haowei Zhang**[*2] **Shengyun Si**[*2]
Linyong Nan[1] **Xiangru Tang**[1] **Arman Cohan**[1,3]
[1]Yale University, [2]Technical University of Munich, [3]Allen Institute for AI
yilun.zhao@yale.edu {haowei.zhang, shengyun.si}@tum.de

Abstract

Tabular data is prevalent across various industries, necessitating significant time and effort for users to understand and manipulate for their information-seeking purposes. The advancements in large language models (LLMs) have shown enormous potential to improve user efficiency. However, the adoption of LLMs in real-world applications for table information seeking remains underexplored. In this paper, we investigate the table-to-text capabilities of different LLMs using four datasets within two real-world information seeking scenarios. These include the LogicNLG and our newly-constructed LoTNLG datasets for *data insight generation*, along with the FeTaQA and our newly-constructed F2WTQ datasets for *query-based generation*. We structure our investigation around three research questions, evaluating the performance of LLMs in table-to-text generation, automated evaluation, and feedback generation, respectively. Experimental results indicate that the current high-performing LLM, specifically GPT-4, can effectively serve as a table-to-text generator, evaluator, and feedback generator, facilitating users' information seeking purposes in real-world scenarios. However, a significant performance gap still exists between other open-sourced LLMs (e.g., Tülu and LLaMA-2) and GPT-4 models. Our data and code are publicly available at https://github.com/yale-nlp/LLM-T2T.

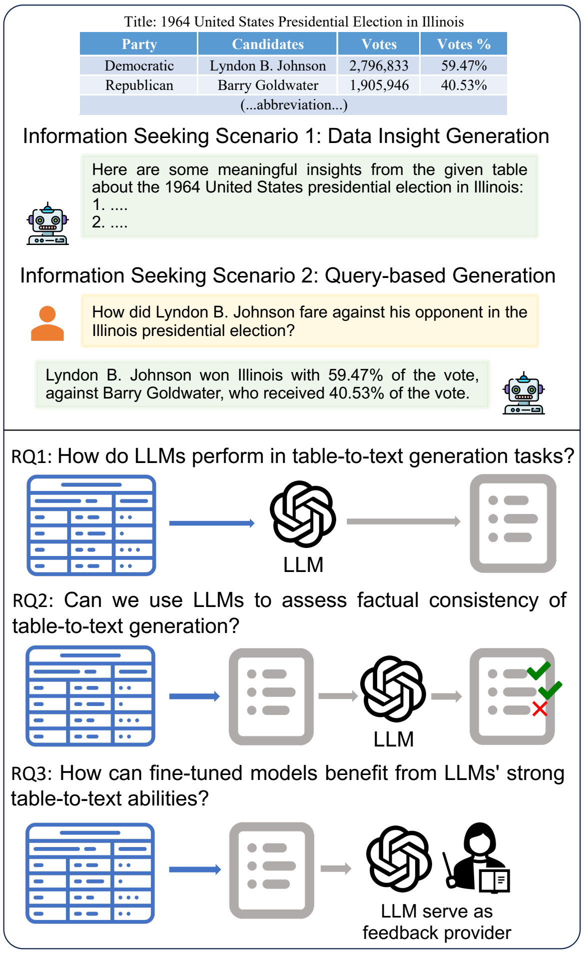

Figure 1: The real-world table information seeking scenarios and research questions investigated in this paper.

1 Introduction

In an era where users interact with vast amounts of structured data every day for decision-making and information-seeking purposes, the need for intuitive, user-friendly interpretations has become paramount (Zhang et al., 2023; Zha et al., 2023; Li et al., 2023). Given this emerging necessity, table-to-text generation techniques, which transform complex tabular data into comprehensible narratives tailored to users' information needs, have drawn considerable attention (Parikh et al., 2020; Chen et al., 2020a; Nan et al., 2022b; Zhao et al., 2023c). These techniques can be incorporated into a broad range of applications, including but not limited to game strategy development, financial analysis, and human resources management. However, existing fine-tuned table-to-text generation models (Nan et al., 2022a; Liu et al., 2022b,a; Zhao et al., 2023b) are typically task-specific, limiting their adaptability to real-world applications.

The emergence and remarkable achievements of LLMs (Brown et al., 2020; Scao et al., 2022; Wang

[*]Equal Contributions.

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 160–175
December 6-10, 2023

Dataset	# Table	# Examples	Control Signal	Rich in Reasoning?
		Data Insight Generation		
LogicNLG (Chen et al., 2020a)	862	4,305	None	✓
LoTNLG (ours)	862	4,305	Reasoning type	✓
		Query-based Generation		
FeTaQA (Parikh et al., 2020)	2,003	2,003	User query	✗
F2WTQ (ours)	4,344	4,344	User query	✓

Table 1: Experimental dataset statistics for the test set. Examples of our newly-constructed LoTNLG and F2WTQ datasets are displayed in Figure 2 and 3, respectively.

et al., 2023; Scheurer et al., 2023; OpenAI, 2023; Touvron et al., 2023a; Taori et al., 2023; Touvron et al., 2023b) have sparked a significant transformation in the field of controllable text generation and data interpretations (Nan et al., 2021; Zhang et al., 2022; Goyal et al., 2022; Köksal et al., 2023; Gao et al., 2023b; Madaan et al., 2023; Zhou et al., 2023). As for table-based tasks, recent work (Chen, 2023; Ye et al., 2023; Gemmell and Dalton, 2023) reveals that LLMs are capable of achieving competitive performance with state-of-the-art fine-tuned models on table question answering (Pasupat and Liang, 2015; Nan et al., 2022b) and table fact checking (Chen et al., 2020b; Gupta et al., 2020). However, the potential of LLMs in generating text from tabular data for users' information-seeking purposes remains largely underexplored.

In this paper, we investigate the table-to-text generation capabilities of LLMs in two real-world table information seeking scenarios: 1) **Data Insight Generation** (Chen et al., 2020a), where users aim to promptly derive significant facts from the table, anticipating the systems to offer several data insights; and 2) **Query-based Generation** (Pasupat and Liang, 2015; Nan et al., 2022b), where users consult tables to answer specific questions. To facilitate a rigorous evaluation of LLM performance, we also construct two new benchmarks: **LoTNLG** for data insight generation conditioned with specific logical reasoning types; and **F2WTQ** for free-form question answering that requires models to perform human-like reasoning over Wikipedia tables.

We provide an overview of table information seeking scenarios and our main research questions in Figure 1, and enumerate our findings as follows:

RQ1: *How do LLMs perform in table-to-text generation tasks?*
Finding: LLMs exhibit significant potential in generating coherent and faithful natural language statements based on the given table. For example, GPT-4 outperforms state-of-the-art fine-tuned models in terms of faithfulness during both automated and human evaluations. The statements generated by GPT-3.5 and GPT-4 are also preferred by human evaluators. However, a significant performance gap still exists between other open-sourced LLMs (e.g., Vicuna and LLaMA-2) and GPT-* models, especially on our newly-constructed LoTNLG and F2WTQ datasets.

RQ2: *Can we use LLMs to assess factual consistency of table-to-text generation?*
Finding: LLMs using chain-of-thought prompting can serve as reference-free metrics for table-to-text generation evaluation. These metrics demonstrate better alignment with human evaluation in terms of both fluency and faithfulness.

RQ3: *How can fine-tuned models benefit from LLMs' strong table-to-text abilities?*
Finding: LLMs that utilize chain-of-thought prompting can provide high-quality natural language feedback in terms of factuality, which includes explanations, corrective instructions, and edited statements for the output of other models. The edited statements are more factually consistent with the table compared to the initial ones.

2 Table Information Seeking Scenarios

Table 1 illustrates the data statistics for the four datasets used in the experiments. We investigate the performance of the LLM in the following two real-world table information-seeking scenarios.

2.1 Data Insight Generation

Data insight generation is an essential task that involves generating meaningful and relevant insights from tables. By interpreting and explaining tabular data in natural language, LLMs can play a crucial role in assisting users with information seeking and decision making. This frees users from the need to manually comb through vast amounts of data. We use the following two datasets for evaluation.

2.1.1 LogicNLG Dataset

The task of LogicNLG (Chen et al., 2020a) involves generating five logically consistent sentences from a given table. It aims to uncover intriguing facts from the table by applying various logical reasoning operations (e.g., count and comparison) across different table regions.

2.1.2 LoTNLG Dataset

Our preliminary experiments revealed that when applied to the LogicNLG dataset, table-to-text generation systems tend to generate multiple sentences that employ the same logical reasoning operations. For instance, in a 0-shot setting, the GPT-3.5 model is more inclined to generate sentences involving numerical comparisons, while overlooking other compelling facts within tables. This lack of diversity in data insight generation poses a significant limitation because, in real-world information-seeking scenarios, users typically expect systems to offer a variety of perspectives on the tabular data. To address this issue, application developers could tailor the table-to-text generation systems to generate multiple insights that encompass different logical reasoning operations (Perlitz et al., 2022; Zhao et al., 2023b). In order to foster a more rigorous evaluation of LLMs' abilities to utilize a broader range of logical reasoning operations while generating insights from tables, we have developed a new dataset, LoTNLG, for logical reasoning type-conditioned table-to-text generation. In this setup, the model is tasked with generating a statement by performing the logical reasoning operations of the specified types on the tables.

LoTNLG Dataset Construction Following Chen et al. (2020b), we have predefined nine types of common logical reasoning operations (e.g., count, comparative, and superlative), with detailed definitions provided in Appendix A.1. We use examples from the LogicNLG test set to construct LoTNLG. Specifically, for each statement from LogicNLG, we assign two annotators to independently label the set of logical reasoning types used in that statement, ensuring that no more than two types were identified per statement. If there are discrepancies in the labels, an expert annotator is brought in to make the final decision. The distribution of logical reasoning types in LoTNLG is illustrated in Figure 4 in Appendix A.1.

Table title: World Golf Championships

Nation	Total Wins	Team wins	Individual Wins	Individual Winners
United States	32	1	31	12
Australia	5	0	5	3
England	5	1	4	3
South Africa	4	2	2	1
Northern Ireland	2	0	2	1
Germany	2	1	1	1
Canada	1	0	1	1
Fiji	1	0	1	1
Sweden	1	0	1	1
Italy	1	0	1	1
Japan	1	1	0	0
Wales	1	1	0	0

Statement1: *Australia and England have the same exact number of Total Win at the World Golf Championship*
Logical label: count

Statement2: *England has 2 more Individual Win than South Africa at the World Golf Championship*
Logical label: comparative

Statement3: *South Africa has the most Team Win of any country at the World Golf Championship*
Logical label: superlative

Statement4: *There are 5 country with only 1 Team Win at the World Golf Championship*
Logical label: count, unique

Statement5: *The United State had 11 more Individual Winner than Northern Ireland had at the World Golf Championship*
Logical label: comparative

Figure 2: An example of LoTNLG, where models are required to generate statements using the specified types of logical reasoning operations

2.2 Query-based Generation

Query-based table-to-text generation pertains to producing detailed responses based on specific user queries in the context of a given table. The ability to answer users' queries accurately, coherently, and in a context-appropriate manner is crucial for LLMs in many real-world applications, such as customer data support and personal digital assistants. We utilize following two datasets to evaluate LLMs' efficiency in interacting with users and their proficiency in table understanding and reasoning.

2.2.1 FeTaQA Dataset

Nan et al. (2022b) introduces a task of free-form table question answering. This task involves retrieving and aggregating information from Wikipedia tables, followed by generating coherent sentences based on the aggregated contents.

2.2.2 F2WTQ Dataset

Queries in the FeTaQA dataset typically focus on *surface-level facts* (e.g., "Which country hosted the 2014 FIFA World Cup?"). However, in real-world information-seeking scenarios, users are likely to consult tables for more complex questions, which require models to perform human-like reasoning over tabular data. Therefore, we have constructed a new benchmark, named F2WTQ, for more challenging, free-form table question answering tasks.

Year	Competition	Venue	Position	Event	Notes
1999	European Junior Championships	Riga, Latvia	4th	400 m hurdles	52.17
2000	World Junior Championships	Santiago, Chile	1st	400 m hurdles	49.23
2001	World Championships	Edmonton, Canada	18th (sf)	400 m hurdles	49.80
2001	Universiade	Beijing, China	8th	400 m hurdles	49.68
2002	European Indoor Championships	Vienna, Austria	1st	400 m	45.39 (CR, NR)
2002	European Indoor Championships	Vienna, Austria	1st	4x400 m relay	3:05.50 (CR)
2002	European Championships	Munich, Germany	4th	400 m	45.40
2002	European Championships	Munich, Germany	8th	4x400 m relay	DQ
2003	World Indoor Championships	Birmingham, United Kingdom	7th (sf)	400 m	46.82
2003	World Indoor Championships	Birmingham, United Kingdom	3rd	4x400 m relay	3:06.61
2003	European U23 Championships	Bydgoszcz, Poland	1st	400 m hurdles	48.45
2003	European U23 Championships	Bydgoszcz, Poland	1st	4x400 m relay	3:03.32
2004	Olympic Games	Athens, Greece	6th	400 m hurdles	49.00
2004	Olympic Games	Athens, Greece	10th (h)	4x400 m relay	3:03.69
2006	European Championships	Gothenburg, Sweden	2nd	400 m hurdles	48.71
2007	World Championships	Osaka, Japan	3rd	400 m hurdles	48.12 (NR)
2007	World Championships	Osaka, Japan	3rd	4x400 m relay	3:00.05
2008	Olympic Games	Beijing, China	6th	400 m hurdles	48.42
2008	Olympic Games	Beijing, China	7th	4x400 m relay	3:00.32
2012	European Championships	Helsinki, Finland	18th (sf)	400 m hurdles	50.77

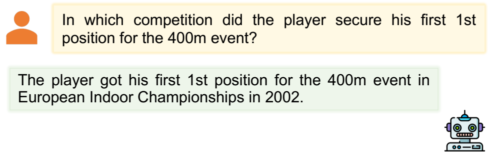

Figure 3: An example of F2WTQ, where models need to perform human-like reasoning to generate response.

F2WTQ Dataset Construction We adopt the WTQ dataset (Pasupat and Liang, 2015) as a basis to construct F2WTQ. The WTQ dataset is a short-form table question answering dataset, which includes human-annotated questions based on Wikipedia tables and requires complex reasoning. However, we do not directly use WTQ for LLM evaluation because, in real-world scenarios, users typically prefer a natural language response over a few words. In the development of F2WTQ, for each QA pair in the WTQ test set, we assign an annotator who assumes the role of an agent that analyzes the table and provides an expanded, sentence-long response. We found that the original questions in the WTQ dataset occasionally contained grammatical errors or lacked a natural linguistic flow. In these cases, the annotators are required to rewrite the question to ensure it was fluent and natural.

3 Evaluation System

3.1 Automated Evaluation

We adopt following popular evaluation metrics for automated evaluation:

- **BLEU** (Papineni et al., 2002) uses a precision-based approach, measuring the n-gram matches between the generated and reference statements.
- **ROUGE** (Lin, 2004) uses a recall-based approach, and measures the percentage of overlapping words and phrases between the generated output and reference one.
- **SP-Acc** (Chen et al., 2020a) extracts the meaning representation from the generated sentence and executes it against the table to verify correctness.
- **NLI-Acc** (Chen et al., 2020a) uses TableBERT fine-tuned on the TabFact dataset (Chen et al., 2020b) as faithfulness classifier.
- **TAPAS-Acc** (Liu et al., 2022a) uses TAPAS (Herzig et al., 2020) fine-tuned on the TabFact dataset as the backbone.
- **TAPEX-Acc** (Liu et al., 2022a) employs TAPEX (Liu et al., 2022b) fine-tuned on the TabFact dataset as the backbone. Recent works (Liu et al., 2022a; Zhao et al., 2023b) have revealed that NLI-Acc and TAPAS-Acc is overly positive about the predictions, while TAPEX-Acc serves as a more reliable faithfulness-level metric.
- **Exact Match & F-Score for Logical Reasoning Type** For LoTNLG evaluation, the exact match measures the percentage of samples with all the labels classified correctly, while the F-Score provides a balanced metric that considers both type I and type II errors.
- **Answer Accuracy** refers to the proportion of correct predictions out of the total number of predictions in F2WTQ generation.

3.2 Human Evaluation

To gain a more comprehensive understanding of the system's performance, we also conduct human evaluation. Specifically, the generated statements from different models are evaluated by humans based on two criteria: *faithfulness* and *fluency*. For *faithfulness*, each sentence is scored 0 (refuted) or 1 (entailed). For *fluency*, scores range from 1 (worst) to 5 (best). We average the scores across different human evaluators for each criterion. We do not apply more fine-grained scoring scales for *faithfulness*-level evaluation, as each statement in LogicNLG consists of only a single sentence.

4 Experiments

In the following subsections, we discuss the three key research questions about adopting LLMs into real-world table information seeking scenarios. Specifically, we explore LLMs' capabilities for table-to-text generation tasks, their ability to assess factual consistency, and whether they can benefit smaller fine-tuned models. The examined systems for each experiment are discussed in Appendix B.

Type	Models	SP-Acc	NLI-Acc	TAPAS-Acc	TAPEX-Acc
Fine-tuned	GPT2-C2F	43.6	71.4	46.2	43.8
	R2D2	53.2	86.2	60.2	61.0
	PLOG	52.8	84.2	63.8	**69.6**
	LOFT	**53.8**	**86.6**	**67.4**	61.4
0-shot*	GPT-3.5	**54.2**	87.6	81.6	79.4
	GPT-4	43.2	**90.4**	**91.8**	**91.0**
1-shot Direct	GPT-3.5	**60.2**	79.0	80.4	79.2
	GPT-4	57.6	**82.0**	**87.6**	**88.0**
1-shot CoT	GPT-3.5	51.6	70.0	81.8	78.2
	GPT-4	**59.8**	**80.8**	**89.4**	**90.8**
2-shot Direct	Pythia-12b	39.4	53.2	39.4	40.4
	LLaMA-13b	47.2	58.4	47.0	43.2
	LLaMA-7b	38.6	63.4	45.8	43.6
	LLaMA2-70b-chat	56.0	52.4	54.6	52.4
	LLaMA-30b	45.4	55.8	53.8	53.0
	Alpaca-13b	44.0	70.6	58.0	54.6
	LLaMA-65b	52.2	57.2	58.4	56.8
	TÜLU -13b	44.4	68.4	63.4	59.6
	Vicuna-13b	51.8	71.4	66.2	65.2
	GPT-3.5	**64.0**	78.4	78.8	81.2
	GPT-4	55.4	**85.8**	**92.0**	**89.6**
2-shot CoT	Pythia-12b	41.8	54.0	41.2	42.8
	LLaMA-7b	38.0	63.2	48.0	43.0
	LLaMA-13b	44.2	53.2	49.2	48.6
	LLaMA-30b	45.0	56.6	60.8	54.2
	LLaMA-65b	48.0	58.8	57.4	57.4
	TÜLU -13b	46.0	69.8	61.6	58.8
	Vicuna-13b	44.6	70.8	63.0	61.6
	Alpaca-13b	45.4	68.2	64.0	64.0
	LLaMA2-70b-chat	52.6	66.8	69.4	69.2
	GPT-3.5	60.4	70.2	84.0	83.4
	GPT-4	**62.2**	**76.8**	**88.8**	**90.4**

Table 2: Faithfulness-level automated evaluation results on the LOGICNLG dataset. Within each experimental setting, we used TAPEX-Acc as the ranking indicator of model performance. *: It is challenging for other LLMs to follow the instructions in 0-shot prompt to generate five statements for the input table.

4.1 RQ1: How do LLMs perform in table-to-text generation tasks?

We experiment with two in-context learning methods, *Direct Prediction* (Figure 5 in Appendix) and *Chain of Thoughts* (CoT, Figure 6 in Appendix), to solve the table-to-text generation tasks.

Data Insight Generation Results The results on the LOGICNLG dataset, as displayed in Table 2 and Table 3, indicate that GPT-* models generally surpass the current top-performing fine-tuned models (i.e., LOFT and PLOG) even in a 0-shot setting. Meanwhile, LLaMA-based models (e.g., LLaMA, Alpaca, Vicuna, TÜLU) manage to achieve comparable performance to these top-performing fine-tuned models in a 2-shot setting. However, when it comes to the more challenging LOTNLG dataset, the automated evaluation result shows that only GPT-4 is capable of generating faithful statements that adhere to the specified logical reasoning types (Table 6 in Appendix). Moreover, increasing the number of shots or applying chain-of-thought approach does not always yield a performance gain, motivating us to explore more advanced prompting methods for data insight generation in future work.

Query-based Generation Results Table 7 and 8 in Appendix display the automated evaluation results for the FeTaQA and F2WTQ datasets, respectively. On FeTaQA, both LLaMA-based LLM and GPT-* models achieve comparable performance to the current top-performing fine-tuned models in a 2-shot setting, indicating the capability of LLMs to answer questions requiring surface-level facts from the table. However, a significant performance gap exists between other LLMs and GPT-* models on the more challenging F2WTQ dataset. Moreover, increasing the number of shots or applying

Model	Fluency (1-5)	Faithfulness (0-1)
GPT2-C2F	3.85	0.54
R2D2	4.29	0.72
PLOG	4.23	0.77
LOFT	4.42	0.81
GPT-4 0-shot	**4.82**	0.90
Vicuna 2-shot Direct	4.69	0.71
Vicuna 2-shot CoT	4.65	0.73
LLaMA2 2-shot Direct	4.75	0.79
LLaMA2 2-shot CoT	4.70	0.83
GPT-4 2-shot Direct	4.71	0.89
GPT-4 2-shot CoT	4.77	**0.92**

Table 3: Human evaluation results on LOGICNLG.

Metric	Acc on Tabfact	Pearson's correlation
SP-Acc	63.5	.458
NLI-Acc	65.1	.526
TAPAS-Acc	81.0	.705
TAPEX-Acc	**84.2**	.804
CoT-3.5-Acc	78.0	.787
CoT-4-Acc	80.9	**.816**

Table 4: System-level Pearson's correlation bettwen each automated evaluation metric and human judgement. We also report the accuracy of automated evaluation metrics on the TabFact dataset for reference.

the chain-of-thought approach can both yield performance gains for query-based generation.

4.2 RQ2: Can we use LLMs to assess factual consistency of table-to-text generation?

In RQ1, we demonstrate that LLMs can generate statements with comparative or even greater factual consistency than fine-tuned models. One natural follow-up question is whether we can employ LLMs to evaluate the faithfulness of table-to-text generation systems. This capability is crucial, as it ensures that tabular data is accurately interpreted for users, thereby preserving the credibility and reliability of real-world applications.

As discussed in Section 3.1, existing faithfulness-level NLI-based metrics are trained on the TabFact dataset (Chen et al., 2020b). Recent work (Chen, 2023) has revealed that large language models using chain-of-thought prompting can achieve competitive results on TabFact. Motivated by this finding, we use the same *2-shot chain-of-thought prompt* (Figure 7 in Appendix) as Chen (2023) to generate factual consistency scores (0 for refuted and 1 for entailed) for output sentences from LogicNLG. We use GPT-3.5 and GPT-4 as the backbones, as they outperforms other LLMs in RQ1 experiments. We refer to these new metrics as *CoT-3.5-Acc* and *CoT-4-Acc*, respectively.

CoT-Acc Metrics Achieve Better Correlation with Human Judgement We leverage the human evaluation results of models (excluding GPT-4 models) in RQ1 as the *human judgement*. We then compare the system-level Pearson's correlation between each evaluation metric and this human judgement. As shown in Table 4, the proposed CoT-4-Acc and CoT-3.5-Acc metrics achieve the highest and third highest correlation with human judgement, respectively. This result demonstrates LLMs' capabilities in assessing the faithfulness of table-to-text generation. It's worth noting that although TAPAS-Acc and TAPEX-Acc perform better than CoT-4-Acc on the TabFact dataset, they exhibit lower correlation with human judgement on table-to-text evaluation. We suspect that this can be largely attributed to over-fitting on the TabFact dataset, where negative examples are created by rewriting from the positive examples. We believe that future work can explore the development of a more robust faithfulness-level metric with better alignment to human evaluation.

4.3 RQ3: How can fine-tuned models benefit from LLMs' strong table-to-text abilities?

In RQ1 and RQ2, we demonstrate the strong capability of state-of-the-art LLMs in table-to-text generation and evaluation. We next explore how fine-tuned smaller models can benefit from these abilities. We believe such exploration can provide insights for future work regarding the distillation of text generation capabilities from LLMs to smaller models (Gao et al., 2023a; Scheurer et al., 2023; Madaan et al., 2023). This is essential as deploying smaller, yet performance-comparable models in real-world applications could save computational resources and inference time.

Generating Feedback for Improving Factual Consistency Utilizing human feedback to enhance neural models has emerged as a significant area of interest in contemporary research (Liu et al., 2022c; Gao et al., 2023a; Scheurer et al., 2023; Madaan et al., 2023). For example, Liu et al. (2022c) illustrates that human-written feedback can be leveraged to improve factual consistency of text summarization systems. Madaan et al. (2023) demonstrates that LLMs can improve their initial outputs through iterative feedback and refinement. This work investigates whether LLMs can provide

Models	TAPAS-Acc	TAPEX-Acc
GPT2-C2F	46.2	43.8
Edit by LLaMA2-70b-chat	58.0 (+11.8)	50.0 (+6.2)
Edit by GPT-3.5	71.0 (+24.8)	68.4 (+24.6)
Edit by GPT-4	81.0 (+34.8)	82.0 (+38.2)
R2D2	60.2	61.0
Edit by LLaMA2-70b-chat	65.0 (+4.8)	60.0 (-1.0)
Edit by GPT-3.5	74.0 (+13.8)	74.0 (+13.0)
Edit by GPT-4	87.0 (+26.8)	89.0 (+28.0)
PLOG	63.8	69.6
Edit by LLaMA2-70b-chat	75.0 (+11.2)	66.0 (-3.6)
Edit by GPT-3.5	70.6 (+6.8)	67.0 (-2.6)
Edit by GPT-4	91.0 (+27.2)	86.0 (+16.4)
LoFT	67.4	61.4
Edit by LLaMA2-70b-chat	72.0 (+4.6)	64.0 (+2.6)
Edit by GPT-3.5	70.0 (+2.6)	65.6 (+4.2)
Edit by GPT-4	81.0 (+13.6)	86.0 (+24.6)

Table 5: Automated evaluation results on LogicNLG using statements pre-edited and post-edited by LLMs.

human-like feedback for outputs from fine-tuned models. Following Liu et al. (2022c), we consider generating feedback with three components: 1) *Explanation*, which determine whether the initial statement is factually consistent with the given table; 2) *Corrective Instruction*, which provide instructions on how to correct the initial statement if it is detected as unfaithful; and 3) *Edited Statement*, which edits the initial statement following the corrective instruction. Figure 8 in Appendix shows an example of *2-shot chain-of-thought* prompts we use for feedback generation.

Feedback from LLMs is of High Quality We assess the quality of generated feedback through automated evaluations. Specifically, we examine the faithfulness scores of *Edited Statements* in the generated feedback, comparing these scores to those of the original statements. We report TAPAS-Acc and TAPEX-Acc for experimental results, as these two metrics exhibit better alignment with human evaluation (Section 4.2). As illustrated in Table 5, LLMs can effectively edit statements to improve their faithfulness, particularly for outputs from lower-performance models, such as GPT2-C2F.

5 Related Work

Table-to-Text Generation Text generation from semi-structured knowledge sources, such as web tables, has been studied extensively in recent years (Parikh et al., 2020; Chen et al., 2020a; Cheng et al., 2022; Zhao et al., 2023a). The goal of the table-to-text generation task is to generate natural language statements that faithfully describe information contained in the provided table region. The most popular approach for table-to-text generation tasks is to fine-tune a pre-trained language model on a task-specific dataset (Chen et al., 2020a; Liu et al., 2022a; Zhao et al., 2022; Nan et al., 2022a; Zhao et al., 2023b). To the best of our knowledge, we are the first to systematically evaluate the performance of LLMs on table-to-text generation tasks.

Large Language Models LLMs have demonstrated remarkable in-context learning capabilities (Brown et al., 2020; Chowdhery et al., 2022; Scao et al., 2022; Chung et al., 2022; OpenAI, 2023), where the model receives a task demonstration in natural language accompanied by a limited number of examples. The Chain-of-Thought prompting methods (Wei et al., 2022; Wang et al., 2022) further empower LLMs to perform complex reasoning tasks (Han et al., 2022; Zhao et al., 2023c; Ye et al., 2023; Chen, 2023). More recent works (Chen, 2023; Nan et al., 2023) investigate in-context learning capabilities of LLMs on table-based tasks, including table question answering (Pasupat and Liang, 2015; Iyyer et al., 2017; Zhong et al., 2018) and table fact checking (Chen et al., 2020b; Gupta et al., 2020). However, the potential of LLMs in generating text from tabular data remains underexplored.

6 Conclusion

This paper investigates the potential of applying LLMs in real-world table information seeking scenarios. We demonstrate their superiority in faithfulness, and their potential as evaluation systems. Further, we provide valuable insights into leveraging LLMs to generate high-fidelity natural language feedback. We believe that the findings of this study could benefit real-world applications, aimed at improving user efficiency in data analysis.

Ethical Consideration

LoTNLG and F2WTQ were constructed upon the test set of LogicNLG (Chen et al., 2020a) and WTQ (Pasupat and Liang, 2015) datasets, which are publicly available under the licenses of MIT[1] and CC BY-SA 4.0[2], respectively. These licenses permit us to modify, publish, and distribute additional annotations upon the original dataset.

[1] https://opensource.org/licenses/MIT

[2] https://creativecommons.org/licenses/by-sa/4.0/

References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

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

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)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, et al. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Ge Gao, Hung-Ting Chen, Yoav Artzi, and Eunsol Choi. 2023a. Continually improving extractive qa via human feedback.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023b. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.

Carlos Gemmell and Jeffrey Stephen Dalton. 2023. Generate, transform, answer: Question specific tool synthesis for tabular data. *ArXiv*, abs/2303.10138.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq R. Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir R. Radev. 2022. Folio: Natural language reasoning with first-order logic. *ArXiv*, abs/2209.00840.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. Longform: Optimizing instruction tuning for long text generation with corpus extraction. *arXiv preprint arXiv:2304.08460*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. 2023. Sheetcopilot: Bringing software productivity to the next level through large language models. *ArXiv*, abs/2305.19308.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022a. PLOG: Table-to-logic pretraining for logical table-to-text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5531–5546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022b. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron L Halfaker, Dragomir R. Radev, and Ahmed Hassan Awadallah. 2022c. On improving summarization factual consistency from natural language feedback. *ArXiv*, abs/2212.09968.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Linyong Nan, Lorenzo Jaime Flores, Yilun Zhao, Yixin Liu, Luke Benson, Weijin Zou, and Dragomir Radev. 2022a. R2D2: Robust data-to-text with replacement detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6903–6917, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. 2022b. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Linyong Nan, Yilun Zhao, Weijin Zou, Narutatsu Ri, Jaesung Tae, Ellen Zhang, Arman Cohan, and Dragomir Radev. 2023. Enhancing few-shot text-to-sql capabilities of large language models: A study on prompt design strategies.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur 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)*, pages 1173–1186, Online. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Yotam Perlitz, Liat Ein-Dor, Dafna Sheinwald, Noam Slonim, and Michal Shmueli-Scheuer. 2022. Diversity enhanced table-to-text generation via type control. *ArXiv*, abs/2205.10938.

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

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

J'er'emy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *ArXiv*, abs/2303.16755.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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 Canton 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. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning. *ArXiv*, abs/2301.13808.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, Tao Zhang, Chen Zhou, Kaizhe Shou, Miao Wang, Wufang Zhu, Guoshan Lu, Chao Ye, Yali Ye, Wentao Ye, Yiming Zhang, Xinglong Deng, Jie Xu, Haobo Wang, Gang Chen, and Junbo Zhao. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yue Ting Zhuang. 2023. Data-copilot: Bridging billions of data and humans with autonomous workflow. *ArXiv*, abs/2306.07209.

Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. 2022. Macsum: Controllable summarization with mixed attributes. *arXiv preprint arXiv:2211.05041*.

Yilun Zhao, Boyu Mi, Zhenting Qi, Linyong Nan, Minghao Guo, Arman Cohan, and Dragomir Radev. 2023a. OpenRT: An open-source framework for reasoning over tabular data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 336–347, Toronto, Canada. Association for Computational Linguistics.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022. ReasTAP: Injecting table reasoning skills during pre-training via synthetic reasoning examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9006–9018, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Lorenzo Jaime Flores, and Dragomir Radev. 2023b. LoFT: Enhancing faithfulness and diversity for table-to-text generation via logic form control. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 554–561, Dubrovnik, Croatia. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Xiangru Tang, Yumo Xu, Arman Cohan, and Dragomir Radev. 2023c. Qtsumm: A new benchmark for query-focused table summarization.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Seq2SQL: Generating structured queries from natural language using reinforcement learning.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. *arXiv preprint arXiv:2303.11315*.

A Table-to-Text Generation Benchmarks

A.1 LoTNLG Dataset

Logical Reasoning Type Definition

- Aggregation: operations involving sum or average operation to summarize the overall statistics. Sentence: The total number of scores of xxx is xxx. The average value of xxx is xxx.
- Negation: operations to negate. Sentence: xxx did not get the first prize.
- Superlative: superlative operations to get the highest or lowest value. Sentence: xxx achieved the most scores.
- Count: operations to count the amount of entities that fulfil certain conditions. Sentence: There are 4 people born in xxx.
- Comparative: operations to compare a specific aspect of two or more entities. Sentence: xxx is taller than xxx.
- Ordinal: operations to identify the ranking of entities in a specific aspect. Sentence: xxx is the third youngest player in the game.
- Unique: operations to identify different entities. Sentence: The players come from 7 different cities.
- All: operations to summarize what all entities do/have in common. Sentence: All of the xxx are more expensive than $25.
- Surface-Level: no logical reasoning type above. Sentence: xxx is moving to xxx.

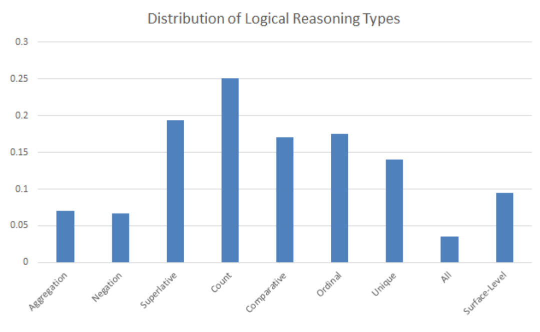

Figure 4: Distribution of logical reasoning types for the LoTNLG dataset.

B Examined Systems

B.1 Fine-tuned Models

- **BART** (Lewis et al., 2020) is a pre-trained denoising autoencoder with transformer-based architecture and shows effectiveness in NLG tasks.
- **Flan-T5** (Chung et al., 2022) enhances T5 (Raffel et al., 2020) by scaling instruction fine-tuning and demonstrates better human-like reasoning abilities than the T5.
- **GPT2-C2F** (Chen et al., 2020a) first generates a template which determines the global logical structure, and then produces the statement using the template as control.
- **R2D2** (Nan et al., 2022a) trains a generative language model both as a generator and a faithfulness discriminator with additional replacement detection and unlikelihood learning tasks, to enhance the faithfulness of table-to-text generation.
- **TAPEX** (Liu et al., 2022b) continues pre-training the BART model by using a large-scale corpus of synthetic SQL query execution data, showing better table understanding and reasoning abilities.
- **OmniTab** (Jiang et al., 2022) uses the same backbone as TAPEX, and is further pre-trained on collected natural and synthetic Table QA examples.
- **ReasTAP** (Zhao et al., 2022) enhances the table understanding and reasoning abilities of BART by pre-training on a synthetic Table QA corpus.
- **PLOG** (Liu et al., 2022a) continues pre-training text generation models on a table-to-logic-form generation task (i.e., T5 model), improving the faithfulness of table-to-text generation.
- **LoFT** (Zhao et al., 2023b) utilizes logic forms as fact verifiers and content planners to control table-to-text generation, exhibiting improved faithfulness and text diversity.

B.2 Large Language Models

- **Pythia** (Biderman et al., 2023) is a suite of 16 open-sourced LLMs all trained on public data in the exact same order and ranging in size from 70M to 12B parameters. This helps researchers to gain a better understanding of LLMs and their training dynamics.
- **LLaMA** (Touvron et al., 2023a,b) is an open-source LLM trained on large-scale and publicly available datasets. We evaluate both LLaMA and LLaMA2 in this paper.
- **Alpaca** (Taori et al., 2023) and **Vicuna** (Chiang et al., 2023) are fine-tuned from LLaMA with instruction-following data, exhibiting better instruction-following capabilities.

- **TÜLU** (Wang et al., 2023) further trains LLaMA on 12 open-source instruction datasets, achieving better performance than LLaMA.
- **GPT** (Brown et al., 2020; Wei et al., 2022) is a powerful large language model which is capable of generating human-like text and performing a wide range of NLP tasks in a few-shot setting. We use the OpenAI engines of `gpt-3.5-0301` and `gpt-4-0314` for GPT-3.5 and GPT-4 models, respectively.

To formulate the prompt, we linearize the table as done in previous work on table reasoning (Chen, 2023) and concatenate it with its corresponding reference statements as demonstrations. We use the table truncation strategy as proposed by Liu et al. (2022b) to truncate large table and ensure that the prompts are within the maximum token limitation for each type of LLMs. For LLM parameter settings, we used a temperature of 0.7, maximum output length of 512, without any frequency or presence penalty.

C Experiments

```
Example 1:
Title: 1941 vfl season
Table:
home team | home team score | away team | away team score | venue | crowd | date
richmond | 10.13 (73) | st kilda | 6.11 (47) | punt road oval | 6000 | 21 june 1941
hawthorn | 6.8 (44) | melbourne | 12.12 (84) | glenferrie oval | 2000 | 21 june 1941
collingwood | 8.12 (60) | essendon | 7.10 (52) | victoria park | 6000 | 21 june 1941
carlton | 10.17 (77) | fitzroy | 12.13 (85) | princes park | 4000 | 21 june 1941
south melbourne | 8.16 (64) | north melbourne | 6.6 (42) | lake oval | 5000 | 21 june 1941
geelong | 10.18 (78) | footscray | 13.15 (93) | kardinia park | 5000 | 21 june 1941

Five generated statements:
1. footscray scored the most point of any team that played on 21 june, 1941.
2. geelong was the home team with the highest score.
3. kardinia park was the one of the six venues that were put to use.
4. north melbourne away team recorded an away score of 6.6 (42) while melbourne
   recorded an away score of 12.12 (84).
5. all six matches took place on 21 june 1941.

Example 2:
Title: {title}
Table:
{table}
```

Figure 5: An example of 1-shot *direct-prediction* prompting for the LOGICNLG task.

```
[INSTRUCTION] Your task is to provide 5 different consistent statements derived from a
table. Consistent means that all information of your statements should be supported by the
corresponding table. Provided 5 statements should be different from each other.
To guide your responses, we have provided two example tables with five statements each.
Use the template to structure your answer, provide reasoning for your statements and
suggest statements. We encourage you to think through each step of the process carefully.

Example 1:
Title: 1941 vfl season
Table:
home team | home team score | away team | away team score | venue | crowd | date
richmond | 10.13 (73) | st kilda | 6.11 (47) | punt road oval | 6000 | 21 june 1941
hawthorn | 6.8 (44) | melbourne | 12.12 (84) | glenferrie oval | 2000 | 21 june 1941
collingwood | 8.12 (60) | essendon | 7.10 (52) | victoria park | 6000 | 21 june 1941
carlton | 10.17 (77) | fitzroy | 12.13 (85) | princes park | 4000 | 21 june 1941
south melbourne | 8.16 (64) | north melbourne | 6.6 (42) | lake oval | 5000 | 21 june 1941
geelong | 10.18 (78) | footscray | 13.15 (93) | kardinia park | 5000 | 21 june 1941

Reasoning 1: looking at both "home team score" column and "away team score" column,
finding the highest score was 13.15 (93) in "away team score" column and then looking for
which team scored 13.15 (93) in "away team" colmun, footscray scored the most point of
any team that played on 21 june.
Statement 1: footscray scored the most point of any team that played on 21 june, 1941.

Reasoning 2: looking at "home team" column and finding the corresponding home team
scores of geelong in "home team score" column, geelong did have the highest score.
Statement 2: geelong was the home team with the highest score.

Reasoning 3: looking at "venue" column, kardinia park was the one of six venues.
Statement 3: kardinia park was the one of the six venues that were put to use.

Reasoning 4: looking at "away team" column and finding the corresponding away team
scores of north melbourne and melbourne in "away team score" column, north melbourne
as away team scored 6.6 (42) while melbourne as away team scored 12.12 (84).
Statement 4: north melbourne away team recorded an away score of 6.6 (42) while
melbourne recorded an away score of 12.12 (84).

Reasoning 5: looking at "date" column, all six matches took place on 21 june 1941.
Statement 5: all six matches took place on 21 june 1941.

Now please give 5 different consistent claims of the new table. Let's think step by step and
follow the given examples.

Title: {title}
Table:
{table}
```

Figure 6: An example of 1-shot *chain-of-thought* prompting for the LOGICNLG task.

```
Read the table below regarding "1919 in brazilian football" to verify whether the provided
claims are true or false.

Table:
date | result | score | brazil scorers | competition
may 11 , 1919 | w | 6 - 0 | friedenreich (3) , neco (2) , haroldo | south american
championship
may 18 , 1919 | w | 6 - 1 | heitor , amílcar (4), millon | south american championship
may 26 , 1919 | w | 5 - 2 | neco (5) | south american championship
may 30 , 1919 | l | 1 - 2 | jesus (1) | south american championship
june 2nd , 1919 | l | 0 - 2 | - | south american championship

Statement: neco has scored a total of 7 goals in south american championship.
Explanation: neco has scored 2 goals on may 11 and 5 goals on may 26. neco has scored
a total of 7 goals, therefore, the claim is true.

Statement: jesus has scored in two games in south american championship.
Explanation: jesus only scored once on the may 30 game, but not in any other game,
therefore, the claim is false.

Statement: brazilian football team has scored six goals twice in south american
championship.
Explanation: brazilian football team scored six goals once on may 11 and once on may 18,
twice in total, therefore, the claim is true.

Read the table below regarding
(...abbreviate the second prompting example...)

Read the table below regarding "{title}" to verify whether the provided claims are true or
false.

Table:
{table}

Statement: {statement_i}
```

Figure 7: An example of 2-shot chain-of-thought prompting adopted from Chen (2023) for faithfulness-level automated evaluation.

Type	Models	SP-Acc	NLI-Acc	TAPAS-Acc	TAPEX-Acc	Type EM	Type F1
0-shot*	GPT-3.5	51.2	77.2	70.8	66.8	59.2	43.8
	GPT-4	**69.2**	**79.4**	**85.6**	**84.2**	**75.2**	**60.0**
1-shot Direct	GPT-3.5	53.8	75.6	71.6	71.0	51.2	38.1
	GPT-4	**60.2**	**72.8**	**83.8**	**84.2**	**76.6**	**63.0**
1-shot CoT	GPT-3.5	50.8	**78.8**	79.2	79.4	46.2	30.2
	GPT-4	**59.2**	74.8	**84.4**	**85.8**	**70.0**	**51.6**
2-shot Direct	Pythia-12b	44.2	60.6	41.8	43.0	19.0	12.2
	LLaMA-7b	41.0	62.2	46.2	46.2	18.2	13.4
	Vicuna-13b	48.6	71.2	57.4	54.4	22.0	15.2
	LLaMA-13b	44.6	62.4	50.8	48.8	22.6	15.8
	Alpaca-13b	46.2	73.8	50.8	54.0	21.8	15.8
	LLaMA2-70b-chat	44.2	60.0	56.0	58.0	24.2	15.8
	LLaMA-30b	40.0	62.6	53.0	52.6	24.2	16.4
	LLaMA-65b	46.2	57.8	54.0	51.8	21.0	17.2
	TÜLU -13b	44.2	72.8	60.8	56.8	26.6	17.4
	GPT-3.5	55.2	**76.2**	70.8	67.6	52.2	35.0
	GPT-4	**61.4**	72.2	**84.6**	**83.2**	**73.4**	**54.8**
2-shot CoT	Pythia-12b	42.0	53.8	41.2	41.0	15.2	11.6
	LLaMA-30b	41.0	60.4	52.6	59.2	20.4	13.2
	LLaMA-7b	37.6	61.2	43.8	45.0	17.2	13.4
	LLaMA2-70b-chat	48.2	64.6	56.0	67.8	20.2	13.4
	LLaMA-13b	45.0	56.6	51.2	51.2	18.8	14.0
	LLaMA-65b	45.2	62.4	59.4	58.8	21.2	15.2
	Vicuna-13b	43.4	72.0	62.2	61.0	18.4	16.0
	Alpaca-13b	40.4	71.6	58.4	57.8	23.0	16.2
	TÜLU -13b	45.8	65.8	60.8	61.0	23.2	16.2
	GPT-3.5	49.2	**74.4**	77.2	75.4	49.4	35.0
	GPT-4	**59.2**	72.0	**85.6**	**83.2**	**67.6**	**55.6**

Table 6: Faithfulness-level automated evaluation results on LOTNLG. We do not evaluate fine-tuned models as LOTNLG does not contain a training set. *: It is challenging for other LLMs to follow the instructions in 0-shot prompt to generate a statement using the specified types of logical reasoning operations.

Type	Models	BLEU-1/2/3	ROUGE-1/2/L	TAPAS-Acc	TAPEX-Acc
Fine-tuned	BART	63.2/50.8/42.0	**67.6/46.0/57.2**	94.8	68.8
	Flan-T5	62.2/49.6/41.0	66.8/45.0/56.2	94.2	69.2
	OmniTab	63.4/50.8/41.8	67.4/45.2/56.2	94.6	71.6
	ReasTAP	**63.6/51.0/42.2**	**67.6**/45.8/**57.2**	94.6	71.4
	TAPEX	**63.6**/50.8/42.0	66.4/45.0/56.2	**96.2**	**73.0**
0-shot	GPT-3.5	**56.4/42.6/33.4**	60.6/38.0/49.4	92.4	72.8
	GPT-4	52.4/40.2/31.8	**63.8/40.4/51.6**	**94.0**	**74.4**
1-shot Direct	GPT-3.5	**56.8**/43.2/34.2	63.0/39.8/51.4	91.8	**74.6**
	GPT-4	56.4/**43.6/34.8**	**66.2/43.0/54.4**	**94.0**	73.8
1-shot CoT	GPT-3.5	43.2/32.4/25.2	57.4/35.8/46.8	**94.2**	67.0
	GPT-4	**59.6/45.8/36.4**	**64.0/41.0/52.4**	91.0	**76.4**
2-shot Direct	Pythia-`12b`	38.8/26.6/19.4	43.2/22.6/35.2	76.6	35.0
	LLaMA-`7b`	40.6/28.6/21.4	48.2/26.6/39.0	86.2	47.8
	LLaMA-`13b`	48.4/35.2/26.8	51.0/29.4/42.2	85.4	57.4
	Alpaca-`13b`	52.2/38.4/29.6	56.4/33.6/46.2	88.4	57.4
	TÜLU -`13b`	50.6/37.4/29.0	54.2/31.8/44.6	86.4	60.0
	LLaMA-`30b`	50.4/37.0/28.2	56.2/33.2/45.4	87.0	60.2
	Vicuna-`13b`	56.0/42.2/32.8	59.0/36.2/48.0	87.6	62.4
	LLaMA-`65b`	53.6/39.8/30.8	57.0/34.0/46.6	88.4	63.0
	LLaMA2-`70b-chat`	54.6/41.0/31.8	58.4/35.8/47.8	89.4	66.2
	GPT-4	55.0/**42.8/34.6**	**66.0/42.8/54.0**	**95.2**	75.8
	GPT-3.5	**55.8/42.8**/34.0	63.2/40.0/51.6	92.2	**76.0**
2-shot CoT	Pythia-`12b`	38.8/25.4/17.8	39.2/18.8/32.2	69.0	36.2
	LLaMA-`7b`	33.0/22.2/16.0	41.0/21.2/33.2	77.6	42.0
	LLaMA-`13b`	43.2/30.4/22.6	45.4/25.2/37.6	82.0	50.8
	Alpaca-`13b`	47.4/34.4/26.2	51.4/30.0/42.0	82.8	54.4
	TÜLU -`13b`	37.0/25.8/18.8	43.6/24.0/35.2	86.2	55.8
	LLaMA-`30b`	45.4/33.2/25.6	52.4/30.8/42.2	86.2	63.6
	Vicuna-`13b`	50.4/37.6/29.4	53.8/32.4/44.6	85.6	65.8
	LLaMA-`65b`	50.2/37.0/28.4	54.8/32.8/44.6	87.8	66.0
	LLaMA2-`70b-chat`	53.8/40.2/31.4	57.4/34.8/47.0	89.2	66.2
	GPT-3.5	50.8/38.8/30.8	60.6/38.2/49.0	**92.8**	70.8
	GPT-4	**62.2/48.6/39.2**	**65.8/42.8/54.4**	91.2	**79.2**

Table 7: Automated evaluation results on the FeTaQA dataset.

Type	Models	BLEU-1/2/3	ROUGE-1/2/L	TAPAS-Acc	TAPEX-Acc	Accuracy
0-shot	GPT-3.5	**63.2/49.2/39.4**	64.4/40.0/**56.4**	73.0	74.6	54.0
	GPT-4	60.6/46.8/37.4	**64.6/40.4**/54.8	**78.6**	**80.6**	**62.4**
1-shot Direct	GPT-3.5	62.0/48.4/39.0	64.0/40.0/56.8	75.0	73.2	51.8
	GPT-4	**63.2/49.8/40.4**	**66.2/42.6/58.0**	**78.4**	**79.0**	**66.0**
1-shot CoT	GPT-3.5	55.0/42.4/33.8	62.8/39.0/54.8	72.4	72.2	55.2
	GPT-4	62.2/49.0/39.6	**66.2/42.2/58.4**	**78.2**	**78.6**	**69.8**
2-shot Direct	Pythia-12b	12.4/7.6/5.2	19.6/9.2/17.4	74.6	62.4	7.8
	LLaMA-7b	14.4/9.6/6.8	26.2/13.4/23.0	71.8	53.0	19.0
	LLaMA-13b	7.6/4.8/3.4	20.2/10.4/18.2	78.4	56.0	21.4
	Vicuna-13b	43.0/31.6/24.4	46.0/27.2/40.6	74.6	64.2	30.2
	Alpaca-13b	40.8/29.2/21.6	46.6/26.2/40.4	71.8	57.6	31.2
	LLaMA-30b	34.0/24.4/18.2	44.6/25.0/39.8	74.0	61.0	31.8
	TÜLU -13b	49.6/36.4/28.0	51.4/29.4/45.8	78.8	60.4	33.8
	LLaMA-65b	45.8/33.8/26.0	48.8/28.2/43.6	73.6	64.4	36.2
	LLaMA2-70b-chat	51.2/38.4/30.0	50.4/29.6/45.4	72.4	68.4	37.6
	GPT-3.5	**63.4/49.8**/40.2	64.8/40.8/57.2	74.8	73.6	51.8
	GPT-4	62.8/49.2/39.6	**65.8/41.8/57.6**	**78.6**	**81.4**	**63.6**
2-shot CoT	Pythia-12b	27.2/18.0/12.8	35.6/17.4/31.4	66.0	48.8	15.8
	LLaMA-7b	13.2/8.4/5.8	28.0/13.2/24.0	73.4	47.8	24.2
	LLaMA-13b	22.2/14.8/10.4	35.2/18.0/31.4	74.0	56.2	26.2
	Alpaca-13b	33.2/23.6/17.8	47.6/26.4/41.2	75.0	55.4	32.2
	LLaMA-30b	37.4/26.2/19.6	46.2/24.8/40.6	72.6	60.0	35.6
	TÜLU -13b	25.8/17.0/12.0	35.4/17.4/31.0	**79.0**	65.6	35.8
	Vicuna-13b	45.2/33.2/25.4	53.6/31.2/47.6	75.6	62.2	38.6
	LLaMA-65b	51.2/37.8/29.0	51.6/29.4/45.6	75.6	67.6	41.6
	LLaMA2-70b-chat	46.2/34.2/26.6	49.6/28.8/44.2	75.8	66.6	43.2
	GPT-3.5	57.4/44.4/35.4	64.0/40.0/55.4	73.6	72.8	58.6
	GPT-4	**63.0/49.6/40.0**	**66.2/42.4/58.8**	76.4	**79.6**	**68.4**

Table 8: Automated evaluation results on the F2WTQ dataset. We do not evaluate fine-tuned models as F2WTQ does not contain a training set.

[INSTRUCTION] Your task is to provide feedback on statements derived from tables. Your feedback should consist of 1) Explanation, which determine whether the initial statement is factually consistent with the given table; 2) Corrective Instruction, which provide instructions on how to correct the initial statement if it is detected as unfaithful; and 3) Edited Statement, which edits the initial statement following the corrective instruction.
There are two types of errors: intrinsic and extrinsic. Intrinsic errors refer to mistakes that arise from within the statement itself, while extrinsic errors are caused by factors external to the statement. To help you provide accurate feedback, we have provided instruction templates for your use. These templates include "remove," "add," "replace," "modify," "rewrite," and "do nothing".
It is important to note that you should be capable of identifying logical operations when reviewing statements. Examples of such operations include superlatives, exclusives (such as "only"), temporal relationships (such as "before/after"), quantitative terms (such as "count" or "comparison"), inclusive/exclusive terms (such as "both/neither"), and arithmetic operations (such as "sum/difference" or "average").
To guide your responses, we have provided two examples with three statements each. Use these templates to structure your answer, provide reasoning for your feedback, and suggest improved statements. We encourage you to think through each step of the process carefully.
Remember, your final output should always include a “Edited Statement” no matter if there is error or not.

Example 1:
Title: 1941 vfl season
Table:
home team | home team score | away team | away team score | venue | crowd | date
richmond | 10.13 (73) | st kilda | 6.11 (47) | punt road oval | 6000 | 21 june 1941
hawthorn | 6.8 (44) | melbourne | 12.12 (84) | glenferrie oval | 2000 | 21 june 1941
collingwood | 8.12 (60) | essendon | 7.10 (52) | victoria park | 6000 | 21 june 1941
carlton | 10.17 (77) | fitzroy | 12.13 (85) | princes park | 4000 | 21 june 1941
south melbourne | 8.16 (64) | north melbourne | 6.6 (42) | lake oval | 5000 | 21 june 1941
geelong | 10.18 (78) | footscray | 13.15 (93) | kardinia park | 5000 | 21 june 1941

Statement: st kilda scored the most point of any team that played on 21 june, 1941
Explanation: footscray scored the most point of any team that played on 21 june, not st kilda. So the statement has instrinsic error.
Corrective Instruction: replace st kilda with footscray.
Edited Statement: footscray scored the most point of any team that played on 21 june, 1941.

Example 2:
(...abbreviate…)

Now please give feedback to the statement of the new table. Let's think step by step and follow the given example. Remember to include “Explanation”, “Corrective Instruction”, and “Edited Statement” parts in the output.

Title: {title}
Table:
{table}
Statement: {sent}

Figure 8: An example of 2-shot *chain-of-thought* prompts for natural language feedback generation on LOGICNLG.