STRUC-BENCH: Are Large Language Models Really Good at Generating Complex Structured Data?

Anonymous submission

Abstract

Despite the impressive capabilities of Large Language Models (LLMs) such as GPT-4, they still encounter challenges when it comes to generating complex, structured outputs. This study aims to assess the current capability of LLMs in generating structured data and proposes a novel structure-aware fine-tuning approach to enhance their ability in this aspect. Here we introduce STRUC-BENCH, a benchmark that includes representative LLMs (GPT-NeoX-20B, GPT-3.5, GPT-4, and Vicuna), encompassing text tables, HTML, and LaTeX formats. To construct the benchmark, we employ FORMATCOT (Chain-of-Thought) to generate format instructions from target outputs. Moreover, considering the lack of task-specific metrics, we introduce two novel metrics: P-Score (**P**rompting Score) and H-Score (**H**euristical Score). Experimental results demonstrate that our structure-aware fine-tuning approach, applied to LLaMA-7B, significantly improves adherence to natural language constraints, surpassing other evaluated LLMs. Our analysis reveals common errors and areas open for improvement. Accordingly, we present an ability map across six dimensions (coverage, formatting, reasoning, comprehension, pragmatics, and hallucination), suggesting promising directions for future research.

1 Introduction

Significant advancements have been made in various natural language processing tasks by Large Language Models (LLMs) (Brown et al., 2020; Scao et al., 2022; Ouyang et al., 2022; Muennighoff et al., 2022; OpenAI, 2023; Zhao et al., 2023a), especially in text generation tasks (Qin et al., 2023). The ability to output structured data, one of the key aspects of generative capability, has also attracted great interest in previous studies (Wu et al., 2022; Zhao et al., 2023c,b; Zha et al., 2023).

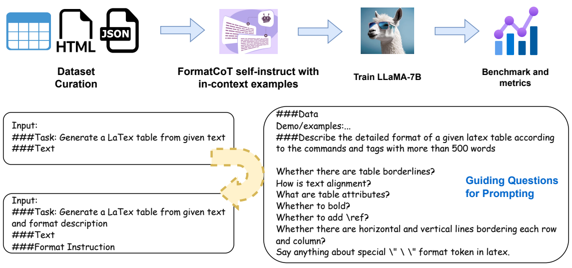


Figure 1: A system for describing complex structured formats and learning to follow this format in human language. We use zero-shot for inference.

However, LLMs still underperform in generating complex structured outputs – a critical ability for various applications ranging from coding assistance to automated report writing. Furthermore, most evaluation of LLMs has been on natural text or code generation, and relatively less research has been conducted to evaluate LLMs on their ability to generate structured output. This leaves it unclear *whether LLMs can generate complex structured data effectively*. We aim to address the following unanswered questions and deliver an in-depth examination of our research.

First, there is a lack of systematic analysis and comprehensive benchmarks of the ability of LLMs to output complex structured data. Previous efforts on evaluating LLMs (Qin et al., 2023; Ma et al., 2023) on structured data primarily centered around simple Information Extraction (IE) tasks: recognizing named entities, extracting relations, and detecting events. Here the goal of IE tasks is to gather the extracted data in a highly structured form (Zhong and Chen, 2020). Much earlier work was considerably more task-centric as opposed to LLM-centric. The focus was predominantly on generating structured data from text (text-to-data) tasks with pre-trained models (He et al., 2023; Rossiello et al., 2022; Whitehouse et al., 2023; Pietruszka et al., 2022) like BART (Lewis et al., 2019) and

T5 (Raffel et al., 2020).

Second, there is a lack of evaluation metrics of structured data generation. Existing benchmarks often rely on rudimentary objective metrics such as word overlap to measure the accuracy of the content generated by the model (Li et al., 2023; Wu et al., 2022; Pietruszka et al., 2022). This may be insufficient for evaluating whether LLMs can generate structured output, as an ideal evaluation metric ought to also consider the format of generated content.

Third, is there potential for enhancing the performance of current LLMs to better *follow natural language inputs and generate outputs with the correct format?*

Our contributions are summarized as:
(1) We introduce STRUC-BENCH, a benchmark specifically designed to generate structured data in Tables, HTML, and LaTeX formats. (2) We evaluate popular LLMs on STRUC-BENCH via two proposed metrics to gain a comprehensive understanding of prevailing error types and limitations. (3) We propose structure-aware instruction tuning, leveraging GPT-3.5 to generate format instructions and training the LLaMA model to follow these formats. The promising results demonstrate that fine-tuning small models can surpass the performance of a large language model in this particular task.

2 Problem Analysis and Benchmark

2.1 Problem Analysis

The task of generating complex structured data presents a notable challenge that tests the capabilities of LLMs in producing intricate, format-specific outputs. This task moves beyond conventional text generation. The complexity lies not only in the need to generate accurate and coherent content but also in maintaining a strict and specific data structure or format. For example, text-to-table is a task that aims to convert unstructured textual data into structured tabular data, by extracting necessary contents from text and following the required structure or format.

In our investigation, we have identified a significant limitation of GPT-3.5 and GPT-4 in handling complex structured output. Despite being state-of-the-art LLMs developed by OpenAI, these models both have demonstrated certain limitations in generating output in more complex formats, examples can be found in Appendix A.

This shortcoming becomes evident when the model is tasked with producing data that adhere to specific structural formats or templates, such as tables. Here, we select the Rotowire dataset (Wiseman et al., 2017) as an investigation, as shown in Appendix B. We collect human annotation by MTurk (See Appendix C) to examine the error types in 100 example instances. Figure 2 presents the proportions of errors and each error type: ELEMENT ERRORS, ELEMENT FORMAT ERRORS, STRUCTURE ERROR, STRUCTURE NAMING ERRORS.

We find that only **3%** of the output of GPT-3.5 is completely correct, while GPT-4 is only **9%**. This observation may be attributed to the inherent design of the GPT family. While the GPT-4 excels at capturing the statistical patterns of human language, it does not specifically account for structured outputs that require maintaining a state across a longer span of tokens.

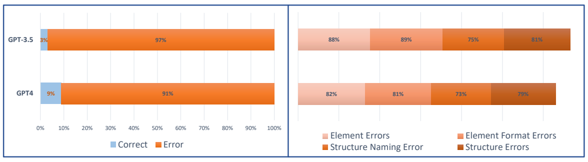


Figure 2: Error analysis by human annotation. Some error types are explained in Appendix A.

2.2 Benchmark

Firstly, we select tables from four prominent data-to-text datasets: Rotowire (Wiseman et al., 2017), E2E (Novikova et al., 2017), WikiTableText (Bao et al., 2018), and WikiBio (Lebret et al., 2016) with dimensions greater than 3x3 to ensure a sufficient level of complexity. Simultaneously, we construct more diverse datasets drawn from broader domains. This includes tables from LaTeX and HTML data strategically sourced from GitHub. Every kind of table format introduces its unique intricacies, layers of complexity, and degrees of structuration.

Table 1 gives statistics for the Rotowire dataset and our constructed datasets. Then we evaluate 4 popular LLMs, including GPT-NeoX-20B (Black et al., 2022), GPT-3.5 (Ouyang et al., 2022), GPT-4 (OpenAI, 2023) and Vicuna-13B (Chiang et al., 2023). For LaTex and HTML data without paired text, we harness GPT-3.5 to construct synthetic descriptions to be utilized as input. To guarantee the quality of our benchmark, we sample 50 tables for each format to ensure the correctness of the descriptions. Initially, we achieved a satisfaction rate of 76%. However, upon incorporating a manual interpretation template (e.g. tab names for HTML) tailored to each format (Appendix E), our satisfaction rate improved significantly, reaching 96%. For example, HTML tables possess their own unique tags and structure, conforming faithfully to the syntax rules of HTML language.

Dataset	# Train	# Test	Format	Rows & Columns
Rotowire (Wiseman et al., 2017)	3.4k	728	Raw tex	7.26 & 8.75
Struc-Bench LaTeX	5.3k	500	LaTeX	2.75 & 4.47
Struc-Bench HTML	5.4k	499	HTML	5.50 & 3.54

Table 1: Struc-Bench data statistics. The number of Rows & Columns has been averaged.

3 Methodology

3.1 Data Generation

As shown in Figure 1, we propose FORMATCOT and self-instruct with GPT-3.5 to generate data, instruction pairs. Here the prompt of FORMATCOT involves guiding models to accurately extract, interpret, and employ the core elements present in a LaTeX table, inspired by (Wang et al., 2023b) in the summarization task. To verify the effectiveness of the FormatCOT, we do an ablation study in Appendix G. In essence, FORMATCOT analyzes a given LaTeX table and generates a comprehensive description that exceeds 500 words. This detailed description encompasses all relevant factors in defining and formatting a LaTeX table, then used as the input.

3.2 Structure-aware Instruction Tuning

Here we propose a structure-aware instruction tuning method to bolster the capability of LLMs in generating structured text (Touvron et al., 2023; Patil et al., 2023). Our ultimate goal is to enable LLaMA to comprehend the task at hand and deliver the output in a conversational mode. The entire pipeline can be found in Figure 1.

3.3 Evaluation Metrics

Evaluating the similarity of generated tables to the ground-truth tables is non-trivial: for instance, the same table can be formatted in many different ways in HTML or LaTeX. Hence, our evaluation metric should ideally capture meaningful differences in the data presented, while being invariant to insignificant differences in formatting.

We propose to break down the similarity of two tables into two coarse components: *content* and *format*. In scoring content similarity, we attempt to parse *content* out the data within the table cells, and compute the similarity. This similarity is computed between the generated and ground-truth table cells by commonly used similarity metrics. In scoring format similarity, we place higher emphasis on components such as the number of columns and rows, cell alignment, and the table caption. Both similarity scores do overlap (e.g. a table with the wrong number of rows/columns would likely score poorly on content), but we find that these two scores allow us to perform more involved analysis on where predicted and ground-truth tables differ.

3.3.1 P-Score

We take two approaches to score each metric. First, we perform model-based evaluation, querying GPT-3.5 with both tables and having it score the similarity of content and format separately. Following Wang et al. (2023a), we prompt the model to perform Chain-of-Thought (Wei et al., 2023) reasoning before outputting its scores, and we query the model with the predicted and ground-truth tables in both orders and average the scores. We report these as the *P-Score* (Prompting Score). The prompt of P-Score can be found in Appendix D.

3.3.2 H-Score

In addition to model-based evaluation, we also implement hand-crafted scoring functions to score the similarity of the tables. Since the tables can be presented in different formats, we implement several heuristics to normalize the tables and to compute their similarity. We use an average of Levenshtein distance and the Ratcliff/Obershelp similarity metric to compute the similarities between strings or data structures. These heuristically normalized metrics are reported as the *H-Score* (Heuristical Score). The implementation of scoring functions for different formats can be found in Appendix D.

4 Experiments

4.1 Basic Settings

For metrics, we use SacreBLEU, ROUGE-L, BERTScore, BARTScore, and BLEURT metrics as they are all classical metrics to evaluate text similarity, as well as two proposed metrics: P-Score and H-score. In our dataset, each item consists of three parts: instruction, input, and output. When generating results, we put each item's instruction and input together as the final input to models. During the inference process, we provide the model with a natural language prompt to describe the form and content of our task, as well as the expected re-

Model	SacreBLEU	ROUGE-L	BERTScore	BARTScore	BLEURT	Content P-Score	Format P-Score	Content H-Score	Format H-Score
					Tables from Raw Text				
GPT-NeoX-20B	35.24	55.78	68.91	-2.34	33.51	3.86	6.10	0.50	-1.32
GPT-3.5	56.92	70.97	91.35	-1.68	36.85	6.19	8.16	0.52	-1.27
GPT-4	68.13	75.44	94.89	-0.99	55.24	6.88	8.30	0.85	0.53
Vicuna-13B	40.12	50.77	75.21	-2.05	40.02	4.07	6.33	0.55	-1.38
Ours-7B	**90.6**	**88.98**	**98.54**	**-0.69**	**66.07**	**7.69**	**8.60**	**1.65**	**3.61**
$w.o.finetune$	9.9	36.56	81.63	-2.50	70.24	4.58	6.00	0.51	-1.01
					LaTeX				
GPT-NeoX-20B	45.92	65.10	76.09	-2.05	40.87	7.23	7.02	0.56	0.72
GPT-3.5	56.94	75.99	86.25	-1.30	42.89	8.22	8.41	0.99	1.27
GPT-4	78.15	85.34	88.07	-1.09	**67.11**	8.78	8.81	1.10	1.35
Vicuna-13B	50.80	69.48	80.44	-1.07	36.74	7.70	8.10	0.78	1.06
Ours-7B	**89.13**	**88.99**	**98.55**	**-0.69**	66.07	**8.94**	**9.05**	**1.14**	**1.52**
$w.o.finetune$	47.24	70.89	73.27	-2.13	38.13	7.10	6.98	0.51	0.69
					HTML				
GPT-NeoX-20B	60.36	72.13	86.88	-1.59	30.06	8.42	8.94	0.81	0.92
GPT-3.5	73.80	85.19	96.76	-1.46	34.81	9.11	9.35	1.10	2.15
GPT-4	**79.25**	85.95	**97.22**	-1.31	41.59	9.17	9.62	1.15	2.29
Vicuna-13B	58.75	70.37	88.65	-1.58	31.11	8.55	8.88	0.79	0.93
Ours-7B	77.50	**86.08**	96.25	**-1.30**	**42.89**	**9.20**	**9.70**	**1.18**	**2.49**
$w.o.finetune$	65.30	78.24	88.12	-1.57	32.78	8.22	8.81	0.92	0.96
					Average				
GPT-NeoX-20B	47.47	64.33	77.29	-1.99	34.81	6.50	7.35	0.62	0.11
GPT-3.5	62.55	77.38	91.45	-1.48	38.18	7.84	8.64	0.87	0.72
GPT-4	68.11	82.24	93.39	-1.13	54.65	8.28	8.91	1.03	1.39
Vicuna-13B	49.89	63.54	81.43	-1.57	35.96	6.77	7.77	0.71	0.20
Ours-7B	**85.74**	**88.02**	**97.78**	**-0.89**	**58.34**	**8.61**	**9.12**	**1.32**	**2.54**
$w.o.finetune$	40.81	61.90	81.00	-2.07	47.05	6.63	7.26	0.64	0.21

Table 2: Automated evaluation results on the test set, involving five types of previous metrics and four proposed ones. $w.o.finetune$ means that we also compared the performance of our model without structure-aware finetuning as an ablation study. 'Average' means calculating each model's average score. 'Ours-7B' means the finetuned LLaMA.

sponse (e.g., "please generate a table given by the following information and format"). Considering the inconsistency observed by different metrics, we also conducted a human evaluation on 100 examples using MTurk. Evaluators rated each example on a scale of 10, assessing both format consistency and content consistency. Our proposed P-Score and Format H-Score have better instance-level Spearman correlation for format accuracy.

4.2 Results

Table 2 provides a comparative analysis of different LLMs based on several metrics. For 'Tables from Raw Text', the Ours-7B outperforms the other models in every metric. Interestingly, without finetuning, the performance drops significantly, particularly in SacreBLEU, ROUGE-L, and BERTScore. The results for 'LaTeX' reveal a similar trend where we again achieve the best results across all metrics, except for the BLEURT metric, where GPT-4 takes the lead. In the 'HTML' category, GPT-4 scores the highest in SacreBLEU and BERTScore. However, these differences are slight and our 7B model comes out on top for the rest of the metrics. The results demonstrate that our approach exhibits superior performance, highlighting the efficacy of finetuning smaller models in surpassing much larger models.

Moreover, we delve into an analysis based on our Mturk annotation, attributing observed shortcomings to several error types, spanning some key dimensions. And we present an ability map, see details in Appendix F.

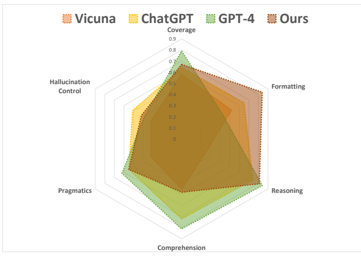


Figure 3: Visualization of LLM capability with human evaluation over STRUC-BENCH.

5 Conclusion

In conclusion, this work presents a comprehensive examination of the limitations of LLMs in generating structured data. We propose new evaluation metrics and incorporate diverse data types to develop a dedicated benchmark. Our analysis identifies several areas of concern, notably in terms of content accuracy, formatting, numerical reasoning, and the handling of long tables.

6 Limitations

Although we present a comprehensive analysis, the exploration of LLMs in structured text generation presented in this paper has several limitations:

Domain-Specific Benchmark Development While we've made strides in constructing benchmarks for structured text generation, it may be beneficial to develop benchmarks that cater to specific domains. Different fields might have unique structural requirements and understanding these nuances can significantly improve the models' applicability across diverse contexts.

Expand the Range of Datasets There are endless data types and sources that can be explored. Incorporating a broader variety of datasets could expose the models to an even wider range of structural formats, ultimately enhancing their overall performance.

Enhancing Numerical Reasoning Capabilities Our study identified inadequate numerical reasoning as one of the challenges faced by LLMs. Investigating techniques to bolster numerical reasoning in these models could lead to significant improvements in their performance.

Developing Advanced Methods While our structure-aware instruction tuning method showed promising results, more sophisticated techniques could be developed. For instance, future work could explore ways of incorporating more explicit structural information into the model or developing methods that allow the model to learn structural patterns more effectively.

Exploring Multimodal LLMs As LLMs continue to evolve, there are opportunities to explore multimodal models that can process and generate both text and other forms of data, such as sound or images (Kamigaito et al., 2023), in a structured manner.

References

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can eae models learn better when being aware of event co-occurrences? *arXiv preprint arXiv:2306.00502*.

Hidetaka Kamigaito, Katsuhiko Hayashi, and Taro Watanabe. 2023. Table and image generation for investigating knowledge of entities in pre-trained vision and language models. *arXiv preprint arXiv:2306.02115*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su. 2023. A sequence-to-sequence&set model for text-to-table generation. *arXiv preprint arXiv:2306.00137*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.

2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*.

Michał Pietruszka, Michał Turski, Łukasz Borchmann, Tomasz Dwojak, Gabriela Pałka, Karolina Szyndler, Dawid Jurkiewicz, and Łukasz Garncarek. 2022. Stable: Table generation framework for encoder-decoder models. *arXiv preprint arXiv:2206.04045*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Gaetano Rossiello, Faisal Chowdhury, Nandana Mihindukulasooriya, Owen Cornec, and Alfio Gliozzo. 2022. Knowgl: Knowledge generation and linking from text. *arXiv preprint arXiv:2210.13952*.

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*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023b. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. *arXiv preprint arXiv:2305.13412*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Chenxi Whitehouse, Clara Vania, Alham Fikri Aji, Christos Christodoulopoulos, and Andrea Pierleoni. 2023. Webie: Faithful and robust information extraction on the web. *arXiv preprint arXiv:2305.14293*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.

Xueqing Wu, Jiacheng Zhang, and Hang Li. 2022. Text-to-table: A new way of information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2518–2533, Dublin, Ireland. Association for Computational Linguistics.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, et al. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *arXiv preprint arXiv:2307.08674*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023a. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. 2023b. Large language models are effective table-to-text generators, evaluators, and feedback providers. *arXiv preprint arXiv:2305.14987*.

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

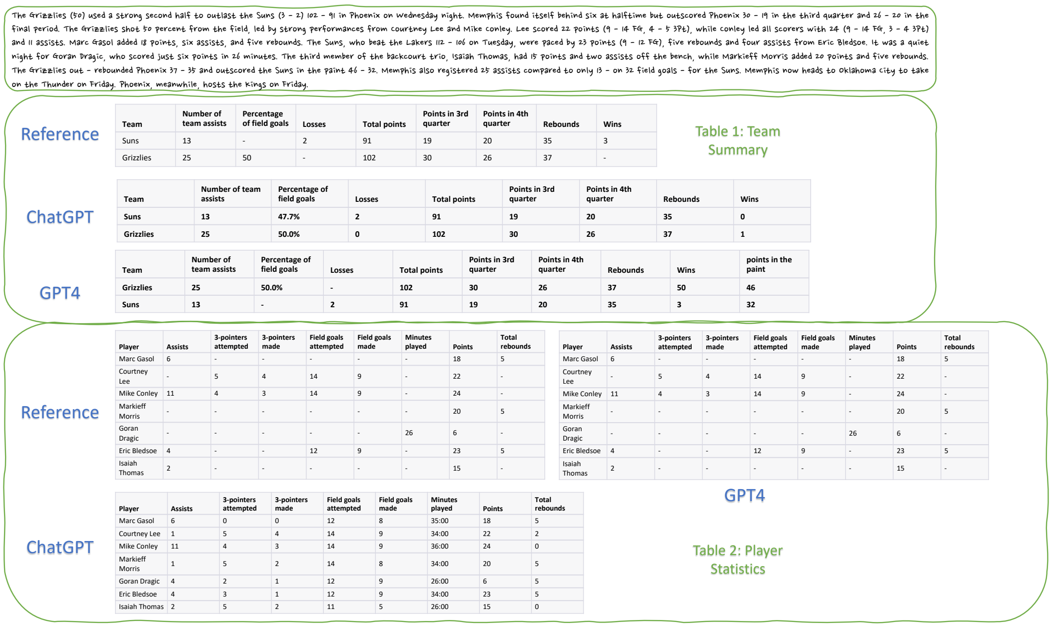
Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for entity and relation extraction. *arXiv preprint arXiv:2010.12812*.

A Analysis with Examples

A.1 Example Table A

The main difference between the reference tables and the tables generated by GPT-3.5 and GPT4 is in the completeness and precision of the data provided.

In the reference tables, all relevant data is fully represented: For the teams (Table 1), each team has a precise number or percentage for every statistic. Similarly, for the players (Table 2), each player has a definite number for every statistic, including minutes played in the format “mm:ss”.



The Grizzlies (50) used a strong second half to outlast the Suns (3 - 2) 102 - 91 in Phoenix on Wednesday night. Memphis found itself behind six at halftime but outscored Phoenix 30 - 19 in the third quarter and 26 - 20 in the final period. The Grizzlies shot 50 percent from the field, led by strong performances from Courtney Lee and Mike Conley. Lee scored 22 points (9 - 14 FG, 4 - 5 3Pt), while Conley led all scorers with 24 (9 - 14 FG, 3 - 4 3Pt) and 11 assists. Marc Gasol added 18 points, six assists, and five rebounds. The Suns, who beat the Lakers 112 - 106 on Tuesday, were paced by 23 points (9 - 12 FG), five rebounds and four assists from Eric Bledsoe. It was a quiet night for Goran Dragic, who scored just six points in 26 minutes. The third member of the backcourt trio, Isaiah Thomas, had 15 points and two assists off the bench, while Markieff Morris added 20 points and five rebounds. The Grizzlies out - rebounded Phoenix 37 - 35 and outscored the Suns in the paint 46 - 32. Memphis also registered 25 assists compared to only 13 - on 32 field goals - for the Suns. Memphis now heads to Oklahoma City to take on the Thunder on Friday. Phoenix, meanwhile, hosts the Kings on Friday.

Table 1: Team Summary

Reference

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins
Suns	13	-	2	91	19	20	35	3
Grizzlies	25	50	-	102	30	26	37	-

ChatGPT

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins
Suns	13	47.7%	2	91	19	20	35	0
Grizzlies	25	50.0%	0	102	30	26	37	1

GPT4

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins	points in the paint
Grizzlies	25	50.0%	-	102	30	26	37	50	46
Suns	13	-	2	91	19	20	35	3	32

Table 2: Player Statistics

Reference

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	-	-	-	-	-	18	5
Courtney Lee	-	5	4	14	9	-	22	-
Mike Conley	11	4	3	14	9	-	24	-
Markieff Morris	-	-	-	-	-	-	20	5
Goran Dragic	-	-	-	-	-	26	6	-
Eric Bledsoe	4	-	-	12	9	-	23	5
Isaiah Thomas	2	-	-	-	-	-	15	-

GPT4

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	-	-	-	-	-	18	5
Courtney Lee	-	5	4	14	9	-	22	-
Mike Conley	11	4	3	14	9	-	24	-
Markieff Morris	-	-	-	-	-	-	20	5
Goran Dragic	-	-	-	-	-	26	6	-
Eric Bledsoe	4	-	-	12	9	-	23	5
Isaiah Thomas	2	-	-	-	-	-	15	-

ChatGPT

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	0	0	12	8	35:00	18	5
Courtney Lee	1	5	4	14	9	34:00	22	2
Mike Conley	11	4	3	14	9	36:00	24	0
Markieff Morris	1	5	2	14	8	34:00	20	5
Goran Dragic	4	2	1	12	9	26:00	6	5
Eric Bledsoe	4	3	1	12	9	34:00	23	5
Isaiah Thomas	2	5	2	11	5	26:00	15	0

Figure 4: Using GPT-3.5 and GPT-4 to generate a table based on the input text, the generated results contain a large number of errors, including format errors and content errors.

In contrast, the generated tables show data that is incomplete and imprecise. For GPT-3.5 generated one, the team statistics table has some statistics missing, as represented by empty cells, and some are not presented as percentages. The player statistics table also has missing data in a similar fashion, and it lacks the "minutes played" statistics entirely. For instance, in the ’team’ table, the "Percentage of field goals" column for the Suns is missing. Similarly, in the ‘player’ table, many key statistics such as "3-pointers attempted", "3-pointers made", "Field goals attempted", "Field goals made", and "Minutes played" are missing for various players. Regarding the format, we observe a lot of format errors. For example, the ‘Percentage of field goals’ column for Grizzlies is represented as "50" instead of "50.0%". Moreover, the ‘Wins’ column for the Suns is represented as "3" instead of "0". This misrepresentation can lead to significant misunderstanding of the data. The ‘Player’ table also has format errors. For instance, the ‘Minutes played’ column is missing the time format (i.e., “00:00”). On the other hand, the reference tables adhere to a standard format. Percentage data is represented with a ‘%’ sign, time data uses the ‘00:00’ format, and numeric data correctly represents each statistic.

The Grizzlies (50) used a strong second half to outlast the Suns (3 - 2) 102 - 91 in Phoenix on wednesday night. Memphis found itself behind six at halftime but outscored Phoenix 30 - 19 in the third quarter and 26 - 20 in the final period. The Grizzlies shot 50 percent from the field, led by strong performances from courtney Lee and Mike conley. Lee scored 22 points (9 - 14 FG, 4 - 5 3Pt), while conley led all scorers with 24 (9 - 14 FG, 3 - 4 3Pt) and 11 assists. Marc Gasol added 18 points, six assists, and five rebounds. The Suns, who beat the Lakers 112 - 106 on Tuesday, were paced by 23 points (9 - 12 FG), five rebounds and four assists from Eric Bledsoe. It was a quiet night for Goran Dragic, who scored just six points in 26 minutes. The third member of the backcourt trio, Isaiah Thomas, had 15 points and two assists off the bench, while Markieff Morris added 20 points and five rebounds. The Grizzlies out - rebounded Phoenix 37 - 35 and outscored the Suns in the paint 46 - 32. Memphis also registered 25 assists compared to only 13 - on 32 field goals - for the Suns. Memphis now heads to Oklahoma city to take on the Thunder on Friday. Phoenix, meanwhile, hosts the Kings on Friday.

Table 1: Team Summary

Reference

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins
Suns	13	-	2	91	19	20	35	3
Grizzlies	25	50	-	102	30	26	37	-

Vicuna-13B

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins
Suns	13	-	3	91	19	20	35	5
Grizzlies	25	50.0%	-	102	30	26	37	7

LLaMA2-7B

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins
Suns	13	40.6%	2	91	19	20	35	0
Grizzlies	25	50.0%	1	102	30	26	37	1

Table 2: Player Statistics

Reference

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	-	-	-	-	-	18	5
Courtney Lee	-	5	4	14	9	-	22	-
Mike Conley	11	4	3	14	9	-	24	-
Markieff Morris	-	-	-	-	-	-	20	5
Goran Dragic	-	-	-	-	-	26	6	-
Eric Bledsoe	4	-	-	12	9	-	23	5
Isaiah Thomas	2	-	-	-	-	-	15	-

LLaMA2-7B

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	3	1	9	4	34	18	5
Courtney Lee	3	5	4	14	9	36	22	3
Mike Conley	11	4	3	14	9	36	24	3
Markieff Morris	2	2	0	12	5	31	20	5
Goran Dragic	4	3	1	12	6	26	6	2
Eric Bledsoe	5	2	1	12	7	35	23	4
Isaiah Thomas	2	2	1	11	4	23	15	2

Vicuna-13B

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	-	-	14	9	32	18	5
Courtney Lee	-	4	5	14	9	32	22	-
Mike Conley	11	3	4	14	9	32	24	-
Markieff Morris	-	2	1	12	6	26	20	5
Goran Dragic	-	2	0	12	6	26	6	-
Eric Bledsoe	-	1	1	12	9	26	23	5
Isaiah Thomas	-	1	0	8	4	18	15	2

Figure 5: Using Vicuna-13B and LLaMA2-7B to generate a table based on the input text, the generated results contain a large number of errors, including format errors and content errors.

For Vicuna-13B results, although it has the correct format for both tables, there are still many element errors. For instance, the ’team’ table has wrong statistics such as “Losses" and “Wins" for the Suns. Besides, in the ’player’ table, many cells shouldn’t have data. However, they actually have, which is obviously a mistake. Some cells like Isaiah Thomas’s and Eric Bledsoe’s ’Assists’ should be 2 and 4, but they are none in Vicuna-13B ’player’ table. Similarly, LLaMA2-7B results, have the same element errors in the ’team’ table and worse errors in the ’player’ table. It fills all cells, many of which should be none. As for some cells that should have data, their data are wrongly filled in like Eric Bledsoe’s ’Assists’ and ’Field goals made’.

A.2 Error Type

Structure Errors: These errors pertain to the structural integrity of the generated tables. Specifically, they include instances where there are excess or missing rows or columns in comparison to the correct table structure. For instance, in figure 6 GPT4 generated result has missing columns like “Wins" and “Losses" in ‘team’ table.

Structure Naming Errors: This category captures errors related to the naming conventions used for rows or columns. Any discrepancies in a row or column names between the generated and correct table are flagged as structure naming errors. For instance, in figure 6 GPT4 generated result has wrong column names like "Half-Time Score" in the ’team’ table.

Element Errors: These are inaccuracies observed at the element level within the generated table. Element errors encompass incorrect numbers, values, or inappropriately empty cells, reflecting discrepancies in individual table entries relative to the correct table. In figure 4 and figure 5, most errors are element errors.

The Grizzlies (50) used a strong second half to outlast the Suns (3 - 2) 102 - 91 in Phoenix on Wednesday night. Memphis found itself behind six at halftime but outscored Phoenix 30 - 19 in the third quarter and 26 - 20 in the final period. The Grizzlies shot 50 percent from the field, led by strong performances from Courtney Lee and Mike Conley. Lee scored 22 points (9 - 14 FG, 4 - 5 3Pt), while Conley led all scorers with 24 (9 - 14 FG, 3 - 4 3Pt) and 11 assists. Marc Gasol added 18 points, six assists, and five rebounds. The Suns, who beat the Lakers 112 - 106 on Tuesday, were paced by 23 points (9 - 12 FG), five rebounds and four assists from Eric Bledsoe. It was a quiet night for Goran Dragic, who scored just six points in 26 minutes. The third member of the backcourt trio, Isaiah Thomas, had 15 points and two assists off the bench, while Markieff Morris added 20 points and five rebounds. The Grizzlies out - rebounded Phoenix 37 - 35 and outscored the Suns in the paint 46 - 32. Memphis also registered 25 assists compared to only 13 - on 32 field goals - for the Suns. Memphis now heads to Oklahoma City to take on the Thunder on Friday. Phoenix, meanwhile, hosts the Kings on Friday.

Table 1: Team Summary

Reference

Team	Number of team assists	Percentage of field goals	Losses	Total points	Points in 3rd quarter	Points in 4th quarter	Rebounds	Wins
Suns	13	-	2	91	19	20	35	3
Grizzlies	25	50	-	102	30	26	37	-

GPT4

Team	Half-Time Score	Final Score	Field Goals (%)	3Pt (%)	Rebounds	Assists	Points in the Paint
Grizzlies	**46**	**102**	**50**	**N/A**	**37**	**25**	**46**
Suns	**52**	**91**	**N/A**	**N/A**	**35**	**13**	**32**

Table 2: Player Statistics

Player	Assists	3-pointers attempted	3-pointers made	Field goals attempted	Field goals made	Minutes played	Points	Total rebounds
Marc Gasol	6	-	-	-	-	-	18	5
Courtney Lee	-	5	4	14	9	-	22	-
Mike Conley	11	4	3	14	9	-	24	-
Markieff Morris	-	-	-	-	-	-	20	5
Goran Dragic	-	-	-	-	-	26	6	-
Eric Bledsoe	4	-	-	12	9	-	23	5
Isaiah Thomas	2	-	-	-	-	-	15	-

Reference

Player	Team	Points	Assists	Rebounds	Field Goals (FG)	Three-Points (3Pt)
Courtney Lee	Grizzlies	22	N/A	N/A	9 - 14	4 - 5
Mike Conley	Grizzlies	24	11	N/A	9 - 14	3 - 4
Marc Gasol	Grizzlies	18	6	5	N/A	N/A
Eric Bledsoe	Suns	23	4	5	9 - 12	N/A
Goran Dragic	Suns	6	N/A	N/A	N/A	N/A
Isaiah Thomas	Suns	15	2	N/A	N/A	N/A
Markieff Morris	Suns	20	N/A	5	N/A	N/A

GPT4

Figure 6: Using GPT-4 to generate a table based on the input text without FORMATCOT, the generated results contain a large number of errors, including format errors and content errors.

B Rationale for Selecting the RotoWire Dataset

Traditional data-to-text datasets include Rotowire (Wiseman et al., 2017), E2E (Novikova et al., 2017), WikiTableText (Bao et al., 2018), and WikiBio (Lebret et al., 2016). Given that only the RotoWire dataset contains tables with more than 2 columns, we specifically opted to utilize this dataset. Furthermore, to maintain a certain level of complexity in our study, we filtered out tables with dimensions smaller than 3x3 in Rotowire.

Dataset	Train	Valid	Test	# of tokens	# of rows	# of columns
E2E	42.1k	4.7k	4.7k	24.90	4.58	2.00
WikiTableText	10.0k	1.3k	2.0k	19.59	4.26	2.00
WikiBio	582.7k	72.8k	72.7k	122.30	4.20	2.00

Table 3: Statistics of E2E, WikiTableText, and WikiBio datasets, including the number of instances in training, validation, and test sets, number of BPE tokens per instance, and number of rows per instance.

C MTurk

About the qualifications of Amazon Mechanical Turk (MTurk) workers, we use the following qualifications to recruit in total of 10 MTurk workers with good track records: HIT approval rate greater than or equal to 98%, number of HITs approved greater than or equal to 500, and located in one of the following English native-speaking countries: Australia, Canada, New Zealand, United Kingdom, United States. Each annotator is limited to annotating 10 examples, including both the output of GPT-3.5 and GPT-4.

Annotators workers were compensated $7, calibrated to equal a $42/hour pay rate. We first annotated examples in-house to determine the required annotation speed. A summary block usually takes around 10 minutes.

To demonstrate our annotation template and facilitate future research, we show the interface for annotations.

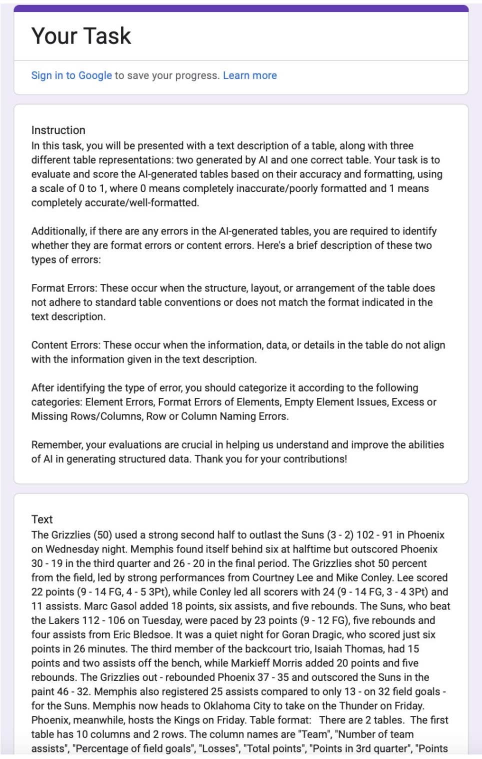


Figure 7: Interface of Mturk.

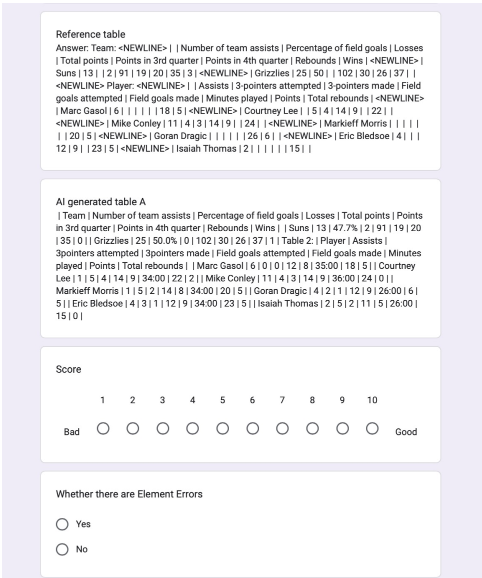


Figure 8: Interface of Mturk.

D Scoring

D.1 P-Score

Our approach involves prompting the model to engage in Chain-of-Thought reasoning prior to issuing its scores. Firstly, we instruct GPT on how to evaluate both "content similarity" and "structural similarity". Following this, the model is guided on the correct procedure to output its answer. In order to calculate the scores, the model is queried with both the predicted table and the ground truth table in varying sequences, after which the scores are averaged. We'll illustrate this process using the P-Scores prompt for raw text tables as an illustrative example:

"Based on the above, we wanted to determine if the above tables are similar. Ideally, they should have identical content and structure. Score the "content similarity" and "structural similarity" between 0 and 10.

- Content similarity: 10 if the contents of the table cells are identical, 0 if they are entirely different. If about 50% of the cells have the same data, the score should be 5.

- Structural similarity: 10 if the tables have the same structure (e.g. same column and rows with identical ordering, same alignment, etc.) although text formatting differences can be ignored (e.g. colors, font).

Output a JSON object such as the following:
"""json
{{
"content_similarity": ...
"structural_similarity": ...
}}
"""
Think carefully, and then output the scores."

D.2 H-Score

LaTeX We use the `pylatexenc` library to parse a given LaTeX table, and walk through the parse-tree structure in the `tabular` environment to identify the table "cells". We score the content similarity based on strings within the cells, and score structural similarity based on having the matching number of rows and columns, the same caption, and the same cell alignment.

HTML We use the `beautifulsoup4` library to parse a given LaTeX HTML snippet and walk through the parse-tree structure in `<table>`, `<ul>` or `<ol>` tags to identify data cells. We separately build a tree of white-listed HTML tags to score the structural similarity, traversing an HTML document tree structure, disregarding the actual content within the tags and simplifying it by focusing only on specific HTML tags (defined in RECOGNIZED_HTML_TAGS). We score the content similarity based on strings within the cells and score structural similarity based on the similarity of the structure tree and the total number of cells matching.

White-listed HTML tags:

```
RECOGNIZED_HTML_TAGS = [
    "table", "tr", "th", "td",
    "ul", "ol", "li",
    "div", "span", "p",
    "a", "img", "embed", "pre",
    "h1", "h2", "h3", "h4", "h5", "h6",
    "input", "button",
]
```

Raw Text Tables In our evaluated dataset, each example consists of two tables (Team and Player). We do a string search for `"Team"` and `"Player"` headers to identify the two tables. We then parse the tables according to Markdown formatting, with newlines and pipes as row and column dividers respectively, to identify the table cells. We score the content similarity based on strings within the cells, and score structural similarity based on the similarity of column names and the number of rows and columns matching.

String Similarity Measurement: Our script includes methods to calculate the similarity between two strings. These methods can be used to compare the structure or content of HTML, latex documents, or any other pair of strings. The similarity is evaluated using well-established algorithms in text analysis: the Levenshtein distance and the SequenceMatcher from Python's difflib module.

E Prompt for Description Generation and Inference

Raw Text Table Description Prompt Traditional data-to-text datasets only have raw text for each table. However, it is not enough for Chatgpt or other LLMs to generate correct tables. As a result, we added some format descriptions to help them generate the correct tables. We use GPT-3.5 to achieve this. We want to get detailed format information without concrete contents in cells, so we explicitly include these requirements in the prompt. Here is our prompt: "Describe details about the given text. First, give the number of tables, and then for each table, describe its format such as the number of columns and rows, column names, and row names."

HTML Table Description Prompt Unlike data-to-text datasets, HTML datasets only have final outputs, so we are required to generate a detailed description of their format and content. For content descriptions, we can simply ask GPT-3.5 to output raw text without HTML tags. For format descriptions, however, we need to ask GPT-3.5 to describe each tag, otherwise, it will leave out some tags and describe the table in general rather than detailed information. Moreover, it is necessary to ask it to use specific numbers instead of 'several' or 'multiple'. Here is our prompt for HTML format descriptions: "Describe the format of this HTML in detail according to each HTML tag of the following HTML code. Be careful and make sure don't miss any HTML tags. Please use more than 300 words to explain the format. Use specific numbers rather than being vague about several."

LaTEX Table Description Prompt Similar to HTML prompt generation, it is necessary to ask GPT-3.5 to generate both format descriptions and content descriptions as latex datasets only have final outputs. For content descriptions, we can simply ask GPT-3.5 to describe the given latex table as detailed as it can and include all cells. For format description, since the latex format is too complex, we need to give it a small example to learn. Then we ask GPT-3.5 to describe the detailed format of a given latex table, including specific questions to help it generate format descriptions. Here is our prompt for latex format descriptions: "Describe the detailed format of a given latex table according to the commands and tags with more than 500 words. Include: Whether there is table border lines? How is text alignment? What are table attributes? Whether to bold? Whether to add \ref? Please clearly explain whether there are horizontal and vertical lines bordering each row and column. Say anything about a special "\" format token in latex if there is one. Don't display latex code directly. Use natural language. And provide enough format information for me to recreate this table based on your output description."

Prompt for Inference When inferencing raw text tables, LLMs tend to output tabular results rather than raw text tables. As a result, we need to give it an example output first, then tell the model that the input consists of two parts, text and format descriptions, and ask the model to generate the output based on them. For HTML and Latex inference, we can simply ask models to infer from the input and specify the format and content sections in the input, since models can generate correct syntax.

F Ability Map

Based on our automated evaluation, we selected Vicuna, ChatGPT, GPT-4, and Ours as representative models and conducted an in-depth analysis of the causes of model errors.

We identified content accuracy, formatting, numerical reasoning, and handling of long tables as the main sources of these errors.

At the fundamental level, we decompose the process of model-generated complex structured outputs into two parts: Content Selection and Format Planning. Initially, the model needs to identify key information from a given vast amount of unstructured input, extract this information, understand it, and organize it. Subsequently, it needs to plan how to summarize these extracted details, devise the format of the table to be generated, and then fill in the information.

Accordingly, we can break down the model's capabilities into Coverage, Formatting Reasoning, Comprehension, Pragmatics, and Hallucination Control.

Coverage entails the model's ability to accurately cover the content in the input. Formatting Reasoning pertains to judgment about the output format, assessing if the model can find the most appropriate and reasonable structured format.

Comprehension reflects whether the model can understand the content of the input, as there are times when it is necessary to infer from a large amount of data (including performing addition or subtraction or comparing multiple elements).

Pragmatics involves the ability to utilize special formats, such as HTML tags and specific syntax in LaTeX.

Finally, Hallucination Control signifies the model's ability to refrain from generating content not present in the input.

We carried out manual annotations and obtained visualized results to demonstrate these aspects.

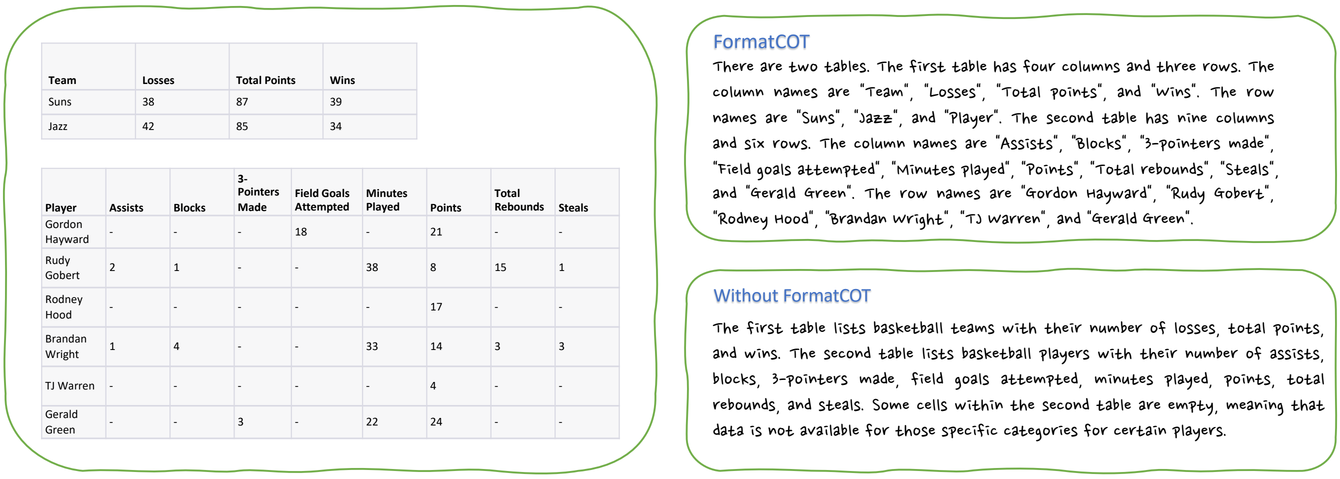
G Ablation Study for FormatCOT

G.1 Contrast between descriptions

In this section, we conduct an ablation study to examine the impact of our proposed FormatCOT. In the generation of table descriptions sans FormatCOT, we simply utilize the prompt: "Provide a description of the following tables." The primary differentiation between results pivots on the extent of details incorporated.

For instance, in the FormatCOT result, the description comprises an array of detailed format information - encompassing row names, column names, and table count. The precision in these details proves substantial enough for models to accurately recreate the tables in question.

Contrastingly, the outcome bereft of FormatCOT conveys considerably less information - providing incomplete column names without the accompaniment of row names. This sparse degree of detail proves insufficient for models seeking to faithfully regenerate the corresponding tables.



Team	Losses	Total Points	Wins
Suns	38	87	39
Jazz	42	85	34

Player	Assists	Blocks	3-Pointers Made	Field Goals Attempted	Minutes Played	Points	Total Rebounds	Steals
Gordon Hayward	-	-	-	18	-	21	-	-
Rudy Gobert	2	1	-	-	38	8	15	1
Rodney Hood	-	-	-	-	-	17	-	-
Brandan Wright	1	4	-	-	33	14	3	3
TJ Warren	-	-	-	-	-	4	-	-
Gerald Green	-	-	3	-	22	24	-	-

Figure 9: Using FormatCOT and normal instructions to ask GPT-3.5 to generate table descriptions based on the input text, FormatCOT results contain more detailed information about row names.

G.2 Contrast between results

In this section, we draw a comparison between two sets of description results. The FormatCOT result showcases a table that stands remarkably close to the correct table, albeit with minor errors. It contains an extra row termed "Player" in the initial table, a discrepancy potentially attributable to the fact that the result comprises two tables, with "Player" denoting the header of the subsequent table. We posit that this error could potentially be circumvented with a different method of integrating table names.

Furthermore, an additional column surfaces in the second table, which in reality represents the final row of that table. Besides these minor inaccuracies, the FormatCOT result accurately replicates the content in each cell as well as maintaining the overall format.

Conversely, the alternative result contains multiple errors that span both content and format. Initially, an additional row is present in the first table, introducing an unrelated basketball team that bears no relevance to the game under consideration. Following this, the second table possesses an excessive number of player names, encompassing unnecessary players along with coaches who did not participate in the game.

Furthermore, its content is not entirely accurate, with discrepancies present in the statistics attributed to both Gordon Hayward and Gerald Green. These shortcomings underscore the efficiency and essentiality of implementing the FormatCOT in order to ensure accuracy and precision.

FormatCOT

Team	Losses	Total Points	Wins
Suns	38	87	39
Jazz	42	85	34
Player			

Player	Assists	Blocks	3-Pointers Made	Field Goals Attempted	Minutes Played	Points	Total Rebounds	Steals	Gerald Greens
Gordon Hayward	-	-	-	-	-	21	-	-	
Rudy Gobert	2	1	-	-	38	8	15	1	
Rodney Hood	-	-	1	-	-	17	-	-	
Brandan Wright	1	4	-	-	33	14	3	3	
TJ Warren	-	-	-	-	-	4	-	-	
Gerald Green	-	-	3	-	22	24	-	-	

Without FormatCOT

Team	Losses	Total Points	Wins
Phoenix Suns	38	87	39
Utah Jazz	42	85	34
Oklahoma City Thunder			

Player	Assists	Blocks	3-Pointers Made	Field Goals Attempted	Minutes Played	Points	Total Rebounds	Steals
Rodney Hood	-	-	-	-	-	-	-	-
Gerald Green	-	-	-	-	-	24	-	-
Brandan Wright	1	4	-	-	33	14	3	3
Trey Burke	-	-	-	-	-	-	-	-
T.J. Warren	-	-	-	-	-	4	-	-
Dante Exum	-	-	-	-	-	-	-	-
Joe Ingles	-	-	-	-	-	-	-	-
Gordon Hayward	2	1		18		21	15	1
Rudy Gobert	2	1			38	8	15	1

Figure 10: Using two descriptions to regenerate table descriptions based on the input text and descriptions, FormatCOT result is more correct in both format and content.