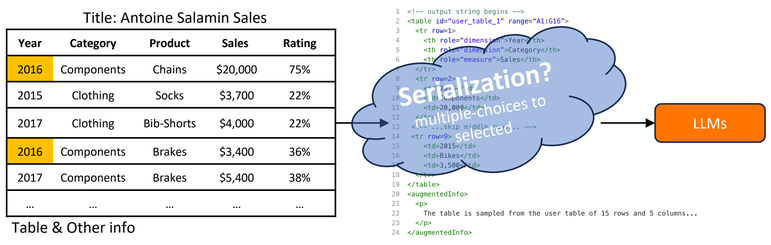
Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study

Yuan Sui*, Mengyu Zhou†, Mingjie Zhou, Shi Han, Dongmei Zhang
*yuan.sui@u.nus.edu, †mezho@microsoft.com

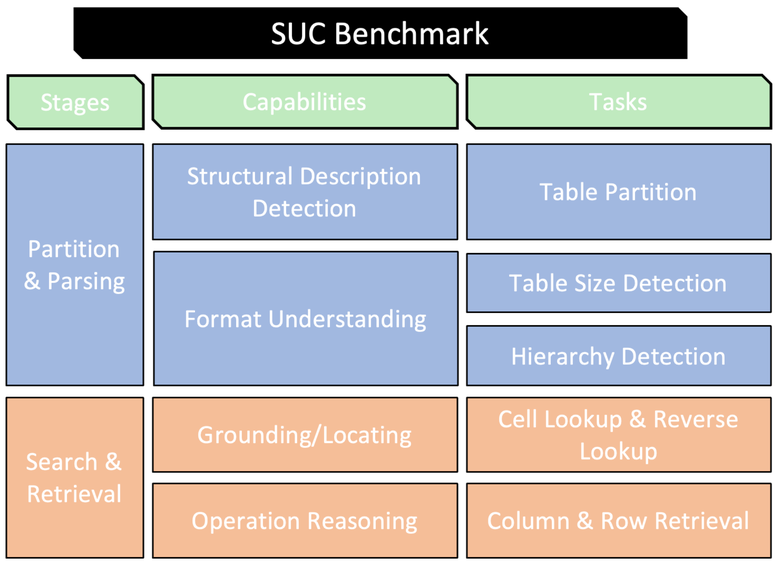
Scope & Motivation

In this digital age, tables are essential for organizing and presenting information, particularly text, in a structured format. They are used to compress recurring information, enhance data manageability, simplify analysis, and improve machine processing. At the same time, large language models (LLMs) have recently emerged as powerful tools for solving natural language-related tasks. However, it's unclear how well these language models understand tables embedded in their prompts. Specifically, we study the following questions: (1) Which input designs most effectively enable LLMs to understand tables? (2) How much do LLMs inherently understand structured data? And (3) How can LLMs' existing knowledge be harnessed to improve this understanding?

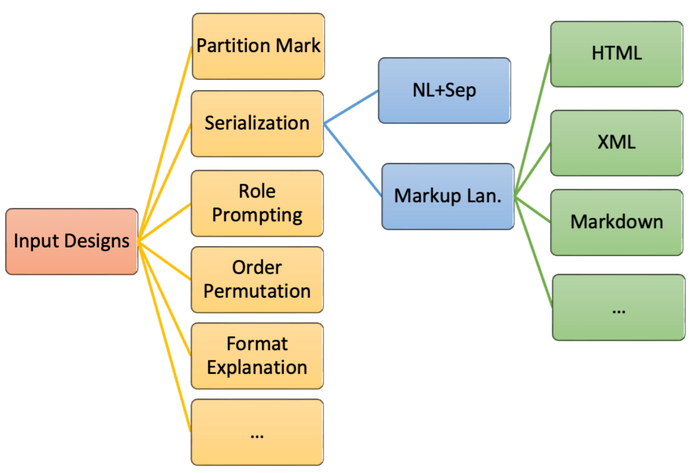


SUC Benchmark

We propose a new benchmark named **Structural Understanding Capabilities (SUC)**, focusing on several low-level fundamental tasks to assess LLMs' ability to understand structured data in tables and to compare different input designs. The taxonomy of SUC is as follows:



We also analyze the complex trade-off among multiple combinations of input designs and overall performance. We conduct a series of experiments using diverse input design variants (represent common practices in the field, shown as follows) with GPT-x models.



Notes

- Yuan Sui and Mingjie Zhou made their contributions during their internships at Microsoft Research Asia, located in Beijing, China.
- The code and data of the paper can be found here: `https://github.com/microsoft/TableProvider`.

Insights and findings using the SUC benchmark

Our evaluations on GPT-3.5 and GPT-4 reveal several notable and unexpected findings as follows:

- Delimiter-separated formats (*e.g.*, CSV, TSV), under-performed compared with HTML 6.76%.
- Using HTML and few-shot learning consistently improved performance. The effectiveness of other approaches, such as format explanation, role prompting, order change, and partition marks, varied depending on task difficulty and the required capacity.
- Despite the simplicity of the benchmark tasks, the highest overall accuracy across seven tasks is only 65.43%. This underscores the need for LLMs to have better awareness of table structures and highlights areas for further improvement in table serialization.

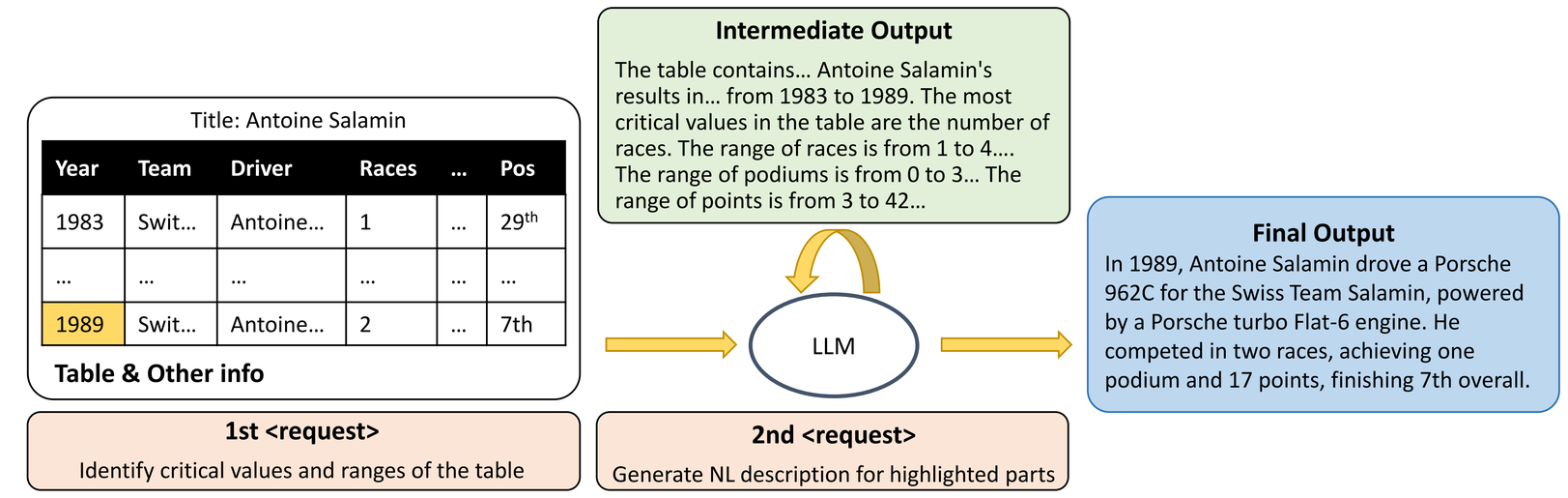
Our exploration suggests that:

- LLMs seem to have a basic understanding of table structures but are far from perfect, even in straightforward task, like detecting table size (number of table columns and rows).
- Choosing the right combination of input designs can significantly enhance LLMs' understanding of structured data. The observation remains even with the use of GPT-4, validating the effectiveness of our benchmark approach.

Format	Table Partition		Cell Lookup		Reverse Lookup		Column Retrieval		Row Retrieval		Size Detection		Merged Cell Detection	
	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4	Acc	GPT-4
NL + Sep	93.00%	96.78%	39.67%	72.48%	52.00%	59.12%	60.67%	66.32%	31.00%	48.67%	42.00%	73.12%	71.33%	74.98%
Markdown	92.33%	**98.32%**	43.33%	71.93%	51.00%	57.32%	35.33%	60.12%	**42.33%**	49.98%	40.67%	82.12%	**78.00%**	**82.64%**
JSON	94.00%	97.12%	42.67%	68.32%	54.33%	58.12%	54.33%	64.32%	29.00%	48.32%	42.67%	76.43%	73.33%	78.98%
XML	96.00%	97.64%	43.33%	72.28%	**55.00%**	**60.32%**	41.33%	68.28%	41.00%	**50.28%**	43.67%	80.21%	75.00%	80.32%
HTML	**96.67%**	**98.32%**	**44.00%**	**73.34%**	47.33%	59.45%	**63.33%**	**69.32%**	42.00%	50.19%	**67.00%**	**83.43%**	76.67%	81.28%

Input Design	Table Partition		Cell Lookup		Reverse Lookup		Column Retrieval		Row Retrieval		Size Detection		Merged Cell Detection	
	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ
Markup Lan. HTML	**96.67%**	0.00%	44.00%	0.00%	47.33%	0.00%	63.33%	0.00%	42.00%	0.00%	67.00%	0.00%	76.67%	0.00%
w/o format explanation	92.00%	-4.67%	52.00%	8.00%	52.33%	5.00%	64.33%	1.00%	36.00%	-6.00%	**78.00%**	11.00%	**77.67%**	1.00%
w/o partition mark	**98.00%**	1.33%	**59.00%**	15.00%	53.00%	5.67%	66.00%	2.67%	39.67%	-2.33%	**72.00%**	5.00%	70.33%	-6.33%
w/o role prompting	95.00%	3.00%	40.67%	-11.33%	44.67%	-7.67%	59.00%	-5.33%	39.33%	3.33%	**69.00%**	-9.00%	76.00%	-1.67%
w/o change order	**96.67%**	0.00%	52.33%	8.33%	40.67%	-6.67%	55.67%	-7.67%	31.67%	-10.33%	52.67%	-14.33%	65.67%	-11.00%
w/o 1-shot	63.00%	-33.67%	9.33%	-34.67%	17.33%	-30.00%	50.00%	-13.33%	30.00%	-12.00%	16.67%	-50.33%	38.00%	-38.67%
GPT-4 w/ Lan. HTML	**98.32%**	1.65%	73.34%	29.34%	59.45%	12.12%	69.32%	5.99%	50.19%	8.19%	83.43%	16.43%	81.28%	4.61%

Improved performance with self-augmented prompting



Type	Choice	TabFact	HybridQA	SQA	Feverous	ToTTo			
		Acc	Acc	Acc	Acc	BLEU-1	BLEU-2	BLEU-3	BLEU-4
1-shot	1-shot	72.04%	46.07%	73.81%	75.56%	72.43%	44.36%	27.01%	17.24%
1-shot	w/o table size	71.33%	45.52%	72.91%	74.66%	72.30%	44.23%	27.14%	17.25%
1-shot	w/o partition mark	71.25%	45.48%	73.09%	75.11%	71.18%	43.17%	26.36%	16.34%
1-shot	w/o format explanation	70.87%	45.39%	71.69%	75.97%	70.54%	43.59%	26.52%	16.74%
1-shot	w/o role prompting	71.35%	46.05%	73.39%	75.52%	70.61%	43.10%	26.02%	16.15%
SA	self format explanation	72.23%	46.12%	73.91%	76.15%	74.18%	45.25%	27.32%	18.34%
SA	self critical values and ranges identification	74.35%	48.20%	76.53%	76.32%	80.83%	47.96%	30.68%	22.92%
SA	self structural information description	73.42%	46.97%	75.97%	77.28%	78.93%	46.91%	28.94%	19.32%

Format	TabFact	HybridQA	SQA	Feverous	ToTTo
	Acc	Acc	Acc	Acc	BLEU-4
NL + Sep	70.26%	45.02%	70.41%	75.15%	**12.70%**
Markdown	68.40%	45.88%	66.59%	71.88%	8.57%
JSON	68.04%	42.40%	70.39%	73.84%	8.82%
XML	70.00%	47.20%	70.74%	73.14%	8.82%
HTML	**71.33%**	**47.29%**	**71.31%**	**75.20%**	12.30%
GPT-4 w/ HTML	78.40%	56.68%	75.35%	83.21%	20.12%

	Manual Prompt	Self-augmented Prompt Generation
	Each table cell is defined by a <td> and a </td> tag.\nEach table row starts with a <tr> and ends with a </tr> tag.\nth stands for table header.	'15 rows and 4 columns, with the first column being an index, the second column being empty, the third column being the name of the Marshal, the fourth column being the date of promotion, and the fifth column being the defence branch.\nThe table is defined by HTML tags, with each table cell being defined by a <td> and a </td> tag, and each table row starting with a <tr> and ending with a </tr> tag.\nThe table header is denoted by the th tag.'

Summary & Looking forward

Our study serves as a key benchmark in advancing the application of LLMs and deepening the understanding of structured table data. By testing more effective input designs and prompting methods on SUC, we aim to improve LLMs' comprehension of structured data. This refinement in LLMs' processing and understanding capabilities will broaden their use in practical downstream tasks.