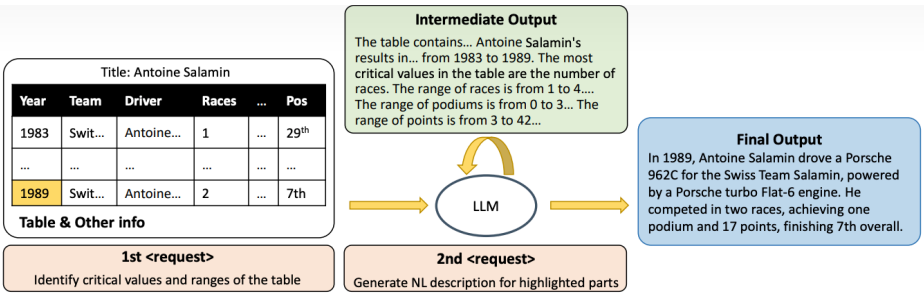


微软2024最新研究 - 大型语言模型（LLM）遇上结构化数据

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我

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

Introduction

结构化数据⁺是由预定义结构组织的文本块组成，用于压缩重复信息。其中，表格是一种常见结构化数据类型，适用于问题回答、事实验证、表格到文本等多个场景。采用结构化数据有助于推动网页挖掘和内容分析⁺中的信息检索和知识提取技术发展。最近研究表明，链式思维、自一致性及混合使用生成和检索方法⁺的方法能有效提高语言模型（LLM）的表现。

具体来说，如GPT-X和FlanT5在zero-shot⁺和 few-shot示例下就能完成复杂的数学推理任务，这显示了LLM在结构化数据应用方面的潜力。因此，采用LLM来处理结构化数据具有新前景。本文主要探讨如何优化语言模型的理解结构化数据的能力。

- 我们提出了一种新的评估基准------SUC，用于衡量LLM的多种结构理解能力。
- 本研究通过对各种结构理解能力的测试，为未来的相关工作提供了宝贵的参考和建议。
- 本文提出了一种名为"自我增强"的方法，该方法利用模型的内部知识来提升LLM的表现。作者已在五个表格推理数据集上验证了这一方法的有效性。这是一种简单而通用的方法。

Preliminaries

Table Structure

表格数据具有广泛的灵活性，能在各种不同结构中展现，如图所示。这种结构可以是简单的水平关系表，也可以是复杂的层次化结构，比如ToTTo。在每个表格中，每一行表示一个记录，而列则代表特定的字段，没有明显的层次顺序。表格数据也有许多不同的格式值方法，如文本、数字、日期/时间、公式和其他相关信息。其中，文本对于捕捉元信息，如标题、注释、图例和数据区域内的单元格至关重要。另外，数字经常涉及到算术关系，如求和和比例，以及统计属性，如分布和趋势。表格数据通常是以一种精心组织的方式呈现数值数据，以便于参考和比较。

Table Serialization & Splitting

表序化是一种将表格数据转换为线性的、顺序的文本格式的方法，这对于训练和使用LLM（语言模型）非常重要，特别是对于像掩码语言建模这样的任务来说，理解并预测语言模式是非常关键的。一个简单的序列化函数是逐行序列化表格。很多研究，如TaPas、MATE、TableFormer、TUTA和TURL都采用了这种方法。TaPEx使用特殊的标记来表示组件，如标题\<HEAD>和行\<ROW>。TABBIE既可以按行也可以按列进行序列化。TableGPT则使用模板方法，以每张表记录中的属性值对进行序列化。

然而，大多数LLM在处理长句子时效率低下，因为自注意力机制⁺具有二次复杂性。虽然@model_tapex 通过简单地根据最大序列长度截断输入来解决了这个问题，但这可能会导致关键

计附加了一个1次示例。 有许多精心设计的序列化函数被提出，它们通常被认为是表格序列化的常用实践，包括@model_tapex。在这篇论文中，我们将这些不同的序列化方法作为基准，并进行公平比较。

SUC Benchmark

1）对于LLMs理解和表格，哪种输入设计和选择最有效？ 2）LLMs是否具有处理结构化数据的能力，且能够理解其结构？ 同时，我们评估了多种不同组合的输入设计，考虑到了复杂的权衡因素。。

Structural Understanding Capabilities

我们将"人类理解表格结构"的需求划分为两类，如图所示。

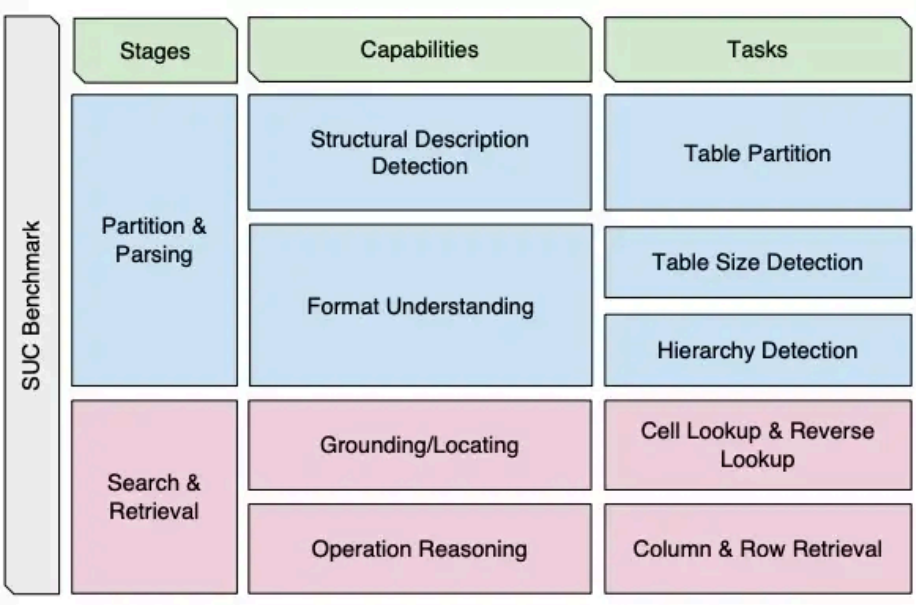


Figure 1: SUC Benchmark Overview

1）数据分隔与解析。在执行下游任务时，往往需要结合其他知识源来获得更多信息，以解决特定的任务。例如，在HybridQA系统中会运用到段落信息；而在TabFact和Feverous中，则会采用人工注释；至于MultiModalQA，则会利用图像信息。 然而，在处理这些下游任务之前，首先需要准确地将数据进行分隔，而且还需要具备区分表格和其他辅助信息*的能力以及对表格结构基本的理解。

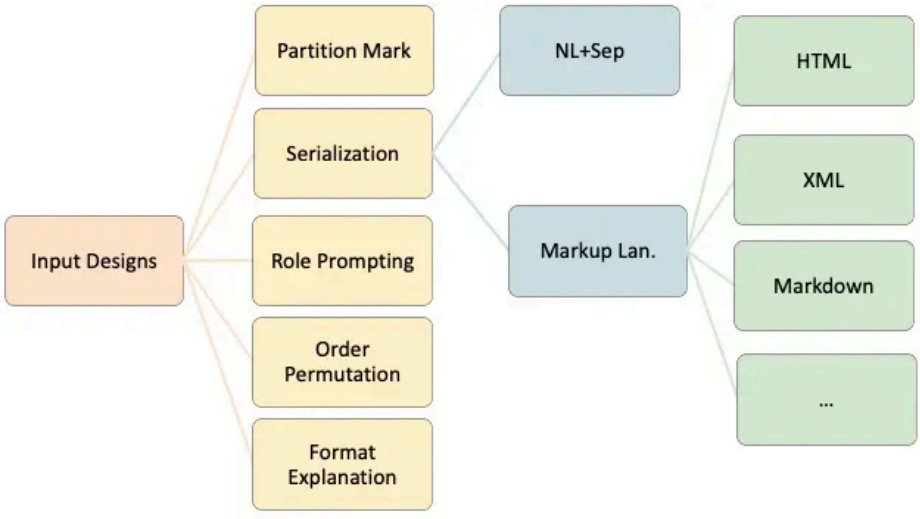


Figure 2: Input Designs for SUC Evaluation

非常相关。

Task Design

此任务旨在测试LLM识别表格结构的能力。LLM需要在给定的用户输入设计中确定表格的边界。该输入设计可能包括各种类型的辅助信息，如"描述"，"上下文"，"陈述"和"用户查询"。形式上，给定一个输入设计，$D = d_1, d_2, \ldots, d_n$，其中每个部分$d_i$都是一个"多功能"的序列，包含补充信息，如描述，上下文，陈述或用户查询。为了便于评估和比较，我们将LLM要求输出一个包括表格内容的表格边界的元组*，作为b_h和b_e。

Table 1: Input design of each task in our benchmark

Task	Input
Table Partition	What is the first token (cell value instead of separator |) of the given table? What is the end token (cell value instead of separator |) of the given table? Answer questions one by one and use | to split the answer.
Cell Lookup	What is the position of the cell value cell_value? Use row index and column index to answer
Reverse Lookup	What is the cell value of row index, column index ? Only output the cell value without other information
Column Retrieval	What is the column name with the index column_idx of the following table? Only give the column name without any explanation
Row Retrieval	What are the cell values of the row_idx row in following table? Only list the cell values one by one using | to split the answers
Size Detection	How many rows in the table? How many columns in the table. Answer the questions one by one and use | to split the answer
Merged Cell Detection	What is the column index of the cell which span is over 1. use | to split the answer (e.g., 3 | 4), the column index starts from 0. If there's no answer, return None

表格大小检测：这个任务十分重要，因为它能揭示LLM正确解析结构信息的能力。表格大小通常被忽略，但其实它是衡量表格行数和列数的直接指标。例如，如果一个表格只有三列，那么输出的答案就不应考虑超出这个范围的答案。形式上，给定一个有m行和n列的表格，LLM的正确答案应该是(m, n)。

合并单元格*检测：这个任务是评估LLM的解析结构信息能力，它可以通过检测表格中是否合并单元格来实现。合并单元格是表格构造中的特殊结构，它是由两个或多个相邻的单元格组合而成的更大的单元格。LLM被要求检测表格中的合并单元格索引，这些索引表示合并单元格的坐标。

查找和检索单元格：这个任务考察了LLM在搜索和检索结构信息方面的能力。LLM需要从指定位置准确地搜索和检索单元格值。这个任务依赖于LLM的信息分割和解析能力。如果找到多个具有相同值的单元格，LLM需要检索它们的位置；反之，对于特定的单元格位置，LLM应该检索对应的单元格值。

列和行检索：这个任务评估了LLM在检索和检索结构信息方面的能力，通过列出单元格值来实现。对于列检索，LLM需要列出给定表格中所有列名的单元格值；对于行检索，LLM需要列出特定行索引的所有单元格值。在这项任务中，我们希望LLM能够预测出结果与实际价值列表一致时的性能更好。我们预期列/行检索任务的性能会优于细胞查找和反向查找任务，因为使用列/行索来进行定位具体的值列表更为常见。

Data Collection and Reformatting of SUC

Question、FileName。每个样本与其对应的唯一问句相连，如"表格中有多少行（列）？"，原始数据集提供参考答案（"GroundTruth"）。

评估时需要遵循特定要求，如"逐个回答问题，并使用 | 分割答案"。我们使用GPT-3.5⁺对问题进行评估，并人工删除在多次生成过程中始终正确的答案。在此实验中，温度设置为0.7，在其他实验中设置为0。仅从ToTTo数据集中采样合并单元检测任务数据，因为这是唯一一个与合并单元相关联的数据源。每个任务设置下，我们从ToTTo数据集中随机抽取1,500张表用于测试，确保表格分布完整。

我们设计了一个基于表格任务的一次性上下文⁺学习基准。这意味着模型可以参考SUC中的示例并在生成答案时获取一些上下文信息。大型语言模型⁺具有随较少提示完成未见过任务的能力，而无需微调。此新兴能力并未被小型语言模型捕捉到。SUC充分利用了这一点来更好地揭示大型语言模型可能缺乏的潜在能力。我们还进行了零度设置的实验进行比较（见表 ）。

Evaluation

我们采用常见的输入设计方法来评估基准，并将其应用于不同的LLM以进行更深入的研究。具体而言，我们考虑CSV、JSON、XML、markdown⁺、HTML和XLSX作为不同的格式选项，每个格式都代表一种不同的信息压缩级别，并对LLM理解和表格内容提出了不同的挑战。

Table 2: Micro results of the benchmark. Change order [39] refers to put external text (like questions, statement) ahead of tables. Noted that "GPT-4" refers to the evaluation outcomes utilizing the GPT-4 model. Given the resource-intensive nature of GPT-4 calls, we only conducting the GPT-4 inference test on a subset of 300 samples (randomly sampled) from each task set. Each column follows the roles of graded color scale, *i.e.*, the deeper color refers to better perf.

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%	78.99%
XML	96.00%	97.64%	43.33%	72.28%	**55.00%**	**60.32%**	41.33%	68.28%	41.00%	**50.28%**	43.37%	80.21%	75.00%	80.52%
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%

此外，我们还考虑使用最常用的特殊令牌串联方式作为分隔符，如|作为基准。数字可以在表中找到。 我们也尝试其他输入设计选项，例如语法解释⁺、分区标记、角色提示和格式解释，作为输入设计的增强。更多细节可以在图和表中找到。 特别地，我们通过准确度来评估每项任务。为确保更好的评估，我们在输出格式上添加了一些限制。

Table 3: Micro ablation results of the input designs over benchmark.

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%	36.67%	-30.33%	48.00%	-28.67%
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%

例如，在表格分区任务中，我们要求用户一次性回答一个问题并使用 | 分割答案。从实证观察来看，超过90%的答案遵循这些特定的格式指示。对于剩下的10%，我们应用了一种使用正则表达式⁺（Re）的语义解析策略来解析答案。

Structural Prompting

我们发现在许多不同类型的表格任务中，LLMs在结构理解方面存在不足，即使是最简单的任务，如表格大小检测。正确选择输入设计组合是提升LLMs理解表格数据的重要方法之一。我们提出了一种简单且通用的方法，即自我增强提示，利用LLMs的自我知识生成额外的约束。这种方法可以显著提高LLMs在各种表格下游任务中的表现，例如表格大小检测（见表 ）。

知乎

Type	Choice	Tabfact Acc	HybridQA Acc	SQA Acc	Feverous Acc	TaTa BLEU-1	TaTa BLEU-2	TaTa BLEU-3	TaTa 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%	78.93%	46.91%	28.94%	19.32%
SA	self structural information description	73.42%	46.97%	75.97%	77.28%	78.93%	46.91%	28.94%	19.32%

我们发现，近期提出的CoT技术可以在LLMs上进行复杂的文本推理，并引发了大量相关研究。通过向模型提供几个示例的推理链，LLMs可以学会使用模板解决未曾见过的难题。受此启发，我们提出了一个简单、通用和有效的自增强提示方法，利用LLM内部检索生成中间的结构性知识。

我们在Table中设计了几种从LLM中提取知识的方法，例如让LLM生成格式规范以澄清输入格式模式是由LLM自身实现的。这些选择与以往的CoT和Zero-shot-CoT技术不同，因为它们关注于确定解锁LLM能够正确理解和处理结构化信息⁺的有效方法。此外，这种方法是模型无关的，这意味着任何标准的结构化数据推理任务都可以作为基础，并且还可以与其他基于提示的方法如自我一致性集成。

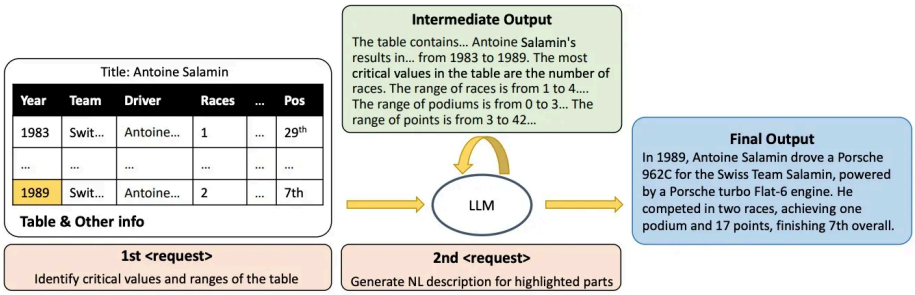


Figure 3: Illustration of self-augmented prompting. This process consists of two phases: 1) using self-augmented prompts to ask the LLM to generate additional knowledge (intermediate output) about the table; 2) incorporating the self-augmented response into the second prompt to request the final answer for a downstream task. As depicted in the figure, the LLM is able to identify important values in the table, which assists in generating a more accurate answer for the downstream task.

实证研究⁺表明，结构化信息对于理解表格至关重要。已有研究如@model_unifiedSKG 尝试将特殊标记植入到结构信息中作为提示。在这个基础上，我们借鉴这些成果和SUC基准测试的结果，探讨手动提示工程⁺作为一种自我增强提示的补充技术。

具体来说，我们会从原始输入中提取结构信息并将它融入输入本身。这可能包括使用单元格地址和明确指示表格中的行数和列数。这种扩展是为了提供额外的知识和限制，从而提高LLM在表格下游任务中的推理能力。 我们已经注意到LLM在表格大小检测任务上的表现不佳，所以我们考虑在输入中添加结构特征。例如，我们可以将关于表格大小和合并单元位置的信息附加到创建一个更具结构意识的学习环境，用于下游任务。

Results

Benchmark Highlights

我们比较了自然语言和带有特定分隔符的标记语言⁺（NL+Sep）的使用情况，如HTML、XML和JSON。尽管NL+Sep常用于表格下游任务，但我们发现使用HTML等标记语言比NL+Sep更优，提高了6.76%。我们假设GPT-3.5的训练过程中涉及代码调整，并且训练数据集包含大量的网页数据。因此，由于GPT-3.5的训练信息（参见 ）。

Downstream Tasks

知乎

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%	53.21%	20.12%

表格展示了使用结构特征检测技术对自我增强提示进行比较的结果。从表格中可以看出，模型在接受自我生成的信息之后，表现得更好，就像"SA"行中的模型一样。

对比1-shot w/o format explanation行和SA自格式解释行，可以看出，人工标记的格式说明可能会对下游任务如FEVEROUS造成负面影响。这是因为FEVEROUS中的表格结构更为复杂，包含许多段落和子表。这种结构性复杂性给GPT-3.5带来了一定的挑战。

同时，人工编写的知识并不能涵盖这种特性下的所有细节。相比之下，自我增强提示可以独立学习模式，并生成更全面、更有效的线索来回答问题。在表格中，我们提供了一个使用两种提示设计的FEVEROUS格式解释的例子。

发布于 2024-03-25 11:48 · IP 属地北京

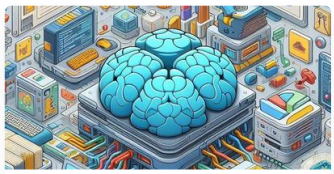
微软（Microsoft） LLM 数据结构化

理性发言，友善互动

发布

还没有评论，发表第一个评论吧

推荐阅读



LLM之新手入门：大语言的概念介绍与应用