Large Language Models are few(1)-shot Table Reasoners

Wenhu Chen
University of Waterloo, Vector Institute
wenhuchen@uwaterloo.ca

Abstract

Recent literature has shown that large language models (LLMs) are generally excellent few-shot reasoners to solve text reasoning tasks. However, the capability of LLMs on table reasoning tasks is yet to be explored. In this paper, we aim at understanding how well LLMs can perform table-related tasks with few-shot in-context learning. Specifically, we evaluated LLMs on popular table QA and fact verification datasets like WikiTableQuestion, FetaQA, TabFact, and FEVEROUS and found that LLMs are competent at complex reasoning over table structures, though these models are not pre-trained on any table corpus. When combined with ‘chain of thoughts’ prompting, LLMs can achieve very strong performance with only a 1-shot demonstration, even on par with some SoTA models. We show that LLMs are even more competent at generating comprehensive long-form answers on FetaQA than tuned T5-large. We further manually studied the reasoning chains elicited from LLMs and found that these reasoning chains are highly consistent with the underlying semantic form. We believe that LLMs can serve as a simple yet generic baseline for future research. The code and data are released in https://github.com/wenhuchen/TableCoT.

1 Introduction

The problem of structured knowledge grounding has been extensively studied for many years. Tables, as one of the most popular (semi)-structured forms to store world knowledge receive significant attention from the natural language processing (NLP) community. Traditional approaches mostly rely on synthesizing executable languages like SQL or SPARQL to access the information inside the table. However, these symbolic languages normally make a rigid assumption about the table and cannot capture the semantics of text chunks inside the table. Such issues are even more pronounced with web tables due to their irregular forms. To fully understand web tables, both structured reasoning and textual reasoning are required. Such challenges have attracted many researchers to work in the field.

Recently, there have been efforts like UnifiedSKG (Xie et al., 2022) aiming to unify these heterogeneous table-based tasks as a generic text-to-text format. UnifiedSKG has shown that using T5-3B (Raffel et al., 2020) with the text-to-text format can already achieve state-of-the-art performance on almost all the table-based tasks without task-specific designs. However, the proposed text-to-text models still need to be fully fine-tuned on the downstream tasks. UnifiedSKG also identified that T0-style (Sanh et al., 2022) cross-task transfer can only achieve almost random performance.

Wei et al. (2022); Wang et al. (2022); Zhou et al. (2022); Drozdov et al. (2022) have recently discovered that large language models (Brown et al., 2020; Chowdhery et al., 2022; Ouyang et al., 2022) can be used to solve complex mathematical and commonsense reasoning tasks with few-shot in-context learning. Inspired by this discovery, we aim at understanding whether these LLMs can also solve complex table-based reasoning tasks. Though the LLMs are not specifically designed to encode ta-
Figure 1: In-context learning for table-related tasks with chain-of-thoughts reasoning.

bles, given the enormous number of tables present in the pre-training corpus, we believe they are also competent at reasoning over table information.

In this paper, we experimented with few-shot in-context learning for LLMs as depicted in Figure 1. Instead of fine-tuning the model, we only provide a few examples to showcase the desired input-output format as the condition for the model to follow to solve unseen test examples. We experiment with several prompting variants including (1) direct prediction, (2) Chain of Thoughts (Wei et al., 2022) (CoT), (3) Chains of thoughts with self-consistency (Wang et al., 2022) (CoT+SC). We evaluate these methods on WikiTableQA (Pasupat and Liang, 2015), FetaQA (Nan et al., 2022), TabFact (Chen et al., 2019) and FEVEROUS (Aly et al., 2021). Our results reveal that LLMs (Ouyang et al., 2022; Chen et al., 2021a; Chowdhery et al., 2022) can achieve striking performance with only 1 or 2 demonstrations, e.g. 48.8% on WikiTableQuestions and 78.8% on TabFact, which are on par some near-SoTA models (Yu et al., 2021; Eisen- schlos et al., 2020). On other datasets like FetaQA with long-form answers, our human evaluation reveals that GPT-3 can significantly outperform the fine-tuned T5-large by more than 30% in terms of correctness and adequacy.

Furthermore, we manually studied the chain of thoughts elicited from LLMs and found that the rationale is highly consistent with the ‘ground truth’ semantic forms when the model predictions are correct. We found that these models are surprisingly competent at performing symbolic operations over the table, like maximum, minimum, counting, comparison, addition, and difference. However, we also identify several issues of the LLMs on these table reasoning tasks: (1) due to the token limitation, the model is unable to generalize to ‘huge’ tables with 30+ rows, which is the major error source, (2) LLMs can sometimes make simple mistakes when performing symbolic operations.

Due to the simplicity and generality, we believe LLMs with CoT should be used as an important baseline for any future table-related research.

2 Related Work

2.1 Reasoning over Tables

Table-based reasoning is traditionally accomplished by semantic parsing to execute commands on tables like WikiTableQuestions (Pasupat and Liang, 2015), WikiSQL (Zhong et al., 2017), and Spider (Yu et al., 2018). These models aim to synthesize SQL/SPARQL to interact with tables. However, these machine languages have a rigorous requirement regarding the tables, e.g. the value in the same column should follow the same data type. Such rigorous assumptions are frequently violated by web tables containing unnormalized free-form text in cells. Therefore, language understanding inside the table is essential to achieve a better score. Recently, Yin et al. (2020); Herzig et al. (2020); Liu et al. (2021); Deng et al. (2022) have proposed to pre-train table and text to learn joint representation. These pre-trained models can use joint representation to perform reasoning implicitly without relying on symbolic execution. By pre-training the model on large-scale crawled or synthesized data, these models can normally achieve the best-known performance on table tasks. However, these models still require a significant amount of fine-tuning on the downstream datasets. Unlike these methods, we are interested in in-context learning, where the model can only learn with a
few examples (demonstration) without any fine-tuning. One contemporary work similar to ours is BINDER (Cheng et al., 2022), which utilizes Codex to synthesize SQL to execute logical forms against tables for question answering. One big difference is that BINDER (Cheng et al., 2022) involves logical form execution, if the execution fails, BINDER will fall back to using language models to answer the question, which is more similar to ours.

2.2 In-context Learning with LLMs

GPT-3 (Brown et al., 2020) and other large language models demonstrated strong abilities to perform few-shot predictions without fine-tuning, where the model is given a description of the task in natural language with few examples. Scaling model size, data, and computing are crucial to enable this learning ability. Recently, (Rae et al., 2021; Smith et al., 2022; Chowdhery et al., 2022; Du et al., 2022) have proposed to train different types of large language models with different training recipes. The LLMs have demonstrated a striking capability utilizing the few-shot prompts to accomplish unseen tasks without any fine-tuning, which is found to be an emergent capability not presented in smaller language models.

2.3 Chain of Thoughts Reasoning

Although LLMs (Brown et al., 2020; Chowdhery et al., 2022) have demonstrated remarkable success across a range of NLP tasks, their ability to demonstrate reasoning is often seen as a limitation. Such capability cannot be acquired simply by scaling up the model size. Recently, the ‘chain of thoughts’ prompting (Wei et al., 2022) has been discovered to empower LLMs to perform complex reasoning over text. By providing the model with several exemplars of reasoning chains, LLMs can learn to follow the template to solve difficult unseen tasks. Later on, Wang et al. (2022) propose to use self-consistency with CoT to further improve performance. Later on, Kojima et al. (2022) discovered that LLMs can even perform reasoning without any demonstration by using appropriate prompts. These recent findings reveal the strong capability of LLMs to perform complex reasoning. However, the current studies are still heavily focused on text-based tasks like question answering, common sense reasoning, etc. The models’ capability to reason over tables is yet unknown. In this paper, we are specifically interested in understanding LLMs’ capability to reason over web tables with CoT prompting.

3 Method

We experiment with different in-context learning methods to solve the table-based reasoning tasks. To formulate the prompt, we linearize the table and concatenate it with a few examples as demonstrations of the language model to predict the output from an unseen test example. The format is described in Figure 2. We mainly investigate three different variants for language model prompting, including (1) Direct Prediction, (2) Chain of Thoughts (CoT), and (3) Chain of Thoughts + Celf-Consistency decoding (CoT+SC). For self-consistency methods, we use LLMs to generate five diverse reasoning paths and then use majority voting to select the most voted answer.

To limit the budget and constrain the input token length, we truncate the input tables to contain only few examples (demonstration) without any fine-tuning. One contemporary work similar to ours is BINDER (Cheng et al., 2022), which utilizes Codex to synthesize SQL to execute logical forms against tables for question answering. One big difference is that BINDER (Cheng et al., 2022) involves logical form execution, if the execution fails, BINDER will fall back to using language models to answer the question, which is more similar to ours.

2.2 In-context Learning with LLMs

GPT-3 (Brown et al., 2020) and other large language models demonstrated strong abilities to perform few-shot predictions without fine-tuning, where the model is given a description of the task in natural language with few examples. Scaling model size, data, and computing are crucial to enable this learning ability. Recently, (Rae et al., 2021; Smith et al., 2022; Chowdhery et al., 2022; Du et al., 2022) have proposed to train different types of large language models with different training recipes. The LLMs have demonstrated a striking capability utilizing the few-shot prompts to accomplish unseen tasks without any fine-tuning, which is found to be an emergent capability not presented in smaller language models.

2.3 Chain of Thoughts Reasoning

Although LLMs (Brown et al., 2020; Chowdhery et al., 2022) have demonstrated remarkable success across a range of NLP tasks, their ability to demonstrate reasoning is often seen as a limitation. Such capability cannot be acquired simply by scaling up the model size. Recently, the ‘chain of thoughts’ prompting (Wei et al., 2022) has been discovered to empower LLMs to perform complex reasoning over text. By providing the model with several exemplars of reasoning chains, LLMs can learn to follow the template to solve difficult unseen tasks. Later on, Wang et al. (2022) propose to use self-consistency with CoT to further improve performance. Later on, Kojima et al. (2022) discovered that LLMs can even perform reasoning without any demonstration by using appropriate prompts. These recent findings reveal the strong capability of LLMs to perform complex reasoning. However, the current studies are still heavily focused on text-based tasks like question answering, common sense reasoning, etc. The models’ capability to reason over tables is yet unknown. In this paper, we are specifically interested in understanding LLMs’ capability to reason over web tables with CoT prompting.

3 Method

We experiment with different in-context learning methods to solve the table-based reasoning tasks. To formulate the prompt, we linearize the table and concatenate it with a few examples as demonstrations of the language model to predict the output from an unseen test example. The format is described in Figure 2. We mainly investigate three different variants for language model prompting, including (1) Direct Prediction, (2) Chain of Thoughts (CoT), and (3) Chain of Thoughts + Celf-Consistency decoding (CoT+SC). For self-consistency methods, we use LLMs to generate five diverse reasoning paths and then use majority voting to select the most voted answer.

To limit the budget and constrain the input token length, we truncate the input tables to contain only
the first 22 rows and the first 8 columns. For each cell, we truncate the word length to contain only the first 10 words. Through such truncation, we can restrict the input token length to within 2000 tokens. We will talk about the impact of input token length on the final performance.

4 Experimental Results

For the GPT-3 experiments, we used the four provided models, Ada, Babbage, Curie, and Davinci with 350M, 1.3B, 6.7B, and 175B parameters respectively. We mainly use Davinci-text-002 (Ouyang et al., 2022) in our experiments. We also report results for Codex (Chen et al., 2021a) (Davinci-code-002) on some datasets. We use a temperature of 0.7 without any frequency penalty and without top-k truncation. We found that the model performance is robust to the sampling strategies and the hyper-parameters. These models are mainly trained on web-crawled data and code data, without any specialized training on table corpus.

4.1 Datasets

Here we list all of our datasets as follows:

**WikiTableQuestions**  Pasupat and Liang (2015) consists of complex questions annotated based on Wikipedia tables. Crowd Workers are asked to compose a series of complex questions that include comparisons, superlatives, aggregation, or arithmetic operations. The annotated dataset is cross-validated by other crowd workers. In our experiments, we use the unseen test set for evaluation. We evaluate the standard test set with roughly 4000 questions. In this dataset, we adopt the answer exact match as our evaluation metric.

**FetaQA**  Nan et al. (2022) consists of free-form table questions. These questions are mostly complex questions that require integrating information from discontinuous chunks in the table. Instead of having short answers, the dataset annotates long free-form answers. Unlike other datasets using copies of short text spans from the source, the questions in FetaQA require a high-level understanding. We adopt sacre-BLEU and human evaluation as our evaluation metrics. The evaluation set contains a total of 2003 examples.

**TabFact**  Chen et al. (2019) consists of both simple and complex claims annotated by crowd workers based on Wikipedia tables. In the simple subset, the claims normally do not involve higher-order operations like max/min/count, etc. While the complex subset mainly contains claims involving higher-order operations. We evaluate the original test set containing 12,779 examples. We report binary classification accuracy on the set.

**FEVEROUS**  Aly et al. (2021) consists of compositional claims annotated by crowd workers regarding Wikipedia tables. Since the dataset contains both table-supported and text-supported claims. We filter out text-supported claims and only keep the 2,295 table-supported claims as our test set. Different from TabFact, FEVEROUS consists of more complex tables with irregular structures like multi-row, multi-column, multi-table, etc. We report dev-set accuracy.

4.2 Baselines

In these experiments, we mainly consider the following baseline models.

**Pre-trained Encoder-Decoder Model** Pre-trained encoder-decoder model is one of our competitors, which aims to encode the table as a plain sequence into the encoder, and then apply the decoder to generate either an answer or a verdict. In this paper, we mainly compare against T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) as our baselines.

**Pre-trained Table Understanding Model** This family of models is specifically pre-trained on the table-related corpus, which utilizes specific architecture to encode table structure and handle symbolic computation. In this paper, we mainly consider TAPAS (Herzig et al., 2020), TABERT (Yin et al., 2020), and TAPEX (Liu et al., 2021).

**Neural Symbolic Model** This family of models includes a non-pre-trained neural symbolic model, which can synthesize machine language to interact with the table. This line of work includes LogicFactChecker (Zhong et al., 2020), Neural-Symbolic Machine (Liang et al., 2018), etc.

4.3 Main Results

Here we show our main results for different datasets as follows.

**WikiTableQuestions**  As can be seen from Table 1, directly asking GPT-3 to generate answers can only lead to 26% EM score. However, if we prompt the model with the CoT demonstrations,
Table 1: Experimental Results on WikiTableQuestions. PT means pre-training and FT means fine-tuning.

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>Test EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Pasupat and Liang (2015)</td>
<td>37.1</td>
</tr>
<tr>
<td>Train</td>
<td>Zhang et al. (2017)</td>
<td>43.7</td>
</tr>
<tr>
<td>Train</td>
<td>Liang et al. (2018)</td>
<td>43.7</td>
</tr>
<tr>
<td>Train</td>
<td>Agarwal et al. (2019)</td>
<td>44.1</td>
</tr>
<tr>
<td>Train</td>
<td>Wang et al. (2019)</td>
<td>44.5</td>
</tr>
<tr>
<td>PT + FT</td>
<td>Herzig et al. (2020)</td>
<td>48.8</td>
</tr>
<tr>
<td>PT + FT</td>
<td>Yu et al. (2021)</td>
<td>52.7</td>
</tr>
</tbody>
</table>

GPT-3 is more likely to follow the logical operation to derive the answers. With two demonstrations, GPT-3 can achieve roughly 46% EM score. By switching from GPT-3 to Codex, we are able to further improve the EM score to over 48.8%. These results are particularly surprising given that TAPAS has a built-in module to complete symbolic operations, while GPT-3 was not trained on any table-specific dataset. These results demonstrate GPT-3’s built-in capabilities to perform diverse types of reasoning over tables.

**FetaQA**  
As demonstrated in Table 2, we compare GPT-3 with different fine-tuned models from Nan et al. (2022). Unlike the other datasets with short phrase answers, the goal of this dataset is to generate a complete long-form answer. Unlike WikiTableQuestion, the questions normally do not involve complex operations like max, min, compare, average, etc. The long-form answer is similar to the role of CoT. Therefore, we only applied ‘direct generation’ in this experiment. In terms of BLEU score (Papineni et al., 2002), GPT-3 is still a bit behind the fine-tuned T5-large. However, the BLEU score cannot reflect the faithfulness and correctness of the model generation. Thus, we follow Nan et al. (2021) to do human evaluation over the four aspects: (1) fluency (whether the generated sentence contains the linguistic error), (2) correctness (whether the generated sentence answers the question correctly), (3) faithfulness (whether the generated sentence is grounded on the input table), and (4) adequacy (whether the generated sentence is comprehensive enough to cover all the answers). We list our results in Table 3. Similarly, we also sample 100 model predictions and manually evaluate their quality and adopt binary scores for each example. As can be seen, GPT-3 can significantly outperform T5-large over all the aspects, i.e. more than 30% improvement over correctness, adequacy, and faithfulness. The evaluation indicates that the model output is almost on par with the average human performance on this dataset.

Table 2: Experimental Results on FetaQA. PT means pre-training and FT means fine-tuning.

<table>
<thead>
<tr>
<th>Source</th>
<th>Fluency</th>
<th>Correct</th>
<th>Adequate</th>
<th>Faithful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipeline</td>
<td>85.2</td>
<td>25.4</td>
<td>23.6</td>
<td>23.6</td>
</tr>
<tr>
<td>T5-large</td>
<td>94.6</td>
<td>54.8</td>
<td>50.4</td>
<td>50.4</td>
</tr>
<tr>
<td>Human</td>
<td>95.0</td>
<td>92.4</td>
<td>95.6</td>
<td>95.6</td>
</tr>
<tr>
<td>GPT-3</td>
<td>98.0</td>
<td>84.0</td>
<td>78.0</td>
<td>90.0</td>
</tr>
</tbody>
</table>

Table 3: Human Evaluation Results on FetaQA.

**TabFact**  
As demonstrated in Table 4, we compare GPT-3 against the other pre-trained and fine-tuned models including TAPAS (Eisenschlos et al., 2020), TAPEX (Liu et al., 2021), etc. We show that GPT-3 direct prediction is already getting a decent accuracy of 72%, which is slightly higher than Logic FactChecker (Zhong et al., 2020). When combined with CoT reasoning, the model accuracy increases to over 77%. Similar to before, we found that Codex can generate more accurate reasoning chains, thus achieving better accuracy of 78.8%, which is only 2% lower than pre-trained table understanding model TAPAS (Eisenschlos et al., 2020). The more intriguing property about LLM + CoT is that the intermediate rationale can be produced without any training. All the existing trained models do not have the capability to produce the intermediate reasoning steps due to the lack of annotation in the dataset.

**FEVEROUS**  
We demonstrate our results on FEVEROUS dev-set in Table 5 and compare different-sized UnifiedSKG models (built with T5). We found that GPT-3’s performance with direct prediction is similar to UnifiedSKG-base. Similar to TabFact, we found that the model performance can be boosted with ‘chain of thoughts’ prompt-
Table 4: Experimental Results on TabFact. PT means pre-training and FT means fine-tuning.

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>Chen et al. (2019)</td>
<td>65.1</td>
</tr>
<tr>
<td>FT</td>
<td>Zhong et al. (2020)</td>
<td>71.1</td>
</tr>
<tr>
<td>FT</td>
<td>Zhang et al. (2020)</td>
<td>73.2</td>
</tr>
<tr>
<td>FT</td>
<td>Yang et al. (2020)</td>
<td>74.4</td>
</tr>
<tr>
<td>FT</td>
<td>Lewis et al. (2020)</td>
<td>82.5</td>
</tr>
<tr>
<td>FT + FT</td>
<td>Eisenschlos et al. (2020)</td>
<td>81.0</td>
</tr>
<tr>
<td>PT + FT</td>
<td>Liu et al. (2021)</td>
<td>84.2</td>
</tr>
</tbody>
</table>

1-shot    | GPT-3 Direct                  | 72.0    |
2-shot    | GPT-3 Direct                  | 73.9    |
1-shot    | GPT-3 CoT                     | 75.5    |
2-shot    | GPT-3 CoT                     | 76.0    |
1-shot    | GPT-3 CoT+SC                  | 77.3    |
2-shot    | Codex CoT                     | 78.8    |

Table 5: Experimental Results on FEVEROUS. PT means pre-training and FT means fine-tuning.

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>Dev Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>Aly et al. (2021)</td>
<td>82.23</td>
</tr>
<tr>
<td>FT</td>
<td>UnifiedSKG-base (Xie et al., 2022)</td>
<td>75.05</td>
</tr>
<tr>
<td>FT</td>
<td>UnifiedSKG-large (Xie et al., 2022)</td>
<td>79.81</td>
</tr>
<tr>
<td>FT</td>
<td>UnifiedSKG-3B (Xie et al., 2022)</td>
<td>82.40</td>
</tr>
<tr>
<td>1-shot</td>
<td>GPT-3 Direct</td>
<td>74.20</td>
</tr>
<tr>
<td>2-shot</td>
<td>GPT-3 Direct</td>
<td>75.22</td>
</tr>
<tr>
<td>1-shot</td>
<td>GPT-3 CoT</td>
<td>75.70</td>
</tr>
<tr>
<td>2-shot</td>
<td>GPT-3 CoT</td>
<td>76.44</td>
</tr>
<tr>
<td>1-shot</td>
<td>GPT-3 CoT+SC</td>
<td>77.22</td>
</tr>
</tbody>
</table>

We also provide a few mistakes made by GPT-3 in Figure 5. In the first example, GPT-3 miscounts the ‘number of countries above 1 billion box office’ because it misidentifies ‘world’ also as a country. In the second example, GPT-3 misunderstood ‘2nd highest’ as ‘highest’, which leads to prediction error. In the last example, GPT-3 misunderstands the semantics of the question and answers ‘left office time’ instead of ‘took office time’. These examples show the typical errors of grounding the inputs to the wrong rows or columns of the table.

4.5 Analysis

Impact of Number of Shots

First of all, we conduct an ablation study to understand the impact of a number of shots in the final performance. In order to control the budget, we only sample 200 samples from WikiTableQuestions, TabFact and FEVEROUS for this ablation study. As can be seen from Figure 7, GPT-3 is not quite sensitive to the number of provided demonstrations. Increasing from 1-shot to 2-shot can often benefit the model, however, increasing the shot number further does not yield more performance gain. We conjecture that instruct fine-tuning used in GPT-3 (Ouyang et al., 2022) can easily extrapolate the task meaning, thus, having a single demonstration is already enough for the model to understand the task.
Quality Evaluation of Reasoning Chains

We conduct a human evaluation to assess whether GPT-3 is making the correct prediction with the correct reasons. Specifically, we sample 100 reasoning paths from the correctly predicted examples and manually study whether these reasoning chains are grounded on the table or simply ‘hallucination’. As can be seen from Figure 7, we found that around 90% of reasoning chains are faithful to the information on the table, and only less than 10% of the reasoning chains are hallucinated. Based on this evaluation, the answer is 7.

Claim: more players have the position of gk than any other position.
There are 2 players with the position of gk and 5 players with the position of d. Therefore, the claim is false.

Figure 4: ‘Correct’ predictions from WikiTableQuestions, TabFact, and FetaQA datasets, where the ‘blue’ text are the outputs from the GPT-3, ‘red’ means the correct rows to reference.

Figure 5: ‘Wrong’ predictions from WikiTableQuestions, TabFact, and FetaQA datasets, where ‘blue’ text are the outputs from the GPT-3, ‘red’ means the region of the correct cell to reference, and ‘green’ means the reference trusted by GPT-3.

Question: how many countries had at least $1 billion in box office? Canada/United States, China, Japan, United Kingdom, France, South Korea, India, Germany and Russia. All world had at least $1 billion in box office. Therefore, the answer is 11.

Claim: knight in retreat is the 2nd highest NO in series.
Knight in strike has no of series of 67, which is the highest no in series, therefore, the claim is false.

Figure 6: ‘Wrong’ predictions from WikiTableQuestions, TabFact, and FetaQA datasets, where ‘blue’ text are the outputs from the GPT-3, ‘red’ means the region of the correct cell to reference, and ‘green’ means the reference trusted by GPT-3.
evaluation, we believe that LLMs are not guessing the answers correctly by chance.

We believe these ‘reasoning chains’ are useful in many aspects: (1) the chains can provide a rationale to humans to justify the decision-making process. (2) one of the notorious annotation tasks is to annotate the ‘underlying’ semantic form for many NLP tasks, which require expertise for human annotators, on the other hand, the annotation cost is huge. Using GPT-3 to demonstrate useful natural language ‘semantic forms’ could potentially greatly lower the annotation burden of these tasks.

**Impact of Table Size** An important factor for model performance is the size of the table. Here we want to understand how relevant the model performance is w.r.t the input table length. We group the table token length into different groups like ‘0-100’, ‘100-200’, etc, and plot the group-wise accuracy for WikiTables and TabFact in Figure 8. As can be seen from the table, we found that GPT-3’s performance is highly sensitive to the table size. As the table size grows, the accuracy almost decreases monotonically. After the table size exceeds 1000 tokens (e.g. 1500 word pieces), GPT-3’s performance almost degrades to random guesses. This ablation study reveals one of the drawbacks of using LLMs for table reasoning. To further enhance LLMs’ performance, we need to develop better methods to maintain more consistent performance across different-sized tables.

**Discussions** In this study, we investigate the possibilities of prompting LLMs to perform complex reasoning tasks over tables. However, we do not believe LLM prompting can replace the existing symbolic methods. LLMs have several favorable properties: (1) no annotation is needed, and (2) the functional coverage is broader than symbolic methods. However, LLM prompting exhibits unpredictable randomness and cannot generalize to large tables. In contrast, symbolic models are (1) agnostic to the table size, and (2) can reliably perform designed functions without much randomness. But they in general require a significant amount of annotated data to learn.

In conclusion, these two types of models are complementary to each other. To push the limit forward, we need to investigate how to combine the merits of these two types of methods. For example, the symbolic methods can perform certain operations to narrow down to a targeted region in the table, and then LLMs can be used to reason over the limited information.

**5 Conclusion**

In this paper, we investigate whether the current LLMs (GPT-3) can be directly utilized to perform table reasoning tasks. Surprisingly, though LLMs are not optimized for table-based tasks, we found these models highly competent in performing complex table reasoning tasks, especially when combined with ‘chain of thoughts’ prompting. We believe this study can open new possibilities for LLM application in table-related tasks to either directly predict the output or to serve as an auxiliary tool for annotating complex intermediate forms.
Limitations

Our approach has several limitations: (1) the proposed approach is still far from state-of-the-art performance, and there is still room for improvements before it can be used as an alternative. (2) the method is still costly, we show that the model can only achieve superior performance when scaling up. Smaller-sized models are still weak at reasoning. Therefore, we need to consider how to empower smaller models with such reasoning capabilities.

References


1129


