

Are Large Language Models Table-based Fact-Checkers?

Hanwen Zhang^{1,2}, Qingyi Si^{1,2}, Peng Fu^{1,2}, Zheng Lin^{1,2}, and Weiping Wang¹

¹Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

²School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

{zhanghanwen, siqingyi, fupeng, linzheng, wangweiping}@iie.ac.cn

Abstract—Table-based Fact Verification (TFV) aims to extract the entailment relation between statements and structured tables. Existing TFV methods based on small-scaled models suffer from insufficient labeled data and weak zero-shot ability. Recently, the appearance of Large Language Models (LLMs) has gained lots of attraction in research fields. They have shown powerful zero-shot and in-context learning abilities on several NLP tasks, but their potential on TFV is still unknown. In this work, we implement a preliminary study about whether LLMs are table-based fact-checkers. In detail, we design diverse prompts to explore how the in-context learning can help LLMs in TFV, i.e., zero-shot and few-shot TFV capability. Besides, we carefully design and construct TFV instructions to study the performance gain brought by the instruction tuning of LLMs. Experimental results demonstrate that LLMs can achieve acceptable results on zero-shot and few-shot TFV with prompt engineering, while instruction-tuning can stimulate the TFV capability significantly.¹ We also make some valuable findings about the format of zero-shot prompts and the number of in-context examples. Finally, we analyze some possible directions to promote the accuracy of TFV via LLMs, which is beneficial to further research of table reasoning.

Index Terms—Table-based Fact Verification, Large Language Models, In-context Learning, Instruction Tuning

I. INTRODUCTION

Fact verification aiming to determine the veracity of given claims has been increasingly popular for downstream applications like fake news detection and misinformation identification. Initial studies on fact verification focused on unstructured textual evidence [1], while evidence with table format has attracted more attention in recent years [2], [3]. Table-based fact verification (TFV) intends to identify whether a statement is supported or refuted by the structured or semi-structured table, which is more challenging due to the rich structure information of tables and complex reasoning types of statements. Fig.1 shows an example of table-based fact verification. To verify the statement “only round 3 is not listed two times”, we should first locate the “round” column in the table, and then count the number of occurrences for each element.

Previous studies have mainly proposed two kinds of methods for table-based fact verification: (1) program-based methods [2], [4]–[6] and (2) pretrain-based methods [7]–[11]. Program-based methods utilize the semantic parser trained under weakly-supervised settings to transform statements into logical forms with logical and numerical semantics. The

Table:

round	player	position	nationality
3	rob bellamy	right wing	united states
4	r j anderson	defense	united states
4	david laliberte	right wing	canada
5	chris zarb	defense	united states
5	gino pisellini	right wing	united states
6	ladislav scurko	center	slovakia
6	frederik cabana	center	canada
8	martin houle	goaltender	canada
8	travis gawryletz	defense	canada
9	triston grant	left wing	canada
9	john carter	center	united states

Statement:

only round 3 is not listed two times

Label:

supported

Fig. 1. An example of the Table-based Fact Verification task.

pretrain-based methods retrain the text-based pre-trained language models (PLMs) on table-aware pre-training tasks [7], [8], [11] or extra model structure [10] to enhance the reasoning ability on tables. However, both two kinds of methods rely on sufficient training data and have difficulties explaining their judgment.

Meanwhile, the closed-source large language models (LLMs) such as GPT-3 [12], Codex [13] and ChatGPT demonstrate a marvelous in-context learning ability (especially zero-shot and few-shot learning) to achieve competitive performances in various NLP tasks [14]–[16]. But they can be used only by calling API interfaces with designed prompts. The open source of LLaMA [17], a competitive LLM, and the success in instruction tuning of LLMs [18], [19] make it possible to deploy and customize LLM for various tasks. The closed-source and open-source LLMs, intuitively, can especially promote research directions of zero-/few-shot learning and instruction tuning for table-based face verification, which is still in its infancy. Existing related works [20], [21] only regard LLMs as an external knowledge base, which is used to enhance small-scaled models. Until now, there has been no research on directly using LLMs to complete TFV task. This leaves the question “Are large language models table-based fact-checkers” unanswered.

To answer the question above, this paper conducts comprehensive research on LLMs for table-based fact verification under three settings. First, we explore the performance of LLMs

¹Our result files can be found at: <https://github.com/Heaven-zhw/LLM-on-tabfact>

under the zero-shot setting, which represents the inherent TFV capability of LLMs. Second, we use LLMs to achieve TFV in few-shot fashion, which reflects how in-context learning promotes TFV. Third, we construct the TFV instructions for tuning LLaMA to induce the full potential of LLMs in TFV.

We conduct extensive experiments on the above three settings and make some valuable findings and conclusions. For zero-shot learning of LLMs in TFV, we discover a more robust design of prompts. For few-shot learning, it is not that more in-context examples are better and 2-shot is a good choice. However, the prompt engineering under the above two in-context learning settings cannot significantly improve LLMs' TFV capability, while instruction-tuning of LLMs can. Finally, we discuss some research directions to improve the performance of TFV via LLMs in future studies.

II. RELATED WORK

A. Table-based Fact Verification

Program-based and pre-training-based methods are two kinds of existing mainstream methods on TFV. Program-based methods synthesize latent programs to access structured tables [2], or utilize programs indirectly to get the logical and numerical semantics [4], [5]. Pretrain-based methods, inspired by the success of general PLMs in natural language tasks, are proposed to jointly learn various table tasks [7], [8], [11] for better table understanding. These models are pre-trained with newly designed table-specific tasks (e.g., SQL execution [8] and sentence-table cloze [11]) on massive table and natural language data [7], [11]. More recently, some works [20], [21] also leverage LLMs in their methods. However, they serve LLMs as auxiliary components (e.g., semantic parser [20], evidence decomposers [21]) to enhance the small-scaled models, while the exploration of using LLMs directly as table-based fact-checkers has not yet been carried out.

B. Prompt Engineering of Large Language Models

Prompt engineering is an important topic in the era of LLMs. Since GPT3 [12] demonstrated excellent few-shot capabilities, researchers have been attempting to continuously guide various abilities of LLMs through the design of prompts, also known as in-context learning. The emergence of ChatGPT further promotes the development of prompt engineering. Because these competitive LLMs have been closed source for a long time (until the leak of LLaMA), researchers can only achieve specific tasks [14]–[16] by calling API interfaces to transmit designed prompts. This paper attempts to leverage LLMs in TFV task under zero-shot and few-shot settings.

C. Instruction Tuning of Large Language Models

Base LLMs are primarily pre-trained with the next word prediction objective, which can be promoted to massive unsupervised texts. This pre-training stage is the main source of knowledge accumulation language understanding and generation ability. Recently, the open source community [18], [22]

has found that instruction fine-tuning base LLMs can significantly improve their instruction-following ability to better achieve complex tasks.

However, instruction-tuning LLMs usually needs huge computing power. Parameter-efficient fine-tuning (PEFT) methods [23], [24] make it possible to fine-tune LLMs at a relatively low cost. These strategies are widely used for fine-tuning LLMs under the condition of limited computational resources.

III. LARGE LANGUAGE MODELS FOR TFV

A. Task Definition

Here is a symbolized definition of the table-based fact verification task: Given a structured table $T = \{t_{ij} | i \leq m, j \leq n\}$ with m rows and n columns and a statement $S = \{w_i | i \leq W_S\}$ with W_S words, we need a verification model f to predict a verdict label $\hat{y} = f(S, T)$, usually $\hat{y} \in \{0, 1\}$ for binary classification datasets.

As the blossom of LLMs for several tasks, we wonder whether LLMs are good table-based fact-checkers. To answer this question, we induce LLMs to directly generate the final answer for TFV. We have explored their capability in TFV under the in-context learning (i.e., zero-shot and few-shot prompt engineering) and instruction tuning settings, which will be discussed in Section IV-A, Section IV-B, and Section V individually.

B. Dataset and Experimental Settings

We evaluate LLMs on TabFact [2], a TFV benchmark dataset, including 16K tables from Wikipedia for 118K natural language statements written by crowdworkers. The statements are labeled from the collection {supported, refuted}. The whole dataset is randomly divided into train, development, and test subsets of splits of the original test sets with the ratio of 8:1:1, wherein a small test set with 1998 samples is further divided from the original test set for human evaluation. To facilitate comparison with humans and reduce the cost, we use the small test for our experiments, noted as “TabFact-small-test”.

For evaluation, we use accuracy as the evaluation metric followed by the original setting. We extract the predicted label from the generative content by keyword matching, but sometimes we can't extract the keywords from the responses. In all our experiments, we regard these unrelated or empty responses as REFUTED class.

This paper mainly uses representative LLMs, ChatGPT and LLaMA, to conduct experiments. The former is the most popular LLM base in the zero-/few-shot learning research of LLMs, which has sparked a series of “API” research in prompt engineering. The latter is the most competitive open source LLM, which has important significance in instruction-tuning research.

For ChatGPT, we conduct the experiments by API version of GPT-3.5-turbo. For LLaMA, the original version [17], the iterative and length-enhanced version [25], and a further fine-tuned and dialogue-adaptive version are involved in our experiments, namely noted as “LLaMA-1”, “LLaMA-2” and

“LLaMA-2-chat”, the latter two of which belong to LLaMA2 family. All versions of LLaMA we use are with 7B parameters. According to the length of context windows, we set the max length of input 2048 for LLaMA-1, 4096 for LLaMA-2 and LLaMA-2-chat.

IV. IN-CONTEXT LEARNING OF LLMs

We first adopt the prompt engineering strategies to observe the gain brought by the in-context learning ability of LLMs on their implementation of table-based fact verification.

A. Zero-shot Learning of LLMs

To leverage these generative models, we need to design elaborate prompts for TFV to stimulate the verification ability of LLMs. Under the zero-shot setting, there is nothing but the TFV task description to provide for LLMs. We explore the following zero-shot prompts for TFV.

1. Sentence. Prompts with one-turn dialogue are the simplest ways to implement ChatGPT on NLP tasks. It just needs to input one sequence and collect the response as the result. In terms of TFV, we can simply construct a long sentence with the task description, the linearized table, and the statement to input into ChatGPT, forming the **Sentence** prompt. The form of the Sentence prompt is “*Give you a statement and a table, please tell me whether the statement is supported or refuted by the table. The table is [LINEARIZED TABLE]. The statement is [STATEMENT].*”.

2. Sentence+word. Considering zero-shot LLMs may generate responses with uncertain length or undesired label words, we can input an extra description to constrain the format of the output. Inspired by [15], we append an additional word-guided sentence “*Just answer only one word “supported” or “refuted” without other tokens.*” after Sentence prompt, named **Sentence+word**. With word-guided sentences, it’s hoped to extract the verdict label more conveniently.

3. Paragraph. Additionally, inspired by [26], we ask the web version of ChatGPT(ChatGPT-web) to provide TFV prompts. From the responses, we choose an accessible and related prompt named **Paragraph** prompt as it consists of several paragraphs. The Paragraph prompt is shown as Fig.2.

```
Statement: [STATEMENT]
Table: [LINEARIZED TABLE]
Fact-verification: Based on the information provided in the table,
is the statement supported or refuted?
```

Fig. 2. Paragraph prompt

4. Dialogue. A more interactive way to use dialogue models is multi-turn dialogue, which inputs all dialogue history and extracts results from the response of the last turn. For TFV, we can exploit a multi-turn dialogue to input task description, table, and statement separately. To simulate real responses, we input requests into ChatGPT-web and extract some responses. Here we will use two kinds of responses generated by ChatGPT-web to fill the prompt, named **Dialogue-simple** and **Dialogue-complex**. The Dialogue-simple prompt

```
Q1: Give you a statement and a table, please tell me the statement is
supported or refuted by the table.
```

```
R1: Please provide me with the statement and the table you are referring to.
```

```
Q2: The table is [LINEARIZED TABLE].
```

```
R2: Please provide me with the statement that you would like me to verify
using the provided table.
```

```
Q3: The statement is [STATEMENT].
```

Fig. 3. Dialogue-simple prompt

```
<s>[INST] <<SYS>>
{system_message}
<</SYS>>
```

```
{history_message_1} [/INST] {history_reply_1}</s><s>[INST]
{user_message} [/INST]
```

Fig. 4. Recommended Instruction prompt of LLaMA-2

is demonstrated as Fig.3. For the Dialogue-complex prompt, the questions are the same as the Dialogue-simple, while the responses R1 and R2 are replaced with more courteous and indicative sentences. In the Dialogue-complex prompt, R1 is “*I’ll do my best to help you with that! Please provide me with the statement and the table, and I’ll let you know if the statement is supported or refuted by the information presented in the table.*”, and R2 is “*Thank you for providing the table. Please provide me with the statement that you want me to check against the table, and I’ll let you know if it’s supported or refuted by the information in the table.*”.

It is worth noticing how to input the dialogue history to LLMs. For ChatGPT, it’s convenient to add dialogue history to API by switching the role to “user” for history input or “assistant” for history output. For LLaMA, we use the recommended dialogue instruction prompt ² (as shown in Figure 4) to imitate a dialogue environment, which is also employed in the fine-tuning procedure of LLaMA-2-chat, a dialogue adaptive version of LLaMA-2. Therefore, we select LLaMA-2-chat and ChatGPT to perform the zero-shot TFV experiment. The accuracy of different methods under zero-shot setting on TabFact-small-test is shown in Table I.

The results show that ChatGPT demonstrates an acceptable performance under the zero-shot setting, which is comparable to early baseline methods. However, the accuracy of LLaMA-2-chat with all prompts is slightly over 0.5, the result of random guessing for binary classification tasks. We analyze that compared with LLaMA (7B), ChatGPT has more parameters and has seen more tabular data within pre-training. Zero-shot LLaMA-2-chat can generate seemingly reasonable responses, but it lacks reasoning ability on tables.

For concrete prompts, there are some interesting findings as follows. First, the Dialogue-complex prompt shows a superior performance among these prompts, which can be attributed to the greater number of interactions they facilitate. Second, the

²<https://huggingface.co/blog/llama2#how-to-prompt-llama-2>

TABLE I
ZERO-SHOT EXPERIMENTAL RESULTS

Model	Prompt	Accuracy(%)
LLaMA-2-chat	Sentence	53.1
	Sentence+word	50.0
	Paragraph	55.5
	Dialogue-simple	51.5
	Dialogue-complex	55.6
ChatGPT	Sentence	73.6
	Sentence+word	69.7
	Paragraph	57.8
	Dialogue-simple	74.2
	Dialogue-complex	74.9
Random Guessing	-	50.0
Human	-	92.1

Sentence+word prompt under-performs the Sentence prompt with both LLaMA and ChatGPT. This is mainly because word-guided prompt discourages LLMs from thinking deeply. Third, we notice the Paragraph prompt with ChatGPT reaches an unexpectedly low accuracy. One possible reason is that the Paragraph prompt deviates from a smooth and natural expression of dialogue, thereby exerting a greater influence on ChatGPT for zero-shot learning.

B. Few-shot Learning of LLMs

Besides task description, a few in-context TFV examples including inputs and labels are provided under the few-shot setting. Different from previous few-shot learning methods for small models, LLMs can imitate the response pattern from a small amount of in-context examples without updating model parameters. Let us assess the in-context learning ability of LLMs on TFV.

We adapt a dialogue-style in-context prompt for few-shot experiments as Fig.5 shows. For the N-shot condition, the number of dialogue turns can be expanded by introducing more question-response pairs after R1, followed by the example that needs to be predicted at the end.

Q1: Give you a statement and a table, please tell me the statement is supported or refuted by the table.
R1: Sure, I can help you with that. Please provide me with the statement and the table.
Q2: The table is [LINEARIZED TABLE 1]. The statement is [STATEMENT 1].
R2: [ANSWER 1].
Q3: The table is [LINEARIZED TABLE]. The statement is [STATEMENT].

Fig. 5. Dialogue-style in-context prompt (1-shot)

We choose LLaMA-2, LLaMA-2-chat, and ChatGPT for our few-shot experiments. Same as zero-shot experiments, LLaMA-dialogue instruction prompts (Fig.4) are used for LLaMA-2 and LLaMA-2-chat. Besides, in-context examples are handpicked from the training set and remain consistent across various LLMs with the same shot number.

Table II shows the accuracy of different methods under the few-shot setting on TabFact-small-test. Compared with the zero-shot setting, ChatGPT gains better accuracy with the

assistance of in-context examples. The victory over Codex demonstrates the superior in-context learning capability of ChatGPT.

However, the LLaMA models still perform poorly with few in-context samples. The accuracy attained by LLaMA-2-chat hovers slightly above 0.5, showing no significant improvement compared with the zero-shot performance. In some cases, LLaMA-2 even performs worse than random guessing. This indicates that LLaMA models with 7B parameters are inadequate in learning sufficient reasoning ability from TFV in-context examples.

Additionally, we observe a decline in accuracy as the number of in-context examples increases from 2 to 4. A possible reason is the excessive length of the in-context examples. Due to the limited contextual memory capability of ChatGPT and LLaMA, an excess of inputs may bring a burden to inference.

TABLE II
FEW-SHOT EXPERIMENTAL RESULTS

Method	Accuracy(%)
LLaMA-2(1-shot)	50.9
LLaMA-2(2-shot)	49.4
LLaMA-2(4-shot)	49.2
LLaMA-2-chat(1-shot)	52.9
LLaMA-2-chat(2-shot)	53.9
LLaMA-2-chat(4-shot)	51.0
ChatGPT(1-shot)	74.2
ChatGPT(2-shot)	75.7
ChatGPT(4-shot)	75.2
Codex(4-shot) [21]	72.6
Random Guessing	50.0
Human	92.1

In this paper, we mainly focus on the original capability of LLMs for TFV, directly using them as table-based fact-checkers. Therefore, the methods [20], [21] using LLMs as additional modules to assist traditional table models are out of the scope of this paper.

V. INSTRUCTION TUNING OF LLMs

In the above in-context learning setting, the LLMs are not trained, where the focus is on the design of prompts. On the contrary, this section discusses the gain brought by model training and parameter updating (i.e., instruction tuning) of LLMs in TFV under the fine-tuning setting.

For instruction tuning, we should first construct TFV instruction data, each of which is composed of instruction, input and output. Followed by Stanford Alpaca [18], we prepare a TFV instruction and fill the table and statement in the instruction-following format shown in Fig.6. Then LLMs are fine-tuned with next-word prediction object to predict the verdict words (e.g. supported) in a supervised manner. At inference stage, we extract the generated content and map it into expected label set. Additionally, considering that tables are long context, we train LLaMA with PEFT method LORA [23] instead of full fine-tuning to reduce training cost, which learns pairs of rank-decomposition matrices with other original weights frozen.

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:

Give you a statement and a table, please determine whether the statement is supported or refuted by, or is unrelated to the table. It's a binary classification problem, you should choose the answer between supported and refuted.

Input:

Table: [TABLE]

Statement: [STATEMENT]

Response:

Fig. 6. Alpaca instruction-following prompt for TFV

We choose LLaMA-1 and LLaMA-2 to perform instruction tuning on the train set of TabFact with 92283 samples. The models are fine-tuned on two A100 GPUs. We compare our instruction tuning methods with existing well-trained LLM-free baselines. The results of different fine-tuning methods on TabFact-small-test are shown in Table III.

TABLE III
EXPERIMENTAL RESULTS OF FINE-TUNING METHODS

Type	Method	Accuracy(%)
LLM-free	Table-BERT [2]	68.1
	LPA [2]	68.9
	LFC (LPA-based) [4]	74.2
	ProgVGAT [5]	76.2
	BART (large) [8]	82.5
	Tapas (large) [7]	83.9
	Tapex (large) [8]	85.9
LLM-based	PASTA [11]	90.6
	LLaMA-1	79.5
Human	LLaMA-2	82.3
	-	92.1

Results show that fine-tuned LLaMA models gain a significant improvement. They outperform certain program-based methods and encoder-only general PLMs but fall short compared to small-scaled table-based PLMs. This discrepancy may be attributed to LLaMA models not having exposure to billions of tabular data.

Moreover, we observe that LLaMA-2 is slightly inferior to BART, a general encoder-decoder PLM with only 0.44 billion parameters. It can be interpreted from two aspects. On the one hand, the LoRA strategy we use only allows a few models parameters to update during fine-tuning, which may prevent to realize the full potential of LLMs. On the other hand, some small language models may be effective enough to perform on individual dataset, while LLMs will exploit their dominance under cross-domain and out-of-domain settings.

VI. ANALYSIS AND DISCUSSION

A. Case Study

Fig.7 upper part reports the qualitative comparison of models under different settings to the same question, which shows

Qualitative Comparison	
Statement:	The only player not from the United States or Canada is from Norway
Ground Truth: Refuted	
<i>In-context Learning:</i>	
① Zero-shot(ChatGPT):	The statement is refuted by the table as there is no player from Norway listed in the table. The only non-North American player listed is Ladislav Scurko from Slovakia.
② Zero-shot(LLaMA2):	Sure, I'd be happy to help! Based on the table provided, the statement "the only player not from the United States or Canada is from Norway" is REFUTED . The table shows that Ladislav Scurko, a center from Slovakia, is not from the United States or Canada. Therefore, the statement is incorrect.
③ Few-shot(ChatGPT):	Refuted . There is no player from Norway in the table. All the players listed are either from the United States or Canada, with one player, Ladislav Scurko, from Slovakia. Therefore, the statement "the only player not from the United States or Canada is from Norway" is refuted by the table.
④ Few-shot(LLaMA2):	Supported . The only player in the table who is not from the United States or Canada is "Ladislav Scurko" from Slovakia, as shown in row 6.
<i>Instruction Timing:</i>	
⑤ LLaMA2:	refuted
Failure Cases	
Statement:	The number of players from Canada outnumber the number of players from the United States
Ground Truth: Refuted	
<i>In-context Learning:</i>	
① Few-shot(ChatGPT):	Supported . According to the table, seven players (laliberte, scurko, cabana, houle, gawryletz, grant, and carter) are from Canada while four players (Bellamy, Anderson, Zarb, and Pisellini) are from the United States. Therefore, the statement "the number of players from Canada outnumber the number of players from the United States" is supported by the table.
② LLaMA2:	supported

Fig. 7. Some real responses for two cases towards the table in Fig.1. We show the responses of best-perform prompt (with two in-context examples) and LLaMA-2 version (LLaMA-2-chat) under the few-shot setting.

that: 1) From zero-shot to few-shot, and then to instruction-tuning settings, the target prediction label (e.g., refuted) gradually move forward in the entire output (from the middle to the beginning), and the generated content of LLaMA2 is more concise and specific, making the answers easier to extract. 2) The response of LLaMA2 under the few-shot setting generating correct explanations but a wrong label, which shows the weak reasoning ability on TFV of LLaMA2 without tuning. 3) Although the accuracy of LLaMA2 is clearly improved after instruction tuning, and the responses are more in line with the format set by TFV task, relevant explanations are no longer generated after instruction tuning due to the lack of explanations annotation during model training. We also showcase the failure examples in Fig.7 lower part. There are actually five players from each of the two countries, however, both the best in-context learning and instruction-tuning results fail to generate the correct labels. This shows that LLMs may suffer severely from the hallucination for complex (e.g., counting and reasoning) questions.

B. Discussion

From the above, we notice that even fine-tuned LLMs still lag behind with some LLM-free methods towards classification accuracy on the TFV task. We would like to discuss some possible research directions in future work to improve the performance on TFV by means of advanced LLMs.

Handling Long Input. The improvement on TFV for LLaMA-2 compared to LLaMA-1 can be primarily attributed to the extension of context windows. It's a universal problem for all tabular tasks to handle excessively long tables. One direct approach is using LLMs with robust extrapolation ability, but it appears to overlook the characteristic of table structure. Besides, a more efficient way is to decompose tables into sub-tables related to their corresponding statements. Datar

[21] first proves the efficacy of LLM-based table decomposers, and more decomposing strategies need to be further explored. **Specifying Inference Procedures.** Followed by the thought of Chain-of-thought (CoT) [27], developing detailed inference procedures in stages, rather than focusing solely on labeling individual words, has potentials to enhance the performance. For instance, it's effective for LLMs to decompose the complex sentences into simpler ones [21], which benefits both the accuracy and interpretability for TFV task. We have conducted some preliminary works about LLMs's ability on TFV, but we suffer from the lack of annotated CoT datasets about tables. Further explorations on CoT of TFV are reserved for the follow-up studies.

Developing table-based LLMs. Inspired by the outstanding performance of small-scaled table-based PLMs, it's promising to fine-tune LLMs with intermediate tables reasoning tasks. To develop table-based LLMs, it's necessary to construct instruction datasets of various table types or formats, such hierarchy tables or HTML-format tables. We infer that TFV under cross-domain and out-of-domain settings will benefit more from table-based LLMs.

VII. CONCLUSION

In this study, we motivate LLMs to perform the table-based fact verification task directly under in-context learning and instruction-funeting settings. Experimental results show that LLMs can be qualified for table-based fact checkers. Larger and intelligent LLMs like ChatGPT can achieve a acceptable result on TFV with elaborate prompts under zero-shot and few-shot setting, while LLMs with small parameters like LLaMA can't. However, the result of LLaMA can be promoted significantly by instruction fine-tuning, though they still lag behind with the most advanced task-specific small-scaled models on TFV. In addition, we also conclude three impressive research directions for the follow-ups.

REFERENCES

- [1] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, “FEVER: a large-scale dataset for fact extraction and verification,” in *NAACL-HLT*, M. A. Walker, H. Ji, and A. Stent, Eds., 2018.
- [2] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang, “Tabfact: A large-scale dataset for table-based fact verification,” in *ICLR*, 2020.
- [3] V. Gupta, M. Mehta, P. Nokhiz, and V. Srikumar, “INFOTABS: inference on tables as semi-structured data,” in *ACL*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020.
- [4] W. Zhong, D. Tang, Z. Feng, N. Duan, M. Zhou, M. Gong, L. Shou, D. Jiang, J. Wang, and J. Yin, “Logicalfactchecker: Leveraging logical operations for fact checking with graph module network,” in *ACL*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020.
- [5] X. Yang, F. Nie, Y. Feng, Q. Liu, Z. Chen, and X. Zhu, “Program enhanced fact verification with verbalization and graph attention network,” in *EMNLP*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., 2020.
- [6] Q. Shi, Y. Zhang, Q. Yin, and T. Liu, “Logic-level evidence retrieval and graph-based verification network for table-based fact verification,” in *EMNLP*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds., 2021.
- [7] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. M. Eisenschlos, “Tapas: Weakly supervised table parsing via pre-training,” in *ACL*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020.
- [8] Q. Liu, B. Chen, J. Guo, M. Ziyadi, Z. Lin, W. Chen, and J. Lou, “TAPEX: table pre-training via learning a neural SQL executor,” in *ICLR*, 2022.
- [9] J. Yang, A. Gupta, S. Upadhyay, L. He, R. Goel, and S. Paul, “Tableformer: Robust transformer modeling for table-text encoding,” in *ACL*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., 2022.
- [10] Y. Zhou, X. Liu, K. Zhou, and J. Wu, “Table-based fact verification with self-adaptive mixture of experts,” in *ACL*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., 2022.
- [11] Z. Gu, J. Fan, N. Tang, P. Nakov, X. Zhao, and X. Du, “PASTA: Table-operations aware fact verification via sentence-table cloze pre-training,” in *EMNLP*, Abu Dhabi, United Arab Emirates, Dec. 2022.
- [12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.
- [13] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, “Evaluating large language models trained on code,” *CoRR*, vol. abs/2107.03374, 2021.
- [14] B. Zhang, D. Ding, and L. Jing, “How would stance detection techniques evolve after the launch of chatgpt?” *CoRR*, vol. abs/2212.14548, 2022.
- [15] J. Wang, Y. Liang, F. Meng, Z. Li, J. Qu, and J. Zhou, “Cross-lingual summarization via chatgpt,” *CoRR*, vol. abs/2302.14229, 2023.
- [16] A. Hendy, M. Abdelrehim, A. Sharaf, V. Raunak, M. Gabr, H. Matushita, Y. J. Kim, M. Afify, and H. H. Awadalla, “How good are GPT models at machine translation? A comprehensive evaluation,” *CoRR*, vol. abs/2302.09210, 2023.
- [17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambo, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [18] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Stanford alpaca: An instruction-following llama model,” https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [19] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, “Training language models to follow instructions with human feedback,” in *NeurIPS*, 2022.
- [20] Z. Cheng, T. Xie, P. Shi, C. Li, R. Nadkarni, Y. Hu, C. Xiong, D. Radev, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu, “Binding language models in symbolic languages,” *CoRR*, vol. abs/2210.02875, 2022.
- [21] Y. Ye, B. Hui, M. Yang, B. Li, F. Huang, and Y. Li, “Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning,” in *SIGIR*, H. Chen, W. E. Duh, H. Huang, M. P. Kato, J. Mothe, and B. Poblete, Eds., 2023.
- [22] Q. Si, T. Wang, Z. Lin, X. Zhang, Y. Cao, and W. Wang, “An empirical study of instruction-tuning large language models in chinese,” 2023.
- [23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.
- [24] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, “GPT understands, too,” *CoRR*, vol. abs/2103.10385, 2021.
- [25] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [26] W. Jiao, W. Wang, J. Huang, X. Wang, and Z. Tu, “Is chatgpt A good translator? A preliminary study,” *CoRR*, vol. abs/2301.08745, 2023.
- [27] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *NeurIPS*, 2022.