TableLlama: Towards Open Large Generalist Models for Tables

Anonymous ACL submission

Abstract

Semi-structured tables are ubiquitous. There has been a variety of tasks that aim to automatically interpret, augment, and query tables. Current methods often require pretraining on tables or special model architecture design, are restricted to specific table types, or have simplifying assumptions about tables and tasks. This paper makes the first step towards developing open-source large language models (LLMs) as generalists for a diversity of table-based tasks. Towards that end, we construct `TableInstruct`, a new dataset with a variety of realistic tables and tasks, for instruction tuning and evaluating LLMs. We further develop the first open-source generalist model for tables, `TableLlama`, by fine-tuning Llama 2 (7B) with LongLoRA to address the long context challenge. We experiment under both in-domain setting and out-of-domain setting. On 7 out of 8 in-domain tasks, `TableLlama` achieves comparable or better performance than the SOTA for each task, despite the latter often has task-specific design. On 6 out-of-domain datasets, it achieves 6-48 absolute point gains compared with the base model, showing that training on `TableInstruct` enhances the model's generalizability. We will open source our dataset and trained model to boost future work on developing open generalist models for tables.[1]

1 Introduction

Semi-structured tables are prevalent data structures to store and present information in almost every domain, ranging from scientific research, business reports, and healthcare records to financial statements. A variety of table-based tasks have been proposed, such as entity linking (Ritze et al., 2015), schema augmentation (Zhang and Balog, 2017), and table-based question answering (Cheng et al., 2022b; Nan et al., 2022; Chen et al., 2020b), which have spurred significant research interest (Deng et al., 2020; Yin et al., 2020; Wang et al., 2021; Iida et al., 2021) in recent years.

Most existing methods for table-based tasks have at least one of the following limitations: (1) Require table pretraining (Liu et al., 2022; Yin et al., 2020; Deng et al., 2020; Iida et al., 2021) and/or special model architecture design for tables (Deng et al., 2020; Wang et al., 2021; Iida et al., 2021), (2) only support limited, specific types of tables and tasks (Chen et al., 2020a; Nan et al., 2022), (3) make strong simplifying assumptions (See the "in-domain" part of Section 2.1) about tables and tasks (Li et al., 2023b).

On the other hand, language models like T5 (Raffel et al., 2020) have been shown to excel in grounding language to structured knowledge (Xie et al., 2022). In addition, instruction tuning (Chung et al., 2022; Wang et al., 2022; Mishra et al., 2022) appears as an important technique that can guide LLMs to follow instructions to complete a variety of table-based tasks.

Under this background, we seek to answer the following question: *Can we build a generalist model to handle a variety of table-based tasks using LLMs and instruction tuning?* Some exemplar tasks are shown in Figure 1. Such a generalist model shall meet the following desiderata: First, **it should not only work well on diverse table-based tasks, but also generalize to unseen tasks.** Since new table data and tasks can be constructed dynamically as new information arrives, it is hard to collect training data that covers all tasks and all tables, which requires a model to be inherently generalizable to tasks and datasets it has never seen before. Second, **it should work on real-world tables and realistic tasks.** The model should not make strong assumptions to only handle simplified synthetic tables and tasks, but must embrace practical challenges such as handling complex numerical reasoning on large hierarchical spreadsheets as well as a large number of candidates for classification

[1] Code, model and data will be available.

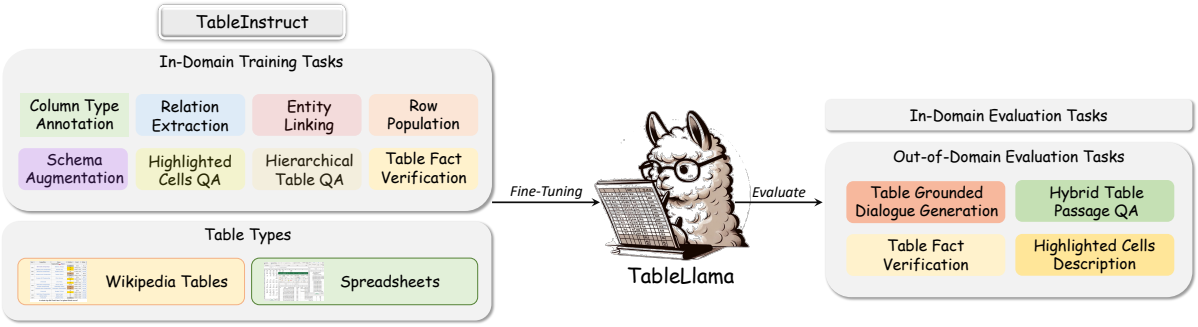

Figure 1: An overview of `TableInstruct` and `TableLlama`. `TableInstruct` includes a wide variety of realistic tables and tasks with instructions. We make the first step towards developing open-source generalist models for tables with `TableInstruct` and `TableLlama`.

and ranking tasks.

In pursuing this goal, we realize there lacks a comprehensive collection of realistic tables and tasks that can support the development and evaluation of generalist models. Therefore, we construct `TableInstruct`, by meticulously selecting representative table-based tasks from widely used datasets, unifying the format for all tasks and manually annotating instructions. `TableInstruct` shown in Table 1 offers the following unique features: (1) **Diverse coverage of tables and tasks**. `TableInstruct` boasts a collection of 14 datasets of 11 tasks in total, with both in-domain and out-of-domain evaluation settings. Our training data includes 8 tasks, which are curated from 1.24M tables containing 2.6M instances spanning from table interpretation, table augmentation, table-based QA, and table-based fact verification. We choose 8 datasets for these 8 tasks for in-domain evaluation and leave the other 6 datasets for 4 tasks for out-of-domain evaluation. The in-domain training tasks can enable the model to learn more fundamental table understanding abilities such as table interpretation and table augmentation, while we choose tasks that require more high-level reasoning abilities such as table QA and cell description to test the model's generalization ability. This extensive range of tables and diverse tasks not only provide valuable resources for table modeling, but also foster a more comprehensive evaluation of generalist models. (2) **The use of real-world tables and realistic tasks**. `TableInstruct` uses authentic real-world instead of overly simplified synthetic task data compared with existing work (Li et al., 2023b). We incorporate a large number of Wikipedia tables and spreadsheets from statistical scientific reports with varied length of contents, realistic and complex semantic types from Freebase (Google.2015) for column type annotation and relation extraction, and a large referent entity corpus with rich metadata from Wikidata (Vrandečić and Krötzsch, 2014) for entity linking. In addition, we include complicated numerical reasoning tasks with hierarchical table structure and existing manually annotated table QA and fact verification tasks. By doing so, we aim to equip models with the capability to cope with realistic and complex table-based tasks.

`TableInstruct` requires models to accommodate long inputs (Table 1). We adopt LongLoRA (Chen et al., 2023b) based on Llama 2 (7B) (Touvron et al., 2023) as our backbone model, which has been shown efficient and effective to handle long contexts. We fine-tune it on `TableInstruct` and name our model `TableLlama`. We conducted extensive experiments under both in-domain and out-of-domain settings. Our experiments show `TableLlama` has strong capabilities for various in-domain table understanding and augmentation tasks, and also achieves promising performance in generalizing to unseen tasks and datasets.

In summary, our main contributions are:

- We construct `TableInstruct`, a large-scale instruction tuning dataset with diverse, realistic tasks based on real-world tables. We unify their format and manually annotate instructions to guarantee quality.

- We develop `TableLlama`, an open-source LLM-based generalist model fine-tuned on `TableInstruct`. Experiments show that compared with the SOTA on each task that often has special pre-training or model architecture design for tables, `TableLlama` can achieve similar or even better performance on

Figure 2: Illustration of three exemplary tasks: (a) Column type annotation. This task is to annotate the selected column with the correct semantic types. (b) Row population. This task is to populate rows given table metadata and partial row entities. (c) Hierarchical table QA. For subfigures (a) and (b), we mark candidates with red color in the "task instruction" part. The candidate set size can be hundreds to thousands in `TableInstruct`.

almost all of the in-domain tasks. For out-of-domain tasks, compared with the base model, `TableLlama` can achieve 6-48 absolute point gains on 6 datasets, and compared with GPT-4, `TableLlama` has less gap or even better zero-shot performance on 4 out of 6 datasets, which demonstrate that `TableInstruct` can substantially enhance model generalizability.

2 `TableInstruct` Benchmark

Unlike existing datasets predominantly designed for training task-specific table models, our objective is to bridge the gap between multiple complex task-specific models and one simple generalist model that can deal with all the table-based tasks without extra model-design efforts. To achieve this, our approach for constructing `TableInstruct` adheres to the following principles. First, instead of collecting multiple datasets from highly homogeneous tasks, we try to diversify the tasks and table types. We pick representative table-based tasks that necessitate different abilities of models, such as table interpretation, table augmentation, table QA and table fact verification from Wikipedia tables and spreadsheets in statistical scientific reports. Second, we select realistic tasks and construct high-quality instruction data in a unified fashion without simplifying assumptions (see "in-domain" part of 2.1). `TableInstruct` will support powerful modeling and realistic evaluation approaches, ensuring a valuable and practical dataset for research.

2.1 Data Collection

`TableInstruct` incorporates samples from 14 table-based datasets of 11 distinctive tasks (Table 1). We separate them and select 8 datasets of 8 tasks for training and in-domain evaluation. We leave the other 6 datasets of 4 tasks as held-out unseen datasets for out-of-domain evaluation.

Task category: Tasks in `TableInstruct` can be categorized into several groups: table interpretation, table augmentation, question answering, fact verification, dialogue generation, and data-to-text. Table interpretation aims to uncover the semantic attributes of the data contained in relational tables, and transform this information into machine understandable knowledge. Table augmentation is to expand the partial tables with additional data. Question answering aims to obtain the answer with tables and optional highlighted cells or passages as evidence. Fact verification is to discriminate whether the tables can support or refute the claims. Dialogue generation is to generate a response grounded on the table and dialogue history. Data-to-text is to generate a description based on the highlighted cells. By choosing the tasks that

Task Category	Task Name	Dataset	In-domain	#Train (Table/Sample)	#Test (Table/Sample)	Input Token Length min	max	median
Table Interpretation	Col Type Annot.	TURL (Deng et al., 2020)	Yes	397K/628K	1K/2K	106	8192	2613
	Relation Extract.		Yes	53K/63K	1K/2K	2602	8192	3219
	Entity Linking		Yes	193K/1264K	1K/2K	299	8192	4667
Table Augmentation	Schema Aug.	TURL (Deng et al., 2020)	Yes	288K/288K	4K/4K	160	1188	215
	Row Pop.		Yes	286K/286K	0.3K/0.3K	264	8192	1508
Question Answering	Hierarchical Table QA	HiTab (Cheng et al., 2022b)	Yes	3K/7K	1K/1K	206	5616	978
	Highlighted Cells QA	FeTaQA (Nan et al., 2022)	Yes	7K/7K	2K/2K	261	5923	740
	Hybrid Table QA	HybridQA (Chen et al., 2020b)	No	–	3K/3K	248	2497	675
	Table QA	WikiSQL (Zhong et al., 2017)	No	–	5K/16K	198	2091	575
	Table QA	WikiTQ (Pasupat and Liang, 2015)	No	–	0.4K/4K	263	2688	709
Fact Verification	Fact Verification	TabFact (Chen et al., 2020a)	Yes	16K/92K	2K/12K	253	4975	630
		FEVEROUS (Aly et al., 2021)	No	–	4K/7K	247	8192	648
Dialogue Generation	Table Grounded Dialogue Generation	KVRET (Eric et al., 2017)	No	–	0.3K/0.8K	187	1103	527
Data-to-Text	Highlighted Cells Description	ToTTo (Parikh et al., 2020)	No	–	7K/8K	152	8192	246

Table 1: Statistics of train/test tasks and datasets in our `TableInstruct`. For each task, we explain its definition and show an example in Appendix D.

require models to learn more fundamental table understanding abilities such as table interpretation and table augmentation for training, we hope the model can demonstrate generalization ability on out-of-domain datasets such as high-level table QA and table cell description tasks.

In-domain: The tasks for training the generalist table model include column type annotation, relation extraction, entity linking, row population, schema augmentation, hierarchical table QA, highlighted cells QA, and table fact verification. These tasks require the model to understand the semantics of table columns, the relation between table column pairs, the semantics of table cells and require the model to gain reasoning ability to answer table-related questions and verify the facts. For the dataset of each task, we intentionally pick up those that enjoy realistic task complexity without simplifying assumptions. For example, for column type annotation and relation extraction, these two tasks are multi-choice classification tasks in essence. We use real-world column semantic types and relation types from Freebase (Google.2015), which contains hundreds of complex choices such as "government.politician.party-government.political_party_tenure.party" shown in Figure 4 in Appendix D. For entity linking, the referent entities are from real-world Wikidata (Vrandečić and Krötzsch, 2014), which contains hundreds of complex metadata, such as "<2011-12 Melbourne Victory season [DESCRIPTION] Association football club 2011/12 season for Melbourne Victory [TYPE] SoccerClubSeason>" as shown in Figure 5 in Appendix D. For schema augmentation and row population, there are a huge number of candidates that LLMs need to rank. For hierarchical table QA, all the tables are engaged with intricate structures with multi-level column names and row names. In addition, it is intensive in numerical reasoning which requires LLMs to understand table structure, identify related cells and do calculations. By doing so, we hope to enable LLMs to become truly powerful generalist models that can handle sophisticated table tasks and `TableInstruct` can be a realistic benchmark to evaluate LLMs' abilities compared with specially designed table models.

Out-of-domain: A powerful generalist table model is expected to not only demonstrate strong performance on in-domain tasks, but also generalize well to unseen tasks or unseen datasets of the same tasks. We choose tasks such as table QA and cell description that require the model's high-level table understanding and reasoning ability as out-of-domain datasets. We involve HybridQA (Chen et al., 2020b), KVRET (Eric et al., 2017), FEVEROUS (Aly et al., 2021), ToTTo (Parikh et al., 2020), WikiSQL (Zhong et al., 2017) and WikiTQ (Pasupat and Liang, 2015) as 6 out-of-domain datasets to test our model's generalization ability.

2.2 Task Formulation and Challenges

The primary objective of `TableInstruct` is to design one generalist model for all table-based tasks. As Figure 2 (a)-(c) shows, each instance in our dataset maps three components: <instruction, table input, question> to an output. The instruction is manually designed to point out the task and give a detailed task description. We concatenate table metadata such as the Wikipedia page title, section title and table caption with the serialized table as

table input. In the question, we put all the information the model needed to complete the task and prompt the model to generate an answer. For example, for the column type annotation task, as Figure 2 (a) shows, the column named "Player" needs to be annotated with its semantic types. In the format, the "instruction" gives the description of the task. The "input" contains the table-related information. Then we provide the entire candidate pool in the "question" and ask the model to choose one or multiple correct semantic types for this column.

Challenges. Since we select realistic tasks and tables, the table length can vary from several to thousands of rows. Besides, for some tasks that are essentially multi-choice classification or ranking, the entire candidate pool can be very large up to thousands. Furthermore, as the candidates are from real-world Freebase and Wikidata, each candidate is long, such as "<2011-12 Melbourne Victory season [DESCRIPTION] Association football club 2011/12 season for Melbourne Victory [TYPE] SoccerClubSeason>" is one candidate for entity linking. These characteristics can not only make it difficult for the model to learn, but also introduce the challenge of handling long contexts.

3 Experimental Setup

Model Construction. Although a few existing LLMs (Chen et al., 2023a; Tworkowski et al., 2023) can handle longer than 4K contexts, their training time is quadratically increasing with context length, which becomes very costly for us to further fine-tune them on `TableInstruct` due to our large data scale. As LongLoRA (Chen et al., 2023b) has been shown as an effective and efficient technique to train long-context LLMs with shift short attention, we adopt it as our backbone model. Shift short attention splits context length into several groups and conducts attention in each group individually. The tokens are shifted by half group size in half attention heads to ensure the information flow between neighboring groups. For example, LongLoRA can use shift short attention with group size 2048 to approximate total 8196 context length training, which leads to less computation cost with similar performance compared to fine-tuning with vanilla attention. We fine-tune LongLoRA on `TableInstruct` to get our generalist model `TableLlama`.

Existing SOTA Models. In our evaluation settings, we have 10 out of 14 SOTA models utilize table pretraining and/or have special model architecture design for tables. The detailed description for each SOTA model is in Appendix B.

Evaluation Metrics. We follow the above baselines to use their evaluation metrics. For column type annotation, relation extraction and KVRET, we use Micro F1. For entity linking, TabFact, FEVEROUS, HybridQA, WikiSQL and WikiTQ, we use accuracy. For row population and schema augmentation, we use MAP. For Hitab, we use execution accuracy (Zhong et al., 2017). For FeTaQA and ToTTo, we use BLEU (Papineni et al., 2002).

Training and Inference Details. We choose LongLoRA 7B, fully fine-tuning version with 8K context length limit as our base model. More training and inference details are in Appendix A.

4 Result Analysis

4.1 Main Results

In-domain Results. As Table 2 shows, we train `TableLlama` on eight table-based tasks and evaluate it on their test sets as the in-domain results. Due to the special semi-structured nature of tables, for most table-based tasks, existing work achieves SOTA results by using pretraining on large-scale tables and/or special model architecture design tailored for tables. Nonetheless, we observe that:

By simply fine-tuning a large language model on `TableInstruct`, `TableLlama` *can achieve comparable or even better performance on almost all the tasks without any table pretraining or special table model architecture design.* For most of the tasks, the performance gap is within 3 absolute points, except for row population. For entity linking, schema augmentation, HiTab and FeTaQA, `TableLlama` can exceed the SOTA performance by up to 17.71 absolute points. This demonstrates that empowering open-source LLMs with more powerful table understanding abilities via instruction tuning can be a promising research direction to further explore.

`TableLlama` *displays advantanges in table QA tasks.* HiTab and FeTaQA are two table question answering tasks we include for training. By comparing the results, we found that `TableLlama` can surpass the SOTA by 5.61 points for FeTaQA and 17.71 points for HiTab, which is full of numerical reasoning on tables. As LLMs have been shown superior in interacting with humans and answering questions, this indicates that the existing underlying strong language understanding ability of LLMs may be beneficial for such table QA tasks despite with semi-structured tables.

In-domain Evaluation						
Datasets	Metric	Base	TableLlama	SOTA	GPT-3.5	GPT-4§
Column Type Annotation	F1	3.01	94.39	**94.54***† (Deng et al., 2020)	30.88	31.75
Relation Extraction	F1	0.96	91.95	**94.91***† (Deng et al., 2020)	27.42	52.95
Entity Linking	Accuracy	31.80	**93.65**	84.90*† (Deng et al., 2020)	72.15	90.80
Schema Augmentation	MAP	36.75	**80.50**	77.55*† (Deng et al., 2020)	49.11	58.19
Row Population	MAP	4.53	58.44	**73.31***† (Deng et al., 2020)	22.36	53.40
HiTab	Exec Acc	14.96	**64.71**	47.00*† (Cheng et al., 2022a)	43.62	48.40
FeTaQA	BLEU	8.54	**39.05**	33.44 (Xie et al., 2022)	26.49	21.70
TabFact	Accuracy	41.65	82.55	**84.87*** (Zhao and Yang, 2022)	67.41	74.40

Table 2: In-domain evaluation results. "Base": LongLoRA model w/o fine-tuning on `TableInstruct`; "*": w/ special model architecture design for tables/tasks; "†": w/ table pretraining; "§": for GPT-4, we uniformly sample 500 examples from test set for each task due to limited budget.

Out-of-domain Evaluation							
Datasets	Metric	Base	TableLlama	SOTA	Δ_{Base}	GPT-3.5	GPT-4§
FEVEROUS	Accuracy	23.66	72.30	82.40 (Xie et al., 2022)	+48.64	60.79	71.60
HybridQA	Accuracy	20.72	27.61	63.40* (Eisenschlos et al., 2021)	+6.89	40.22	58.60
KVRET	Micro F1	38.90	48.73	67.80 (Xie et al., 2022)	+9.83	54.56	56.46
ToTTo	BLEU	10.39	20.77	48.95 (Xie et al., 2022)	+10.38	16.81	12.21
WikiSQL	Accuracy	14.84	41.68	89.50† (Liu et al., 2022)	+26.84	41.91	47.60
WikiTQ	Accuracy	25.48	31.63	57.50† (Liu et al., 2022)	+6.15	53.13	68.40

Table 3: Out-of-domain evaluation results. "Base": LongLoRA model w/o fine-tuning on `TableInstruct`; "*": w/ special model architecture design for tables/tasks; "†": w/ table pretraining; "§": for GPT-4, we uniformly sample 500 examples from test set for each task due to limited budget. We put the SOTA performances here in grey for reference and note that they were achieved under full-dataset training for each task while `TableLlama` is zero-shot.

For entity linking which requires the model to link the mention in a table cell to the correct referent entity in Wikidata, `TableLlama` also presents superior performance with 8 points gain over SOTA. Since the candidates are composed of referent entity name and description, we hypothesize LLMs have certain abilities to understand the description which help identify the correct entities.

Row population is the only task that `TableLlama` has a large performance gap compared to the SOTA. Here we provide a large number of candidates for the model to rank given table metadata and the seed row entity. By analyzing the errors, we found that the model can easily identify the entities containing similar numbers in sequence, such as the first example shown in Table 5 in Appendix C. However, for entities that share high similarities, such as the second example in Table 5 shows, the target row entities are the competitions which "Oleg Veretelnikov" got achievements in. To correctly populate the entities from the given plenty of candidates highly related to "competitions", it requires the model to understand the inherent relation between the athlete and each given candidate, which is still challenging for the current model.

Out-of-domain results. We evaluate `TableLlama` on six out-of-domain datasets. We observe that:

By comparing with the base model, `TableLlama` *can achieve 6-48 points gain on 6 out-of-domain datasets, which demonstrates* `TableInstruct` *can enhance the model's generalization ability.* By learning from the table-based training tasks, the model has acquired essential underlying table understanding ability, which can be transferred to other table-based tasks/datasets and facilitate their performance. Among these 6 datasets, we found that FEVEROUS, a table fact verification dataset exhibits the largest gain over the other 5 datasets. This is likely because the fact verification task is an in-domain training task, despite the dataset unseen during training. Compared with cross-task generalization, it may be easier to generalize to different datasets belonging to the same tasks.

Although there is still some gap between our performance and the previously reported SOTA for each dataset, we note those SOTAs were achieved under full-dataset training while `TableLlama` is zero-shot, hence it is reasonable to see such a gap. Nevertheless, we hope our work can inspire future work to further improve the zero-shot performance.

Open-source vs. closed-source. We compare `TableLlama` and closed-source LLMs (i.e., GPT-

3.5 and GPT-4) and observe that:

`TableLlama` *achieves better performance on in-domain tasks compared with closed-source LLMs.* It shows that even if closed-source LLMs have demonstrated strong performance in general, fine-tuning open-source LLMs on task-specific table-based data still has better performance.

`TableLlama` *shows less gap or even better zero-shot performance than closed-source LLMs on 4 out of 6 out-of-domain datasets (i.e., FEVEROUS, KVRET, ToTTo and WikiSQL), which shows TableLlama has gained generalization ability.* But closed-source LLMs are still stronger at table-based QA tasks that require more complex reasoning.

GPT-4 has better results than GPT-3.5 on all the in-domain and out-of-domain datasets except for FeTaQA and ToTTo. This is because GPT-4 generates longer output than GPT-3.5, so for FeTaQA and ToTTo which are evaluated using BLEU to compare the generated sentence the ground truth sentence, GPT-3.5 performs better.

4.2 Ablation Study

To better understand how `TableInstruct` helps enhance the model's generalizability, we conduct an ablation study to show the transfer between individual datasets.

The model trained on table-based QA tasks generalizes better than that trained on other tasks. As Table 4 shows, the model trained on HiTab scores more than 20 points on 7 out of 13 unseen datasets, and that trained on FeTaQA scores more than 10 points on 7 out of 13 unseen datasets, which can surpass models trained on the other 6 datasets individually by a large gain. We hypothesize that the general forms of table-based QA tasks can encourage models to gain general QA ability, which is beneficial when transferring to other tasks or datasets, since instruction tuning requires models to answer the question in essence. However, the models that are individually trained on other tasks may have learned strong superficial regularities as their formats have unique characteristics specially designed for themselves. Therefore, when evaluating on other unseen datasets or tasks, the models are too obfuscated to generate the correct answer.

Incorporating other tasks helps enhance the model's underlying generalization ability within the same task category. Comparing the model trained on TabFact and `TableInstruct`, when evaluating on FEVEROUS, which is the same task transfer for TabFact, we found `TableLlama` achieves 72.30 accuracy while the model trained on TabFact only achieves 56.15 accuracy. This indicates that other tasks in the training set also play an important role in engaging the model to obtain stronger table fact verification ability. Besides, if we compare the performance on two out-of-domain table QA datasets (i.e., WikiSQL and WikiTQ) among `TableLlama` and models individually trained on two table-based QA datasets (i.e., HiTab and FeTaQA), we can see `TableLlama` achieves better zero-shot performance. This indicates that including the other tasks (i.e., `TableInstruct`) to train the model can further enhance the model's underlying table question answering ability.

Individually fine-tuning models on tasks that are highly different from others tends to make models overfit and hardly generalize to others. As Table 4 shows, the model individually fine-tuned on 4 tasks: column type annotation, relation extraction, entity linking and TabFact tends to have weaker performance when evaluated on other tasks. We hypothesize that these four tasks are highly different from others, so the model individually trained on such tasks will overfit to the task itself, thus becoming hard to generalize to other unseen tasks.

5 Related Work

Table Representation Learning. Given the vast amount of knowledge stored in tables, various table-based tasks have been proposed (Pujara et al., 2021), such as column type annotation (Hulsebos et al., 2019), row population (Zhang and Balog, 2017), table QA (Sun et al., 2016; Pasupat and Liang, 2015; Cheng et al., 2022b; Nan et al., 2022), etc. In order to handle the semi-structured tables, existing work puts their efforts into designing special model architectures, such as TURL with structure-aware attention (Deng et al., 2020), TUTA with tree-based attention (Wang et al., 2021) and TaBERT with vertical self-attention mechanism (Yin et al., 2020); or designing special encodings such as table position encoding (Herzig et al., 2020; Wang et al., 2021; Eisenschlos et al., 2021), and numerical encoding (Wang et al., 2021) to better encode the table structure and infuse more information to the neural architecture. In addition, some work focuses on table pretraining (Liu et al., 2022; Yin et al., 2020; Deng et al., 2020; Iida et al., 2021) to encode knowledge in large-scale tables. However, although such existing works have shown promising progress, they are still data-specific and

Training Data	In-domain								Out-of-domain					
	ColType	RelExtra	EntLink	ScheAug	RowPop	HiTab	FeTaQA	TabFact	FEVER.	HybridQA	KVRET	ToTTo	WikiSQL	WikiTQ
	F1	F1	Acc	MAP	MAP	Acc	BLEU	Acc	Acc	Acc	Micro F1	BLEU	Acc	Acc
Base	3.01	0.96	31.80	36.75	4.53	14.96	8.54	41.65	23.66	20.72	38.90	10.39	14.84	25.48
ColType	94.32	0	0	0	0	0.13	0.52	0	0	0	0	1.11	0.35	0.21
RelExtra	45.69	**93.96**	0.45	8.72	0.99	7.26	1.44	0	2.38	8.17	5.90	5.60	7.02	9.58
EntLink	0.86	0.03	88.45	2.31	0.94	5.37	4.79	0	39.04	3.06	0	1.76	3.42	7.07
ScheAug	-	-	-	80.00	-	-	-	-	-	-	-	-	-	-
RowPop	-	-	-	-	53.86	-	-	-	-	-	-	-	-	-
HiTab	0.20	0.14	7.15	40.81	5.45	63.19	2.07	49.46	46.81	24.70	38.70	2.45	32.86	27.97
FeTaQA	0	0.40	0	30.23	0.15	19.57	38.69	1.20	1.21	**33.79**	**50.69**	**23.57**	13.79	27.12
TabFact	0	0	0	0	0	0	0	74.87	56.15	0	0	0	0	0
TableInstruct	**94.39**	91.95	**93.65**	**80.50**	**58.44**	**64.71**	**39.05**	**82.55**	**72.30**	27.61	48.73	20.77	**41.68**	**31.63**

Table 4: Transfer between different datasets. Bold numbers are the best results for each evaluation dataset. For models trained on schema augmentation (ScheAug) and row population (RowPop), their predictions on other datasets tend to repeat the candidates in the training data, which means they cannot generalize to other datasets, and hence we use "-" to represent their performances.

downstream task-specific, which requires special design tailored for tables and table-based tasks.

Our work proposes `TableInstruct` to unify different table-based tasks and develops a one-for-all LLM `TableLlama` to reduce those extra efforts during modeling. This high-level insight is similar to UnifiedSKG (Xie et al., 2022), which unifies a diverse set of structured knowledge grounding tasks into a text-to-text format. However, UnifiedSKG deals with different knowledge sources such as databases, knowledge graphs and web tables and does not explore instruction tuning, while we focus on a wide range of realistic tasks based on real-world tables via instruction tuning. In addition, a concurrent work (Li et al., 2023b) synthesizes diverse table-related tasks and finetunes close-source LLMs such as GPT-3.5 via instruction tuning. Compared to theirs, we collect more realistic and complex task data such as HiTab as well as classification and ranking tasks with candidates from Freebase and Wikidata and develop open-source LLMs for table-based tasks. We believe both our constructed high-quality table instruction tuning dataset and the trained model can be valuable resources for facilitating this line of research.

Instruction Tuning. Instruction tuning that trains LLMs using <instruction, output> pairs in a supervised fashion is a crucial technique to enhance the capabilities and controllability of LLMs (Chung et al., 2022; Wang et al., 2022; Mishra et al., 2022). The instructions serve to constrain the model's outputs to align with the desired response characteristics or domain knowledge and can help LLMs rapidly adapt to a specific domain without extensive retraining or architecture designs (Zhang et al., 2023). Therefore, different instruction tuning datasets have been proposed to guide LLMs' behaviors (Wang et al., 2022; Honovich et al., 2022; Longpre et al., 2023; Xu et al., 2023). Different instruction tuning models such as InstructGPT (Ouyang et al., 2022), Vicuna (Zheng et al., 2023) and Claude[2] emerge and demonstrate boosted performance compared with the pre-trained models. In addition, instruction tuning has been applied to different modalities such as images, videos and audio (Li et al., 2023a) and has shown promising results. This signals that instruction tuning can be a promising technique to enable large pre-trained models to handle various tasks. However, how to utilize instruction tuning to guide LLMs to complete tables-based tasks is still under-explored. Our work fills this gap by constructing a high-quality table instruction tuning dataset: `TableInstruct`, which covers large-scale diverse and realistic tables and tasks to enable both modeling and evaluation. We will also release `TableLlama`, an open-source LLM-based generalist model fine-tuned on `TableInstruct` to promote this avenue of research.

6 Conclusion

This paper makes the first step towards developing open-source large generalist models for a diversity of table-based tasks. Towards that end, we construct `TableInstruct` and develop the first open-source generalist model for tables, `TableLlama`. We evaluate both in-domain and out-of-domain settings and the experiments show that `TableLlama` has gained strong table understanding ability and generalization ability.

[2]https://www.anthropic.com/index/introducing-claude

7 Limitations

Although we strive to increase the diversity of our dataset and have collected 14 datasets of 11 tasks for tables, there are still some table-based tasks such as data imputation and table classification which are not included in `TableInstruct`. Therefore, even if `TableLlama` has demonstrated the generalization ability on different out-of-domain datasets and tasks, the model's performance may vary based on the complexity and specifics of the new unseen table tasks and datasets. As we have made the first step towards building an open large generalist model for tables, we encourage future work to further explore this line of research and to further enhance the model's generalization ability for tables.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020a. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv:2309.12307*.

Zhoujun Cheng, Haoyu Dong, Ran Jia, Pengfei Wu, Shi Han, Fan Cheng, and Dongmei Zhang. 2022a. Fortap: Using formulas for numerical-reasoning-aware table pretraining. *Association for Computational Linguistics*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022b. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: table understanding through representation learning. *Proceedings of the VLDB Endowment*, 14(3):307–319.

Julian Martin Eisenschlos, Maharshi Gor, Thomas Müller, and William W. Cohen. 2021. Mate: Multi-view attention for table transformer efficiency.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Google.2015. Freebase data dumps. https://developers.google.com/freebase/data.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1500–1508.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023b. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Jay Pujara, Pedro Szekely, Huan Sun, and Muhao Chen. 2021. From tables to knowledge: Recent advances in table understanding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4060–4061.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models.

Dominique Ritze, Oliver Lehmberg, and Christian Bizer. 2015. Matching html tables to dbpedia. In *Proceedings of the 5th international conference on web intelligence, mining and semantics*, pages 1–6.

Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table cell search for question answering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 771–782.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran,

Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey.

Shuo Zhang and Krisztian Balog. 2017. Entitables: Smart assistance for entity-focused tables. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 255–264.

Guangzhen Zhao and Peng Yang. 2022. Table-based fact verification with self-labeled keypoint alignment. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1401–1411, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.

A Training and Inference Details

We choose LongLoRA 7B (Chen et al., 2023b), fully fine-tuning version with 8K context length limit as our base model. The fully fine-tuning version replaces the vanilla attention in Llama 2 with shift short attention. We fine-tune the model with Huggingface transformers library (Wolf et al., 2020). We merge all eight datasets and repeat three smaller datasets (i.e., FeTaQA, HiTab and TabFact) for six times and randomly shuffle them as our final training data. We use a learning rate of 2e-5 and set the batch size at 3. We streamingly train the model on 48 A100 80GB GPUs and use a cosine scheduler with a 3% warm-up period for 2 epochs. To efficiently train the model, we employ DeepSpeed training with ZeRO-2 stage (Rajbhandari et al., 2020). For both training and inference, we set the input length as 8192. For inference on `TableLlama`, as different tasks have different lengths of the ground truth, we use 64 as the output length for column type annotation, relation extraction, entity linking, HiTab, TabFact, FEVEROUS, HybridQA, WikiSQL and WikiTQ, 128 for schema augmentation, FeTaQA, KVRET and ToTTo, and 512 for row population. For column type annotation and entity linking, we uniformly sample a subset from the original test data as our test set due to the large test size. For row population, we filter out the examples with more than 500 candidate entities from the original test set and randomly sample a subset as our test set. For all the down-sampled test set, we reproduce the SOTA results using the SOTA model.

For closed-source LLMs, we use the gpt-4-1106-preview version for GPT-4, which is the latest version that supports 128K context and reports the best performance. For GPT-3.5, we use the gpt-3.5-turbo-1106 version, which supports 16K context.

B Existing SOTA Models

TURL (Deng et al., 2020) is an encoder-based BERT-like model pre-trained on 570K tables. Though TURL has shown SOTA performance on various table tasks such as column type annotation, relation extraction, entity linking, row population and schema augmentation, it requires fine-tuning task-specific modules on labeled data. The SOTA method for HiTab builds on 1) TUTA (Wang et al., 2021), which uses tree attention as the encoder to capture table structures and 2) FORTAP (Cheng et al., 2022a), which leverages spreadsheet formulas for table pre-training to better handle numerical reasoning. The SOTA method for TabFact designs a self-labeled keypoint alignment (Zhao and Yang, 2022) to align salient evidence and aggregate essential information between the statement and table. For HybridQA, the SOTA method MATE (Eisenschlos et al., 2021) uses sparse attention for Transformer architecture which allows heads to efficiently attend to either rows or columns in a table. The SOTA method for WikiSQL and WikiTQ is TAPEX (Liu et al., 2022), which fuses table pre-training by learning a neural SQL executor over a synthetic corpus. For FeTaQA, FEVEROUs, KVRET and ToTTo, the SOTA results come from T5-3B fine-tuned on their own individual training data (Xie et al., 2022).

C Case Study

Query Caption	Seed	Candidates	Target	AP	Predicted
concord quarry dogs	2002_NECBL_season	2003_Amsterdam_Admirals_season The_Young_Punx 2011_FCBL_season ...	2003_NECBL_season 2004_NECBL_season 2005_NECBL_season 2006_NECBL_season	1.0	2003_NECBL_season 2004_NECBL_season 2005_NECBL_season 2006_NECBL_season
oleg veretelnikov achievements	1993_Asian_Athletics _Championships	New_York_City_Marathon Friendship_Games 1998_Asian_Games ...	1997_World_Championships_in _Athletics-2013_Men's_decathlon 1994_Asian_Games 1999_World_Championships_in_Athletics 1998_Asian_Games	0.2	1994_Asian_Games 1995_Asian_Athletics_Championships Athletics_at_the_1995_All-Africa_Games ...

Table 5: Case study for row population task. "Query Caption" refers to the table metadata such as Wikipedia page title and table caption. "AP" means average precision.

D Example Prompts

Column Type Annotation

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:
This is a column type annotation task. The goal for this task is to choose the correct types for one selected column of the table from the given candidates. The Wikipedia page, section and table caption (if any) provide important information for choosing the correct column types.

Input:
[TLE] The Wikipedia page is about 1958 Nippon Professional Baseball season. The Wikipedia section is about Central League. The table caption is Pitching leaders. [TAB] col: | stat | player | team | total | [SEP] row 1: | Wins | Masaichi Kaneda | Kokutetsu Swallows | 31| [SEP] row 2: | Losses | Noboru Akiyama | ...

Question:
The column 'player' contains the following entities: <Masaichi Kaneda>, <Noboru Akiyama>, etc. The column type candidates are: tv.tv_producer, astronomy.star_system_body, location.citytown, sports.pro_athlete, biology.organism, medicine.muscle, baseball.baseball_team, baseball.baseball_player, aviation.aircraft_owner, people.person, ... What are the correct column types for this column (column name: player; entities: <Masaichi Kaneda>, <Noboru Akiyama>, etc)?

Response:
sports.pro_athlete, baseball.baseball_player, people.person.

Figure 3: **Column type annotation** task. This task is to annotate the selected column with the correct semantic types. We mark candidates with red color in the "task instruction" part. The candidate size can be up to hundreds to thousands in `TableInstruct`.

Relation Extraction

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:
This is a relation extraction task. The goal for this task is to choose the correct relations between two selected columns of the table from the given candidates. The Wikipedia page, section and table caption (if any) provide important information for choosing the correct relation types.

Input:
[TLE] The Wikipedia page is about Yukon Legislative Assembly. The Wikipedia section is about Current members. [TAB] col: | | name | party | riding | row 1: | | Kevin Barr | New Democratic Party | Mount Lorne-Southern Lakes | [SEP] row 2: | | Brad Cathers | ...

Question:
The two selected column names are: <(name),(party)>. The entity pairs for these two columns are: <(Kevin Barr),(New Democratic Party)>, <(Brad Cathers),(Yukon Party)>, <(Currie Dixon),(Yukon Party)>, <(Darius Elias),(Yukon Party)>, ... The relation type candidates are: location.location.contains, aviation.airline.hubs, film.film.written_by, time.event.instance_of_recurring_even , people.person.place_of_birth, music.composer.compositions, sports.sports_team.roster- sports.sports_team_roster.player, location.location.containedby, soccer.football_player.statistics- soccer.football_player_stats.team... What are the correct relation types for the two selected columns (column names: <(name),(party)>. entity pairs: <(Kevin Barr),(New Democratic Party)>, <(Brad Cathers),(Yukon Party)>, <(Currie Dixon), (Yukon Party)>, <(Darius Elias),(Yukon Party)>, etc)?

Response:
government.politician.party-government.political_party_tenure.party.

Figure 4: **Relation extraction** task. This task is to annotate the selected column pairs with the correct relations. We mark candidates with red color in the "task instruction" part. The candidate size can be up to hundreds to thousands in `TableInstruct`.

Entity Linking

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: This is an entity linking task. The goal for this task is to link the selected entity mention in the table cells to the entity in the knowledge base. You will be given a list of referent entities, with each one composed of an entity name, its description and its type. Please choose the correct one from the referent entity candidates. Note that the Wikipedia page, Wikipedia section and table caption (if any) provide important information for choosing the correct referent entity.

Input: [TLE] The Wikipedia page is about A-League all-time records. The Wikipedia section is about Average season attendances. [TAB] col: | season | league average | total gate receipts | highest club | average | lowest club | average | row 1: | 2005-06 | 10,955 | 920,219 | Sydney FC | 16,669 | New Zealand Knights | 3,909 | [SEP] row 2: | 2006-07 | 12,927 | ...

Question: The selected entity mention in the table cell is: Melbourne Victory. The column name for 'Melbourne Victory' is highest club. The referent entity candidates are: <Melbourne Victory FC W-League [DESCRIPTION] None [TYPE] SoccerClub>, <2016-17 Melbourne Victory FC season [DESCRIPTION] None [TYPE] SoccerClubSeason>, <2011-12 Melbourne Victory season [DESCRIPTION] Association football club 2011/12 season for Melbourne Victory [TYPE] SoccerClubSeason>, ... What is the correct referent entity for the entity mention 'Melbourne Victory' ?

Response: <Melbourne Victory [DESCRIPTION] association football team from Australia [TYPE] SoccerClub>.

Figure 5: **Entity linking** task. This task is to link the selected entity mention in the table cells to the entity in the knowledge base. We mark candidates with red color in the "task instruction" part. The candidate size can be up to hundreds to thousands in `TableInstruct`.

Row Population

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: This is a table row population task. The goal of this task is to populate the possible entities of the selected column for a table, given the Wikipedia page title, Wikipedia section title, table caption (if any) and table headers. You will be given a list of entity candidates. Please rank them so that the most likely entities come first.

Input: [TLE] The Wikipedia page is about NBA conference finals. The Wikipedia section is about eastern conference finals. The table headers are: | year | champion | coach | result | runner-up | coach |. You need to populate the column: year. [SEED] The seed entity is <1971_NBA_playoffs>.

Question: The entity candidates are: <2003_NBA_playoffs>, <1982-83_Washington_Bullets_season>, <2004_NBA_playoffs>, <Philadelphia_76ers>, <1983-84_Washington_Bullets_season>, <1952_NBA_playoffs>, <1972_NBA_playoffs>, <1999-2000_Dallas_Mavericks_season>, <1985- 86_Sacramento_Kings_season>, <2000-01_Vancouver_Grizzlies_season>, <Toronto_Raptors>, <Vancouver_Grizzlies>, <1976_NBA_playoffs>, ...

Response: <1972_NBA_playoffs>, <1973_NBA_playoffs>, <1974_NBA_playoffs>, <1975_NBA_playoffs>, <1976_NBA_playoffs>, <1977_NBA_playoffs>, ...

Figure 6: **Row population** task. This task is to populate the possible entities of the selected column for a table given partial table and table metadata. We mark candidates with red color in the "task instruction" part. The candidate size can be up to hundreds to thousands in `TableInstruct`.

Schema Augmentation

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:
This is a table schema augmentation task. The goal of this task is to populate the possible headers for a table, given the table caption and the seed table header. You will be given a list of table header candidates. Please rank them so that the most likely headers come first.

Input:
[TLE] The table caption is 2010-11 rangers f.c. season. [SEED] The seed table header is <competition>.

Question:
The header candidates are: <from>, <fee (\u00a3)>, <opponents>, <final position / round>, <started round>, <player>, <fee>, <scorers>, <position (s)>, <name>, <venue>. Please rank the headers in the header candidates.

Response:
<town/city>, <summary>, <suburb>, <county>, <region>, <district>, <stadium>, <city>, <home team>, <remarks>, <city name>, <film name>

Figure 7: **Schema augmentation** task. This task is to populate the possible headers for a table, given the table caption and the seed table header. The targets in this case are: <started round>, <final position/round>.

Hierarchical Table QA

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: This is a hierarchical table question answering task. The goal for this task is to answer the given question based on the given table. The table might be hierarchical.

Input: [TLE] The table caption is department of defense obligations for research, development, test, and evaluation, by agency: 2015-18. [TAB] | agency | 2015 | 2016 | 2017 | 2018 | [SEP] | department of defense | department of defense | department of defense | department of defense | department of defense | [SEP] | rdt&e | 61513.5 | 69306.1| 70866.1 | 83725 | [SEP] | total research | 6691.5 | 7152 | 7178 | 7652.7 | [SEP] | basic research | 2133.4 | 2238.7 | 2110.1 | 2389.9 | [SEP] | defense advanced research projects agency | defense advanced research projects agency | defense advanced research projects agency | ...

Question: How many dollars are the difference for total research of department of the air force increase between 2016 and 2018?

Response:
142.3.

Figure 8: **Hierarchical table QA** task. This task is to answer the question based on the tables with complex hierarchical structures.

Highlighted Cells QA

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:
This is a free-form table question answering task. The goal for this task is to answer the given question based on the given table and the highlighted cells.

Input:
[TLE] The Wikipedia page title of this table is Holly Dunn. The Wikipedia section title of this table is Singles. [TAB] | Year | Single | Peak chart positions | Peak chart positions | Album | [SEP] | Year | Single | US Country | CAN Country | Album | [SEP] | 1985 | ...

Question:
The highlighted cells of the table are: [HIGHLIGHTED_BEGIN] [1988], [Across the Rio Grande in 1988 included the singles \"That's What Your Love Does to Me\" and \"(It's Always Gonna Be) Someday\".], [\"That's What Your Love Does to Me\"], [Across the Rio Grande], [1988], [\"(It's Always Gonna Be) Someday\"], [Across the Rio Grande] [HIGHLIGHTED_END] What singles were Included in Across the Rio Grande in 1988?

Response:
Across the Rio Grande in 1988 included the singles \"That's What Your Love Does to Meänd \"(It's Always Gonna Be) Someday\".

Figure 9: **Highlighted cells QA** task. This task is to answer the question based on the tables with highlighted cells.

Table Fact Verification (TabFact)

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:
This is a table fact verification task. The goal of this task is to distinguish whether the given statement is entailed or refuted by the given table.

Input:
[TLE] The table caption is about tony lema. [TAB] | tournament | wins | top - 5 | top - 10 | top - 25 | events | cuts made [SEP] | masters tournament | 0 | 1 | 2 | 4 | 4 | 4 | [SEP] | us open | 0 | 2 | 3 | 4 | 6 | 5 | [SEP] | the open championship | 1 | 2 | 2 | 2 | 3 | 3 | [SEP] | pga championship | 0 | 0 | 1 | 2 | 5 | 4 | [SEP] | totals | 1 | 5 | 8 | 12 | 18 | 16 |.

Question:
The statement is: <tony lema be in the top 5 for the master tournament, the us open, and the open championship>. Is it entailed or refuted by the table above?

Response:
entailed.

Figure 10: **Table fact verification** task. This task is to discriminate whether the claim can be entailed or refuted by the given table.

Hybrid Question Answering

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:
This is a hybrid question answering task. The goal of this task is to answer the question given tables and passages.

Input:
col : rank | player | team (s) by season | carries | yards | average row 1 : 1 | emmitt smith | dallas cowboys (1990 - 2002) arizona cardinals (| 4,409 | 18,355 | 4.2 row 2 : 2 | walter payton | chicago bears (1975 - 1987) | 3,838 | 16,726 | 4.4 | ... **passages**: emmitt smith: smith led the league in rushing and won the super bowl in the same year three times (1992, 1993, and 1995) when to that point it had never been done. | walter payton: Walter jerry payton (july 25, 1954 - november 1, 1999) was an american professional football player who was a running back for the chicago ... | san francisco 49ers: the team began play ....

Question:
What is the middle name of the player with the second most national football league career rushing yards?

Response:
jerry.

Figure 11: **HybridQA** task. This task is to answer the question based on the table and passages.

Table Grounded Dialogue Generation

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:
This is a dialogue response generation task grounded on tables. The goal of this task is to generate response based on the given dialogue history and the given table. The dialogues are grounded through underlying tables and span three distinct tasks in the in-car personal assistant space: calendar scheduling, weather information retrieval, and point-of-interest navigation.

Input:
col : poi | distance | traffic_info | poi_type | address row 1 : chevron | 5_miles | moderate_traffic | gas_station | 783_arcadia_pl row 2 : town_and_country | 5_miles | no_traffic | shopping_center | 383_university_ave

Question:
The dialogue history is: <what is the address ? || taking you to chevron | that s good ! please pick the quickest route to get there and avoid all heavy_traffic ! | there is a chevron | what gas_station are here ?>. Please generate the response based on the given table and the given dialogue history.

Response:
783_arcadia_pl is the address for chevron gas_station.

Figure 12: **Table grounded dialogue generation** task. This task is to generate the response based on the given table and the dialogue history.

Highlighted Cells Description

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:
This is a highlighted cells description task. The goal of this task is to generate the language description given table cells.

Input:
<page_title> List of Governors of South Carolina </page_title> <section_title> Governors under the Constitution of 1868 </section_title> <table> <cell> 76 <col_header> # </col_header> <col_header> 74 </col_header> <col_header> 75 </col_header> </cell> <cell> Daniel Henry Chamberlain <col_header> Governor </col_header> <row_header> 76 </row_header> </cell> <cell> December 1, 1874 <col_header> Took Office </col_header> <row_header> 76 </row_header> </cell> </table>.

Question:
Please generate one natural language description to describe the given highlighted table cells.

Response:
Daniel Henry Chamberlain was the 76th Governor of South Carolina from 1874.

Figure 13: **Highlighted cells description** task. This task is to generate the language description for the highlighted table cells.

Table Fact Verification (FEVEROUS)

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:
This is a table fact verification task. The goal of this task is to distinguish whether the given statement is entailed or refuted by the given table.

Input:
col : logical system | lindenbaum-tarski algebra row 1 : classical sentential logic | boolean algebra | row 2 : intuitionistic propositional logic | heyting algebra ...

Question:
The statement is: <algebraic logic has five logical system and lindenbaum-tarski algebra which includes physics algebra and nodal algebra (provide models of propositional modal logics).>. Is it entailed or refuted by the table above? If you think the current information can not provide enough evidence for determining it, please choose 'not enough info', otherwise please choose the answer from 'supports' or 'refutes'.

Response:
refutes.

Figure 14: **Table fact verification** task. This task is to discriminate whether the claim can be entailed or refuted by the given table.

Table QA (WikiSQL)

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:
This is a table QA task. The goal of this task is to answer the question given the table.

Input:
col : player | no. | nationality | position | years in toronto | school/club team row 1 : aleksandar radojevi\u0107 | 25 | serbia | center | 1999-2000 | barton cc (ks) row 2 : shawn respert | 31 | united states | guard | 1997-98 | michigan state ...

Question:
What is terrence ross' nationality?

Response:
united states.

Figure 15: **Table QA** task. This task is to answer the question based on the given table.

Table QA (WikiTQ)

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:
This is a table QA task. The goal of this task is to answer the question given the table.

Input:
col : series # | season # | title | notes | original air date row 1 : 1 | 1 | "the charity" | alfie, dee dee, and melanie are supposed to be helping | october 15, 1994 row 2 : 2 | 1 | "the practical joke war" | alfie and goo unleash harsh practical jokes on dee dee | october 22, 1994 ...

Question:
Alfie's birthday party aired on january 19. What was the airdate of the next episode?

Response:
january 26, 1995.

Figure 16: **Table QA** task. This task is to answer the question based on the given table.